

# RECUPERATORIO PARCIAL 1

Cátedra Estructuras de Datos I

22 de Junio de 2017

## 1. Presentación del ejercicio

En el día de ayer se comunicó la ministra de Seguridad de la Nación, la Dra. Patricia Bullrich para organizar la asistencia al evento del día 20 de junio en la ciudad de Rosario. La idea es poder tener un control y un orden que permita saber los invitados que asistirán al evento nos comentó. De cada asistente se va a almacenar su dni, nombre y apellido, la descripción del cargo y una prioridad que debe ser usada para, de alguna manera, tener un cierto orden entre los invitados.



```
typedef struct {  
    char* dni, *nombre, *descripcionCargo;  
    int prioridad;  
} _Invitado;  
  
typedef _Invitado* Invitado;
```

Dado que se desconocía la cantidad de invitados que va a haber es que se nos ocurrió proponer una lista doblemente enlazada; a lo cual nos comentó que le parecía excelente.

```
typedef struct _Nodo {  
    Invitado invitado;  
    struct _Nodo* sig, *ant;  
} Nodo;  
  
typedef Nodo* Lista;
```

Nos pidió que se pueda ir agregando los invitados en forma ordenada según su prioridad y, como lo indica el protocolo, cuando se da por aprobado una prioridad de invitados determinada, que esos se invitados se eliminen de la lista y, los retorne en una nueva lista.

Para poder realizar esto pensamos en los siguientes prototipos para las funciones que se encargarían de realizar estas acciones:



```
Lista agregar(Invitado invitado, Lista listaDeInvitados)

Lista eliminar(Lista listaDeInvitados, int prioridadAprobada)
```

y, toda esta información la vamos a almacenar en un archivo con extensión .h.

Amablemente nos pidió que generemos una prueba de que funcionara el programa por lo que acordamos que íbamos a tener un main que verifique que:

- el retorno de eliminar es NULL cuando listaDeInvitados es NULL;
- el retorno de agregar es distinta de NULL sea listaDeInvitados NULL o no;

Luego, nos agradeció por resolver esto a la brevedad y, cortó la comunicación.

## 2. Evaluación

Se pide que escriba un código que pueda implementar las funciones solicitadas respetando los tipos definidos y los prototipos indicados anteriormente.

Se tomará como criterio de aprobación:

1. funcionamiento del programa;
2. claridad y eficiencia del código escrito;