

```

1  #ifndef LISTA_H
2  #define LISTA_H
3
4  template <typename T>
5  class Lista
6  {
7      private:
8          struct nodo
9          {
10              T elemento;
11              nodo* sig;
12          };
13          nodo* primero;
14          nodo* ultimo;
15          nodo* iterator;
16          int cantidad;
17
18          //metodos para el uso recursivo
19          void agregarEnPos(nodo* &punt, int pos, T elemento, int c);
20          bool esta(nodo* punt, T elemento);
21          const T & devolver(nodo* punt, int pos, int c);
22
23      public:
24          Lista();
25          virtual ~Lista();
26          void agregarAlPrincipio(T elemento);
27          void agregarAlFinal(T elemento);
28          void agregarEnPos(int pos, T elemento);
29          int size() const; //retorna el tamaño de la lista
30          bool esVacia() const; // verifica que si la lista es vacia
31          bool esta(T elemento); //verifica si esta el elemento
32          const T & devolver(int pos); //devuelve elemento segun posicion
33          void eliminar(int pos); //elimina un nodo dado una posicion
34
35          // metodo para recorrer de manera iterativa
36          void inic(); //reinicia el puntero iterator a la posicion primera
37          void sig(); //hace que el puntero iterator apunte al siguiente
38          const T & elemento(); //retorna el elemento del nodo en el cual iterator esta apuntando
39          bool final(); // verifica si iterator esta en el ultimo nodo
40  };
41
42 #endif // LISTA_H

```