

| | | | | | |
|--|-----|--|-----|--------------------------|----------------|
| <div style="display: flex; justify-content: space-between; align-items: center;"> <div> Universidad Tecnológica Nacional Facultad Regional Avellaneda </div>  </div> | | | | | |
| Técnico Universitario en Programación - Técnico Universitario en Sistemas Informáticos | | | | | |
| Materia: Laboratorio de Computación I | | | | | |
| Apellido: | | | | Fecha: | 2/12/2021 |
| Nombre: | | | | Docente ⁽²⁾ : | Baus/Scarafilo |
| División: | 1°G | | | Nota ⁽²⁾ : | |
| Legajo: | | | | Firma ⁽²⁾ : | |
| Instancia ⁽¹⁾ : | PP | | RPP | | SP |
| | | | | | RSP |
| | | | | | X |
| | | | | | FIN |

Se dispone de un archivo con datos obtenidos de una red social, que tiene el siguiente formato:
id, user, likes, dislikes, followers

por ejemplo

```
1,lmalletratt0,0,0,0
2,skilmurray1,0,0,0
3,vhowgego2,0,0,0
```

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 los posteos.
- 3) Asignar estadísticas:** Se deberá hacer uso de la función map. la cual recibirá el linkedlist y una función que asignará a cada post un valor de likes entre 500 y 3000, dislikes con valores entre 300 y 3500 y followers entre 10000 y 20000. calculados de manera aleatoria.
- 4) filtrar por mejores posteos:** Se deberá generar un archivo igual al original, pero donde solo aparezcan posteos con más de 5000 likes.
- 5) filtrar por heaters:** Se deberá generar un archivo igual al original, pero donde solo aparezcan posts donde la cantidad de dislikes supere a la de likes.
- 6) ordenar por nivel de cantidad de followers:** Se deberá mostrar por pantalla un listado de los posteos ordenados por cantidad de followers descendente.
- 7) mostrar más popular:** Informar el nombre del user o user con el posteo más likeado. Y cuál es ese número.
- 8) salir.**

Requerimientos del desarrollo. • Se deberá crear la entidad “ePost” con todos sus campos correspondientes. • se deberá utilizar la biblioteca linkedlist para almacenar los posts 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

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 asignará a los campos likes, dislikes y followers un número aleatorio. el retorno de “pFunc” que es el posteo con todos sus campos cargados se agregará a la lista resultado. esta nueva lista será devuelta por ll_map.

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.

Condiciones para la aprobación directa (nota ≥ 6), se deberá tener el programa funcionando en su totalidad.

Para la aprobación con final (nota = 4 ó 5), se deberá realizar el parseo del archivo, la función ll_map, y haber resuelto algún filtrado.