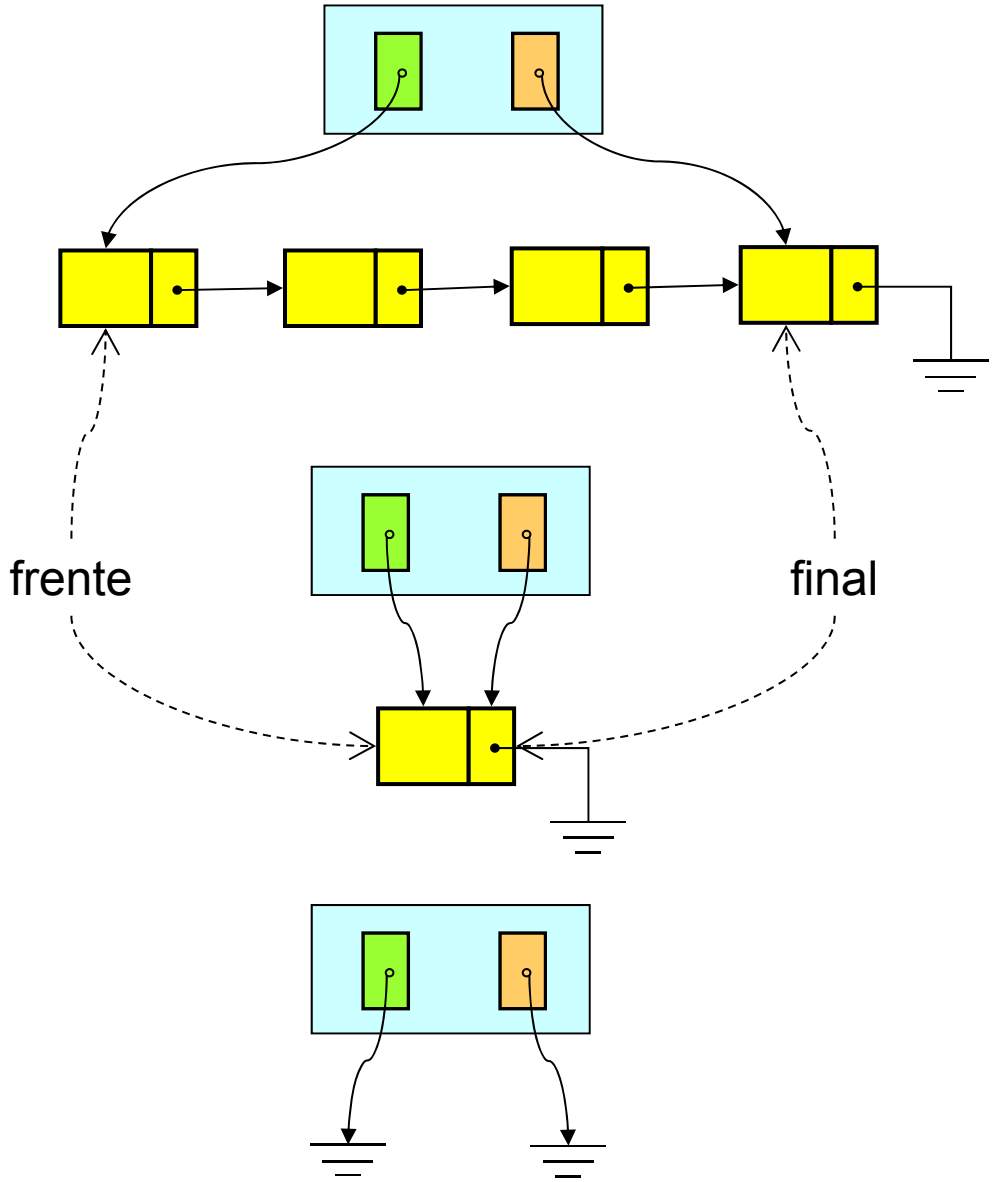
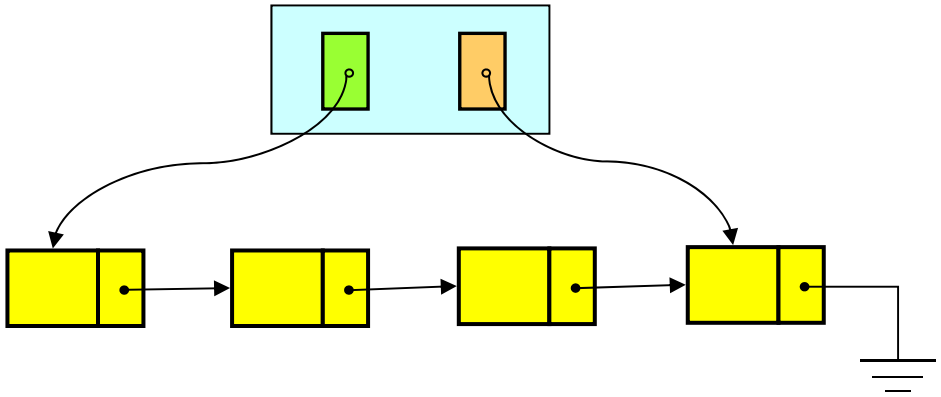
Cola
Implementada mediante una Lista Simplemente Ligada Lineal Con Encabezado

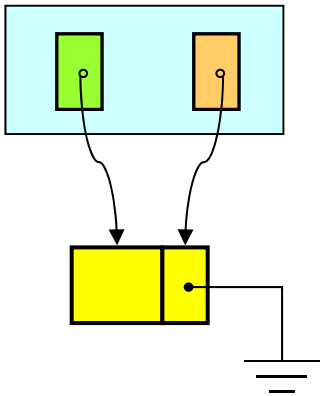

frente
final

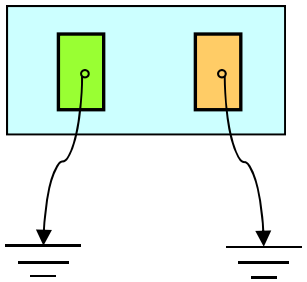
```
typedef  _________ tipo_dato;

typedef struct tipo_nodo {
        tipo_dato elem;
        tipo_nodo *sig;
        } tipo_nodo;

typedef struct {
        tipo_nodo *frente;
        tipo_nodo *final;
        } tipo_cola;
```



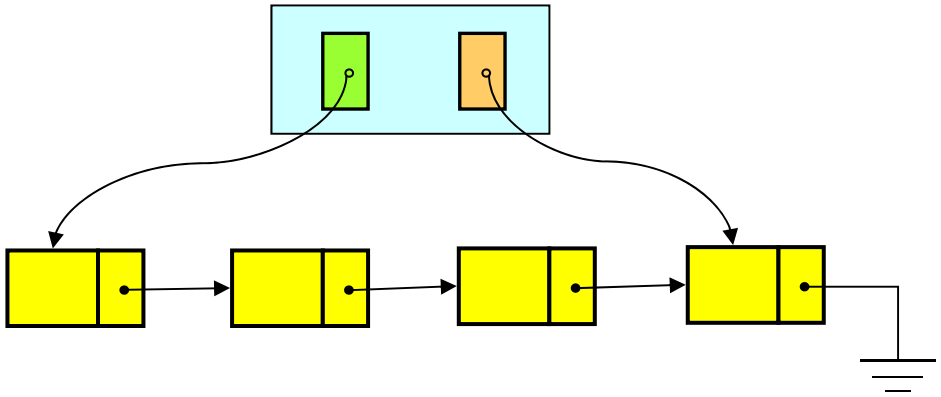


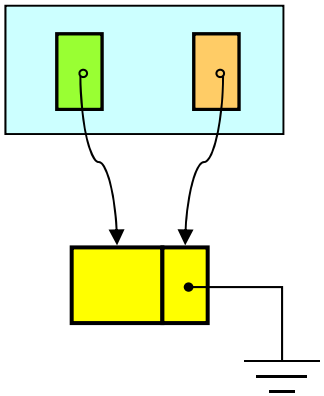


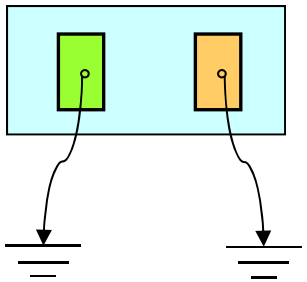
función: **inicializa**
recibe: *cola*(*frente*, *final*)
regresa: nada

frente = NULO

final = NULO

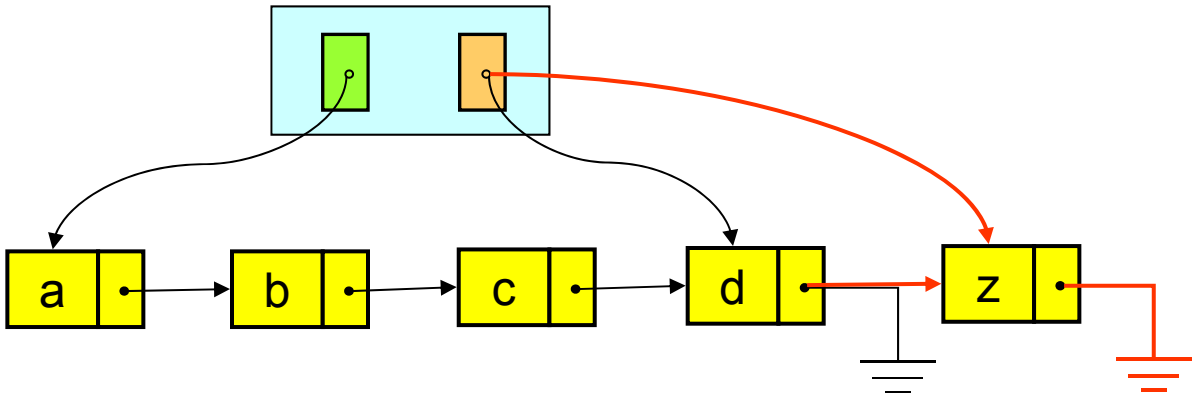


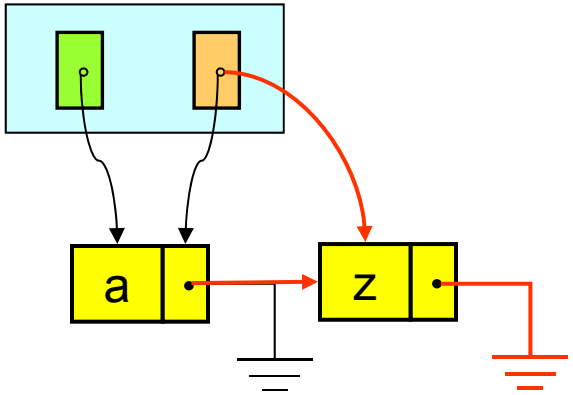


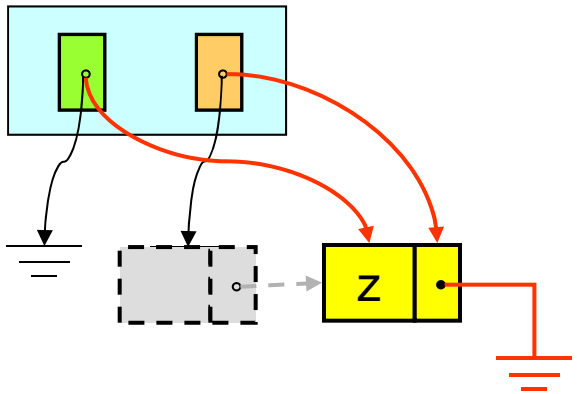


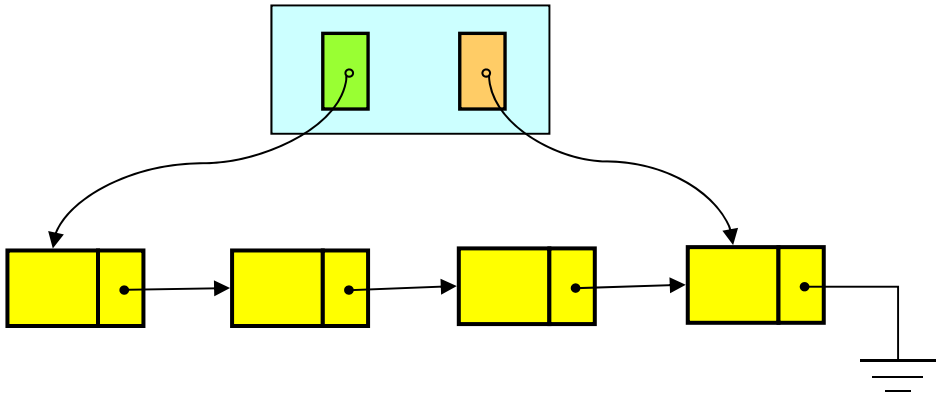
función: **vacía**
recibe: *cola*(*frente*, *final*)
regresa: booleano

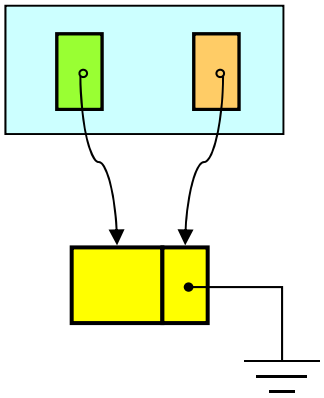
¿*frente* = NULO?
Sí: regresar: *verdadero*
No: regresar: *falso*

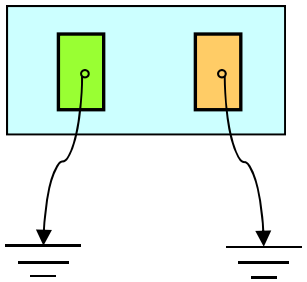

a
b
c
d
z


a
z


Z







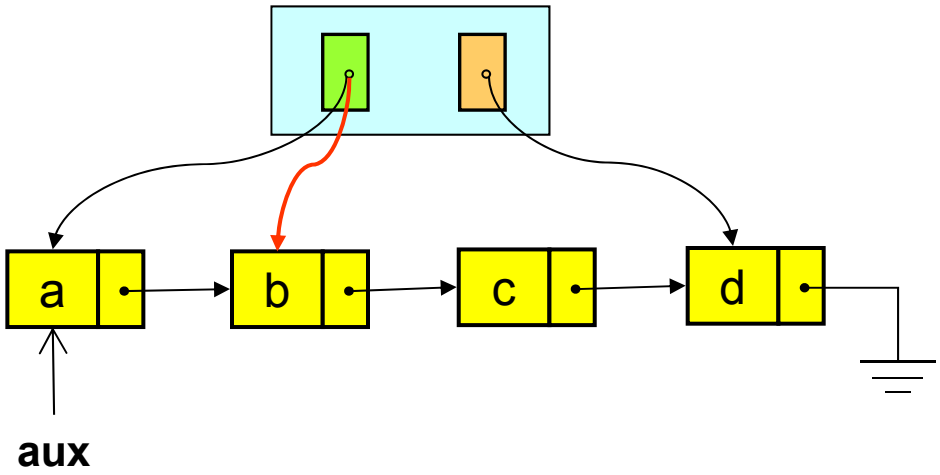
función: **enqueue**
recibe: *elemento*, *cola*(*frente*, *final*)
regresa: nada

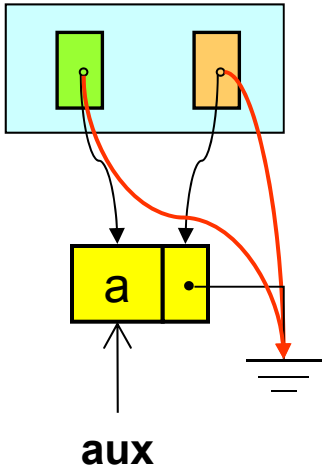
aux = nuevo nodo
aux→elem = *elem*
aux→sig = NULO

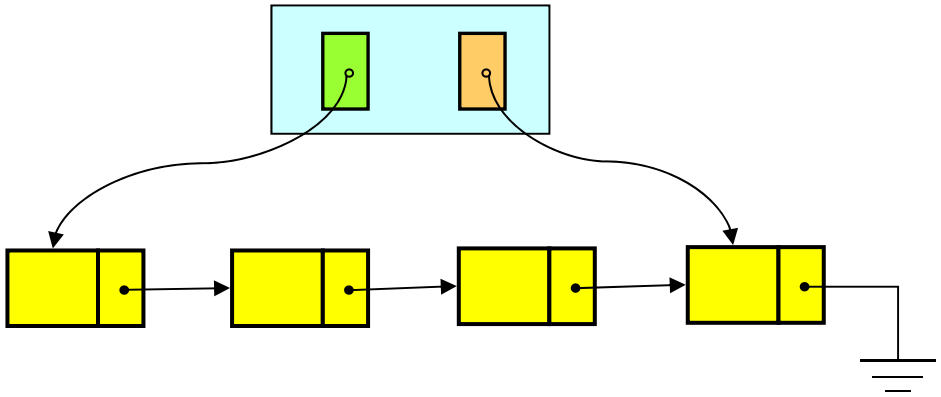
¿*final* ≠ NULO?
Sí: *final*→sig = aux

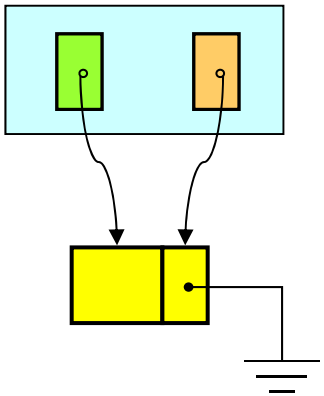
final = aux

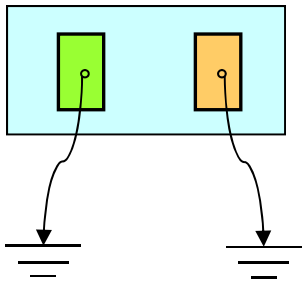
¿*frente* = NULO?
Sí: *frente* = aux


a
b
c
d
aux


a
aux







función: **dequeue**
recibe: *cola*(*frente*, *final*)
regresa: nada

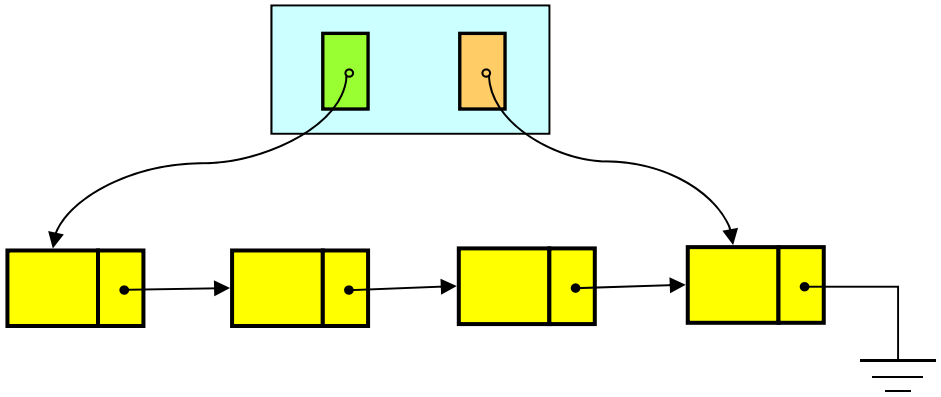
¿**vacía**(*cola*)?
Sí: Insuficiencia de datos
Terminar

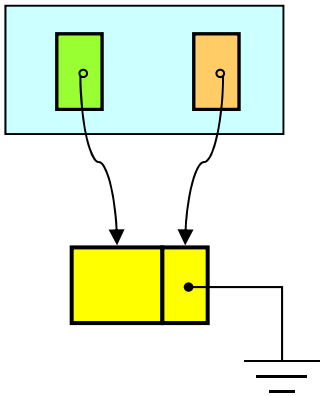
aux = *frente*

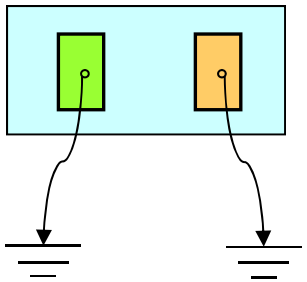
frente = *frente*→sig

¿frente = NULO*?*
Sí: *final* = NULO

liberar espacio de memoria de aux







función: **front**
recibe: *cola*(*frente*, *final*)
regresa: elemento

¿**vacía**(*cola*)?
Sí: ¡Error de excepción!
Insuficiencia de datos
Terminar

No: regresar: *frente*→elem