

Se dispone de un archivo con datos obtenidos de una red social, que tiene el siguiente formato:

id, user, likes, dislikes, followers

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

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 600 y 5000, 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 4000 likes.
- 5) **filtrar por haters:** 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.**

Detalle de la función “ll filter ()” prototipo de la función:

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.