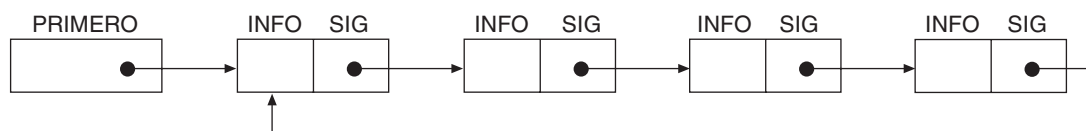


12.5. LISTAS CIRCULARES

Las listas simplemente enlazadas no permiten a partir de un elemento acceder directamente a cualquiera de los elementos que le preceden. En lugar de almacenar un puntero **NULL** en el campo **SIG** del último elemento de la lista, se hace que el último elemento apunte al primero o principio de la lista. Este tipo de estructura se llama *lista enlazada circular* o simplemente *lista circular* (en algunos textos se les denomina listas en anillo).



Las listas circulares presentan las siguientes *ventajas* respecto de las listas enlazadas simples:

- Cada nodo de una lista circular es accesible desde cualquier otro nodo de ella. Es decir, dado un nodo se puede recorrer toda la lista completa. En una lista enlazada de forma simple sólo es posible recorrerla por completo si se parte de su primer nodo.
- Las operaciones de concatenación y división de listas son más eficaces con listas circulares.

Los *inconvenientes*, por el contrario, son:

- Se pueden producir lazos o bucles infinitos. Una forma de evitar estos bucles infinitos es disponer de un nodo especial que se encuentre permanentemente asociado a la existencia de la lista circular. Este nodo se denomina *cabecera* de la lista.

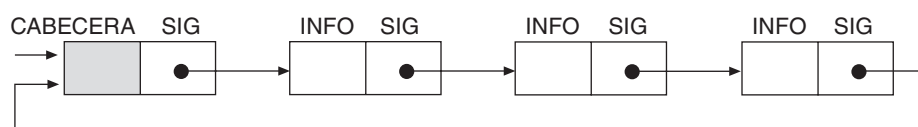


Figura 12.7. Nodo cabecera de la lista.

El nodo cabecera puede diferenciarse de los otros nodos en una de las dos formas siguientes:

- Puede tener un valor especial en su campo **INFO** que no es válido como datos de otros elementos.
- Puede tener un indicador o bandera (*flag*) que señale cuando es nodo cabecera.

El campo de la información del nodo cabecera no se utiliza, lo que se señala con el sombreado de dicho campo. Una lista enlazada circularmente *vacía* se representa como se muestra en la Figura 12.8.

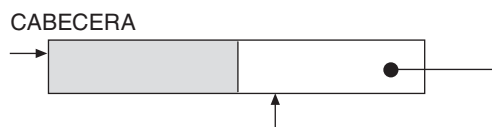


Figura 12.8. Lista circular vacía.

12.6. LISTAS DOBLEMENTE ENLAZADAS

En las listas lineales estudiadas anteriormente 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*. 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 (Fig. 12.9). 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.

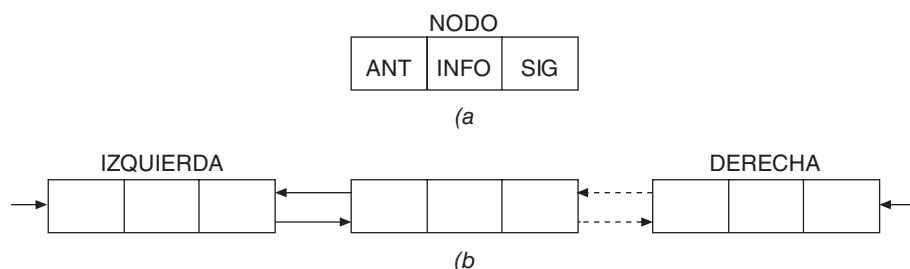


Figura 12.9. Lista doblemente enlazada.

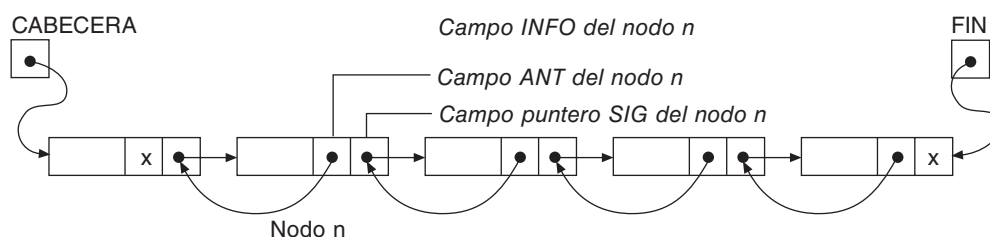


Figura 12.10. Lista doble.

Como se ve en la Figura 12.11, una propiedad fundamental de las listas doblemente enlazadas es que para cualquier puntero p de la lista:

```
nodo [nodo[p].sig].ant = p
nodo [nodo[p].ant].sig = p
```

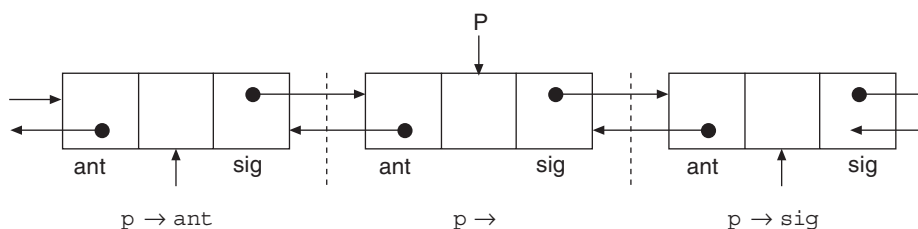


Figura 12.11.

12.6.1. 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 CABECERA y FIN son también NULO. Una inserción indica que CABECERA se debe fijar con la dirección del nuevo nodo y los campos ANT y SIG también se establecen en 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 FIN sea modificado.

² Se adoptan estos términos a efectos de normalización, pero el lector puede utilizar IZQUIERDA y DERECHA.

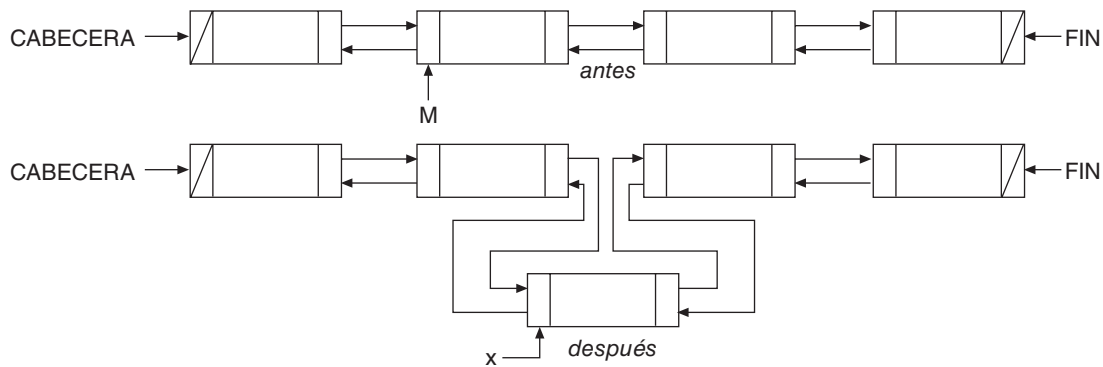


Figura 12.12. Inserción en una lista doblemente enlazada.

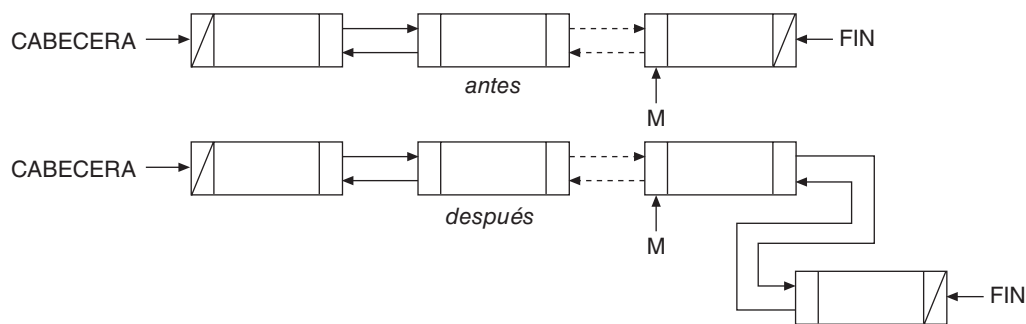


Figura 12.13. Inserción en el extremo derecho de una lista doblemente enlazada.

12.6.2. 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 (Figura 12.14).

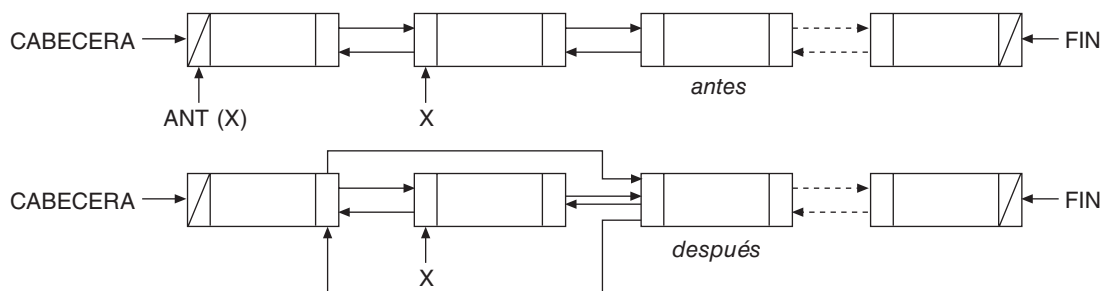


Figura 12.14. Eliminación de un nodo *X* en una lista doblemente enlazada.

12.7. PILAS

Una *pila* (*stack*) es un tipo especial de lista lineal en la que la inserción y borrado de nuevos elementos se realiza sólo por un extremo que se denomina *cima* o *tope* (*top*).

La pila es una estructura con numerosas analogías en la vida real: una pila de platos, una pila de monedas, una pila de cajas de zapatos, una pila de camisas, una pila de bandejas, etc.