

**Instrucciones.** Dispone de 2 horas y 40 minutos para realizar el examen. Registre su grupo, carné y nombre en su cuaderno de examen. Es obligatorio apagar el celular y cualquier otro dispositivo de comunicación. Utilice únicamente los conceptos y técnicas vistos en clases para resolver el problema. Solo puede usar bibliotecas (librerías) cubiertas en el curso. Se evaluarán las buenas prácticas de programación, las convenciones de estilo y la indentación. Puede consultar sus notas en máximo dos hojas manuscritas originales previamente elaboradas, las cuales debe entregar junto con este enunciado y su cuaderno de examen al finalizar la prueba. Asegúrese de firmar la hoja de asistencia al salir. Se podrán realizar actividades adicionales posteriores al examen como entrevistas con efecto en la calificación del mismo.

Una empresa distribuidora de equipo médico abrió una sucursal en Costa Rica. En otros países ha enfrentado dificultades con sus inventarios, como extravío de productos en sus bodegas, o carencia de producto por no solicitar a tiempo unidades a sus proveedores, entre otros. Con el fin de prevenir estos inconvenientes, se ha iniciado un plan piloto en Costa Rica de crear una solución digital que permita mantener el inventario al día, por medio de programación orientada a objetos.

El histórico de compras y ventas de producto realizadas por la distribuidora están guardadas en un archivo de texto con extensión `.csv`. En este archivo, cada línea registra una transacción (compra o venta). Para cada transacción se especifica: código de producto, descripción del producto, tipo de transacción (*compra o venta*), y la cantidad del producto. Estos valores se separan uno del otro por un carácter de punto y coma (;). A continuación, se muestra un trozo de un archivo llamado `Flujo.csv`.

```
1;Aguja 23;compra;500
2;Kit de entubacion;compra;200
3;Mascarilla laringea;compra;1250
4;Tablero masaje cardiaco;compra;350
3;Mascarilla laringea;venta;750
2;Kit de entubacion;venta;180
1;Aguja 23;venta;300
1;Aguja 23;venta;200
```

Se quiere un programa en Python3 que lleve el inventario al día de cada producto. Adelante se muestra una posible ejecución del programa, donde los valores en **negrita** representan entradas del usuario.

```
Archivo de inventario: Registro
Error: No se puede abrir Registro.csv.
Archivo de inventario: Ejemplo.csv
Error: No se puede abrir Ejemplo.csv.
Archivo de inventario: Flujo
Se ha cargado Flujo.csv al inventario.
```

```
Opción [A=Registrar B=Inventario C=Salir]: B
Estado [D=Dispnb1 L=Limitado A=Agotado T=Todos]: T
Prod. 1, Aguja 23, 0 existencias
Prod. 2, Kit de entubacion, 20 existencias
Prod. 3, Mascarilla laringea, 500 existencias
Prod. 4, Tablero masaje cardiaco, 350 existencias
```

```
Opción [A=Registrar B=Inventario C=Salir]: A
Tipo de transacción [compra o venta]: compra
Código de producto: 5
Descripción del nuevo producto: Jeringa 125
Cantidad comprada de producto: 50
```

```
Opción [A=Registrar B=Inventario C=Salir]: A
Tipo de transacción [compra o venta]: compra
Código de producto: 2
Prod. 2, Kit de entubacion, 20 existencias
Cantidad comprada de producto: 80
```

```
Opción [A=Registrar B=Inventario C=Salir]: A
```

```
Tipo de transacción [compra o venta]: venta
Código de producto: 50
Error: producto no existe
```

```
Opción [A=Registrar B=Inventario C=Salir]: A
Tipo de transacción [compra o venta]: venta
Código de producto: 5
Prod. 5, Jeringa 125, 50 existencias
Cantidad a vender de producto: 45
```

```
Opción [A=Registrar B=Inventario C=Salir]: A
Tipo de transacción [compra o venta]: venta
Código de producto: 5
Prod. 5, Jeringa 125, 50 existencias
Cantidad a vender de producto: 70
Error: No hay suficientes existencias
```

```
Opción [A=Registrar B=Inventario C=Salir]: B
Estado [D=Dispnb1 L=Limitado A=Agotado T=Todos]: L
Producto 5, Jeringa 125, 5 existencias
Producto 6, Kit de suturas, 8 existencias
```

```
Opción [A=Registrar B=Inventario C=Salir]: B
Estado [D=Dispnb1 L=Limitado A=Agotado T=Todos]: A
Producto 1, Aguja 23, 0 existencias
```

```
Opción [A=Registrar B=Inventario C=Salir]: C
```

Para efectos del programa, cada producto tiene un código numérico secuencial que lo identifica en el inventario, un texto descriptivo para ayudar a los humanos a reconocerlo, y la cantidad de existencias actuales en el inventario. Suponga que los usuarios son cuidadosos con mantener la secuencia correcta de los códigos de producto y el programa no requiere verificarlo. El sistema debe mantener actualizada la cantidad de existencias de cada producto de acuerdo a los movimientos que de él se hagan. Cuando se registra una venta de un producto, las existencias se verán reducidas del inventario. Cuando se hace una compra a los proveedores, habrá un aumento en las existencias del producto en el inventario. De acuerdo con sus existencias, un producto puede estar en estado: *Disponible* si hay 10 o más unidades del producto en el inventario, *Limitado* si hay entre 1 y 9 unidades, y *Agotado* si no hay unidades en el inventario.

Al iniciar, el programa deberá cargar el inventario a partir del archivo de registros. Debe pedir al usuario un nombre de archivo y el usuario podría o no omitir su extensión. Si el archivo no existe o no se puede abrir, el programa debe mostrar un mensaje de error y pedir de nuevo el nombre del archivo hasta que se provea un nombre de un archivo correcto. El programa carga del archivo al inventario como una colección de objetos *Producto*, y luego muestra el menú.

El menú tiene tres opciones: (A) Registrar una compra o venta, (B) Consultar el inventario, o (C) Salir del programa. La opción A del menú pregunta al usuario el tipo de transacción (*compra* o *venta*) que va a registrar, luego el código del producto que podría corresponder a un producto nuevo en el inventario o uno existente. De acuerdo a estos dos valores se debe tomar una acción como se indica en la siguiente tabla.

Transacción	de producto inexistente	de producto existente
<b>Compra</b>	Solicita al usuario la descripción del nuevo producto y la cantidad de unidades compradas. Agrega el nuevo producto al inventario.	No pregunta la descripción del producto, sino que muestra su detalle (código, descripción y existencias). Pregunta la cantidad de unidades compradas, las agrega al producto.
<b>Venta</b>	Presenta un mensaje de error y retorna al menú.	Muestra detalle del producto (código, descripción, y existencias). Pregunta la cantidad de unidades a vender. Si hay suficientes unidades, las reduce de las existencias, de lo contrario, presenta un mensaje de error y no registra la venta.

La opción B del menú permite al usuario hacer consultas sobre cuatro posibles estados (D por Disponible, L por Limitado, A por Agotado, y T por Todos). El sistema mostrará un detalle de cada producto que coincida con el estado escogido por el usuario. Los productos se deben mostrar en orden ascendente de código.

*Evaluación.* Su solución debe ser orientada a objetos, con al menos las dos clases que se detallan:

**Clase Producto** (modelo), con las siguientes responsabilidades:

1. **[10%]** Todo producto tiene de atributos: código, descripción, y existencias. Todo producto debe cargarse inicialmente del archivo, o de los datos que provienen de una línea del archivo de registros.
2. **[10%]** Modificar las existencias. Recibe una cantidad positiva si es una compra, o negativa si es una venta. Retorna un booleano que indica si la operación se pudo realizar o no (según existencias).
3. **[5%]** Retornar el estado del producto, de acuerdo con su cantidad de existencias.
4. **[5%]** Describirse, que retorna un texto: "Prod. <código>, <descripción>, <cantidad> existencias."

**Clase Inventario** (de control), con las siguientes responsabilidades:

5. **[10%]** El inventario tiene de atributo único una colección de objetos tipo *Producto*, inicialmente vacía.
6. **[20%]** Solicitar el nombre de archivo, validarlo, y cargar los datos del archivo a la colección de objetos.
7. **[15%]** Registrar una compra o venta sobre el inventario, pidiendo los datos que se necesiten en cada caso (se debe considerar el tipo de transacción, y si el producto existe o es nuevo).
8. **[10%]** Imprimir las descripciones de los Productos cuyo estado coincide con el indicado por parámetro.
9. **[10%]** Mostrar repetitivamente el menú, e invocar a los métodos respectivos.

**Función principal:**

10. **[5%]** Incorpore una función principal que crea una instancia del inventario (clase de control), e invoca en esta instancia al método que carga el inventario y luego al método que corre el menú.