

Diseño y programación orientada a objetos - ELO-329

Primer semestre 2021 - Campus Casa Central - Campus San Joaquín

Información sobre la entrega

Debe subir a aula un solo archivo comprimido con el código que haya realizado. El nombre del archivo debe tener el siguiente formato: *ROL_NOMBRE_APELLIDO.zip*.

Certamen 1

Pregunta de Desarrollo

- Descargue el código de ayuda desde el siguiente repositorio gitlab: <https://gitlab.com/francisco.cabezas/elo-329-c1-example>

Dada la contingencia, se hace estrictamente necesario que hayan servicios que puedan permitir comercio electrónico.

Todo comercio electrónico dispone de un inventario de productos, cuyos atributos van desde el *SKU* o *id*, hasta el precio. Además, los usuarios del sitio pueden realizar búsqueda de productos y agregar sus productos favoritos a una lista personal, más bien conocida como *carrito de compras*.

El desafío del día de hoy será construir un servicio que permita realizar operaciones básicas de compra. Para ello, se ha dispuesto de un código inicial que permite leer un archivo de entrada en formato *csv*. Este archivo tiene 1000 registros, en donde cada fila tiene los siguientes atributos:

id (String)	timestamp (String)	category (String)	product (String)
brand (String)	size (String)	price (int)	availability (String)

En donde cada ítem tiene entre paréntesis el tipo de dato que contiene. Actualmente, el código de ejemplo puede leer el archivo, y guardar cada fila como un **String** dentro de un **ArrayList**.

Se pide lo siguiente:

- a) **(25 puntos)** Implemente la funcionalidad de búsqueda que permita buscar según lo que se exprese en los parámetros de entrada, de la siguiente forma:

Buscador por marca

```
> java InventoryManager -s -brand Parle
```

El ejemplo corresponde a una búsqueda (-s) por marca (-brand), para lo cual deberá buscar aquellos productos de la marca ingresada en el tercer atributo. Lo anterior deberá imprimir por pantalla los atributos *id*, *product*, *brand* y *price*, por ejemplo:

```
> Se encontró 1 registro:  
> Item Id: 8687c5317c6629f89fe62e0233563246, product: Parle Melody Chocolatey  
Toffee (195.5 g), brand: Parle, price: 50
```

En el caso de no encontrar registros, imprimir:

```
> No se han encontrado registros
```

Punto importante: Solo debe devolver aquellos productos con disponibilidad, es decir `availability = TRUE`

b) **(25 puntos)** Implemente funcionalidad para ordenar según precio (de Mayor a Menor):

```
> java InventoryManager -o
```

Los ejemplos corresponden a un ordenamiento (-o) según precios. La salida debería ser la siguiente:

```
> Item Id: 0633d9fd9a3271730fae687f105c7a3a, product: B Natural Dakshin Pink  
Guava (750 ml), brand: Parle, price: 95, availability: TRUE  
  
> Item Id: 8687c5317c6629f89fe62e0233563246, product: Parle Melody Chocolatey  
Toffee (195.5g), brand: Parle, price: 50, availability: TRUE  
  
...
```

c) **(20 puntos)** Implemente una vista usando `JavaFX`, en la cual exista un input de texto que actúe como buscador de marca, y un botón *Buscar*. Los resultados los devolverá por consola, en el mismo formato que a).

Hint

En IntelliJ también puede usar argumentos para arrancar su programa, vea la siguiente documentación: <https://users.drew.edu/bburd/JavaForDummies6/CommandLineIntelliJIDEA.pdf>