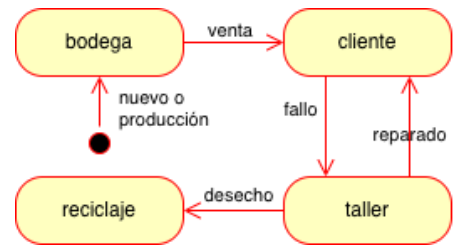


Instrucciones. Dispone de 2 horas y 40 minutos para realizar el examen. Registre su grupo, carné y nombre en su cuaderno de examen. Es obligatorio apagar el celular y cualquier otro dispositivo de comunicación. Utilice únicamente los conceptos y técnicas vistos en clases para resolver el problema. Solo 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 fabrica varios tipos de productos y necesita su ayuda para rastrearlos a través de su ciclo de vida. El diagrama de estados de la derecha muestra en rectángulos redondeados los lugares donde un producto podría encontrarse y con flechas los eventos que lo podrían hacer cambiar de estado. Una vez que la empresa crea un **nuevo** producto, le asigna un código secuencial y las unidades iniciales pasarán a bodega, donde estarán listas para la venta. Cuando un cliente compra una o varias unidades de un producto, la empresa registra una **venta**, el producto es reducido de las existencias en bodega y pasa a manos del cliente. Si el producto es un éxito comercial, su demanda se mantendrá y la empresa fabricará más unidades, lo que se registra con un evento de **producción**.



Un producto podría regresar a la empresa en caso de **fallo** dentro del periodo de garantía. En tal caso, el producto se recibe en taller donde es valorado. Si el producto está en buenas condiciones o es efectivamente **reparado**, es regresado al cliente. Si el producto no puede ser reparado, se le considera pérdida total y es **desecho** para reciclaje. En tal caso, si en bodega aún hay existencias del producto, se le da otra unidad al cliente, de lo contrario, se le reembolsa el dinero.

Se tiene un histórico de eventos guardados en un archivo de texto con extensión **.csv**. En este archivo, cada línea registra un evento con los siguientes campos: el código de producto, el tipo de evento (nuevo, producción, venta, fallo, reparado, o desecho), y la cantidad de unidades involucradas de producto. Sólo para el caso de un producto nuevo, se agrega un campo más con la descripción del producto. Estos campos se separan uno del otro por un carácter de coma (,). A continuación, se muestra un trozo de un archivo llamado **productos.csv**.

```
1,nuevo,50,dron espía
2,nuevo,100,parlante impermeable
2,venta,80
3,nuevo,25,proyector portátil
2,fallo,70
1,venta,50
1,fallo,3
1,desecho,1
1,reparado,2
3,venta,22
1,producción,200
2,desecho,55
1,venta,160
```

Se quiere un programa en Python3 que pueda responder ciertas preguntas importantes que la gerencia tiene sobre los productos, en especial (1) ¿Cuál es el estado actual de todos los productos? (2) Entre tantos productos ¿a cuáles debe la gerencia prestar atención urgente?

Al iniciar, el programa deberá cargar un estado de los productos a partir del archivo de eventos, que le permitirá responder las consultas a la gerencia. 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. Conforme el

programa carga eventos del archivo, irá creando una colección de objetos `Producto`, y los actualizará de acuerdo a los eventos. Una vez terminada la carga, mostrará el menú de consultas.

El menú tiene tres opciones: (E) estado de los productos, (U) productos urgentes, y (S) salir del programa. La opción (E) muestra el estado de todos los productos, uno por línea y en orden secuencial de código. El estado de un producto consta de: su código, su descripción, y la cantidad de unidades de ese producto en cada estado, todos estos datos separados por comas. La opción (U) muestra sólo aquellos productos que se encuentran en una condición que debe recibir atención de la gerencia que es al menos una de las siguientes:

1. Productos con menos de 2% de sus unidades en bodega, dado que si hay demanda la empresa perderá ventas o no habrá existencias para reemplazar unidades desechadas.
2. Productos con menos del 80% de sus unidades vendidas a los clientes.
3. Productos que fueron desechados y tuvo que reintegrarse dinero por no haber unidades en bodega.

La siguiente es una posible ejecución del programa, donde los valores en negrita son entradas del usuario.

```
Archivo de inventario: registro
Error: No se puede abrir registro.csv.
Archivo de inventario: Eventos.csv
Error: No se puede abrir Eventos.csv.
Archivo de inventario: productos
Se ha cargado productos.csv al inventario.
```

Opción [E=Estado U=Urgentes S=Salir]: **E**

```
1: dron espía, bodega=40, cliente=209, taller=0, reciclaje=1, total=250
2: parlante impermeable, bodega=0, cliente=30, taller=15, reciclaje=55, total=100
3: proyector portátil, bodega=3, cliente=22, taller=0, reciclaje=0, total=25
```

Opción [E=Estado U=Urgentes S=Salir]: **U**

```
1: dron espía, 1 reintegro(s) de dinero
2: parlante impermeable, 0% en bodega, 30% en clientes, 25 reintegro(s) de dinero
```

Opción [E=Estado U=Urgentes S=Salir]: **S**

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 cantidad de unidades en: bodega, clientes, taller, reciclaje, y cantidad de reintegros de dinero. Todo producto debe cargarse inicialmente del archivo, o de los datos que provienen de una línea del archivo, correspondiente al evento de un nuevo producto.
2. **[20%]** (1) *Producir*: recibe la cantidad de unidades producidas y las agrega a las existencias en bodega. (2) *Vender*: recibe la cantidad de unidades vendidas que son reducidas de la bodega y trasladadas a los clientes. (3) *Fallar*: recibe la cantidad de unidades que pasan de los clientes al taller. (4) *Reparar*: recibe la cantidad de unidades que regresan del taller a los clientes.
3. **[10%]** *Desear*: Recibe la cantidad de unidades del producto que no se pudieron reparar y son descontadas del taller. Si hay unidades en bodega, son trasladadas a los clientes como reemplazo. Si las unidades en bodega se acaban, se reintegra el dinero al cliente por las que faltan.
4. **[5%]** Retornar el estado del producto como un texto: "<código>: <descripción>, bodega=<cantidad>, cliente=<cantidad>, taller=<cantidad>, reciclaje=<cantidad>, total=<cantidad>".
5. **[10%]** Retornar un texto con las atenciones urgentes que deben realizarse al producto, separadas por comas. Si no tiene urgencias, retorna una cadena vacía.

Clase Bitácora (de control), con las siguientes responsabilidades:

6. **[10%]** La bitácora tiene de atributo único una colección de objetos tipo `Producto`, inicialmente vacía.
7. **[15%]** Solicitar el nombre de archivo, validarlo, cargar los datos del archivo, crear un producto cuando aparece un evento tipo nuevo, actualizar el producto de acuerdo al resto de sus eventos.
8. **[10%]** (1) Imprimir el estado de todos los productos. (2) Imprimir sólo los productos de atención urgente.
9. **[5%]** Mostrar repetitivamente el menú, e invocar a los métodos respectivos.

Función principal:

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