

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación

Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de computación I

Apellido:					Fecha:	11/07/2019				
Nombre:					Docente:					
División:					Nota:					
Legajo:					Firma:					
Instancia	PP		RPP		SP		RSP	X	FIN	X

PARTE 1

Una empresa de soporte técnico telefónico debe analizar los llamados recibidos, los cuales se encuentran registrados en un archivo con el siguiente formato:

ID_Llamada, Fecha, Numero_Cliente, ID_Problema, Solucionado

Por ejemplo:

33,11/07/2019, 3591364,2, SI

34,11/07/2019, 5974941,5, NO

Los IDs de los problemas son:

1	No enciende PC
2	No funciona mouse
3	No funciona teclado
4	No hay internet
5	No funciona teléfono

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 incluidas las llamadas de un problema 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 Llamadas:** Se imprimirá por pantalla la tabla con los datos de las llamadas indicando el problema en forma de texto (no de número)
- 3) **Generar archivo de llamadas:** Se pedirá que se ingrese el número de problema (1 a 5), y se generará un archivo CSV con el mismo formato que el original, pero solo con las llamadas de dicho tipo de problema.
- 4) **Salir.**

Requerimientos del desarrollo.

- Se deberá crear la entidad “Llamada” con todos sus campos correspondientes.
- Se deberá utilizar la biblioteca LinkedList para almacenar las llamadas 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
- Se deberá utilizar la función “ll_filter()” para generar el archivo pedido.

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.

PARTE 2

Preguntas oral:

- Estructura de la LinkedList.
- Función filter.
- Otra función de la LinkedList.

CONDICIONES DE APROBACIÓN

Para la aprobación directa (nota ≥ 6), se deberá tener el programa funcionando en su totalidad como se pide en la parte 1 del examen y haber contestado todas las preguntas de la parte 2.

Para la aprobación con final (nota = 4 o 5), se deberá realizar el parseo del archivo, la función ll_filter, y haber contestado la pregunta oral de la estructura de la LinkedList.