

```

1  #ifndef LISTA_H_
2  #define LISTA_H_
3
4  /**
5      Definición del Tipo de Dato para manejo de Listas.
6      Atributos:
7      * nro,
8      * dato,
9      * nodoSiguiente,
10
11  */
12
13  /*****
14
15
16  typedef struct nodo* Nodo;
17
18  typedef struct listaStruct* Lista;
19
20  struct nodo {
21      int nro;
22      void* dato;
23      Nodo siguiente;
24  };
25
26  struct listaStruct {
27      int tamano;
28      Nodo inicio;
29  };
30
31  *****/
32  /* Definicion de Primitivas */
33  /*-----*/
34
35  /*
36      PRE : El Lista no debe haber sido creado.
37      POST: El Lista queda creado y preparado para ser usado.
38  */
39  Lista crearLista();
40
41  /*
42      PRE : El Lista debe haber sido creada.
43      POST: Crea nuevo nodo en la lista con el dato enviado por parametro.
44  */
45  void agregarNodo(Lista lista, void* dato);
46
47  /*
48      PRE : El Lista debe haber sido creada.
49      POST: Elimina el nodo de nro igual al que se envio por parametro
50  */
51  void eliminarNodo(Lista lista, int nro);
52
53  /*
54      PRE : El Lista debe haber sido creada.
55      POST: Elimina todos los nodos de la lista liberando la memoria
56  */
57  void eliminarLista(Lista lista);
58
59  bool listaVacía(Lista lista);
60
61  Nodo nodoSeleccionado(Lista lista, int nro);
62
63  Nodo ultimoNodo(Lista lista);
64
65  Nodo anteriorNodo(Lista lista, Nodo nodo);
66

```

```
67 void mostrarLista(Lista lista);  
68  
69 int ultimoAnio();  
70 #endif
```