**EJERCICIO**

En una tienda de venta de productos tenemos un fichero con los datos de los productos, y otro fichero con los datos de la ventas de esos productos. Los ficheros son los siguientes:

**Productos.dat**. Fichero **aleatorio** con datos de productos, cada registro representa un producto, no existen productos con el mismo código. El formato es el siguiente:

int **codigoPro**; //campo identificativo – código del producto – debe ser de 1 a 99

String nombrePro; //nombre de producto, 15 caracteres, se graba sin writeUTF

int existencias; //existencias del producto

double pvp; //precio del producto

(46 bytes)

El código de producto **codigoPro** puede variar de 1 a 99, mayor o igual a 100 no se permite.

El campo clave para acceder a cada registro es **codigoPro,** que se identifica con la posición en el fichero.

En el fichero puede haber huecos pues se ha grabado de forma aleatoria. Los datos iniciales para este fichero son estos:

CODIGO NOMBRE EXISTENCIAS PVP

---------- -------------------- ---------- ----------

4 Mesas 10 90,50

8 Cuadernos 300 3,50

9 Bolígrafos 100 1,50

13 Lapiceros 10 500 3,50

14 Portátil 22 250,20

15 Pupitres 20 150,70

17 Tablet Acer 20 225,50

18 Tabletas 20 200,00

21 Estuches 250 10,50

24 Tarjetero 50 2,50

29 Funda Portatil 10 10,50

33 Libro BD 20 24,50

42 Teclado 80 40 15,50

54 Espejo baño 10 50,50

62 Folios A4-100 20 2,50

74 Tijeras 15 2,50

83 Reloj cocina 10 19,70

87 Impresora AT 25 100,60

90 Cubiertos 230 15,50

Disponemos también del fichero **DatosdeVentas.dat,** que contiene las ventas de los artículos. La información de este fichero **está grabada de manera secuencial y el registro es de longitud fija.** La cadena se ha grabado sin utilizar writeUTF.

El fichero contiene las ventas realizadas por los productos, el producto se puede vender varias veces, con lo que puede aparecer repetido el codigoPro. El formato del registro es el siguiente:

int **codigoPro**; // código del producto

int uniVen; //Unidades vendidas del producto

String fecha; //fecha de la venta, 10 caracteres.

En este fichero podemos encontrarnos **códigos de producto erróneos**, es decir que no existen en el fichero **Productos.dat,** pueden ser huecos, puede ser que no existan (por ejemplo el 95 y 98) y puede ser que el código es mayor de 99.

Los datos iniciales para este fichero son estos:

|  |
| --- |
| Codigo UNI VEN FECHA  ------ ------- -------  2 10 12/01/2019  6 15 15/01/2019  8 100 15/01/2019  98 1 16/01/2019  83 3 14/01/2019  13 30 15/01/2019  74 25 12/01/2019  24 10 15/01/2019  4 14 12/01/2019  17 5 17/01/2019  8 5 16/01/2019  21 5 18/01/2019  42 100 18/01/2019  87 1 16/01/2019  95 3 19/01/2019  24 30 19/01/2019  13 30 19/01/2019  14 25 16/01/2019  21 10 19/01/2019  62 25 19/01/2019  17 5 20/01/2019  9 50 22/01/2019  4 10 25/01/2019  49 5 25/01/2019  42 10 25/01/2019  83 1 26/01/2019  100 4 27/01/2019  90 4 01/02/2019  83 4 02/02/2019  120 10 02/02/2019  87 4 03/02/2019  4 1 03/02/2019 |

**EJERCICIO 1**

1 – Realizar un método para actualizar las **existencias** del fichero ***Productos.dat.*** La actualización consistirá en **restar a las existencias del producto, las unidades vendidas** que aparecen en **DatosdeVentas.dat**.

El proceso consistirá en ir leyendo de forma secuencial las ventas de **DatosdeVentas.dat**, e ir actualizando las existencias del producto leído en la venta, en ***Productos.dat.***

Tras una primera ejecución ***Productos.dat*** actualizado contendrá lo siguiente:

CODIGO NOMBRE EXISTENCIAS PVP

---------- -------------------- ---------- ----------

4 Mesas -15 90,50

8 Cuadernos 195 3,50

9 Bolígrafos 50 1,50

13 Lapiceros 10 440 3,50

14 Portátil -3 250,20

15 Pupitres 20 150,70

17 Tablet Acer 10 225,50

18 Tabletas 20 200,00

21 Estuches 235 10,50

24 Tarjetero 10 2,50

29 Funda Portatil 10 10,50

33 Libro BD 20 24,50

42 Teclado 80 -70 15,50

54 Espejo baño 10 50,50

62 Folios A4-100 -5 2,50

74 Tijeras -10 2,50

83 Reloj cocina 2 19,70

87 Impresora AT 20 100,60

90 Cubiertos 226 15,50

* Leer correctamente los registros de ventas visualizando el registro leído con su código, unidades vendidas y fecha,
* Comprobar los tres casos de error: que existe el producto de la venta en ***Productos.dat,*** que el producto no existe, que el producto es un hueco, o que el producto sobrepasa 99, visualizando el código de producto y el mensaje de error correspondiente.

**LOS FICHEROS SE TIENEN QUE LEER, NO SE ALMACENAN EN LISTAS (VECTORES , SET ARRAYS, etc.) PARA SU PROCESO**

**EJERCICIO 2**

2 – Utilizando la API JAXB, realizar un método para obtener el fichero ***Productos.XML,*** con el siguiente formato:

<productos>

<producto>

<codigo> </codigo>

<nombre> </nombre>

<existencias> </existencias>

<precio> </precio>

<unidadesvendidas> </unidadesvendidas>

<importe> </importe>

<estado> </estado>

</producto>

<producto>

<codigo> </codigo>

<nombre> </nombre>

<existencias> </existencias>

<precio> </precio>

<unidadesvendidas> </unidadesvendidas>

<importe> </importe>

<estado> </estado>

</producto>

. . . . . . . . . .

</productos>

**DONDE:**

Las ***unidadesvendidas*** es la suma de las unidades vendidas de ese producto.

El ***importe*** será la suma de las unidades vendidas del producto \* el precio del producto.

El ***estado*** indicará si hay que reponer stock del artículo, aparecerá ***A reponer*** si es necesario reponer, o vacío si no lo es. El stock se calcula con las **existencias – la suma de las unidades vendidas**, si el stock es menor de 2 debe aparecer A reponer, si no lo es, no debe de aparecer.

El ejercicio se evalúa con 5 puntos:

* Crear las clases del documento XML, 1 punto
* Calcular ***unidadesvendidas***, ***importe*** y ***estado*** de cada producto y visualizarlos junto con el id del producto: 2,5 puntos
* Cargar los datos en los objetos correspondientes a las clases del XML y obtener el fichero como se pide, 1,5 puntos

**El ejercicio se evalúa con 0 puntos si se entrega con errores de compilación, o si salen errores de ejecución. Controlar las excepciones que surjan.**

**GENERAR OTRO DOCUMENTO XML ejercicio2bis.**

<productos>

<producto>

<codigo> </codigo>

<nombre> </nombre>

<existencias> </existencias>

<precio> </precio>

<listaventas>

<venta>

<unidadesvendidas> </unidadesvendidas>

<fecha> </importe>

</venta>

<venta>

<unidadesvendidas> </unidadesvendidas>

<fecha> </importe>

</venta>

<venta>

<unidadesvendidas> </unidadesvendidas>

<fecha> </importe>

</venta>

. . . . . . . . . . .. .

</producto>

<producto>

<codigo> </codigo>

<nombre> </nombre>

<existencias> </existencias>

<precio> </precio>

<listaventas>

<venta>

<unidadesvendidas> </unidadesvendidas>

<fecha> </importe>

</venta>

<venta>

<unidadesvendidas> </unidadesvendidas>

<fecha> </importe>

</venta>

<venta>

<unidadesvendidas> </unidadesvendidas>

<fecha> </importe>

</venta>

. . . . . . . . . . .. .

</producto>

. . . . . . . . . .

</productos>