

## T12.Listas doblemente enlazadas

En las listas simplemente enlazadas, hay un solo sentido en el recorrido de la lista. Puede resultar útil el poder avanzar en ambos sentidos, de tal forma que los términos predecesor y sucesor no tengan significado puesto que la lista es completamente simétrica. En la lista doblemente enlazada, cada elemento contiene dos punteros, aparte del valor almacenado en el elemento. En cada nodo de una lista doblemente enlazada existen dos enlaces, uno al siguiente nodo y otro al nodo anterior.

Las operaciones básicas que se definen en el tipo abstracto lista doble son básicamente las mismas que las definidas en una lista simple y tendrán las siguientes funciones:

- Declaración de los tipos nodo y puntero a nodo.
- Inicialización o creación.
- Insertar elementos en una lista.
- Eliminar elementos de una lista.
- Buscar elementos de una lista.
- Recorrer una lista enlazada.
- Comprobar si la lista está vacía.

Una lista doblemente enlazada con valores de tipo int necesita dos punteros y el valor del campo de datos. En una estructura se agrupan estos datos del modo siguiente:

```
typedef int Item;
struct unnodo
{
    Item dato;
    struct unnodo *adelante;
    struct unnodo *atras;
};
typedef struct unnodo Nodo;
```

**Bibliografía:**

- Joyanes A. L., Zahonero M. I. (2004) *Algoritmos y estructuras datos. Una perspectiva en C*. España: McGraw-Hill.