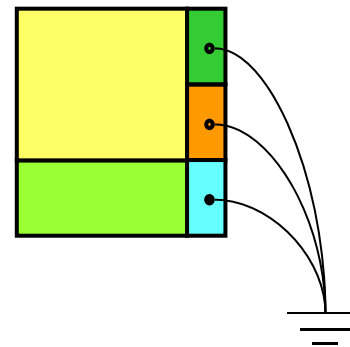
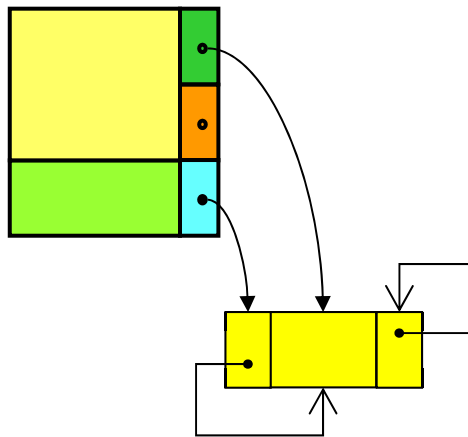
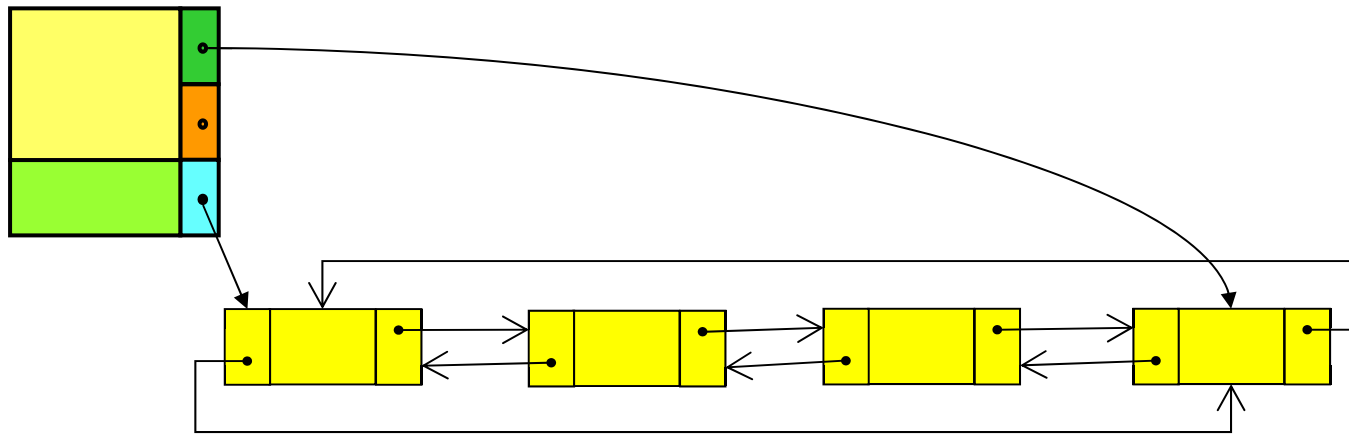


# Lista Doblemente Ligada Circular Con Encabezado (LDLCCE)

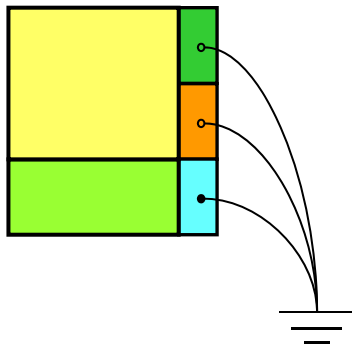
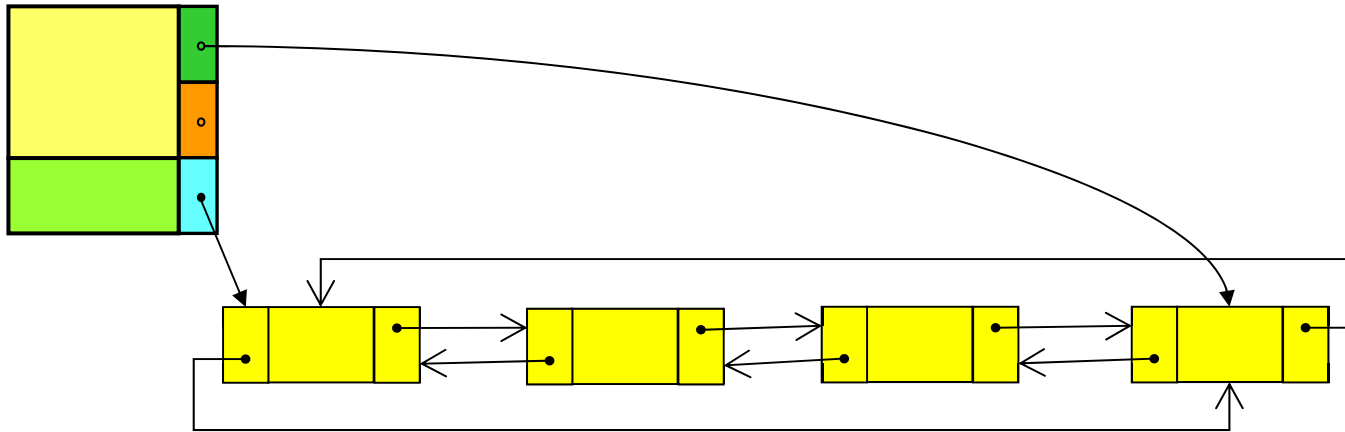


```
typedef _____ tipo_dato;
```

```
typedef struct tipo_nodo  
    tipo_dato elem;  
    struct tipo_nodo *ant, *sig;  
} tipo_nodo;
```

```
typedef tipo_nodo *tipo_pos;
```

```
typedef struct {  
    tipo_pos ancla;  
    tipo_pos aux;  
    tipo_pos ultimo;  
    unsigned int cantidad;  
    char descripcion[40];  
} tipo_lista;
```



función: **inicializa**

recibe: *lista (ancla, aux, ultimo, cantidad, descripción),  
descr*

regresa: nada

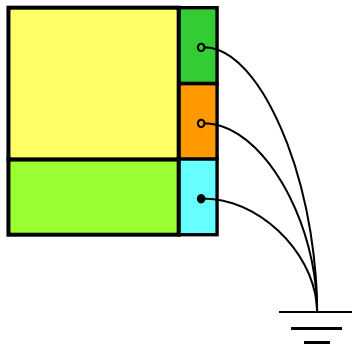
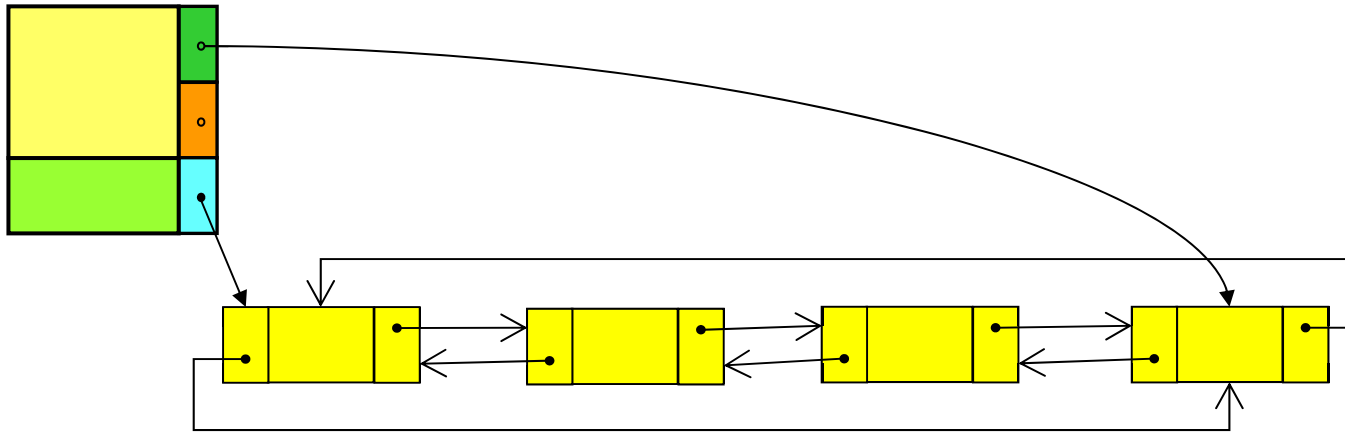
*ancla* = NULO

*aux* = NULO

*ultimo* = NULO

*cantidad* = 0

*descripción* = *descr*



función: **vacía**

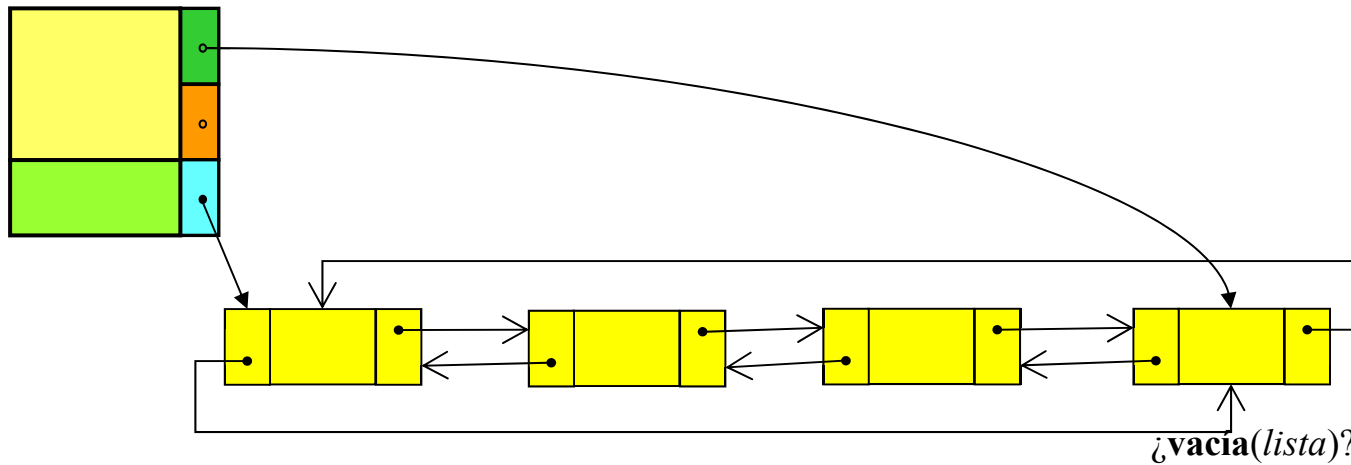
recibe: *lista* (*ancla*, *aux*, *ultimo*, *cantidad*, *descripción*)

regresa: nada

$\text{¿}ancla = \text{NULO?}$

Sí: regresar: *verdadero*

No: regresar: *falso*

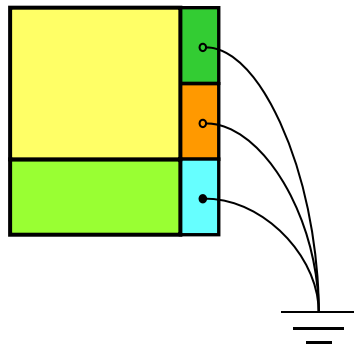


función: **inserta**

recibe: *elem*, *pos*,  
*lista* (*ancla*, *aux*, *ultimo*, *cantidad*, *descripción*)

regresa: nada

*aux* = nuevo nodo  
*aux*→*elem* = *elem*



Sí: *aux*→*sig* = *aux*

*aux*→*ant* = *aux*

*ancla* = *aux*

No: ¿*pos* = NULO?

Sí: *aux*→*sig* = *ancla*

*aux*→*ant* = **ultimo**(*ancla*)

**ultimo**(*lista*)→*sig* = *aux*

*ancla* = *aux*

No: *aux*→*ant* = *pos*

*aux*→*sig* = *pos*→*sig*

*pos*→*sig*→*ant* = *aux*

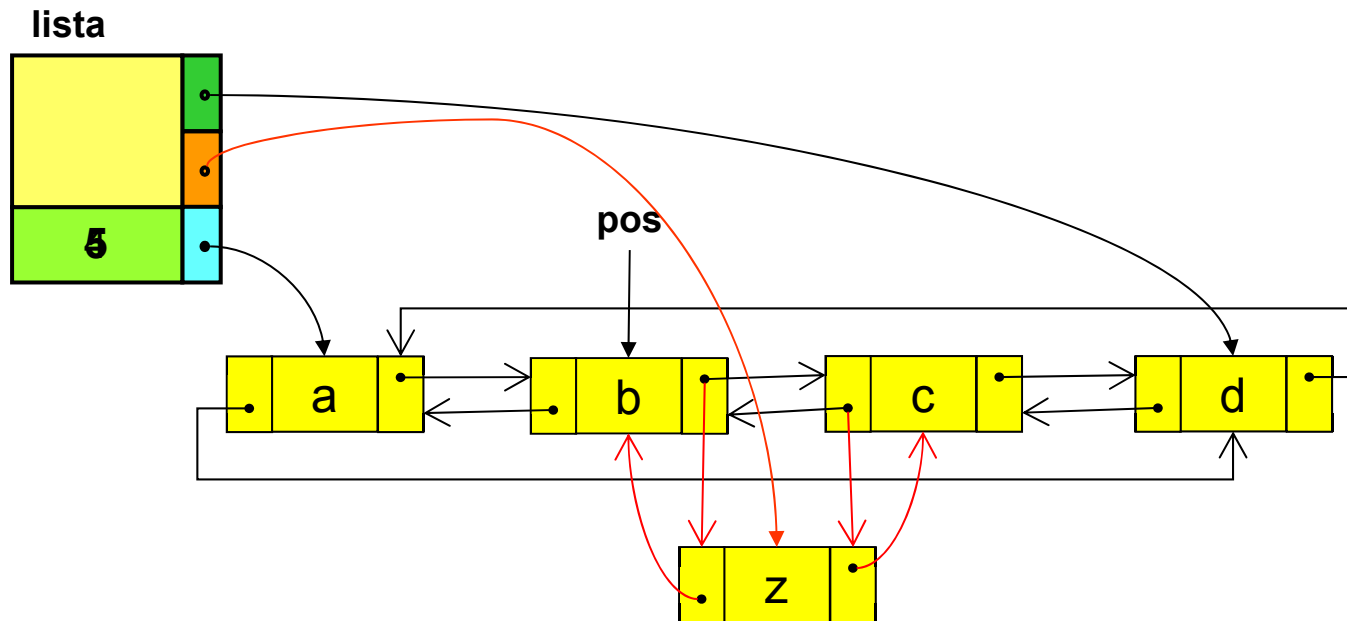
*pos*→*sig* = *aux*

¿*aux*→*sig* = *ancla*?

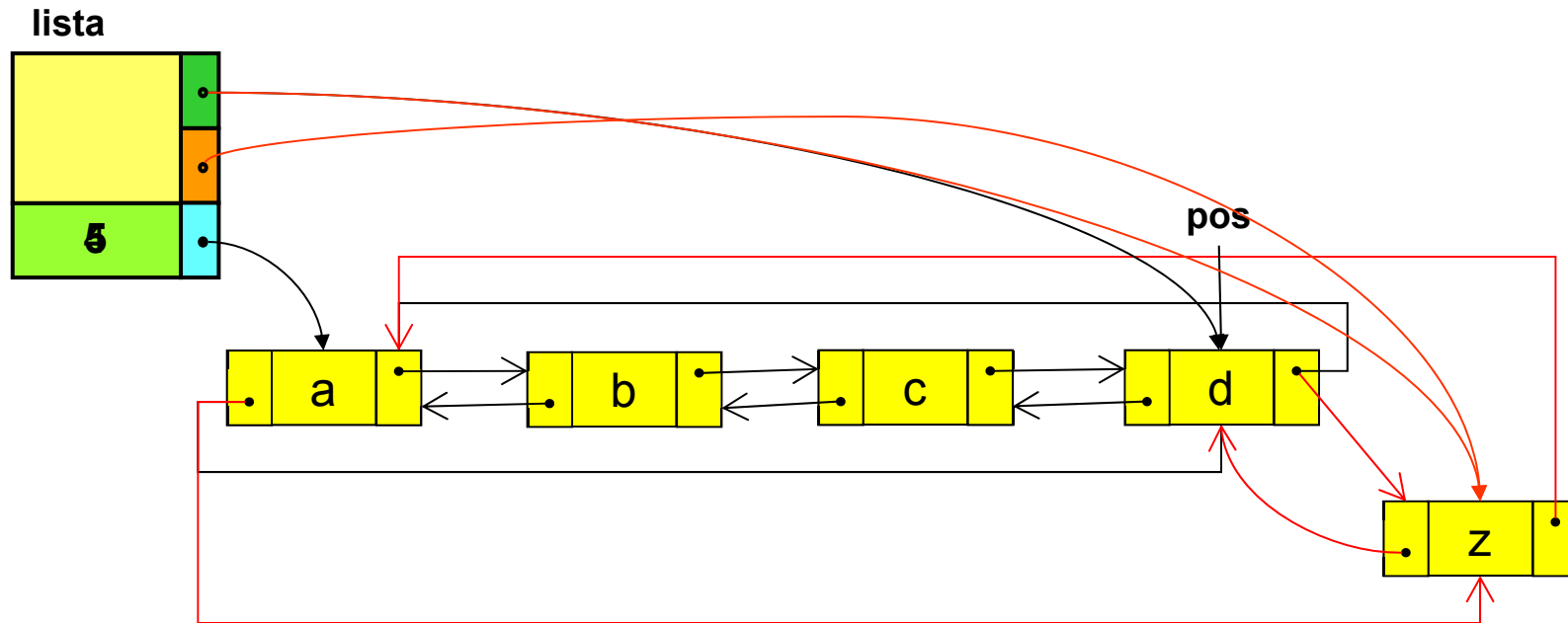
Sí: *ultimo* = *aux*

*cantidad* = *cantidad* + 1

Insertión  
en cualquier posición,  
excepto al principio

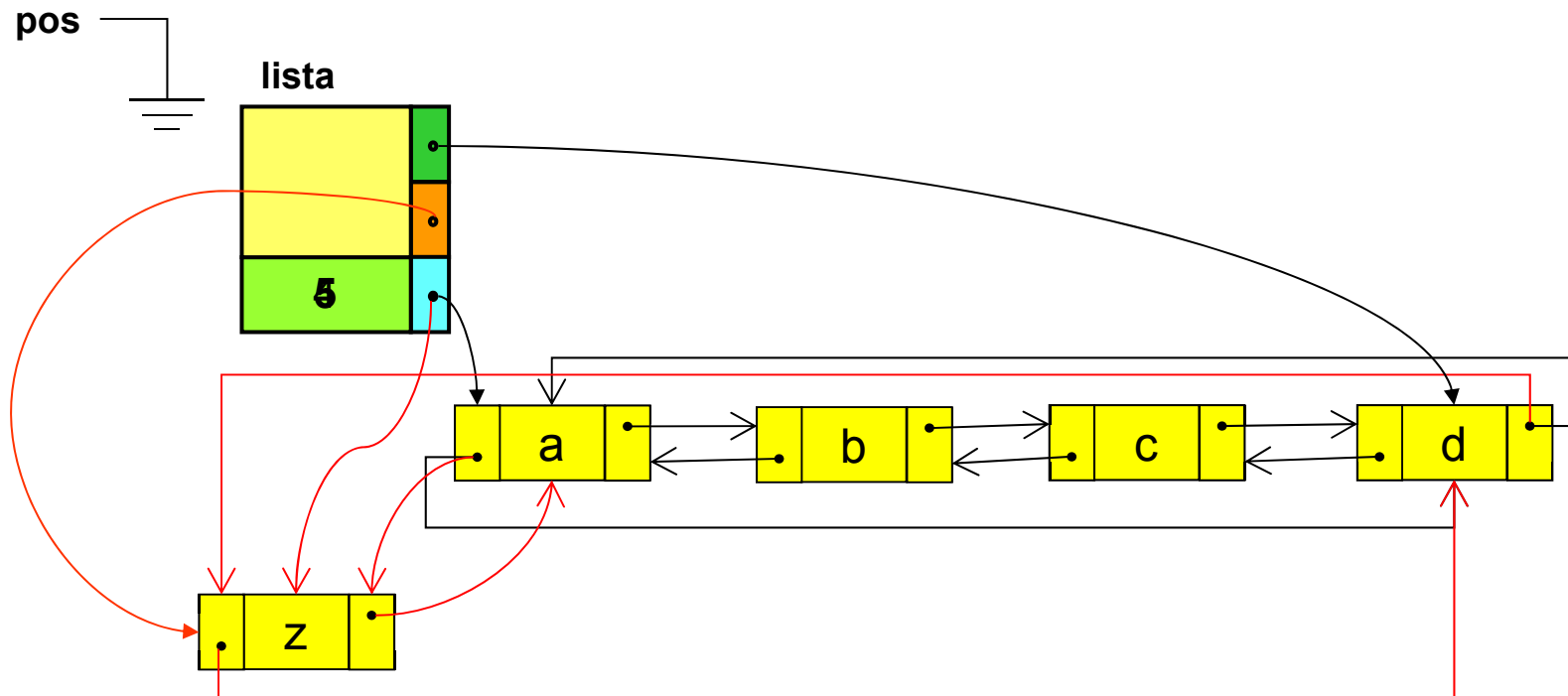


Insertión  
en cualquier posición,  
excepto al principio

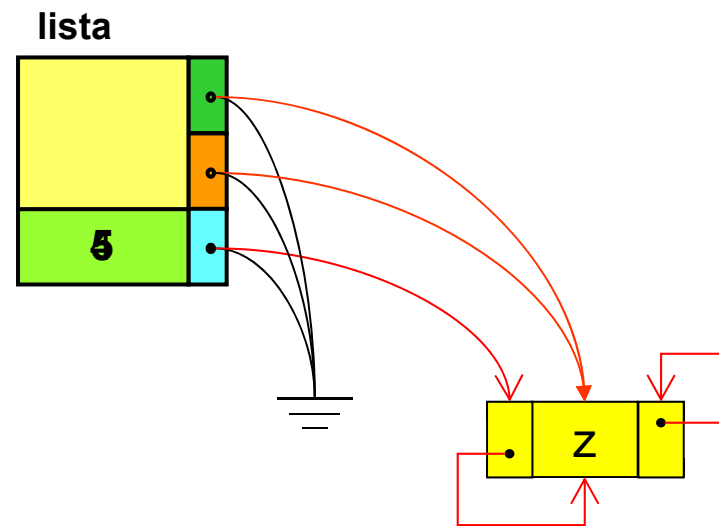


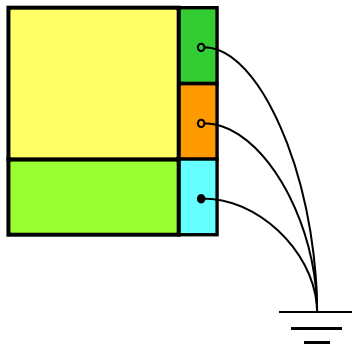
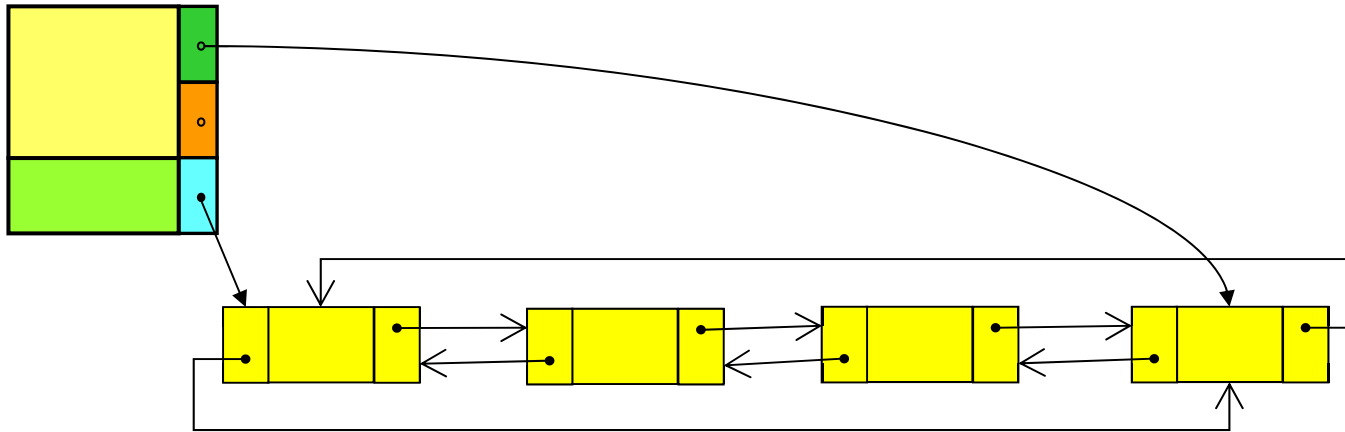


Insertión  
al principio



## Inserción En lista vacía





función: **elimina**

recibe: *pos*,

*lista* (*ancla*, *aux*, *ultimo*, *cantidad*, *descripción*)

regresa: nada

¿**vacía**(*lista*) ó *pos* = NULO?

Sí: terminar

¿*pos*→sig = *pos*?

Sí: *ancla* = NULO

*aux* = NULO

*ultimo* = NULO

No: *pos*→*ant*→sig = *pos*→sig

*pos*→sig→*ant* = *pos*→*ant*

¿*pos* = *ancla*?

Sí: *ancla* = *ancla*→sig

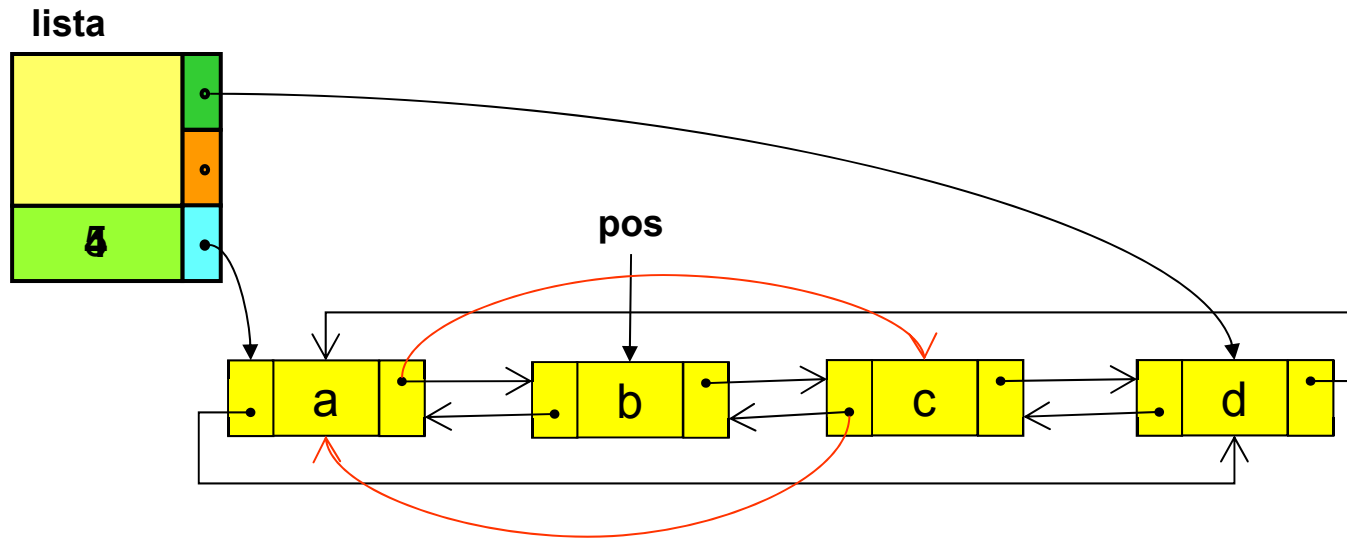
¿*pos*→sig = *ancla*?

Sí: *ultimo* = *pos*→*ant*

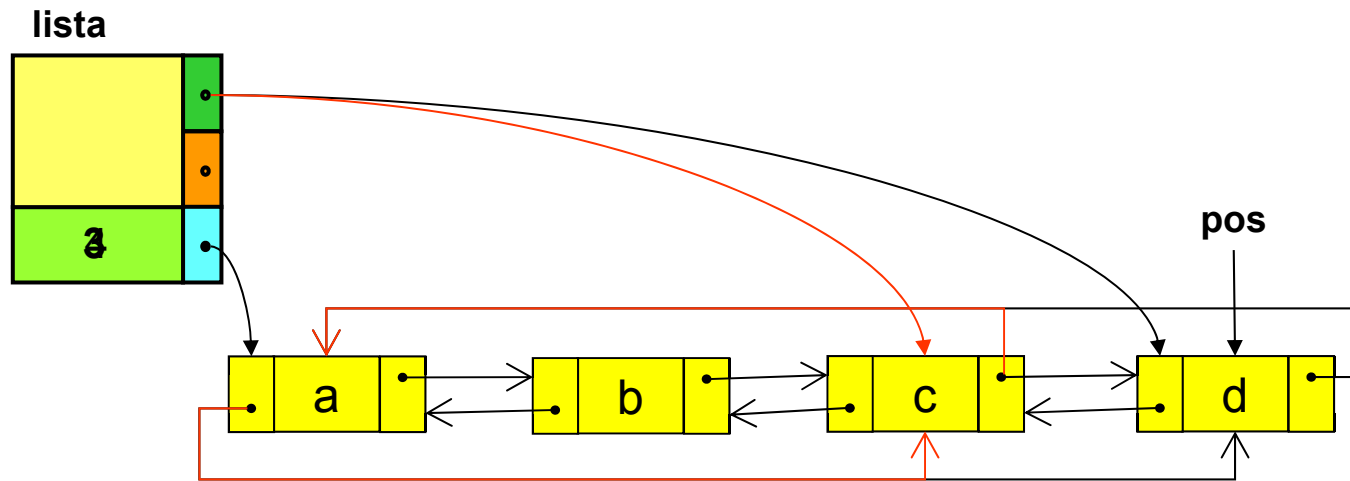
*cantidad* = *cantidad* - 1

liberar espacio de memoria de *pos*

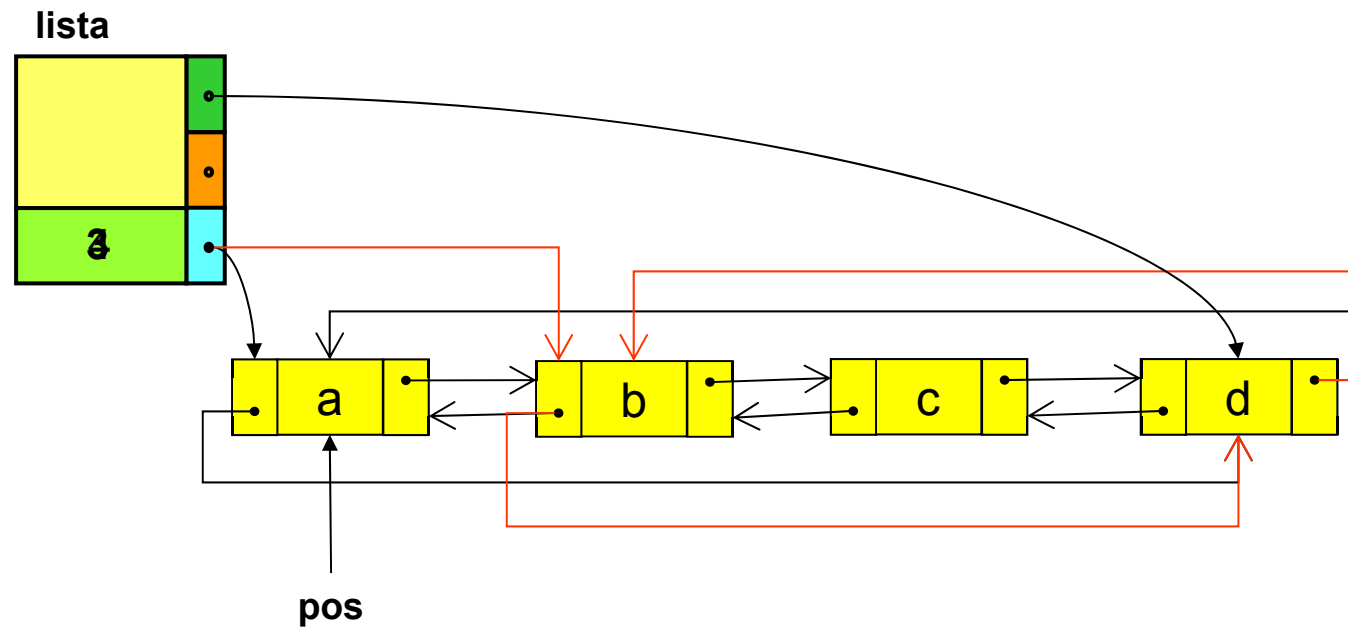
## Eliminación en cualquier posición



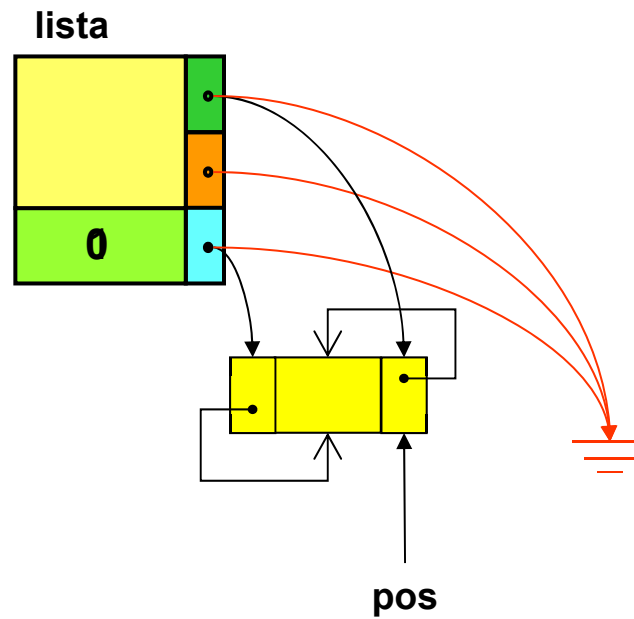
## Eliminación en cualquier posición

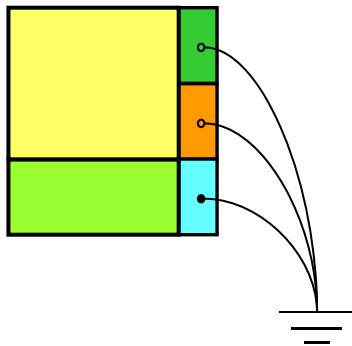
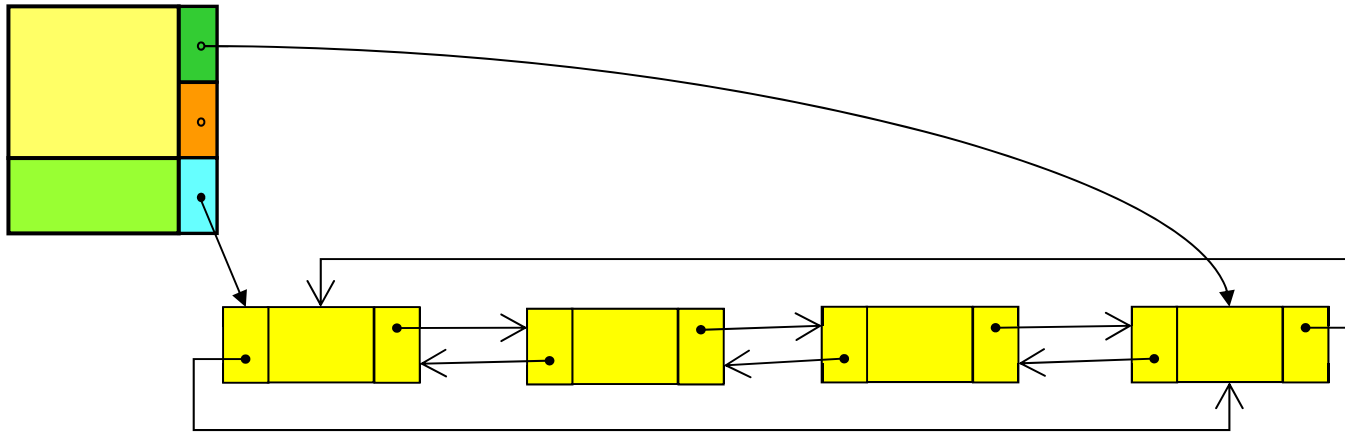


## Eliminación en cualquier posición



Eliminación  
del único elemento  
en la lista





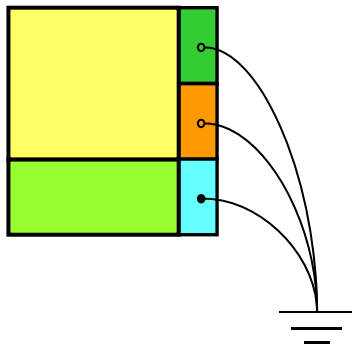
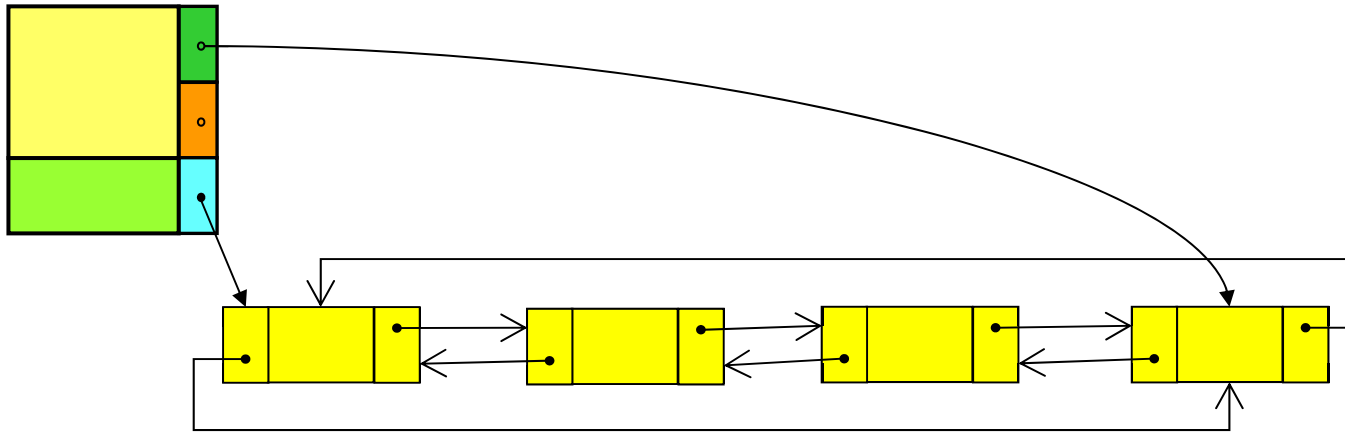
función: ***primero***

recibe: *lista (ancla, aux, ultimo, cantidad, descripción*

regresa: posición

regresar: *ancla*





función: **último**

recibe: *lista* (*ancla*, *aux*, *ultimo*, *cantidad*, *descripción*)

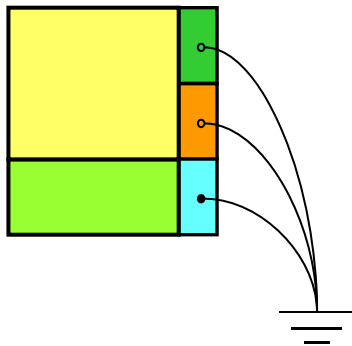
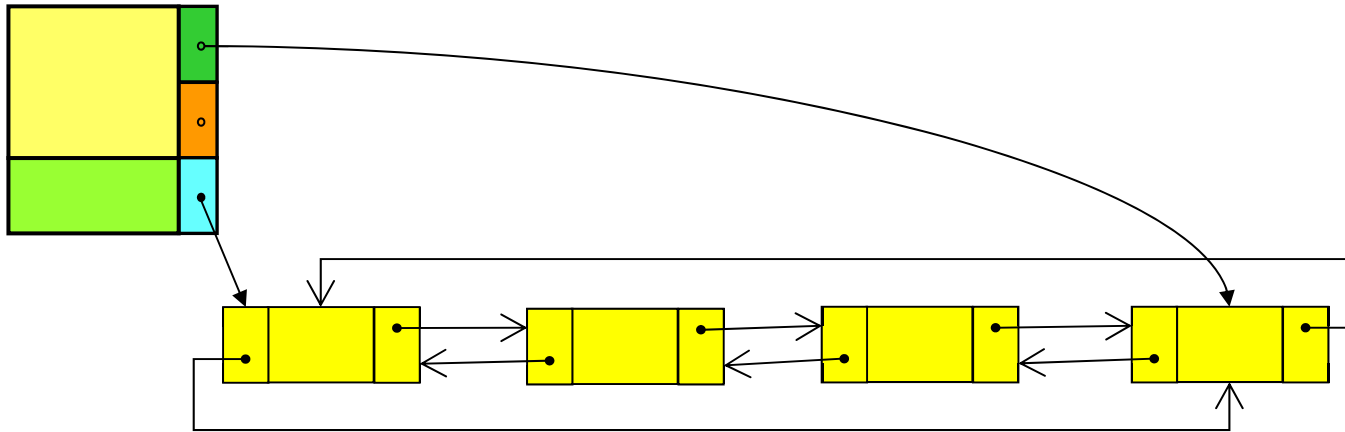
regresa: posición

¿vacía(*lista*)?

Sí: regresar: NULO

terminar

regresar: *ancla* → ant



función. **anterior**

recibe: *pos*,

*lista (ancla, aux, ultimo, cantidad, descripción*

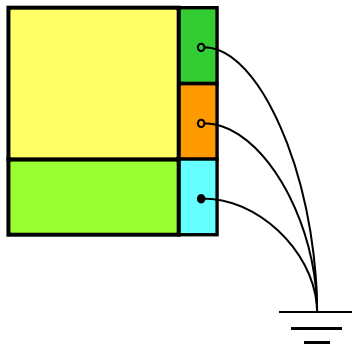
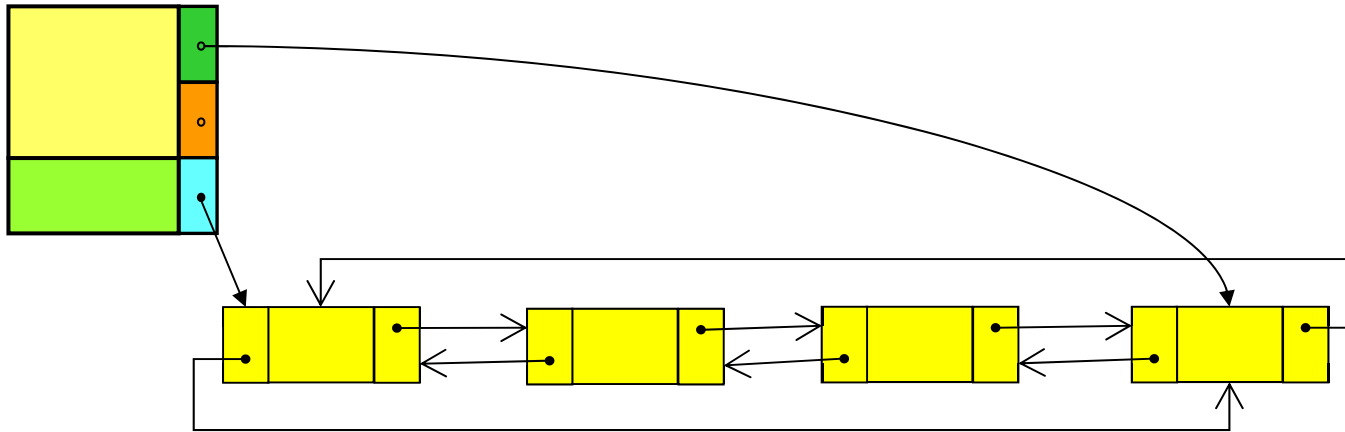
regresa: posición

¿**vacía**(lista) ó *pos* = NULO?

Sí: regresar: NULO

terminar

regresar: *pos* → ant



función: **siguiente**

recibe: *pos*,

*lista (ancla, aux, ultimo, cantidad, descripción)*

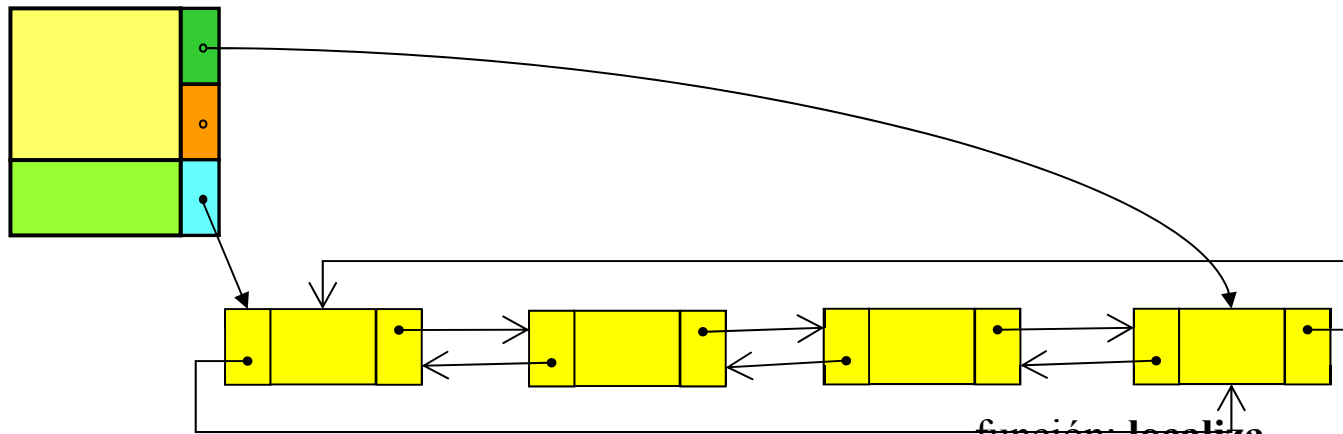
regresa: posición

¿vacía(*lista*) ó *pos* = NULO?

Sí: regresar: NULO

terminar

regresar: *pos* → sig

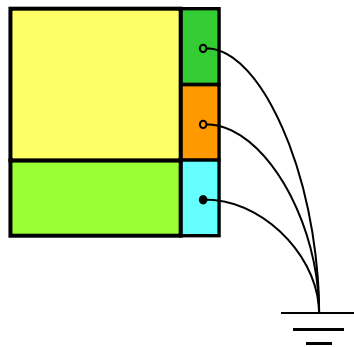


función: **localiza**

recibe: *elem*,

*lista (ancla, aux, ultimo, cantidad, descripción )*

regresa: posición



¿**vacía**(*lista*)?

Sí: regresar: NULO

*aux* = *ancla*

hacer:

¿*aux*→*elem* = *elem*?

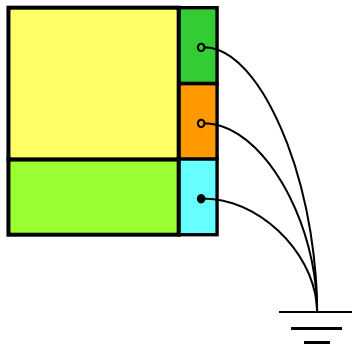
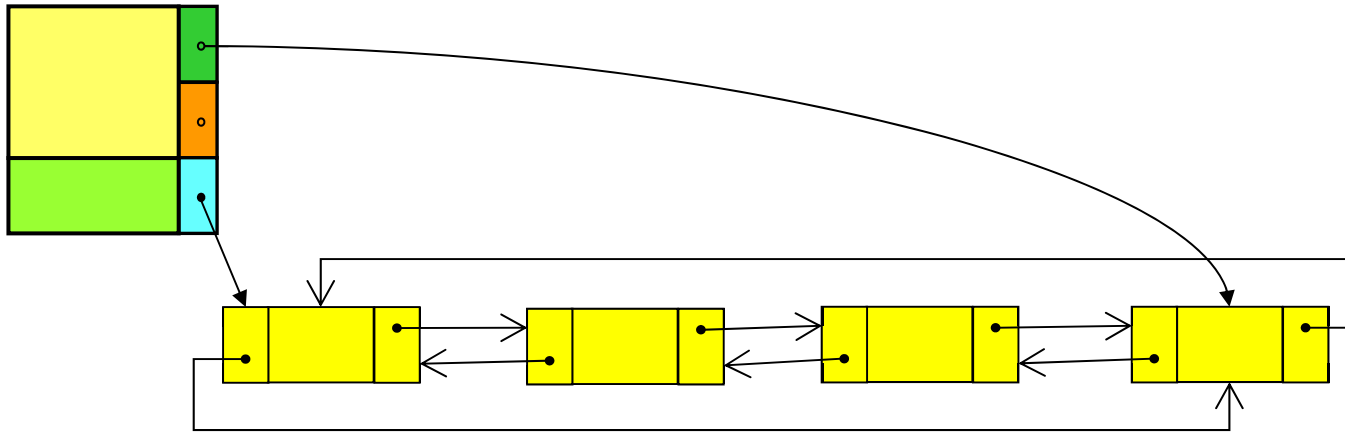
Sí: regresar: *aux*

salir

*aux* = *aux*→*sig*

mientras *aux* ≠ *ancla*

regresar: NULO



función: **recupera**

recibe *pos*,

*lista* (*ancla*, *aux*, *ultimo*, *cantidad*, *descripción* )

regresa: elemento

¿vacía(lista) ó *pos* = NULO?

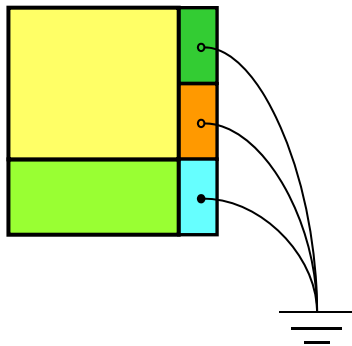
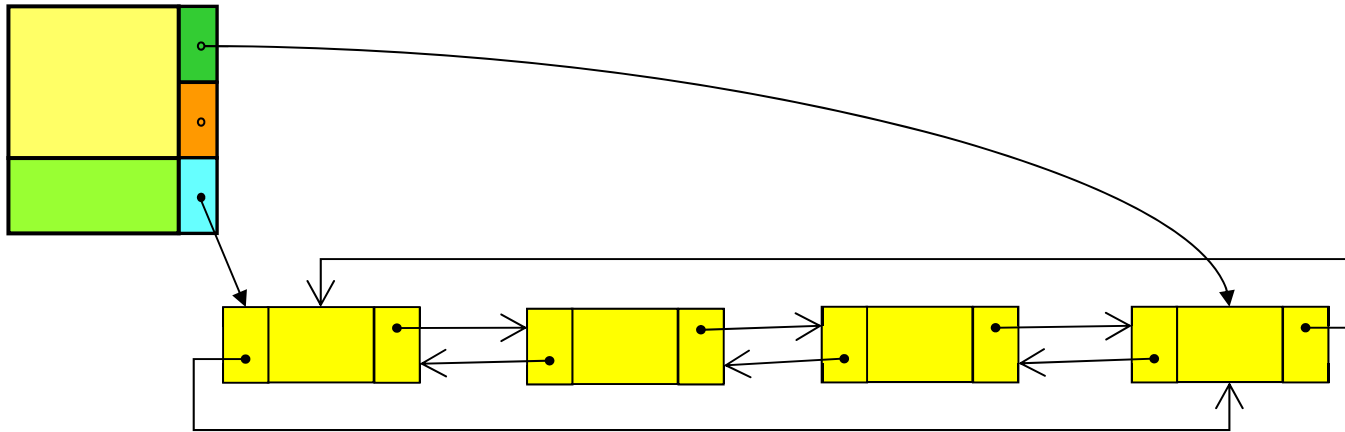
Sí: ¡error de excepción!

Insuficiencia de datos

terminar



No: regresar: *pos* → elem



función: **imprime**

recibe: *lista (ancla, aux, ultimo, cantidad, descripción )*

regresa: nada

¿**vacía**(*lista*)?

Sí: terminar

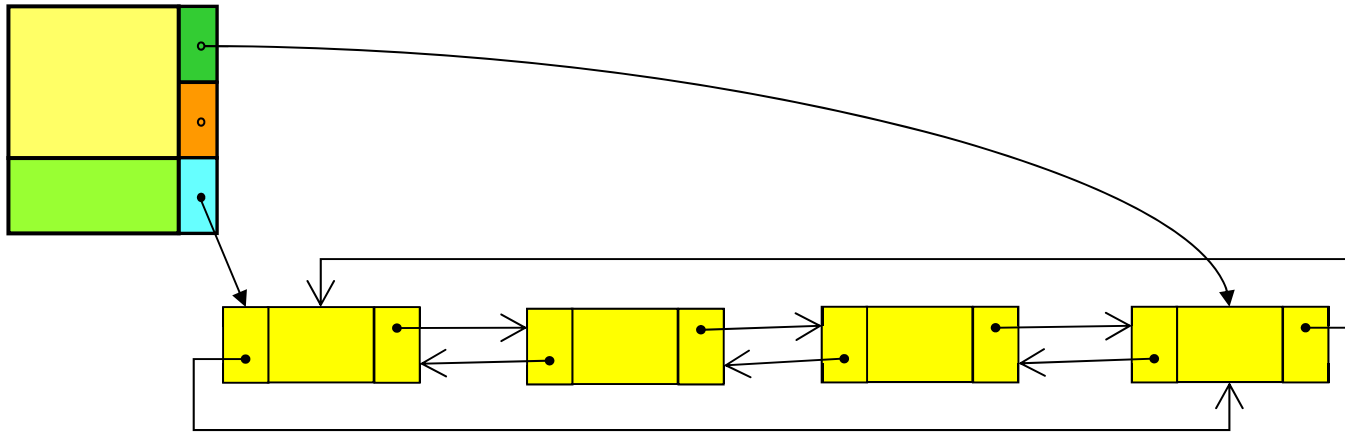
*aux* = *ancla*

hacer:

imprimir *aux*→elem

*aux* = *aux*→sig

mientras *aux* ≠ *ancla*



función: **anula**

recibe: *lista (ancla, aux, ultimo, cantidad, descripción )*

regresa: nada

¿**vacía**(*lista*)?

Sí: terminar

marca = *ancla*

hacer:

*aux* = *ancla*

*ancla* = *ancla* → sig

liberar espacio de memoria de *aux*

mientras *ancla* ≠ marca

*ancla* = NULO

*aux* = NULO

*ultimo* = NULO

*cantidad* = 0

