

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

Se dispone de un archivo con datos acerca de los participantes de la carrera off-road más importante de Latinoamérica: el Enduro del Verano, que tiene el siguiente formato:

**id\_cuatri, nombre** (del dueño), **categoria** (MX1, MX2, MX3, SUPER\_ATV), **tiempo, promedio**

por ejemplo:      50, Warren, MX1,10,0  
                          51, Kania, SUPER\_ATV,15,0  
                          52, Alla, MX3,23,0

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 participantes.
- 3) Asignar promedios:** Se deberá hacer uso de la función map la cual recibirá el linkedlist y una función que asignará a cada participante el promedio calculado de la siguiente forma: para tiempos menor a 15 se le asignará un promedio de 6 segundos, para tiempos entre 15 y 20 se le asignará un promedio de 8 segundos y para tiempos mayores a 20 se le asignará un promedio de 10 segundos.
- 4) Filtrar por tipo:** Se deberá generar un archivo igual al original, pero donde solo aparezcan participantes de la categoría seleccionada.
- 5) Mostrar posiciones:** Se deberá mostrar por pantalla un listado de los participantes ordenados por categoría y dentro de la misma categoría que aparezcan ordenadas por promedio ascendente.
- 6) Guardar posiciones:** Se deberá guardar el listado del punto anterior en un archivo de texto.
- 7) Salir.**

**Requerimientos del desarrollo.** • Se deberá crear la entidad “eParticipantes” con todos sus campos correspondientes. • se deberá utilizar la biblioteca linkedlist para almacenar los participantes leídos 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 promedio, 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 archivos 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.