

<div style="display: flex; justify-content: space-between; align-items: center;"> <div> Universidad Tecnológica Nacional Facultad Regional Avellaneda </div>  </div>					
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos					
Materia: Laboratorio de Computación I					
Apellido:				Fecha:	
Nombre:				Docente (2):	
División:				Nota (2):	
Legajo:				Firma (2):	
Instancia (1):	PP		RPP		SP
					RSP
					FIN

Se dispone de un archivo con datos acerca de las compras de un importador, que tiene el siguiente formato:

id_articulo, descripción (del artículo), **tipo** (1-JUGUETERIA, 2-ELECTRONICA, 3-INDUMENTARIA, 4-CALZADO), **precioUnitario, cantidad, totalBulto**

por ejemplo:

50,Pelota,1,250,200,0
 51,Camisa azul hombre,3,860,250,0
 52,Notebook HP,2,450000,60,0
 53,Zapatillas Nike,4,15436,35,0

se deberá realizar un programa que permita el análisis de dicho archivo y sea capaz de generar nuevos archivos de salida de formato similar filtrados por varios criterios:

El programa contará con el siguiente menú:

- 1) Cargar archivo:** Se pedirá el nombre del archivo y se cargará en un linkedlist los elementos del mismo.
- 2) Imprimir lista:** Se imprimirá por pantalla la tabla con los datos de las compras.
- 3) Asignar totales:** Se deberá hacer uso de la función map la cual recibirá el linkedlist y una función que asignará a cada compra el totalBulto de la siguiente forma: cantidad x precioUnitario.
- 4) Filtrar por tipo:** Se deberá generar un archivo igual al original, pero donde solo aparezcan servicios del tipo seleccionado.
- 5) Mostrar compras:** Se deberá mostrar por pantalla un listado de las compras ordenadas por descripción de manera ascendente.
- 6) Guardar compras:** Se deberá guardar el listado del punto anterior en un archivo de texto.
- 7) Salir.**

Requerimientos del desarrollo. • Se deberá crear la entidad “eCompras” con todos sus campos correspondientes. • se deberá utilizar la biblioteca linkedlist para almacenar las compras leídas del archivo. • se deberá agregar a la biblioteca la función “ll_filter ()” la cual devolverá una nueva linkedlist que contenga alguno de los elementos de la lista original, según algún criterio.

Detalle de la función “ll_filter ()” prototipo de la función:

linkedlist* ll_filter (linkedlist* this, int (*pFunc) (void* element))

la función “ll_filter” recibirá una lista y una función “pFunc”. se deberá iterar todos los elementos de la lista y pasárselos a la función “pFunc”. la función “pFunc” devolverá 1 si ese ítem se debe agregar a la lista resultado o 0 si no debe agregarse. la función “ll_filter” generará la nueva lista resultado, agregará a la misma los ítems correspondientes y la devolverá.

Detalle de la función “ll_map ()” prototipo de la función:

```
linkedlist* ll_map (linkedlist* this, void*(*pFunc) (void* element))
```

la función “ll_map” recibirá una lista y una función “pFunc”. se deberán iterar todos los elementos de la lista y pasárselos a la función “pFunc” que recibirá el elemento y le calculará el campo totalBulto, el retorno de “pFunc” se agregará a la lista resultado. esta nueva lista será devuelta por ll_map.

nota: el código deberá tener comentarios con la documentación de cada una de las funciones y respetar las reglas de estilo de la cátedra.

nota bis: separar en distintas bibliotecas las entidades, parser y generador de informes.

Para la aprobación, se deberá realizar como mínimo el parseo del archivo, imprimir la lista, la función ll_map o ll_filter y el guardado en el archivo de texto.