



Universidad Tecnológica Nacional Facultad Regional Avellaneda

Técnico Universitario en Programación

Técnico Universitario en Sistemas
Informáticos

Materia: Laboratorio de computación I

Apellido:					Fecha:	26/11/2019				
Nombre:					Docente:					
División:					Nota:					
Legajo:					Firma:					
Instancia	PP		RPP		SP	X	RSP		FIN	

PARTE 2

Una empresa de Aerolíneas debe gestionar los diferentes vuelos que serán efectuados en los próximos días, para ello cuenta con un registro que presenta el siguiente formato:

idVuelo, idAvion, idPiloto, fecha, destino, cantPasajeros, horaDespegue, horaLlegada

Por ejemplo:

49, 3, 3, 28/11/2019, Portugal, 63, 16, 21

Los pilotos que trabajan en la Aerolínea son los siguientes:

ID	Nombre
1	Alex Lifeson
2	Richard Bach
3	John Kerry
4	Erwin Rommel
5	Stephen Coonts

Se deberá realizar un programa que permita el análisis de dicho archivo y sea capaz de generar un nuevo archivo de salida de formato similar en el cual solo sean incluidos los vuelos de un piloto en particular especificado por el usuario.

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 vuelos:** Se imprimirá por pantalla la tabla con los datos de los vuelos indicando el nombre del piloto (no el número)
 - 3) **Cantidad de pasajeros:** Cantidad de pasajeros totales
 - 4) **Cantidad de pasajeros a Irlanda:** cantidad de pasajeros a Irlanda.
 - 5) **Filtrar vuelos cortos:** Generar un archivo de iguales características que el original, con los datos de los vuelos cuya duración sea inferior a 3 horas.
 - 6) **Listar vuelos a Portugal:** Generar un listado que solo incluya los vuelos con destino a Portugal.
 - 7) **Filtrar a Alex Lifeson:** Necesitamos un listado de vuelos que no incluya al chofer Alex Lifeson
- Salir.

Detalle de la función "ll_count()"

Prototipo de la función:

int ll_count(LinkedList* this, int (*fn)(void* element))

La función "ll_count" recibirá una lista y una función "fn". Se deberá iterar todos los elementos de la lista y pasárselos a la función "fn". La función "fn" devolverá la cantidad que debe contarse. La función "ll_count" almacenará un acumulador al cual sumará el valor de retorno de "fn" en cada iteración. Al finalizar las iteraciones, la función "ll_count" devolverá el valor acumulado.

Requerimientos del desarrollo.

- Se deberá agregar a la biblioteca LinkedList la función "ll_filter()" la cual devolverá una nueva LinkedList que contenga alguno de los elementos de la lista original, según algún criterio
- Se deberá utilizar la función "ll_filter()" para los informes que lo requieran.

Detalle de la función "ll_filter()"

Prototipo de la función:

LinkedList* ll_filter(LinkedList* this, int (*fn)(void* element))

La función "ll_filter" recibirá una lista y una función "fn". Se deberá iterar todos los elementos de la lista y pasárselos a la función "fn". La función "fn" 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.

Nota 0: El código deberá tener comentarios con la documentación de cada una de las funciones y respetar las reglas de estilo de la cátedra.

Nota 1: Separar en archivos las entidades, parser y generador de informes.