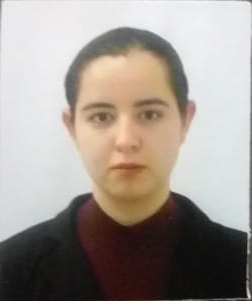
**Protocolo:**

/\*  
Instituto Tecnológico de León  
Ingeniería en Sistemas Computacionales  
Estructuras de datos

( Pablo Vargas Bermúdez, Marcela Janette Mosiño Salas )



Tarea #: 6  
Fecha pactada: 4 de diciembre de 2019  
Fecha de entrega: 4 de diciembre de 2019  
\*/

**Especificación:**

Lista ADT

Matemáticamente, una lista es una secuencia de cero o más elementos de un determinado tipo.

(a1, a2, a3, . . . , an)

donde n ≥ 0, si n = 0 la lista es vacía.  
 Los elementos de la lista tienen la propiedad de que sus elementos están ordenados de forma lineal, según las posiciones que ocupan en la misma. Se dice que ai precede a:  
ai+1 para i = 1, . . . , n − 1 y que ai sucede a ai+1 para i = 2, . . . , n.

Lista Circular Genérica Simplemente Ligada

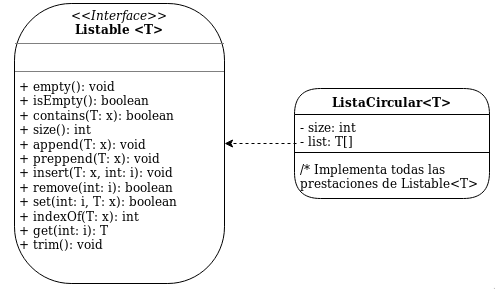
En una lista circular, el campo apuntador del último elemento deberá apuntar al nodo inicial; de esta forma, an apunta a a1. Incluso si an == a1.  
 La particularidad que tienen las listas genéricas es que a la hora de utilizar las prestaciones que provee la clase, es posible definir el tipo de dato que se manejará en dicha estructura, con esto se tendrá la seguridad de que el retorno de los métodos será del tipo de dato especificado y se evitarán las continuas conversiones de los retornos Object. Los paréntesis angulares “<>” denotan el tipo de las clases genéricas.

Las siguientes operaciones se restringen con:

∀L ∈ Lista  
∀x ∈ Objeto <T>  
∀p ∈ Apuntador

|  |  |
| --- | --- |
| Nombre | Descripción |
| Empty() | Elimina todos los valores que conforman a la lista “L”  L.empty() |
| IsEmpty() | Retorna un booleano, verdadero si la lista contiene elementos, falso en caso contrario  Boolean b = L.isEmpty() |
| Contains(Object: x) | Retorna un booleano, verdadero si la lista contiene el objeto “x”, falso en caso contrario  Boolean b = L.contains(x) |
| Size() | Retorna un entero que indica la cantidad de elementos dentro de la lista “L”  Int i = L.size() |
| Append(Object: x) | Inserta un elemento “x” al final de la lista  L.append(x) |
| Preppend(Object: x) | Inserta un elemento “x” al inicio de la lista  L.preppend(x) |
| Insert(Object: x, int: i) | Inserta un elemento “x” en el indice indicado  L.insert(x, i) |
| Remove(int: i) | Elimina el elemento “x” en el indice indicado  L.remove(i) |
| Set(int: i, Object: x) | Sobre escribe el valor posicionado en el indice i de la lista “L”  L.set(0, x) |
| IndexOf(Object: x) | Retorna la posición del objeto “x” en la lista “L”, si no se encuentra el objeto retorna (-1)  Int i = L.indexOf(x) |
| Get(int: i) | Retorna el objeto “x” ubicado en la posición (i % size) de la lista “L”  Object x = L.get(i) |
| Trim() | Ajusta el tamaño de la lista “L” al número de elementos que contenga  L.trim() |

**Diagrama UML:**



**Algoritmo:** // Algoritmo.txt  
**Clases encapsuladas:** // ListaCircular.java; Listable.java (interface);   
**Clases de prueba:**  // Presentador.java  
**Clases auxiliares:** // Graphics.java (Manejo de imágenes); FileHandler.java (Persistencia)  
**Resultados de corrida:**