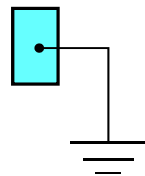
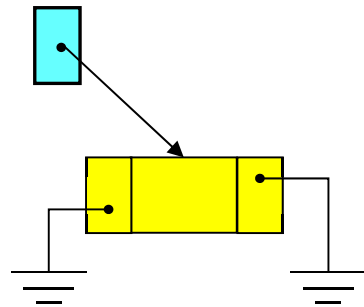
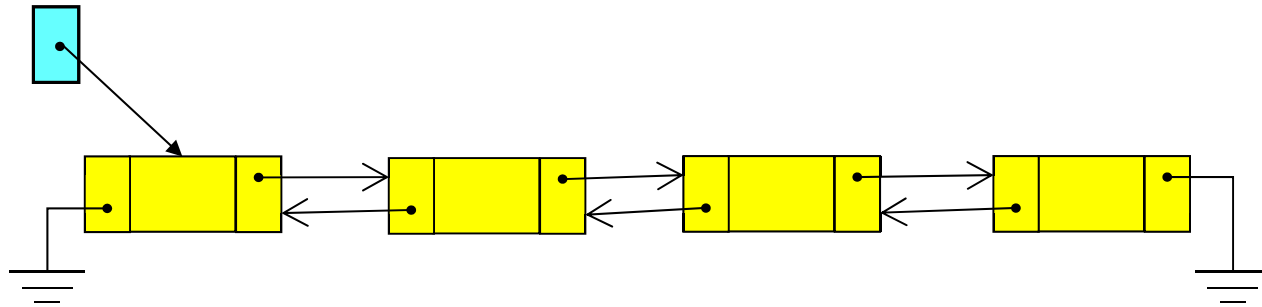
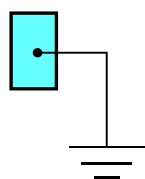
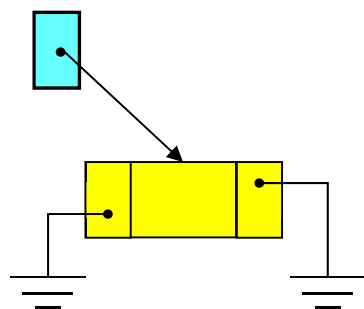
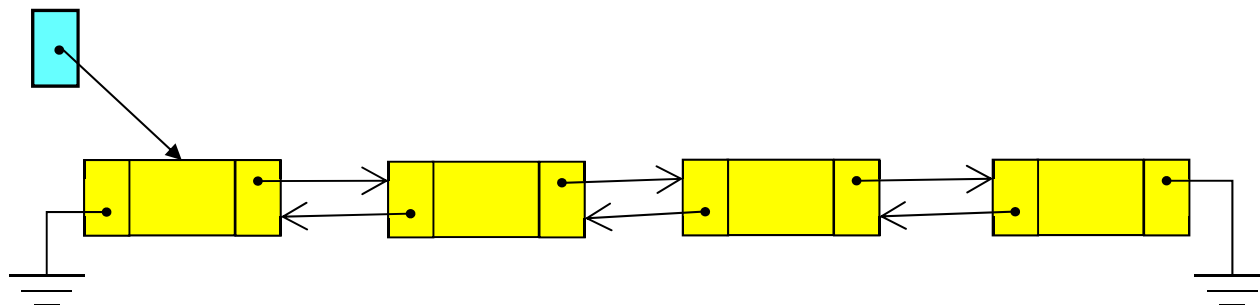


# Lista Doblemente Ligada Lineal Sin Encabezado (LDLLSE)

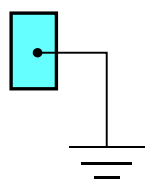
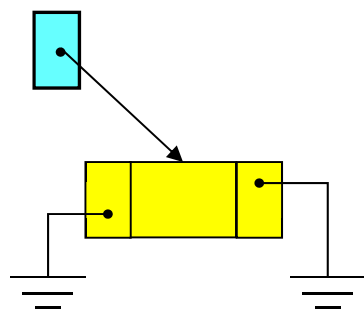
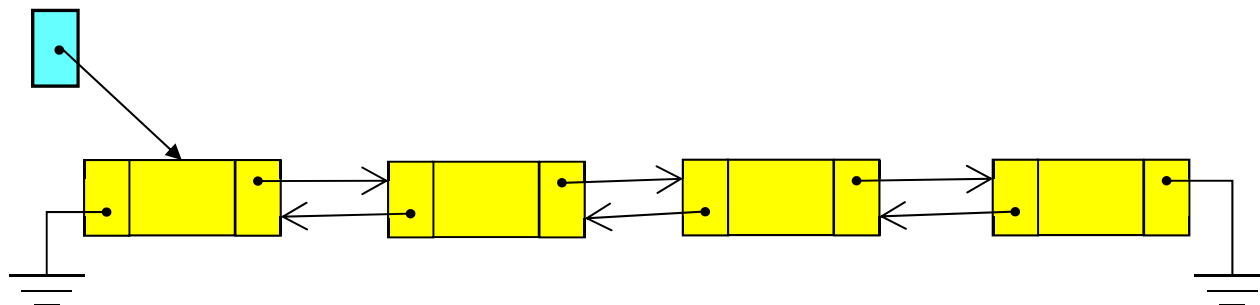


```
typedef _____ tipo_dato;  
  
typedef struct tipo_nodo {  
    tipo_dato elem;  
    struct tipo_nodo *ant;  
    struct tipo_nodo *sig;  
    } tipo_nodo;  
  
typedef tipo_nodo *tipo_lista;  
  
typedef tipo_nodo *tipo_pos;
```



función: **inicializa**  
 recibe: *lista*  
 regresa: nada

*lista* = NULO

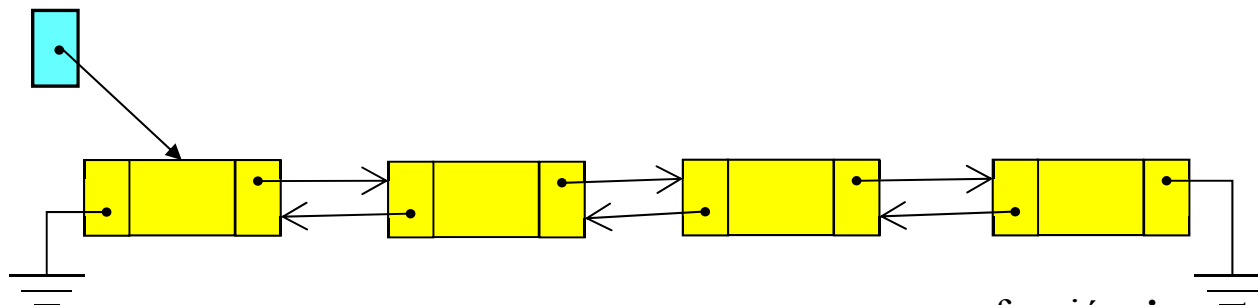


función: **vacía**  
 recibe: *lista*  
 regresa: booleano

*¿lista = NULO?*

Sí: regresar: *verdadero*

No: regresar: *falso*



función: **inserta**

recibe: *elem*, *pos*, *lista*

regresa: nada

*aux* = nuevo nodo

*aux* → *elem* = *elem*

¿*pos* = NULO?

Sí: *aux* → *ant* = NULO

*aux* → *sig* = *lista*

¿*lista* ≠ NULO?

Sí: *lista* → *ant* = *aux*

*lista* = *aux*

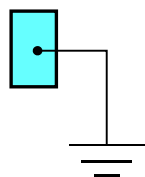
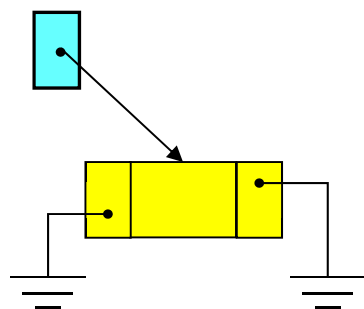
No: *aux* → *ant* = *pos*

*aux* → *sig* = *pos* → *sig*

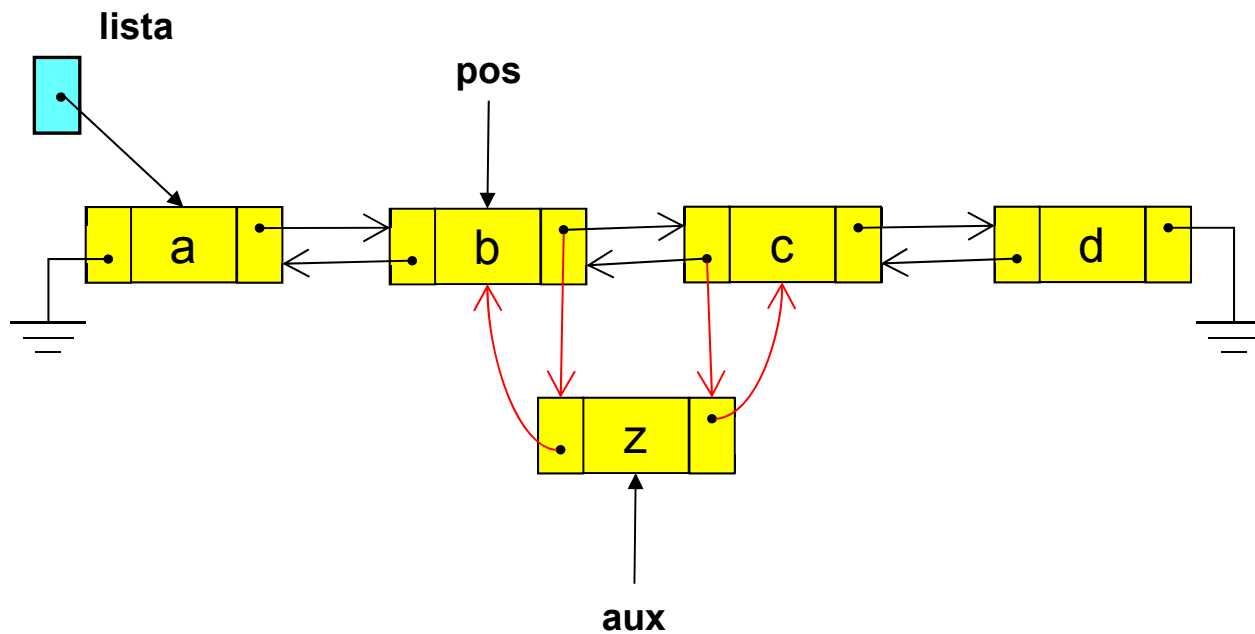
¿*pos* → *sig* ≠ NULO?

Sí: *pos* → *sig* → *ant* = *aux*

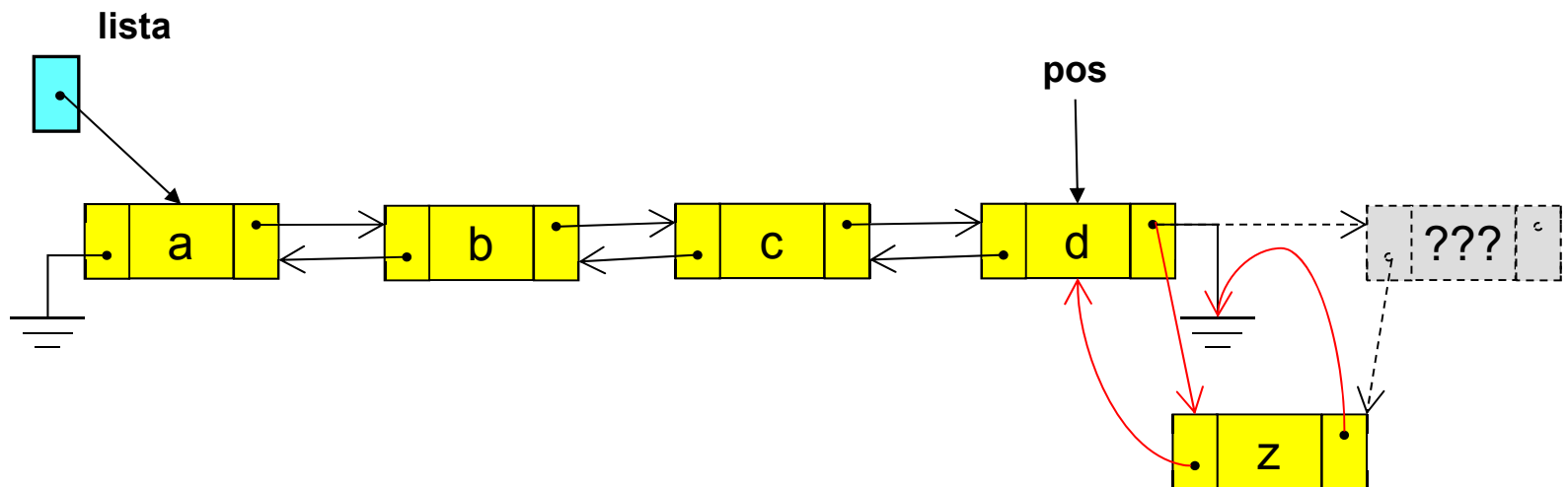
*pos* → *sig* = *aux*



Insertión  
en cualquier posición,  
excepto al principio

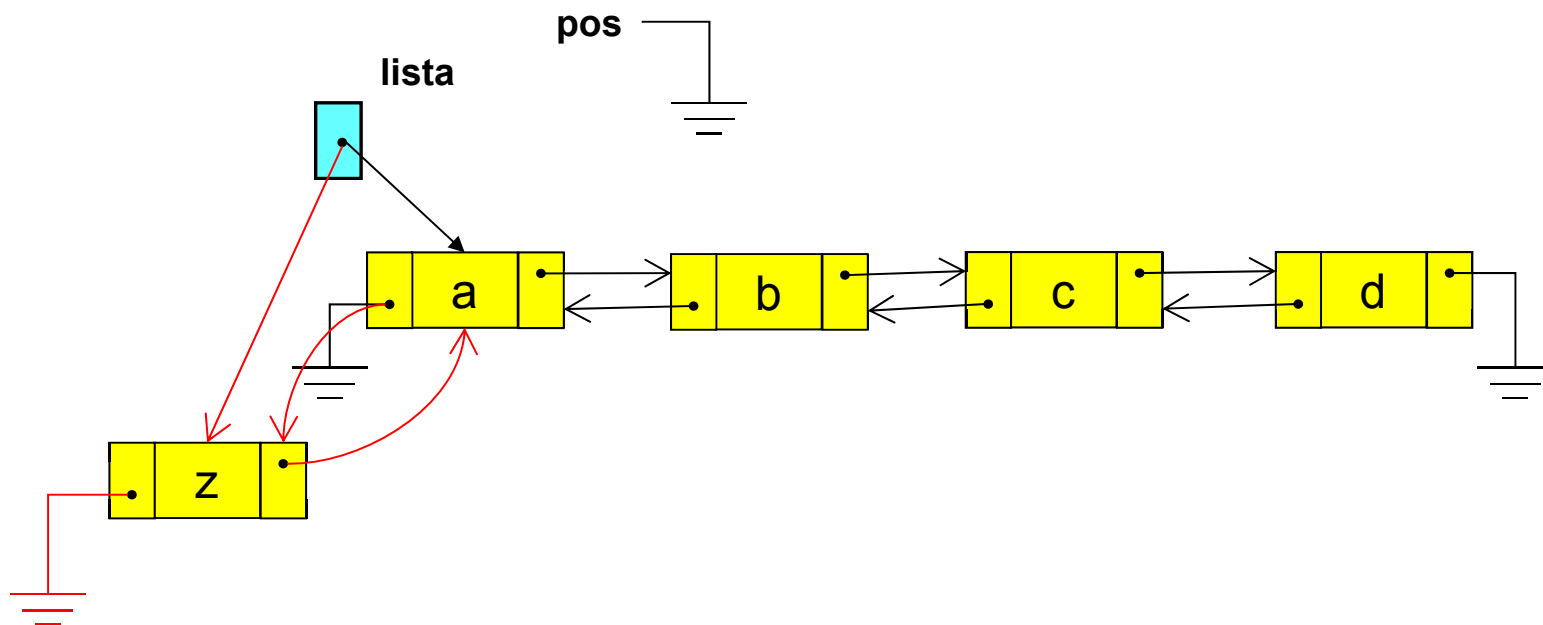


Insertión  
en cualquier posición,  
excepto al principio

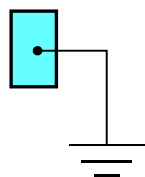
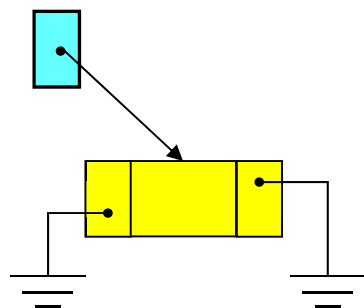
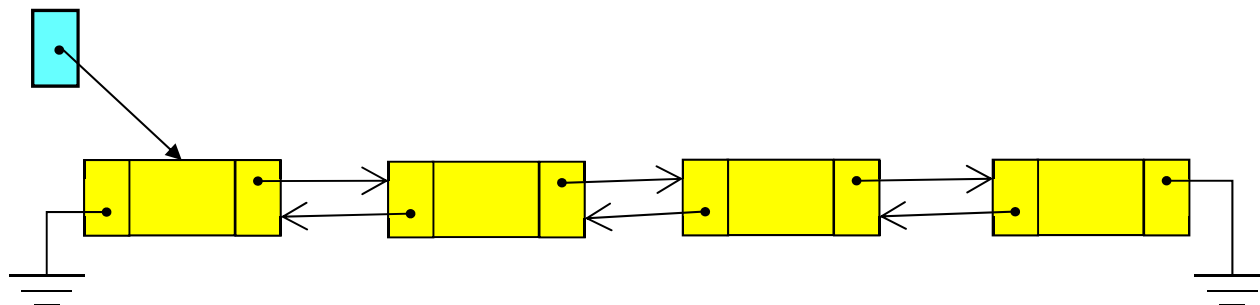




## Inserción al principio







función: **elimina**

recibe:  $pos$ ,  $lista$

regresa: nada

¿**vacía**( $lista$ ) ó  $pos = \text{NULO}$ ?

Sí: terminar

$pos \rightarrow \text{ant} \neq \text{NULO}$ ?

Sí:  $pos \rightarrow \text{ant} \rightarrow \text{sig} = pos \rightarrow \text{sig}$

$pos \rightarrow \text{sig} \neq \text{NULO}$ ?

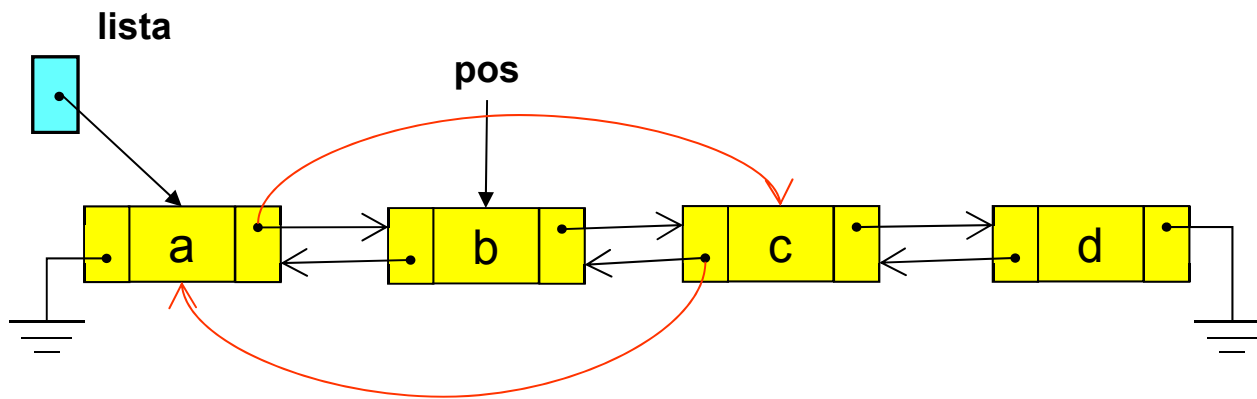
Sí:  $pos \rightarrow \text{sig} \rightarrow \text{ant} = pos \rightarrow \text{ant}$

¿ $pos = lista$ ?

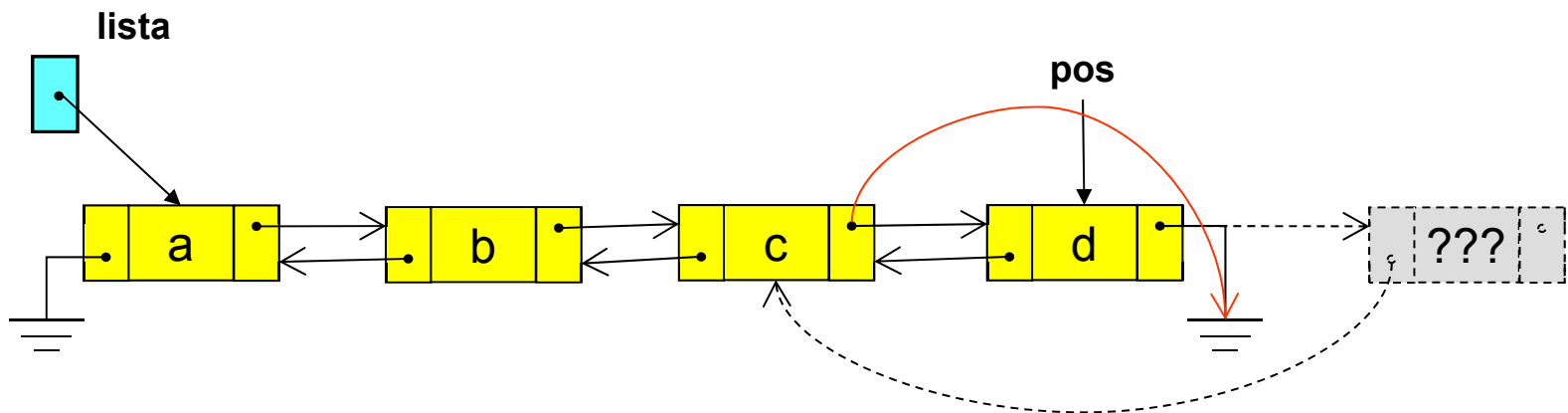
Sí:  $lista = lista \rightarrow \text{sig}$

liberar espacio de memoria de  $pos$

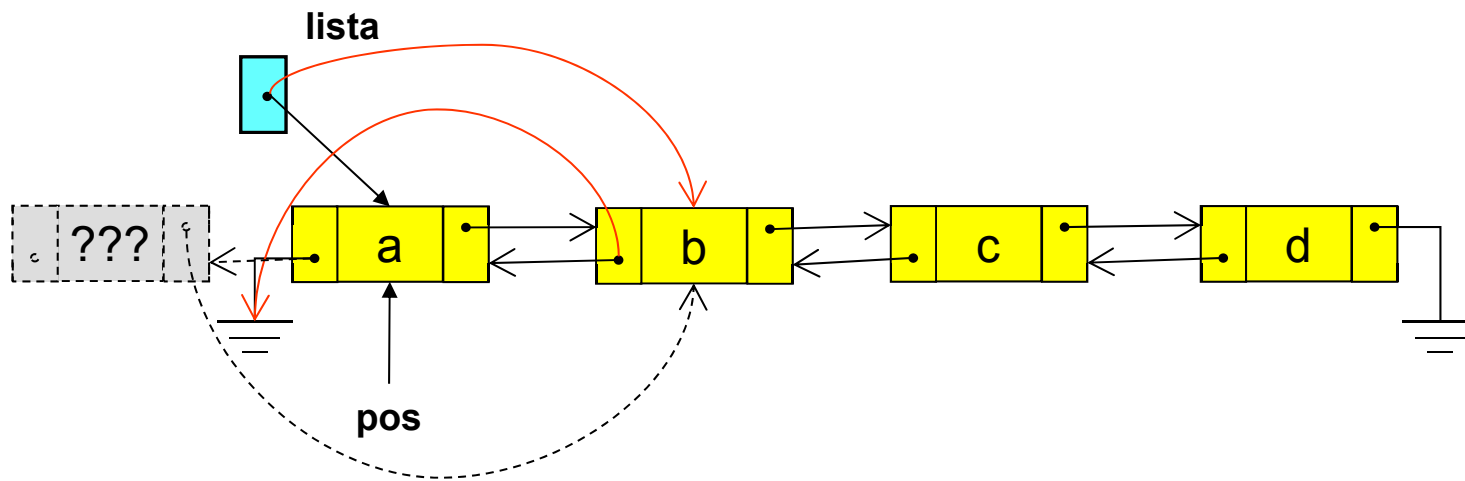
## Eliminación en cualquier posición

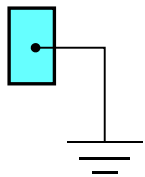
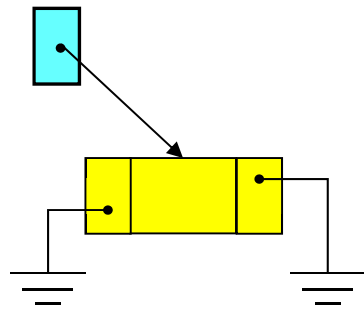
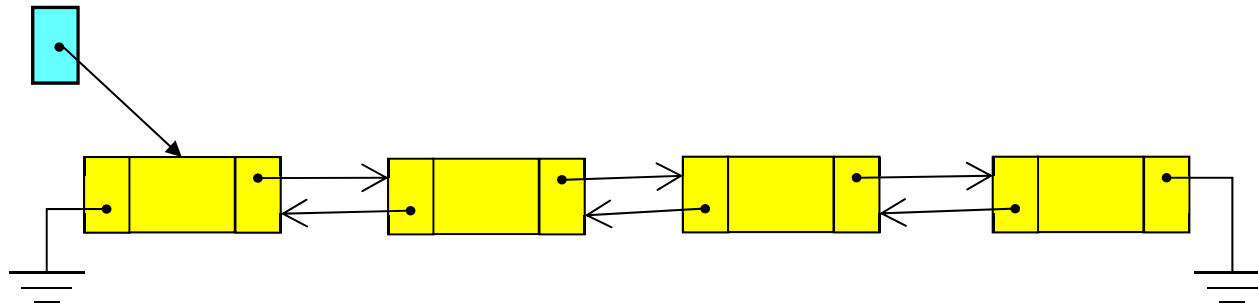


## Eliminación en cualquier posición

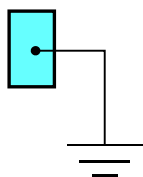
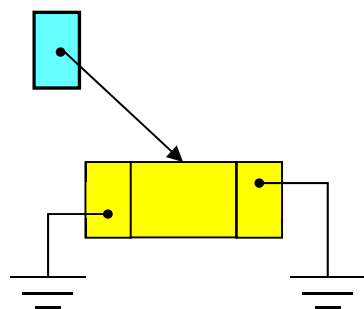
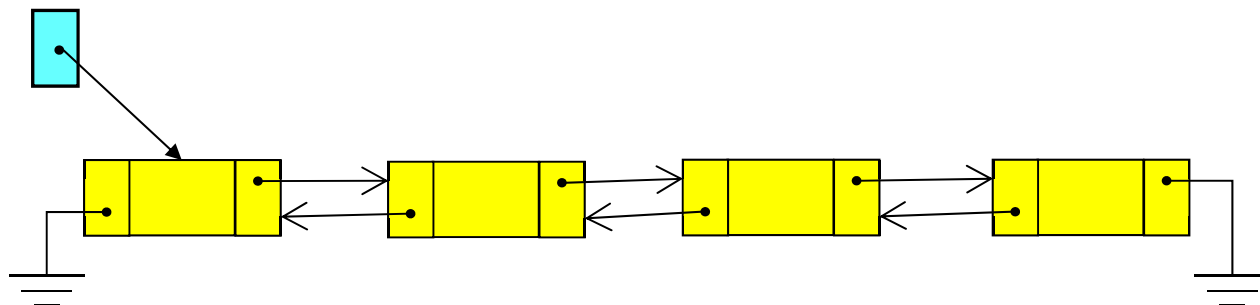


## Eliminación en cualquier posición





función: ***primero***  
 recibe: *lista*  
 regresa: posición  
 regresar: *lista*



función: **último**

recibe: *lista*

regresa: posición

¿**vacía**(*lista*)?

Sí: regresar: NULO

terminar

$aux = lista$

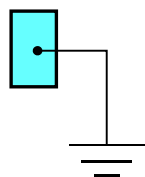
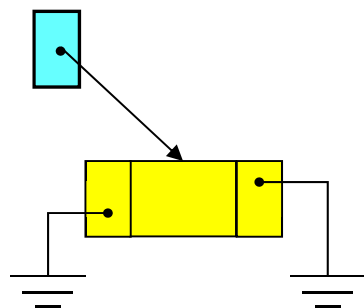
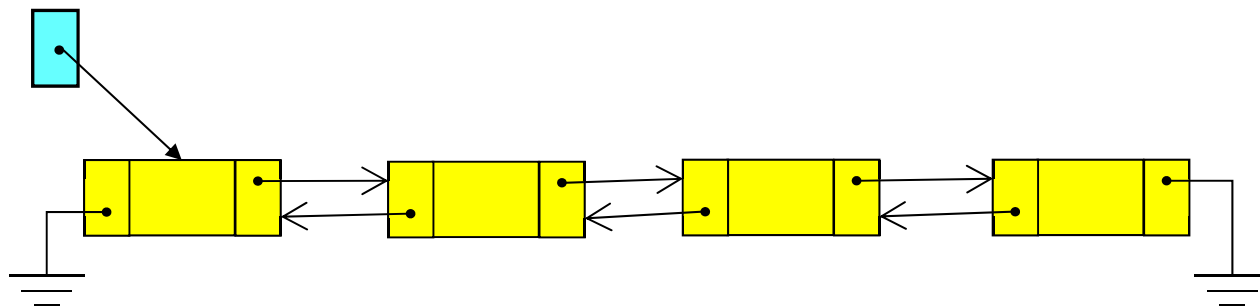
mientras  $aux \rightarrow sig \neq NULO$

$aux = aux \rightarrow sig$

fin mientras

regresar:  $aux$



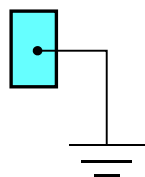
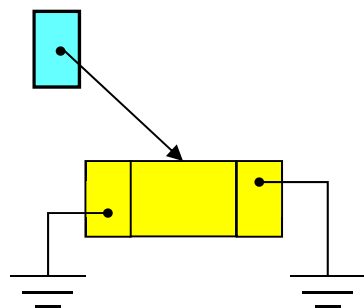
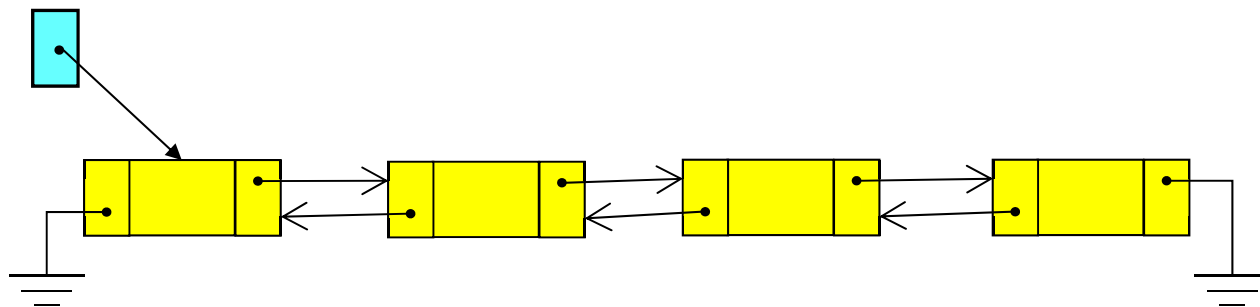


función. **anterior**  
 recibe: *pos*, *lista*  
 regresa: posición

¿**vacía**(lista) ó *pos* = NULO?

Sí: regresar: NULO  
 terminar

regresar:  $pos \rightarrow ant$



función: **siguiente**

recibe: *pos*, *lista*

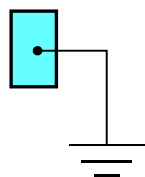
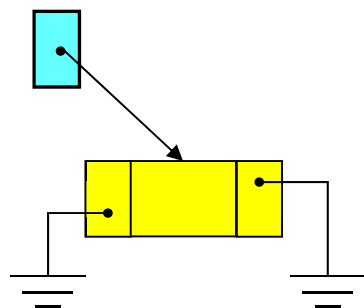
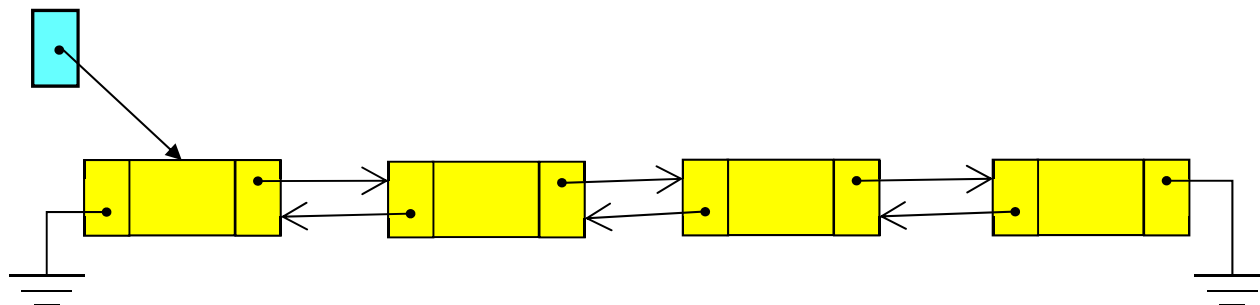
regresa: posición

¿vacía(*lista*) ó *pos* = NULO?

Sí: regresar: NULO

terminar

regresar: *pos* → sig



función: **localiza**

recibe: *elem*, *lista*

regresa: posición

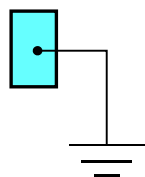
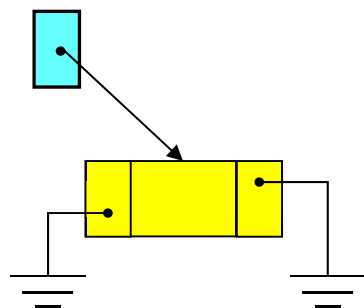
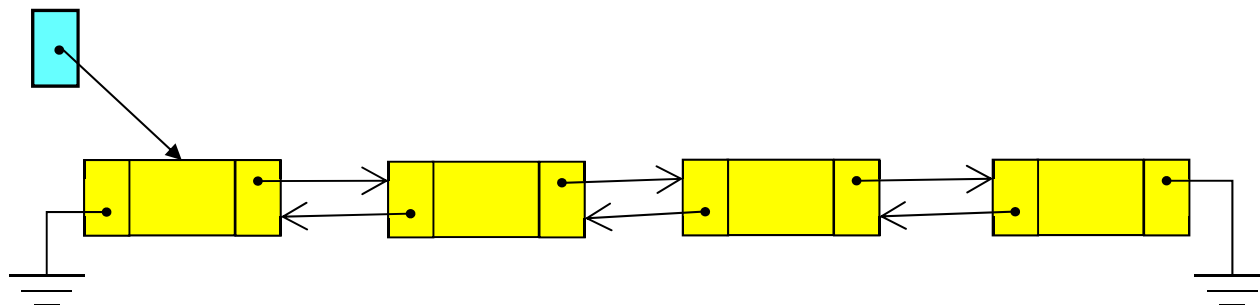
*aux* = *lista*

mientras *aux* ≠ NULO y *aux*→*elem* ≠ *elem*

*aux* = *aux*→*sig*

fin mientras

regresar: *aux*



función: **recupera**  
 recibe *pos*, *lista*  
 regresa: elemento

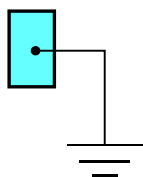
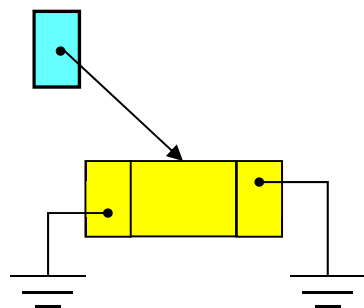
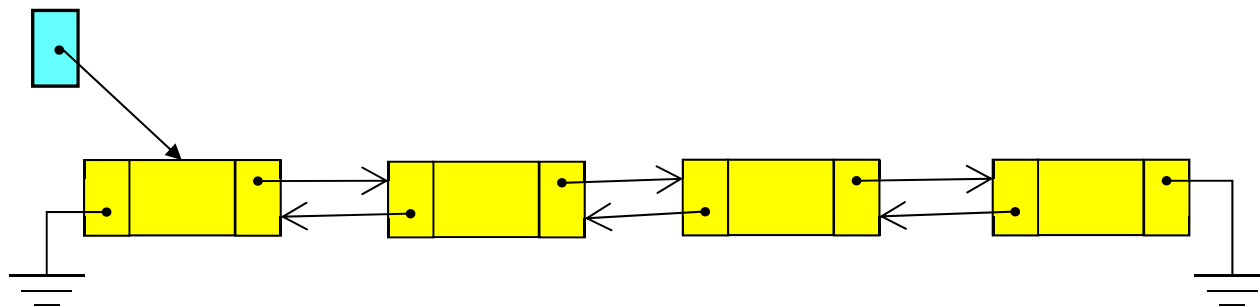
¿vacía(*lista*) ó *pos* = NULO?

Sí: ¡error de excepción!

Insuficiencia de datos  
 terminar



No: regresar: *pos*→elem



función: **imprime**

recibe: *lista*

regresa: nada

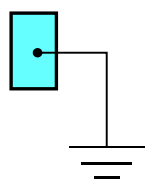
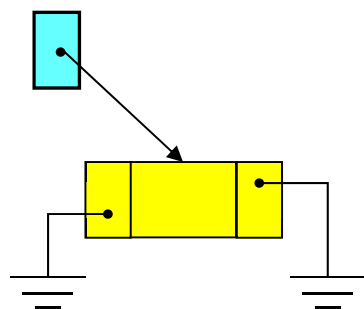
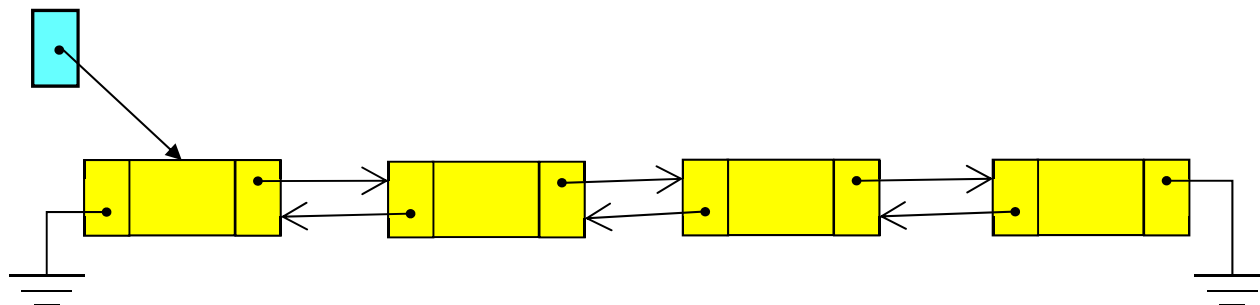
*aux* = *lista*

mientras *aux* ≠ NULO

imprimir *aux*→elem

*aux* = *aux*→sig

fin mientras



función: **anula**

recibe: *lista*

regresa: nada

mientras  $lista \neq \text{NULO}$

$\text{aux} = lista$

$lista = lista \rightarrow \text{sig}$

    liberar espacio de memoria de aux

fin mientras

