

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1 E
SEGUNDO SEMESTRE 2022

CORTO 1 – parte práctica

CONTENIDO

Objetivos	1
General	1
Específico	1
Tienda	2
Requerimientos.....	3
Consideraciones.....	5
Restricciones.....	6
Entregables	6

OBJETIVOS

GENERAL

1. Practicar el pensamiento orientado a objetos

ESPECÍFICO

1. Usar el paradigma de programación orientada a objetos para analizar un caso de negocio
2. Abstraer un problema y generar una solución en código
3. Entrenar la capacidad de análisis y resolución de problemas

TIENDA

Se le presenta el escenario de una tienda que actualmente tiene problemas llevando el control de inventario de su mercadería. El negocio ha ido bien y parece que las ventas han ido subiendo. Conforme las ventas suben, el producto va saliendo de bodega. Últimamente, es común decirle al cliente que ya no hay existencia de un producto porque el inventario no se actualiza en tiempo real.

Dado el anterior contexto se solicita que modele el comportamiento de las salidas y entradas de mercadería de una bodega y genere una solución informática en Java usando una aplicación de ventanas. Considerar los siguientes objetos:

- Productos
 - Nombre
- Entrada de producto
 - Fecha y hora de entrada
 - Cantidad de producto que entró
- Salida de producto
 - Fecha y hora de salida
 - Cantidad de producto que salió

Si así lo considera, puede usar otros objetos u otra forma de modelar el negocio. Lo importante es que cumpla con los requerimientos que se listan en la siguiente sección.

REQUERIMIENTOS

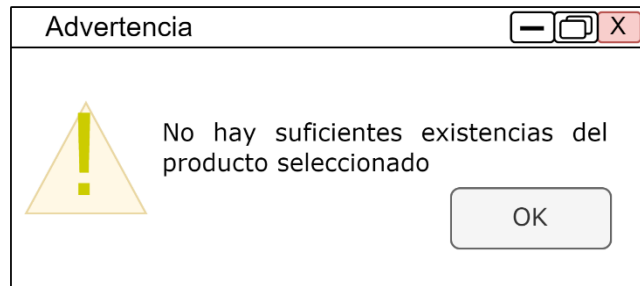
1. Para registrar la entrada y salida de una mercadería, primero deberá registrar la información del producto. Deberá almacenar al menos el nombre del producto.

A screenshot of a software window titled "Crear producto". It features a text input field labeled "Producto" and a button labeled "Crear".

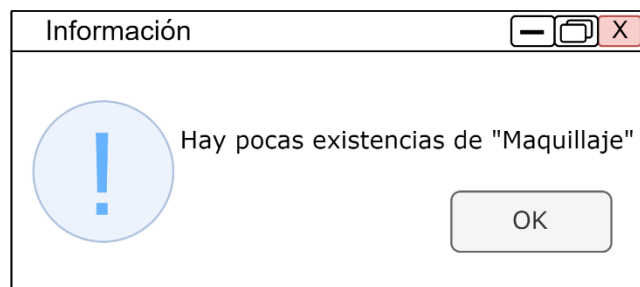
2. Para dar entrada o salida a un producto, deberá almacenarse al menos la fecha y hora, y la cantidad de producto que está entrando o saliendo.

A screenshot of a software window titled "Entradas y salidas". It contains three input fields: a dropdown menu for "Producto" with options "Maquillaje", "Pinta uñas", and "Rizador"; a dropdown menu for "Tipo de operación" with options "Entrada" and "Salida"; and a text input field for "Cantidad" containing the value "100". At the bottom is an "Aceptar" button.

3. Para que haya una salida de mercadería primero deberá existir suficiente producto en existencias. Es decir, debe validar que la cantidad de producto de salida no sea mayor a la cantidad de producto en existencias.



4. Luego de que se registre la salida de producto, el programa deberá revisar que la cantidad de inventario en bodega no sea menor a 10 unidades. Si es menor a 10 unidades deberá lanzar una advertencia al usuario indicando que pronto quedará desabastecido de X producto y que deberá comprar más.



5. Se deberá visualizar el historial de entradas y salidas de un producto en particular. Para mayor orden y claridad, cada producto tendrá su propio historial de entradas y salidas.

En una sola tabla se deberá visualizar de manera legible la fecha, detalle, cantidad de entrada, cantidad de salida y existencias. El detalle puede ser simplemente la palabra "Entrada" o "Salida", dependiendo de la operación.

La columna "existencias" deberá mostrar las existencias luego de realizada la operación descrita en cada fila. Esa columna saldo deberá incrementar o disminuir conforme a la cantidad especificada de entrada o salida.

Historial				
Producto		Mostrar inventario		
Maquillaje				
Fecha	Detalle	Entrada	Salida	Existencias
03/08/2022 21:14	Entrada	100	0	100
03/08/2022 21:15	Salida	0	50	50
03/08/2022 21:17	Salida	0	25	25
03/08/2022 21:17	Entrada	75	0	100

CONSIDERACIONES

- En ningún momento y por ninguna razón la existencia de un producto será negativo.
- Se tomará en cuenta la tolerancia a fallos de la solución. Deberá procurar validar cualquier posible excepción de la solución desarrollada, independientemente si se mencionó en los requerimientos o no.
- En ningún momento y por ninguna razón una operación tendrá valores 0 en “entrada” y “salida” al mismo tiempo. Así como tampoco deberá tener valores mayores 0 al mismo tiempo. Estos campos deberán ser mutuamente excluyentes: uno u otro, pero no ambos.
- Las imágenes que se muestran en este enunciado sirven de ejemplo. No es necesario que la solución generada sea exactamente igual. Únicamente deberá asegurarse que se cumpla con los requerimientos descritos.
- La aplicación deberá ser desarrollada completamente en Java. Se recomienda usar las librerías java.awt y javax.swing para construir la interfaz gráfica.
- El IDE por utilizar queda a discreción del estudiante.
- No se tomará en cuenta el código como “procedimiento” para ponderar un ítem. Sin embargo, si se revisará para asegurarse que se cumpla con los requerimientos y restricciones mencionados en este enunciado.

RESTRICCIONES

- No se podrá usar librerías que implementen la clase `java.util.Collection` o extiendan la clase `java.util.Iterable`. Es decir, no se podrá usar objetos como `LinkedList`, `Set`, `Vector`, `ArrayList`, etc. Únicamente deberá hacer uso de variables, constantes, arreglos, funciones, ciclos (`for`, `while`, `do-while`), estructuras de selección (`if-else`, `switch`), clases generadas por el estudiante, entre otras.
- Las clases creadas por el estudiante y usadas para modelar objetos deberán cumplir con las propiedades de la programación orientada objetos. Se penalizará el uso inadecuado o desproporcionado de las variables y funciones de clase (`static`) cuando la información deba ser almacenada en variables de instancia o manejada por funciones de instancia, respectivamente.
- Copias detectas tendrán 0 en el corto práctico.

ENTREGABLES

- Fecha de entrega: **27/08/2022 23:59 pm.**
- Entregar en el espacio habilitado un archivo comprimido (zip o rar) con todo el código fuente de la aplicación. Colocarle por nombre el carné.