

**PRUEBA:** Programación.  
**Evaluación:** Tema 7 y tema 8  
**Fecha:** 08/05/2024.

*Nombre y Apellidos:*

---

*Grupo:* Mañana

---

Normativa:

1. Entregar un fichero zip con el nombre **1ºAPELLIDO 2ºAPELLIDO NOMBRE.zip** donde irá el proyecto exportado.
2. Si el proyecto no se importa correctamente la nota será un 0.
3. Si el proyecto no compila la nota será un 0.
4. Que el programa del resultado correcto no implica tener la máxima nota.

Se desea implementar una aplicación que actualice el inventario de componentes hardware de una tienda. El cliente nos comenta lo siguiente:

### Primer fichero

La tienda a través de la TPV genera un archivo llamado **ventas.dat** para almacenar las ventas en tiempo real que el negocio genera. **Los datos se encuentran seguidos, no es una línea por venta.**

Dicho fichero tiene el siguiente formato:

*ID**Fecha\_y\_hora\_de\_venta**Componente**Modelo**Cantidad\_vendida**Precio\_unitario**Precio\_total\_de\_venta*

A continuación se muestra un ejemplo:

*12024-04-30 10:30:49**Placa base**ASUS ROG Strix Z590-E Gaming**43001200*  
*22024-05-01 11:59:22**Disco SSD**Samsung 970 EVO Plus 1TB**101501500*  
*32024-05-02 12:35:31**Memoria RAM**Corsair Vengeance LPX 16GB DDR4**580400*  
*42024-05-03 22:33:15**Disco SSD**Samsung 970 EVO Plus 1TB**2150300*  
*52024-05-04 00:54:11**Procesador**Intel Core i7-10700K**33501050*

El cliente nos comenta que la TPV registra la venta en el fichero a través de los valores **de sus tipos en Java**. Los tipos de cada campo son:

- ID: int
- Fecha\_y\_hora\_de\_venta: String
- Componente: String
- Modelo: String
- Cantidad\_vendida: int
- Precio\_unitario: double
- Precio\_total\_de\_venta: double

### Segundo fichero

El cliente nos comenta que para gestionar el inventario de la tienda lo tienen que hacer manualmente, de forma que al final del día un empleado ve que han vendido y modifica el **inventario.dat** actualizando el inventario.

Dicho fichero tiene el siguiente formato:

*Componente;Modelo;Cantidad\_en\_stock;Precio\_unitario;Proveedor*

Hay que tener en cuenta lo siguiente:

- La primera línea del fichero se encuentra la última actualización del inventario que se hizo.
- Cada modelo de un componente aparece una única vez
- Una línea por componente,

A continuación se muestra un ejemplo:

*Ultima actualización: 2024-05-04 00:54:11*

*Procesador;Intel Core i7-10700K;15;350;Distribuidora de Hardware S.A.*

*Tarjeta gráfica;NVIDIA GeForce RTX 3080;8;800;Tech World Inc.*

*Memoria RAM;Corsair Vengeance LPX 16GB DDR4;20;80;Componentes Informáticos Eficaces*

*Disco SSD;Samsung 970 EVO Plus 1TB;12;150;Almacenamiento Rápido Ltda.*

*Placa base;ASUS ROG Strix Z590-E Gaming;10;300;Componentes de Alto Rendimiento S.L.*

### **Tercer fichero**

Tras la ejecución del software que vamos a desarrollar para el cliente se debe crear (si no existe) o modificar un fichero llamado **errores.log** donde se almacenan los errores que se van produciendo actualizando el fichero de inventario cuando el vendedor ejecute nuestra aplicación.

Al leer del fichero **ventas.dat** se deben realizar las siguientes validaciones:

- Cada línea del fichero debe tener todos los campos (7)
- Fecha\_y\_hora\_de\_venta: no puede ser vacío, debe cumplir el formato yyyy-mm-dd HH:mm:ss
- Componente: no puede ser vacío, sus valores posibles son: Procesador, Placa base, Disco SSD, Tarjeta gráfica, Memoria RAM, Fuente de alimentación, Disco HDD
- Modelo: no puede ser vacío, longitud entre 1 y 50 caracteres.
- Cantidad\_vendida: no puede ser vacío, debe ser un número entero mayor que 1.
- Precio\_unitario: no puede ser vacío, debe ser un número decimal mayor que 0,01
- Precio\_total\_de\_venta: no puede ser vacío, debe ser un número decimal mayor que 0,01

**Si alguno de las validaciones no se cumple no se tendrá en cuenta dicha venta para la actualización del inventario.**

Al actualizar el fichero **inventario.dat** se deben realizar las siguientes validaciones para los registros a actualizar:

- Cantidad\_en\_stock: si tras actualizar el inventario no queda stock o es negativo se deberá de avisar en el fichero de logs.

Deberá de escribirse una línea por cada error generado tanto al leer del fichero **ventas.dat** como al actualizar el fichero **inventario.dat**.

Dicho fichero tiene el siguiente formato:

*ID:xxx → Campo XXXXXXXXXX incorrecto. Valor: xxxxxx*

Donde el ID es el id de la venta, campo es campo que no ha pasado la validación y valor el valor erróneo.

Por ejemplo:

*ID:23 → Campo Fecha\_y\_hora\_de\_venta incorrecta. Valor: 01-12-2024*

### **Se pide**

- 1) Crear un programa que reciba **dos parámetros de entrada**, un primer parámetro con la ruta del fichero **ventas.dat** y un segundo parámetro con la ruta del fichero **inventario.dat**. Si alguno de los ficheros no existe se deberá mostrar un mensaje de error a través de la clase **ErrorFicheroNoEncontrado**.
- 2) El programa deberá de actualizar el inventario a través del fichero de **ventas.dat** a partir de la última actualización y hasta la fecha de la última venta en el momento que se ejecuta. Además, deberá actualizar la fecha del fichero **inventario.dat** para que cuando se vuelva a ejecutar siga actualizando por donde iba.
- 3) El programa deberá de generar y/o modificar el fichero **errores.log** con todos los errores detectados cuando se ejecute. **No deberá de borrar los errores anteriores.**

- 4) Cuando finalice deberá de mostrar un mensaje por consola de que el programa ha terminado de actualizar el inventario con el número de productos que se han actualizado correctamente y cuantos productos han dado error.