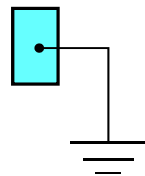
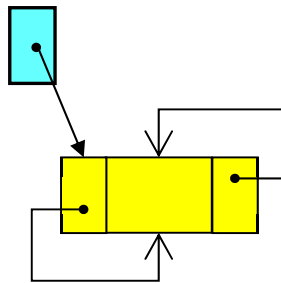
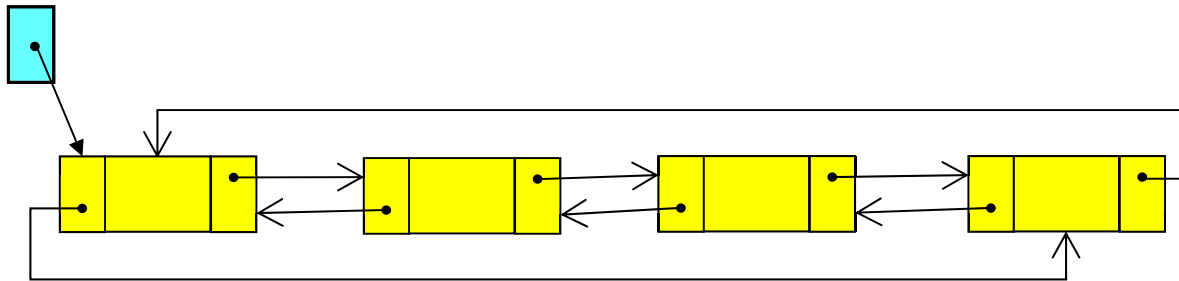
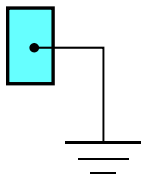
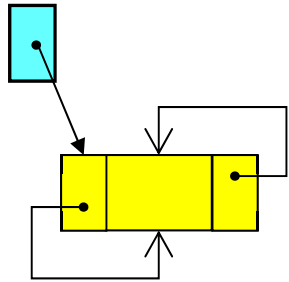
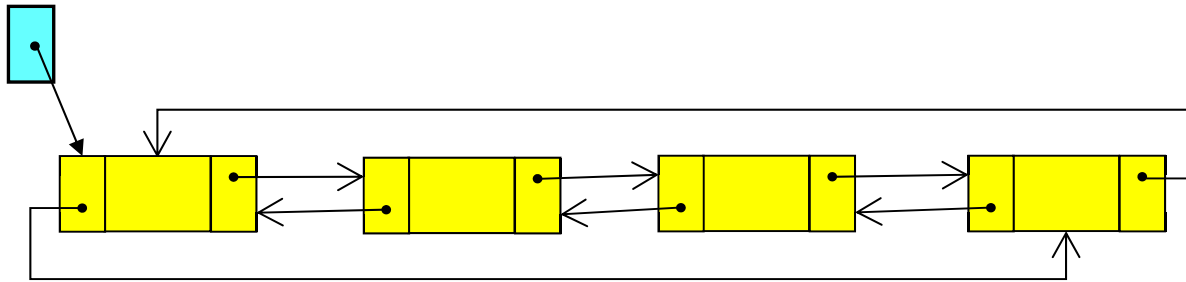


Lista Doblemente Ligada Circular Sin Encabezado (LDLCSE)



```
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;
```

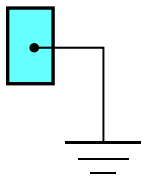
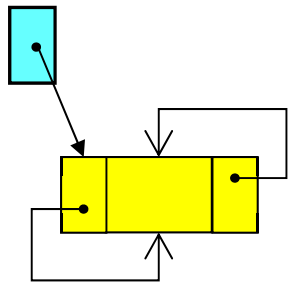
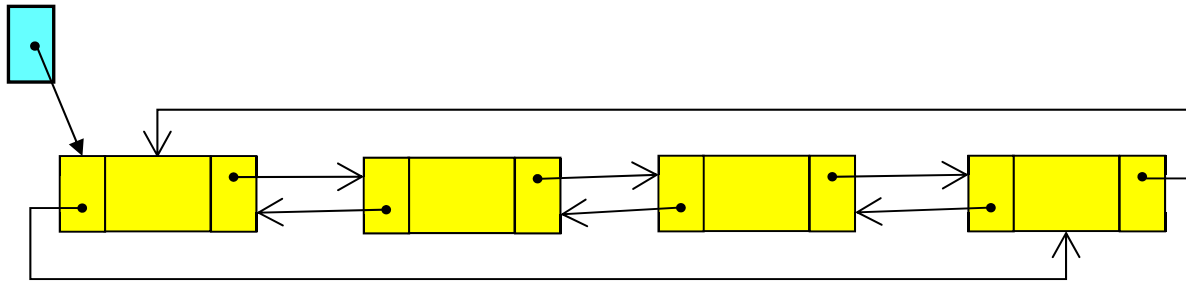


función: **inicializa**

recibe: *lista*

regresa: nada

lista = NULO

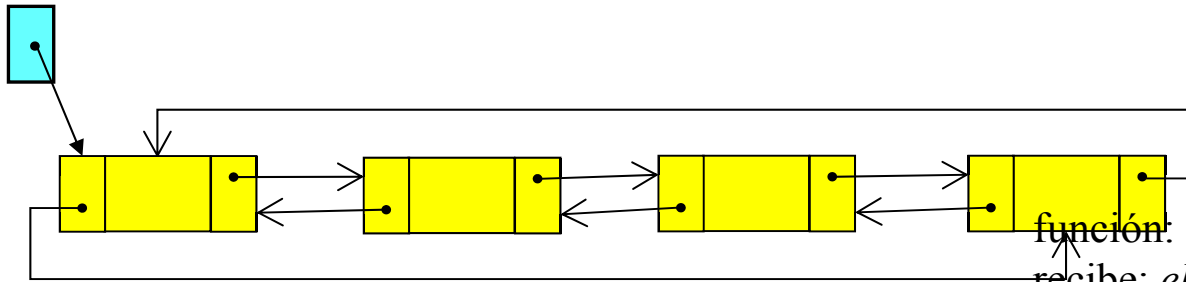


función: **vacía**
 recibe: *lista*
 regresa: booleano

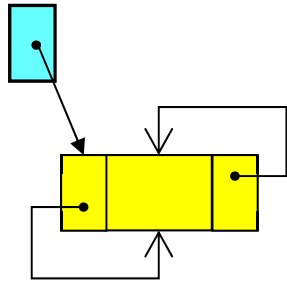
$\text{¿lista} = \text{NULO?}$

Sí: regresar: *verdadero*

No: regresar: *falso*



función: **inserta**
 recibe: *elem*, *pos*, *lista*
 regresa: nada



aux = nuevo nodo
 $aux \rightarrow elem = elem$

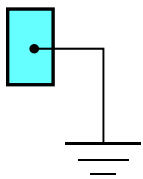
¿**vacía**(*lista*)?

Sí: $aux \rightarrow sig = aux$
 $aux \rightarrow ant = aux$
 $lista = aux$

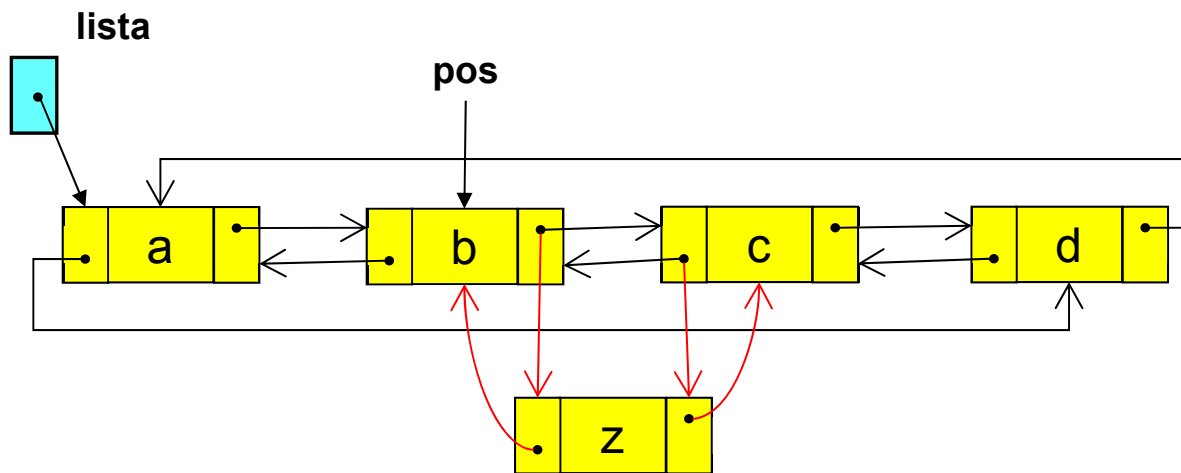
No: ¿*pos* = NULO?

Sí: $aux \rightarrow sig = lista$
 $aux \rightarrow ant = ultimo(lista)$
 $ultimo(lista) \rightarrow sig = aux$
 $lista = aux$

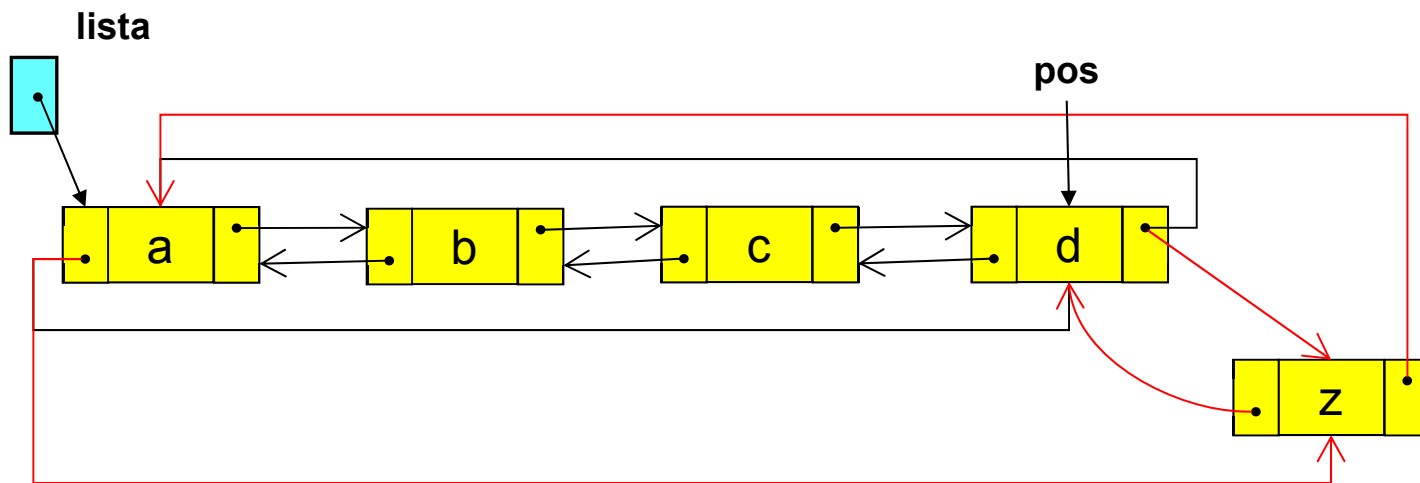
No: $aux \rightarrow ant = pos$
 $aux \rightarrow sig = pos \rightarrow sig$
 $pos \rightarrow sig \rightarrow ant = aux$
 $pos \rightarrow sig = aux$



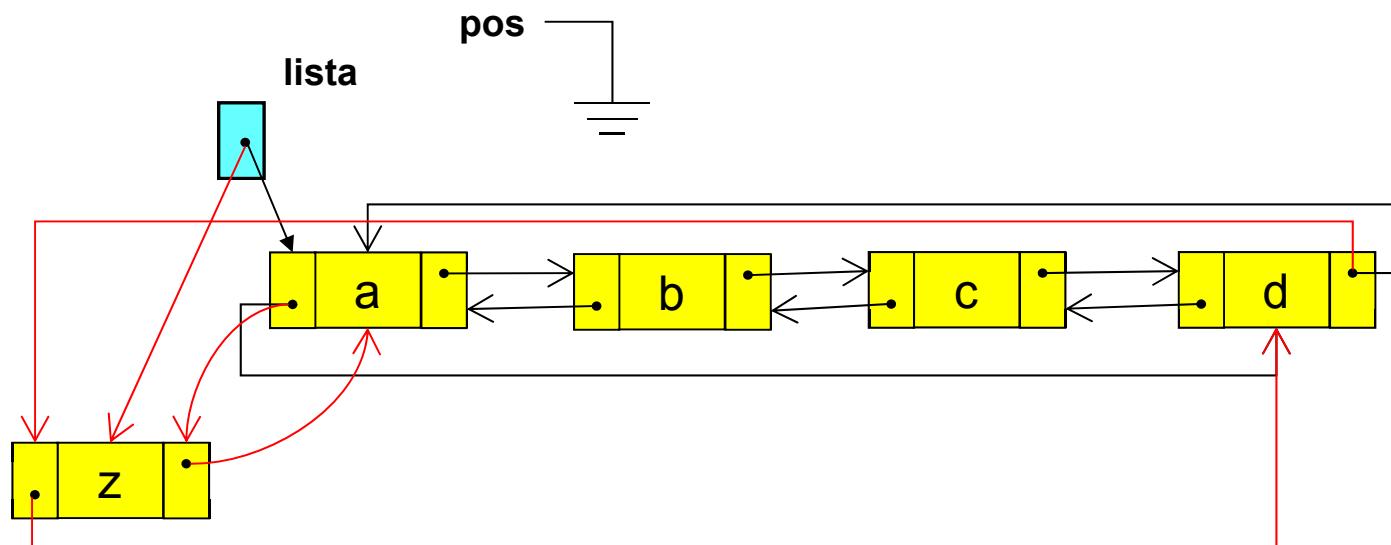
Insertión
en cualquier posición,
excepto al principio



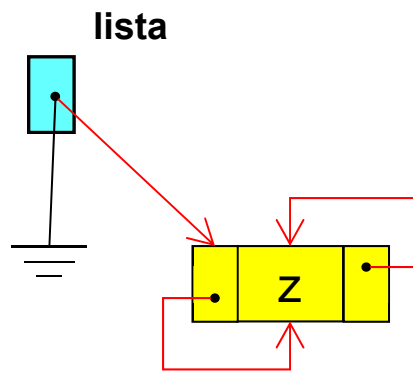
Insertión
en cualquier posición,
excepto al principio

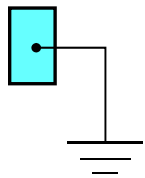
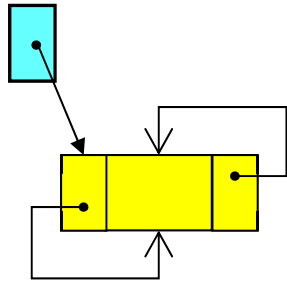
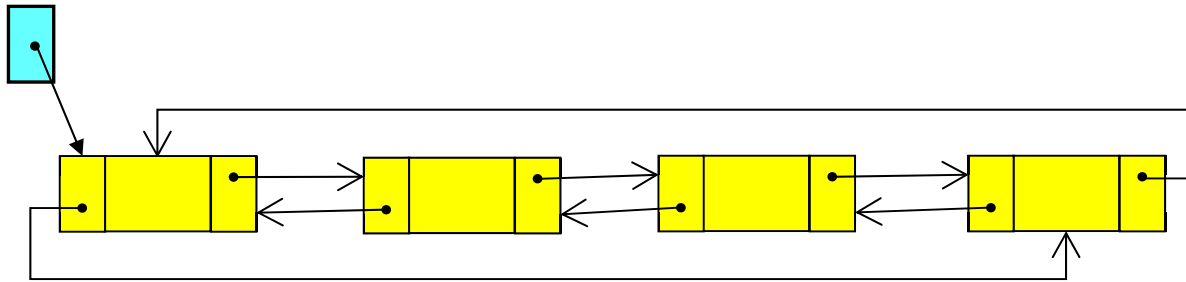


Inserción al principio



Inserción En lista vacía





función: **elimina**
 recibe: pos , $lista$
 regresa: nada

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

Sí: terminar

¿ $pos \rightarrow \text{sig} = pos$?

Sí: $lista = \text{NULO}$

No: $pos \rightarrow ant \rightarrow \text{sig} = pos \rightarrow \text{sig}$

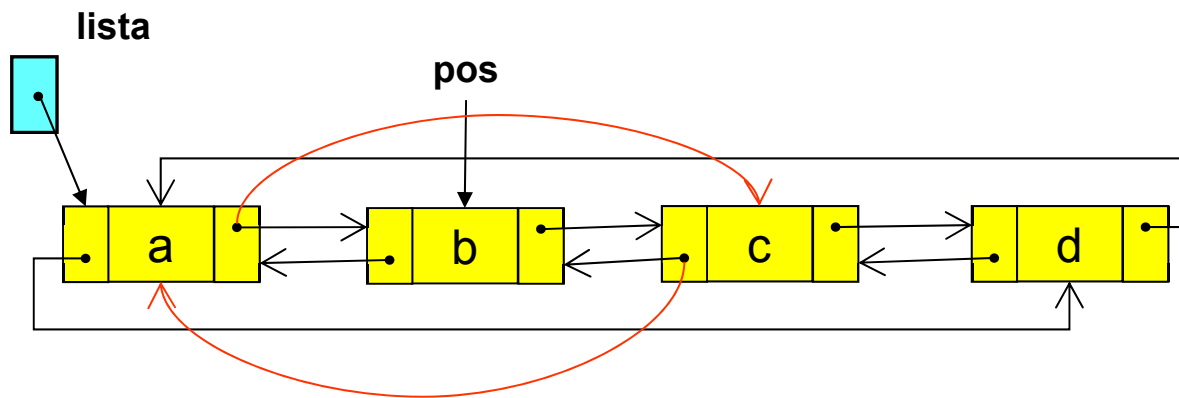
$pos \rightarrow \text{sig} \rightarrow ant = pos \rightarrow ant$

¿ $pos = lista$?

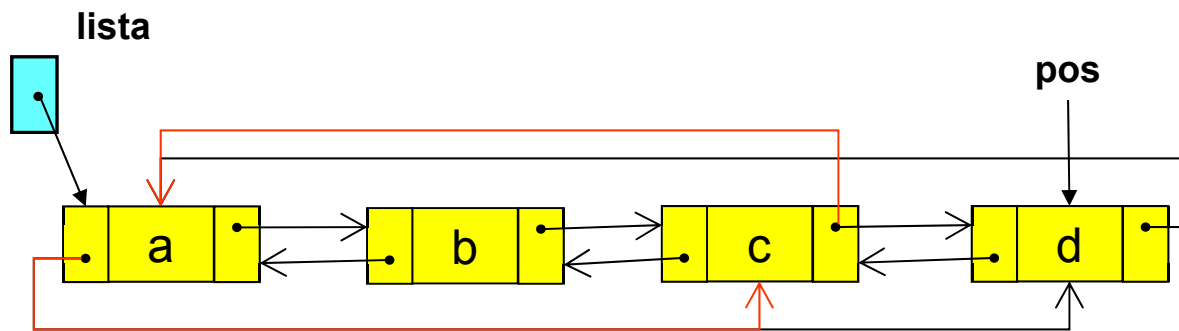
Sí: $lista = lista \rightarrow \text{sig}$

liberar espacio de memoria de pos

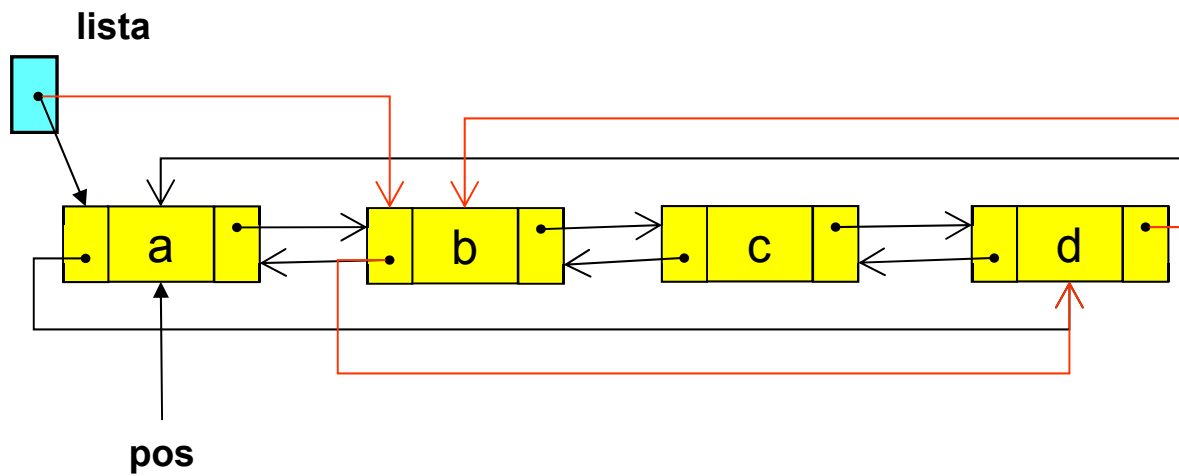
Eliminación en cualquier posición



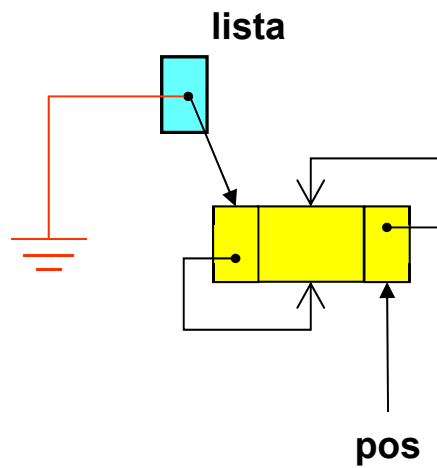
Eliminación en cualquier posición

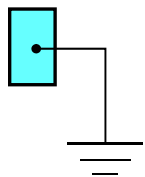
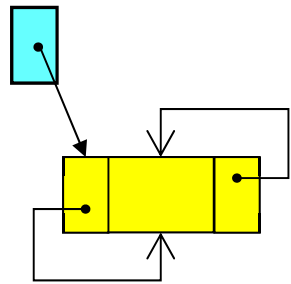
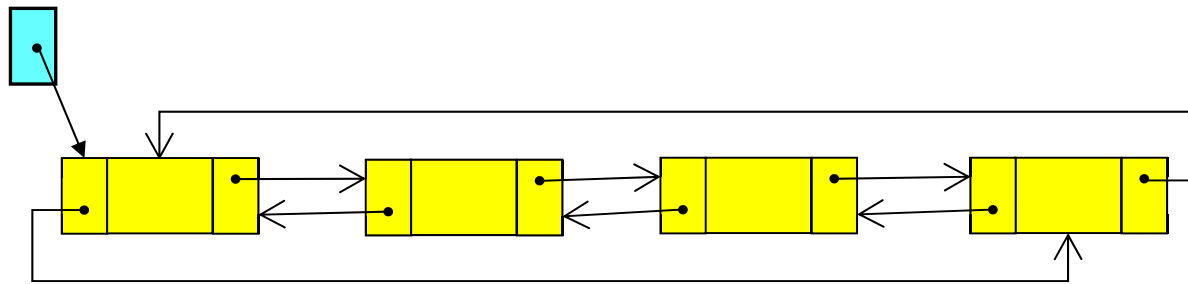


Eliminación en cualquier posición

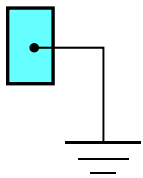
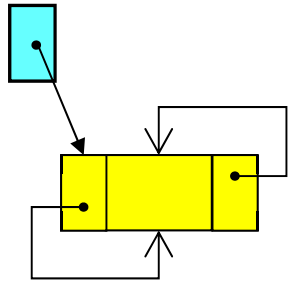
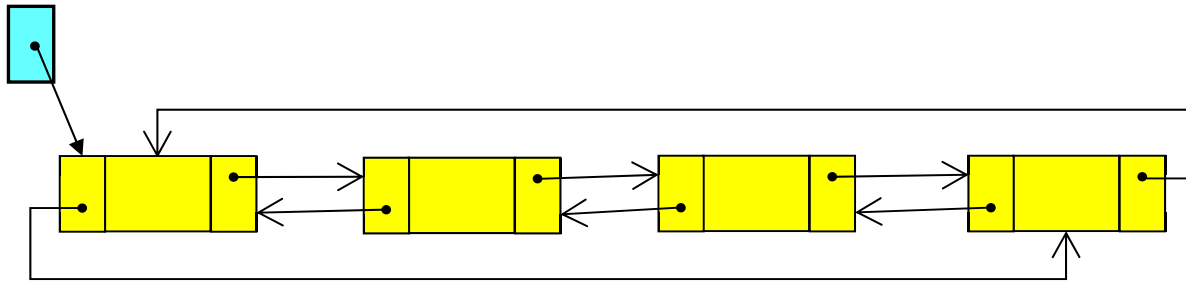


Eliminación del único elemento en la lista





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

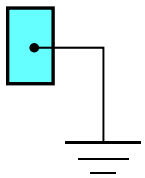
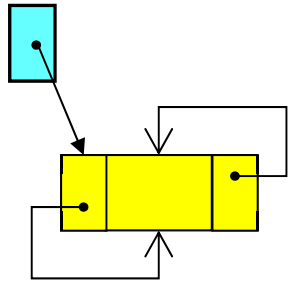
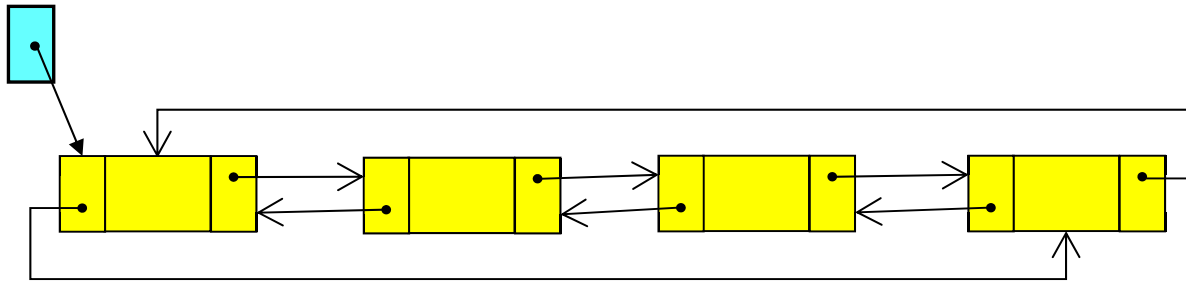


función: **último**
 recibe: *lista*
 regresa: posición

¿vacía(*lista*)?

Sí: regresar: NULO
 terminar

regresar: *lista* → ant

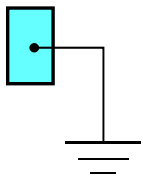
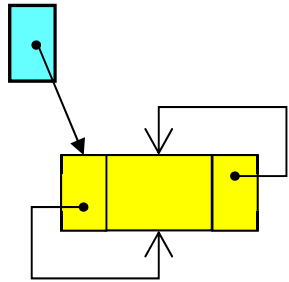
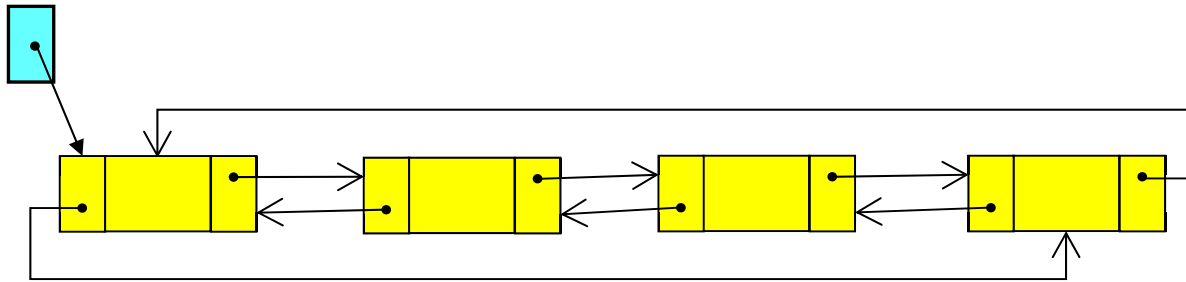


función. **anterior**
 recibe: *pos*, *lista*
 regresa: posición

¿**vacía**(lista) ó *pos* = NULO?

Sí: regresar: NULO
 terminar

regresar: $pos \rightarrow ant$



función: **siguiente**

recibe: *pos*, *lista*

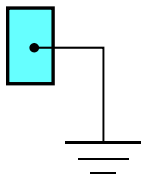
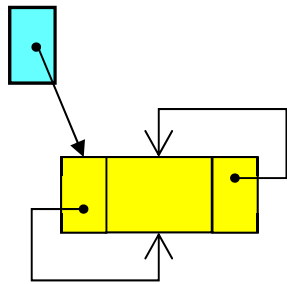
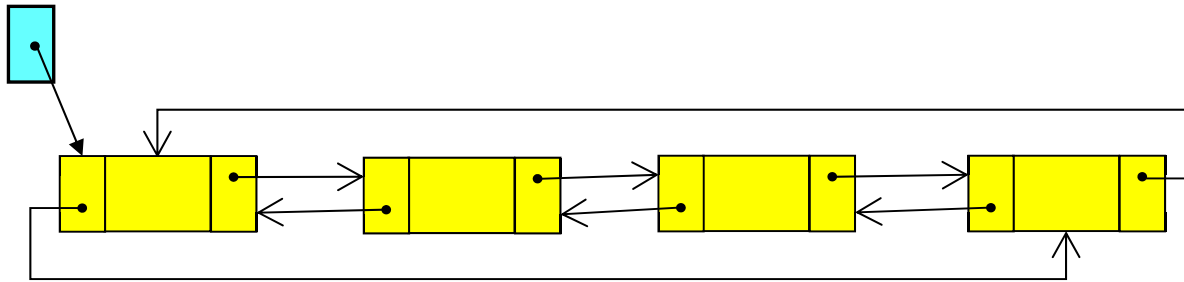
regresa: posición

¿vacía(*lista*) ó *pos* = NULO?

Sí: regresar: NULO

terminar

regresar: *pos* → sig



función: **localiza**

recibe: *elem*, *lista*

regresa: posición

¿**vacía**(*lista*)?

Sí: regresar: NULO

aux = *lista*

hacer:

¿*aux*→*elem* = *elem*?

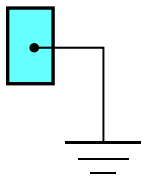
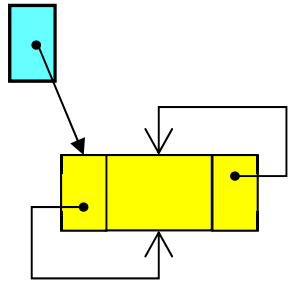
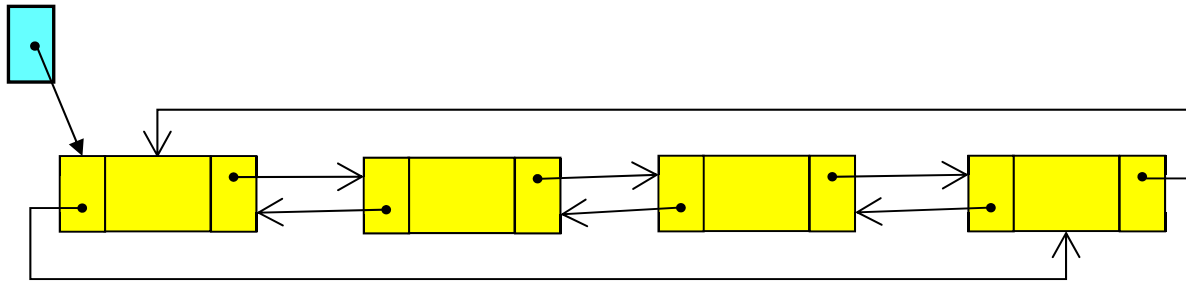
Sí: regresar: *aux*

salir

aux = *aux*→*sig*

mientras *aux* ≠ *lista*

regresar: NULO



función: **recupera**

recibe *pos*, *lista*

regresa: elemento

¿vacía(*lista*) ó *pos* = NULO?

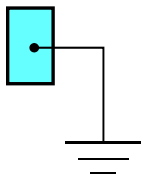
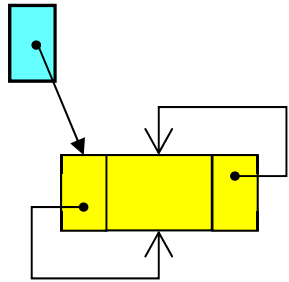
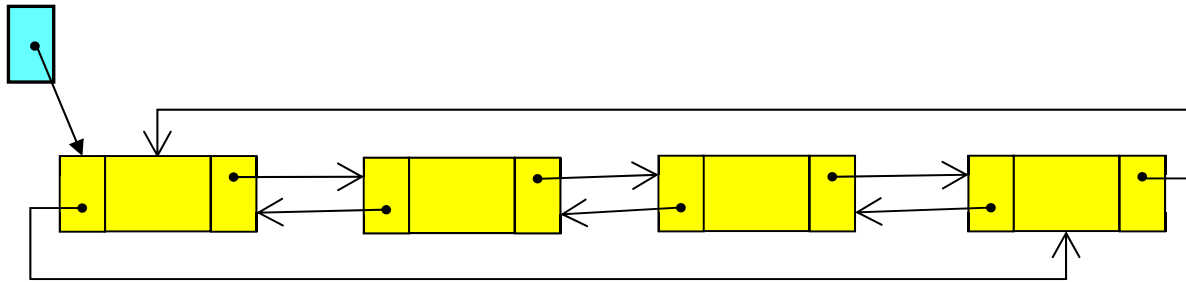
Sí: ¡error de excepción!

Insuficiencia de datos

terminar



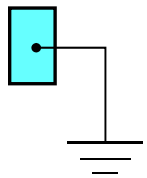
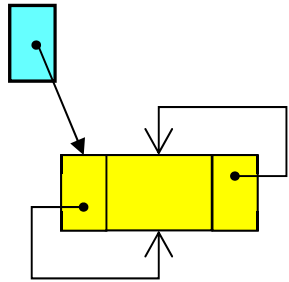
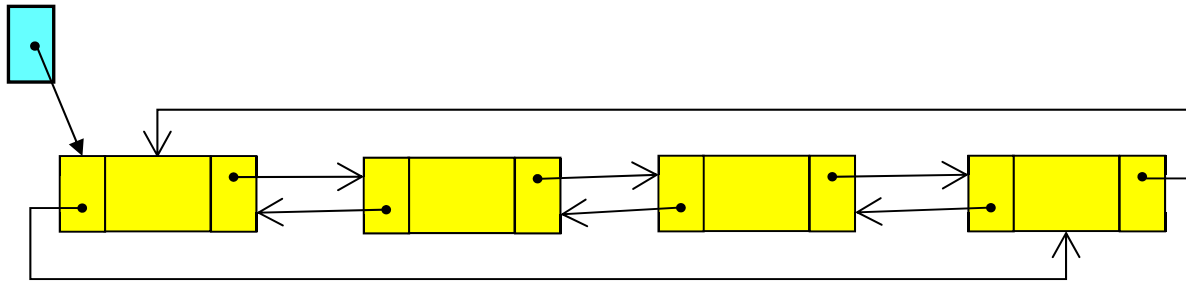
No: regresar: *pos* → elem



función: **imprime**
 recibe: *lista*
 regresa: nada

¿**vacía**(*lista*)?
 Sí: terminar

aux = *lista*
 hacer:
 imprimir *aux* → elem
 aux = *aux* → sig
 mientras *aux* ≠ *lista*



función: **anula**

recibe: *lista*

regresa: nada

¿**vacía**(*lista*)?

Sí: terminar

marca = *lista*

hacer:

aux = *lista*

lista = *lista* → sig

liberar espacio de memoria de aux

mientras *lista* ≠ marca

lista = NULO

