Descripción general: La aplicación nos permite gestionar el stock de productos de la cadena de comida rápida BurguerDonald. Podremos insertar, eliminar y modificar productos o menús.

Un menú no es más que una agrupación de productos bajo un mismo nombre. También podremos realizar búsquedas de productos por tipos diferentes y por productos veganos.

Los lunes el precio del café pasa a ser de 0€. Esto se realiza de manera automática.

Clases utilizadas en la aplicación:

**ImplProducto**

* Id entero, consultable y modificable.
* Nombre Cadena, consultable y modificable.
* Tipo EnumTipo, consultable y modificable.
* Precio real, consultable y modificable.
* Descripción Cadena, consultable y modificable.
* Vegano booleano, consultable y modificable.

Requisitos:

* id debe ser mayor o igual que 0.
* Tipo debe ser: BEBIDA, POSTRE, SANDWICH, COMPLEMENTO, ENSALADA o DESAYUNO.
* Precio debe ser mayor o igual a 0.

**EnumTipo** puede tener uno de los siguientes valores:

* Bebida
* Postre
* Sándwich
* Complemento
* Ensalada
* Desayuno

**ImplStockProducto**

* Producto ImplProducto, consultable y modificable.
* Stock entero, consultable y modificable.

Requisitos:

-stock debe ser mayor o igual que 0.

**ImplMenú**

* Id entero, consultable y modificable.
* Nombre Cadena, consultable y modificable.
* Productos int[], consultable y modificable.
* Descripción Cadena, consultable y modificable.
* Precio real, consultable y modificable.

Requisitos:

* ID debe ser igual o mayor a 0.
* int[] productos no debe contener ningún elemento nulo.
* Precio de ser mayor o igual a 0.

Los errores por inserción, eliminación o modificación no se guardan. Se muestra en pantalla un mensaje explicativo.

**Criterio de ordenación:**

Cuando el usuario desee sincronizar los datos de los archivos, el fichero de movimiento es ordenado de forma ascendente respecto a su ID.

**Criterio de sincronización:**

Se ordenará el fichero de movimientos, de la manera especificada anteriormente.

Leemos un registro de cada fichero.

Mientras ninguno de los dos alcance el fin de fichero.

Si el registro leído del maestro tiene una id menor, se inserta en un maestro actualizado.

Si el registro leído en el de movimientos tiene una id menor, se buscará el último movimiento de ese producto. Si no está marcado se inserta en el maestro actualizado.

Si ambos registros tienen la misma id, se buscará en el fichero de movimientos el último movimiento de ese producto, si no se encuentra marcado se inserta en el maestro actualizado.

Leemos nuevos registros de ambos ficheros.

Almacenamos los registros restantes del fichero inacabado.

Por último, vaciamos los ficheros maestro y de movimientos y renombramos el maestro actualizado a maestro.

**Funciones automatizadas:**

Los lunes el precio del café (con ID = 1) será de 0$. El resto de días tendrá su precio de venta normal: 1$. Ello en caso de existir en el almacén.

Todos los sábados entre las 13:00 y 14:00 se realizará una sincronización automática.

En caso de no existir algún fichero se creará automáticamente.

**Recursos externos:**

JLayer – Librería usada para reproducir sonidos en java.

Más información: <http://www.javazoom.net/javalayer/javalayer.html>