

01 JAVA.IO- PRÁCTICA FINAL.

FICHEROS ACCESO DIRECTO, SECUENCIAL

Se tiene una compleja aplicación para la **gestión de un vivero**. En esta aplicación se juntan varios tipos de ficheros para gestionar distintas cosas. Por un lado se tiene **un fichero de caracteres** con la siguiente forma:

```
<?xml version="1.0" encoding="UTF-8"?>
<plantas>
  <planta>
    <codigo>1</codigo>
    <nombre>captus</nombre>
    <foto>X</foto>
    <descripcion>Ninguna</descripcion>
  </planta>
  <planta>
    <codigo>2</codigo>
    <nombre>Geranio</nombre>
    <foto>X</foto>
    <descripcion>Ninguna</descripcion>
  </planta>
</plantas>
```

En este fichero cuyo **nombre es plantas.xml** se guardan los datos de todas las plantas del vivero que se muestran en la página web, estos datos **se cargan al inicio de la aplicación** y se manejan en una **estructura de datos durante la ejecución del proyecto**. Pero la empresa **guarda otros datos** sobre las plantas en un fichero de acceso directo. El **archivo de acceso directo se llama plantas.dat** y se accede directamente a los datos sin almacenarlos, a través de su código. Éste código coincide con el código leído en el documento xml y permite relacionar los dos ficheros a través de ese campo. Los campos que contiene este fichero son:

- codigo:int
- precio:float
- stock:int.

Se conoce que el código de las plantas que están colocadas de forma ordenada y secuencial se puede acceder a esos datos rápidamente. Pero al ser un fichero de acceso directo, para dar de baja un campo se podrá los campos precio y cantidad a cero. Y esos datos al darse de baja se copiará en otro fichers de acceso secuencial para guardarlos en el caso de recuperar el stock, los datos borrados no están ordenados.

Además de las plantas que son vendidas en el vivero. Se debe tener en cuenta que éstas son vendidas por los empleados. Los datos de los empleados se guardan en un fichero binario que guarda un array de objetos del tipo empleado. La aplicación carga todos los empleados al iniciarse.

El objeto tipo de empleado tendrá los siguientes campos:

- identificación: int → Se genera de forma aleatoria mediante una función que genera un número entero de 4 dígitos.
- nombre: String
- contraseña (entre 5 y 7 caracteres) : String → Debe cumplir las normas de seguridad
- cargo: String (solo hay dos cargos vendedores y gestores) → Se debe comprobar que lo se puede introducir otros cargos

Al iniciar la aplicación el sistema pedirá al empleado que introduzca su número de identificación y la contraseña. Solo en el caso de ser válido se podrá utilizar la aplicación.

El menú de los usuarios dependerá del cargo, según este cargará un menú u otro. Estas son las opciones que ven cada uno de ellos:

- Vendedores
 - Visualizar catálogo de plantas (primer fichero) junto al precio y a la cantidad de stock (segundo fichero)
 - Generar ventas: Esta opción se puede acceder desde el catálogo, que existirá el concepto cesta de la compra (ArrayList) donde se selecciona la planta y la cantidad o desde la opción de venta que habrá que introducir el código de la planta y la cantidad. Antes de

confirmar la compra, hay que aceptar que se va a realizar y saldrá el resumen antes de imprimirlo (generando el ticket)

- **Genera un ticket de venta** con los siguientes datos en un fichero de caracteres que tiene como un nombre un número secuencial de las ventas. Hay que tener en cuenta que cuando se cierra la aplicación puede haber tickets

- Datos

- Código empleado
- Nombre del empleado
- Fecha de venta
- CódigoProducto, unidades, cantidad
- Total

El stock se decrementa cuando se formaliza la venta, es decir cuando se hace el ticket

El usuario tiene que confirmar en la visualización del ticket que tiene que confirmar la compra

- Cuando se realiza una venta hay que calcular el total y decrementar la cantidad del stock de las plantas.
- Los tickets se almacenan en una carpeta que se llama TICKETS

Poner el precio en negativo

- **Generar devolución**

- **Marcar el ticket en negativo** y con la línea devuelto al final
- Sumar el stock

- Buscar por el número de ticket
- Automáticamente una planta se da de baja si el stock es cero y no se puede realizar ninguna compra sobre ella.

- Gestores

- Dar de alta las plantas, se almacena el array y al final se carga en el fichero (Reescribir el archivo XML)
- Dar de baja plantas, poner los campos a cero
 - Las plantas dadas de baja se tiene que guardar los datos existentes para poder recuperar el dato en caso de que vuelva a ver stock
- Modificar campos de las plantas en ambos archivos.
 - Incluye modificar el stock de las plantas
- Dar de alta los empleados SOLO se tendrá los empleados que se encuentren en activo.
- Dar de baja empleados estos se guardan en una carpeta que se llama BAJA
- Recuperar empleado de BAJA o recontratar
- Estadísticas (datos extraídos de los tickets)

- Total recaudado
- Planta que se vende más
- Los tickets devueltos se mueven a una carpeta distinta que se llama DEVOLUCIONES

La jerarquía de los ficheros será la siguiente

- PLANTAS
 - plantas.xml
 - plantasBaja.xml
 - plantas.dat
 - plantasbaja.dat
- EMPLEADOS
 - empleados.dat
 - BAJA
 - empleadosBaja.dat
- TICKETS
 - 1.txt
 - 3.txt
 - 4.txt
- DEVOLUCIONES
 - 2.txt

El modelo del ticket es el siguiente:

Número Ticket:123

-----//-----

Empleado que ha atendido: 1

Nombre del empleado: Oriana

CodigoProducto	Cantidad	PrecioUnitario
1	2	1
2	1	2

-----//-----

Total: 4 €

Se sobrescribe el plantas.xml con plantas.dat al final del archivo para que las plantas con 0 stock no salgan, se hace al final del archivo

El ticket tiene una variable que se inicializa a 0, cuando se crean se van incrementando, pero si cierro y abro la aplicación el primer ticket no sería 0, sería el último ticket creado, por ejemplo 9, entonces el siguiente sería el 10