

## Listas Doblemente enlazadas:

En algunas aplicaciones podemos desear recorrer la lista hacia adelante y hacia atrás, o dado un elemento, podemos desear conocer rápidamente los elementos anterior y siguiente. En tales situaciones podríamos desear darle a cada celda sobre una lista un puntero a las celdas siguiente y anterior en la lista.



Otra ventaja de las listas doblemente enlazadas es que podemos usar un puntero a la celda que contiene el  $i$ -ésimo elemento de una lista para representar la posición  $i$ , mejor que usar el puntero a la celda anterior aunque lógicamente, también es posible la implementación similar a la expuesta en las listas simples haciendo uso de la cabecera. El único precio que pagamos por estas características es la presencia de un puntero adicional en cada celda y consecuentemente procedimientos algo más largos para algunas de las operaciones básicas de listas.

Es importante notar que aunque la estructura física de la lista puede hacer pensar que mediante la operación siguiente podemos alcanzar de nuevo un nodo de la lista, la estructura lógica es la de una lista y por lo tanto habrá una posición primero y una posición fin de forma que al aplicar una operación anterior o siguiente respectivamente sobre estas posiciones el resultado será un error.

### Bibliografía:

**Abstracción y estructuras de datos en C++**  
Escrito por Antonio Garrido Carrillo, Joaquín Fernández-Valdivia