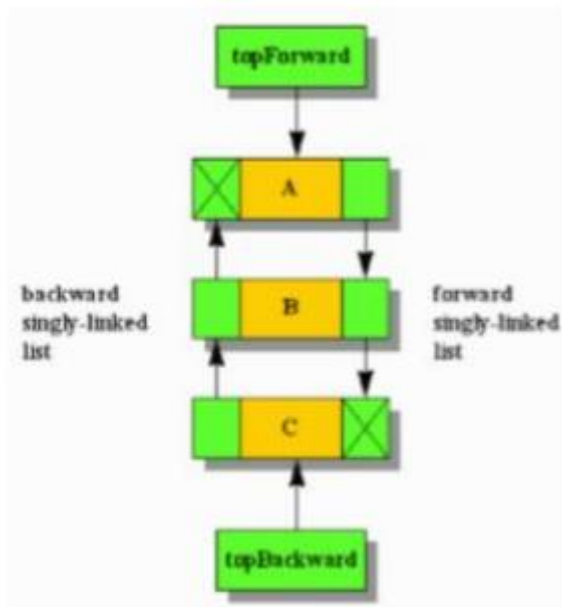


## Listas doblemente enlazadas

En las listas lineales el recorrido de ellas sólo podía hacerse en un único sentido: de izquierda a derecha (principio a final). En numerosas ocasiones se necesita recorrer las listas en ambas direcciones. Las listas que pueden recorrerse en ambas direcciones se denominan listas doblemente enlazadas.

De forma similar, para la dirección contraria, una variable de referencia contiene una referencia al último nodo de la dirección normal (hacia adelante), lo que se interpreta como el primer nodo. Cada nodo se enlaza con el anterior mediante el campo de enlace previous, y el primer nodo de la dirección hacia adelante, contiene null en su campo previous para indicar el fin de la lista. La siguiente figura representa una lista doblemente enlazada de tres nodos, donde topForward referencia el primer nodo en la dirección hacia adelante, y topBackward referencia el primero nodo la dirección inversa.



En estas listas cada nodo consta del campo INFO de datos y dos campos de enlace o punteros: ANTERIOR(ANT) y SIGUIENTE(SIG) que apuntan hacia adelante y hacia atrás

Como cada elemento tiene dos punteros, una lista doblemente enlazada ocupa más espacio en memoria que una lista simplemente enlazada para una misma cantidad de información.

La lista necesita dos punteros CABECERA y FIN2 que apuntan hacia el primero y último nodo. La variable CABECERA y el puntero SIG permiten recorrer la lista en el sentido normal y la variable FIN y el puntero ANT permiten recorrerla en sentido inverso.

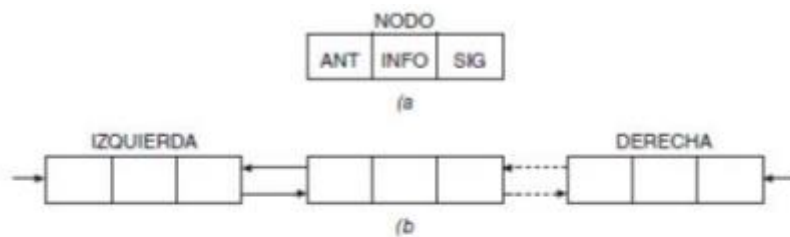
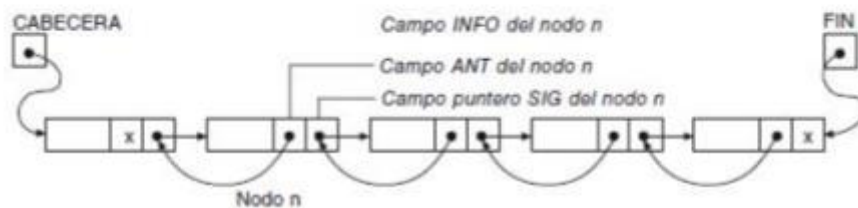


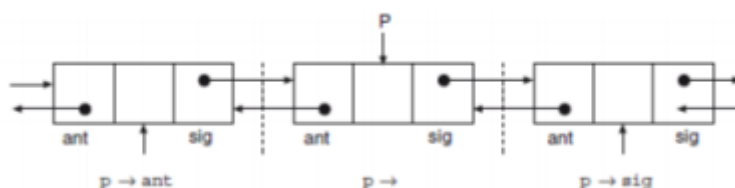
Figura 12.9. Lista doblemente enlazada.



Una propiedad fundamental de las listas doblemente enlazadas es que para cualquier puntero P de la lista:

$\text{nodo}[\text{nodo}[p].\text{sig}].\text{ant} = p$

$\text{nodo}[\text{nodo}[p].\text{ant}].\text{sig} = p$

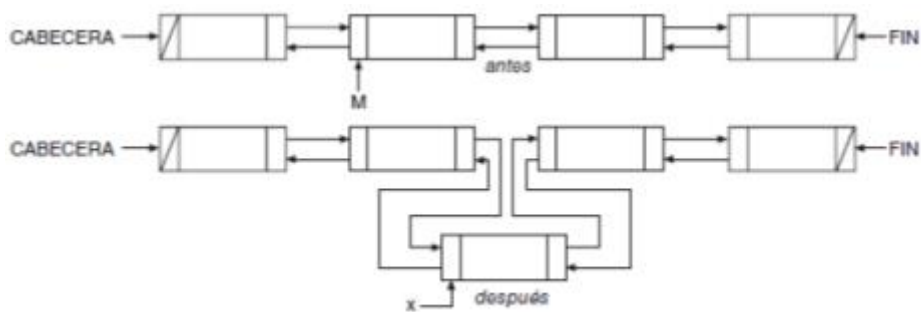


## Operaciones con listas doblemente enlazadas

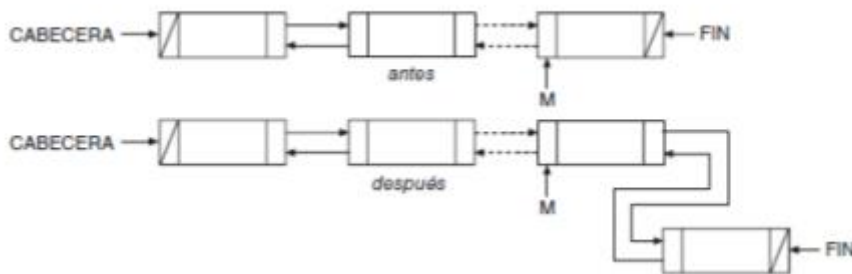
### Inserción

La inserción de un nodo a la derecha de un nodo especificado, cuya dirección está dada por la variable M, puede presentar varios casos:

1. La lista está vacía; se indica mediante  $M = \text{NULO}$  y  $\text{CABECERA}$  y  $\text{FIN}$  son también  $\text{NULO}$ . Una inserción indica que  $\text{CABECERA}$  se debe fijar con la dirección del nuevo nodo y los campos  $\text{ANT}$  y  $\text{SIG}$  también se establecen en  $\text{NULO}$ .
2. Insertar dentro de la lista: existe un elemento anterior y otro posterior de nuevo nodo.
3. Insertar a la derecha del nodo del fin de la lista. Se requiere que el apuntador  $\text{FIN}$  sea modificado.



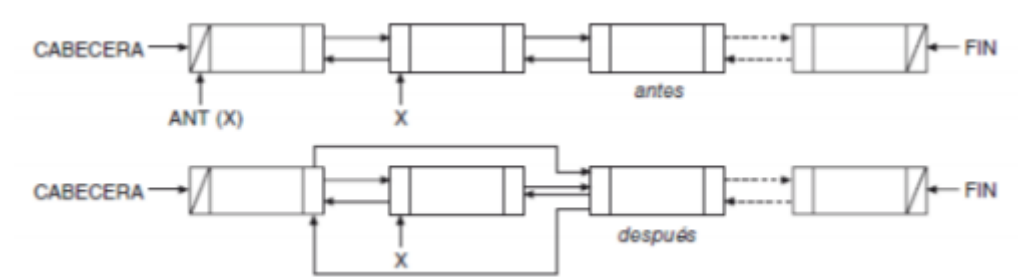
**Figura 12.12.** Inserción en un lista doblemente enlazada.



## Eliminación

La operación de eliminación es directa. Si la lista tiene un simple nodo, entonces los punteros de los extremos izquierdo y derecho asociados a la lista se deben fijar en NULO. Si el nodo del extremo derecho de la lista es el señalado para la eliminación, la variable FIN debe modificarse para señalar el predecesor del nodo que se va a borrar de la lista. Si el nodo del extremo izquierdo de la lista es el que se desea borrar, la variable CABECERA debe modificarse para señalar el elemento siguiente.

La eliminación se puede realizar dentro de la lista.



## Referencias

1. Universidad Privada TELESUP. Lista simple doblemente enlazada. Algorítmica y estructura de datos(130). Perú.
2. Joyanes Aguilar, Luis (1996) Fundamentos de programación, Algoritmos y Estructura de datos. McGraw-Hill, México.
3. Cairó , O., & Guardati, S. (2006). Estructuras de datos (3a. ed.). McGraw-Hill Interamericana.
4. Brassard, G., y Bratley, P.: Fundamentos de algoritmia Madrid. Prentice-Hall, 1997.
5. Heileman, Gregory L.: Estructuras de datos, algoritmos, y programación orientada a objetos, Madrid, McGraw-Hill, 1997.
6. Estructura de datos: Algoritmos, abstracción y objetos, Luis Joyanes, McGraw-Hill, España, 1998.
7. Collins, William J, Data Structures, An object-Oriented Approach, Addison-Wesley, 1992.