# 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.

Las siguientes operaciones se restringen con:

∀L ∈ Lista  
∀x ∈ Objeto  
∀p ∈ Apuntador

|  |  |
| --- | --- |
| Nombre | Descripción |
| Vaciar() | Quita todos los elementos de la lista “L”  L.vaciar() |
| isEmpty() | Retorna un valor booleano (verdadero si la lista está vacía, falso en caso contrario)  Boolean b = L.isEmpty() |
| Contiene(Object: x) | Retoran un valor booleano (verdadero si la lista contiene el objecto “x”, falso en caso contrario)  Boolean b = L.contains(x) |
| Tamaño() | Retorna un entero significativo del número de elementos contenidos en la lista “L”  Int tamaño = L.tamaño() |
| Insert(Object: x) | Inserta un objecto “x” al principio de la lista “L”  L.insert(x) |
| insertLast(Object: x) | Inserta un objecto “x” al final de la lista “L”  L.insertLast(x) |
| insertaOrden(String o) | Inserta una cadena de caracteres en orden lexicográfico  L.insertaOrden(palabra) |
| Buscar(Object: x) | Retorna un apuntador al objecto “x” que conforma la lista “L”  Node p = L.buscar(x) |
| Eliminar(Object: x) | Elimina el objeto “x” de la lista “L”  L.eliminar(x) |
| Primero() | Retorna un apuntador al primer nodo “p” de la lista “L”  Nodo p = L.primero() |
| Anterior(String: o) | Retorna el nodo lexicográficamente anterior a la cadena de caracteres “o”  Nodo p = L.anterior(o) |
| Siguiente(Node: p, MySympleLinkedList L) | Retorna el nodo posterior a el apuntador “p” en la lista “L”  L.siguiente(p, L) |
| Sustituir(Object x1, Object x2) | Sustituye el objeto “x1” por el objecto “x2” en la lista “L”  L.sustituir(x1, x2) |

# UML

