

Práctica: aplicación para gestionar la cosecha de un agricultor

Enunciado práctica 1:

Dado el exponencial incremento de recursos electrónicos se propone la creación personal de un gestor de alimentos para que los agricultores tengan presente la cosecha recolectada durante la campaña.

Se propone centralizar todos los alimentos recogidos en una única interfaz y que se refresque en función del día en el que se ejecuta. Para ello, se quiere recoger la información referente a la cantidad de kilos que posee cada usuario de su producto en la cosecha personal. La aplicación debe permitir un registro de todos los alimentos y guardará:

- Id del alimento.
- Nombre del alimento.
- Valor diario.
- Cosecha máxima del cereal.

El valor de cada alimento fluctúa diariamente debido a la oferta y la demanda según la siguiente tabla:

Alimento	Precio fijo /kg	Precio variable /kg	Descuentos	Cosecha máxima
Garbanzos	500 pts	+20 pts [SAB]	-10% JA y +10% HO	10 toneladas
Lentejas	300 pts	-50 pts [MAR y JUE]	-50% JU y -15% MU	50 toneladas
Patatas	100 pts	-10 pts [LUN]	-25% JA y -10% JU	90 toneladas
Cebollas	150 pts	+5 pts [VIER y SAB]	-30% JU	20 toneladas

donde:

- JA – Joven Agricultor, -35 años
- JU – Jubilado, +67 años
- MU – mujeres
- HO – Hombres
-

La aplicación guardará los datos personales del usuario:

- ID de usuario.
- Nombre.
- Apellidos.
- Sexo (un *enum* que puede ser “Masculino” o “Femenino”)
- Fecha de nacimiento.
- Fecha de registro.

La aplicación deberá manejar para cada agricultor la cantidad de alimentos que posee, la fecha en la que el usuario obtuvo su primer activo y un histórico de transacciones.

El inicio de la aplicación para un usuario ejemplo sería la siguiente:

Usuario	Sexo	Edad	Nombre	Cantidad [kg]	Precio [EUR]	Máximo [kg]
Fran	H	30	Garbanzos	200	600	10.000
Fran	H	30	Lentejas	25.000	50.000	50.000
Fran	H	30	Patatas	10.000	10.000	90.000
Fran	H	30	Cebollas	1.000	1.200	10.000

La aplicación debe permitir realizar acciones con usuarios, como la creación, modificación y eliminación de usuarios, el ingreso (por minado, es decir, sin realizar una transacción) de los diferentes tipos de alimentos y la realización de transacciones (ventas o compras de alimentos para cada usuario).

Todos los datos que se pidan al usuario deberán ser validados (excepciones). Todas las clases deben contener conjuntos de *tests* en el *main* de cada módulo. La información de los alimentos se almacenará en un fichero *csv* que puede estar fijado por el programador y la información de los usuarios en otro fichero, que debe contener la información correspondiente a si tiene abierta una cuenta de cada tipo de alimento y la cantidad que tiene (no es lo mismo tener 0 unidades que no tener una cuenta) y el histórico de transacciones, que debe guardarse en un fichero diferente para cada usuario.

La aplicación deberá cargar la información de los activos y de los usuarios al iniciarse desde los ficheros *csv* correspondientes. La estructura del fichero debe ser definida por el estudiante, y se recomienda consensuarla antes con el profesor. Posteriormente deberá tener un menú principal con las siguientes opciones:

1. Listar la información diaria de todos los tipos de alimentos.
2. Listar todos los usuarios y sus IDs.
3. Cargar un usuario en memoria (por defecto o pidiendo el nombre al usuario). Para ello se debe mostrar los nombres de usuario y su referencia y el usuario introducirá la referencia de su usuario para cargarlo en memoria. Cargar un usuario en memoria implica que se carguen en las estructuras de datos correspondientes los datos del usuario solicitado desde el fichero *.csv* (o la estructura de datos donde se han cargado todos los datos de todos los usuarios) donde se encuentra la información. La estructura del fichero debe ser definida por el estudiante, y se recomienda consensuarla antes con el profesor.
4. Crear un usuario nuevo, indicando todos sus datos y el primer ingreso de un alimento y su valor.
5. Añadir un nuevo usuario indicando su nombre, apellidos y fecha de nacimiento, regalando al usuario una cantidad de 10 euros en patatas. La cantidad de patatas puede variar en función del valor en de cada momento.
6. Realizar una compra de un alimento existente y actualizar su valor en el *csv*.
7. Realizar una venta de una cantidad de kg de un activo.
8. Realizar una venta/compra de un usuario origen a un usuario destino. Se deben listar los usuarios al seleccionar esta opción.
9. Modificar la información del usuario (todos los campos a excepción del documento de identidad, el ID de usuario y la fecha de registro).
10. Ver información detallada de un usuario, con el número de unidades de cada activo y el valor en EUR de cada una de ellas en el momento de hacer la consulta, así como el histórico de ventas/compras de las últimas 10.

11. Guardar la información actual en memoria de todos los alimentos, los usuarios y la información de cada usuario en el fichero *csv* correspondiente donde se guarda toda la información de la aplicación.

Cuando el usuario seleccione la opción 9, o 10 deberá pedírsele al usuario el ID del usuario a modificar y se deberán listar todos los nombres e IDs.

Consejos

Habla con el profesor para determinar cuál es la mejor forma para representar y almacenar los datos, y de estructurar la práctica. Organiza la práctica en al menos dos paquetes. Uno se encargará de la interfaz de usuario, mientras que el otro se encargará de cargar y guardar datos en el disco duro, añadir ratones, etc.... Utiliza abundantes funciones de pequeño tamaño.

Sobre la entrega de la práctica

La práctica debe ser entregada en formato electrónico antes del 12 de noviembre (inclusive). Por cada semana que se retrase la entrega de la práctica a partir de esta fecha, la nota final de la práctica se le descontarán 2 puntos, con un máximo de 2 semanas de retraso. La práctica debe realizarse a través de una consola.

Junto con la práctica, el alumno deberá entregar un listado de los fallos que existen en la práctica, y/o la funcionalidad que no le ha dado tiempo de implementar. A la hora de calificar la práctica, se penalizarán menos la existencia de un fallo/carencia de funcionalidad conocido por el alumno (y que por tanto forma parte del listado de fallos), pero que no le ha dado tiempo a resolver, que un fallo/carencia de funcionalidad desconocido por el alumno (y que por tanto no forma parte del listado de fallos).

El alumno tendrá que realizar una defensa de la práctica; si el alumno es incapaz de defender la práctica y explicar su funcionamiento, su nota en la práctica será un 0.