

# Recuperatorio 2do Parcial Laboratorio

## Antonio Aguirre 1°C

### Choferes

int	id	PK
char[]	nombre	
int	edad	
char[]	vehiculo	

### Viajes

int	id	PK
Fecha	fecha	
int	pasajeros	
int	idChofer	FK
int	kilometros	

### Fecha

(anidada)

int	dia
int	mes
int	anio

- 1) Filtro: Viajes a partir de julio 2021, con mas de 70km
- 2) Filtro: Choferes con auto de mas de 25 años.
- 3) Cantidad de kilometros realizados por Antonio
- 4) Promedio de edad de choferes con combi

Las dos primeras funciones utilizan la herramienta ll\_filter que devuelve una linked list, la cual crea recorriendo la linked list pasada como parámetro añadiendo elementos si cumplen el criterio puesto en la función (también pasada como parámetro).

```
int viajes_filtrarJulioKm(void* pViaje)
{
    int retorno = 0;
    Fecha fechaAux;
    int kilometrosAux;

    if(pViaje!=NULL)
    {
        viajes_getFecha(pViaje,&fechaAux);
        viajes_getKilometros(pViaje,&kilometrosAux);
        if(fechaAux.anio>=2021 &&
           fechaAux.mes>=7 &&
           kilometrosAux>70)
        {
            retorno = 1;
        }
    }

    return retorno;
}
```

```
int chofer_contarEdadCombi(void* pChofer)
{
    int retorno = 0;
    char vehiculo[101];

    if(pChofer!=NULL)
    {
        chofer_getVehiculo(pChofer,vehiculo);
        if(strcmp(vehiculo,"Combi")==0)
        {
            chofer_getEdad(pChofer,&retorno);
        }
    }

    return retorno;
}
```

Las ultimas dos utilizan la herramienta ll\_count que devuelve un número entero, este se determina recorriendo la linked list enviada como parámetro, acumulando según la función también parametrizada.

```
int viajes_countKmAntonio(void* pViaje)
{
    int retorno = 0;
    int idChofer;

    if(pViaje!=NULL)
    {
        viajes_getIdChofer(pViaje,&idChofer);
        if(idChofer==1)
        {
            viajes_getKilometros(pViaje,&retorno);
        }
    }

    return retorno;
}
```

```
int chofer_contarCombi(void* pChofer)
{
    int retorno = 0;
    char vehiculo[101];

    if(pChofer!=NULL)
    {
        chofer_getVehiculo(pChofer,vehiculo);
        if(strcmp(vehiculo,"Combi")==0)
        {
            retorno = 1;
        }
    }

    return retorno;
}
```

## Prototipos principales

```
int controller_loadChoferesCsv(LinkedList* listaChoferes);
```

```
int controller_loadViajesCsv(LinkedList* listaViajes);
```

Reciben la lista, la vacían, piden al usuario el nombre del archivo y cargan sus datos con la función parser.

Retorna (-1) si la lista es NULL o el parser falla

(0) si todo OK

```
int parser_choferesFromText(FILE*,LinkedList*);
```

```
int parser_viajesFromText(FILE*,LinkedList*);
```

Lee los datos del archivo enviado como parámetro, los almacena en memoria dinámica con la función newParametros y lo asigna a la linked list.

Retorna (-1) si la lista es NULL o no se pudo crear espacio en memoria dinámica

(0) si todo OK

```
int controller_listChoferes(LinkedList* listaChoferes);
```

```
int controller_listViajes(LinkedList* listaViajes,LinkedList* listaChoferes);
```

Recorren la linked list, obtienen el elemento y lo imprimen en una lista, en el caso de listViajes, llama a una función que busca el nombre del chofer según su ID y muestra solo el nombre.

Retorna (-1) Si la lista es NULL

(0) si todo OK

```
int controller_filtroUno(LinkedList* listaViajes,LinkedList*
listaFiltrada,LinkedList* listaChoferes);
```

```
int controller_filtroDos(LinkedList* listaFiltrada,LinkedList*
listaChoferes);
```

Cargan en la lista listaFiltrada, los elementos deseados utilizando la función ll\_filter y la muestran, preguntan al usuario si desea guardar dicha lista en un archivo .csv

Retorna (-1) si las listas son NULL

(0) si todo OK

```
int controller_saveFilter(LinkedList* listaFiltrada, int opcion);
```

Recibe la lista filtrada y la imprime en un nuevo archivo .csv, recibe un int que si es (1) guarda la lista de viajes y si es (0) guarda una lista de choferes

Retorna (-1) si no se pudo crear el archivo

(0) si todo OK

```
int controller_countKmAntonio(LinkedList* listaViajes);
```

Utiliza la función ll\_count para acumular en un int los kilómetros según el criterio de la función que recibe ll\_count, muestra el dato.

Retorna (-1) si la lista es NULL

(0) si todo OK

```
int controller_promCombiChoferes(LinkedList* listaChoferes);
```

Utiliza la función ll\_count dos veces, en una acumula y en otra cuenta, para sacar un promedio de edad y mostrarlo.

Retorna (-1) si la lista es NULL

(0) si todo OK

**Enlace a defensa en video**

<https://drive.google.com/file/d/1wQeQ1VgTYMmUI7gN2u7IADSHoyc0Lfqb/view?usp=sharing>

El trabajo está siendo entregado unos minutos tarde debido a una lenta conexión a internet, pero fue finalizado el día 21 antes de las 12, mil disculpas.