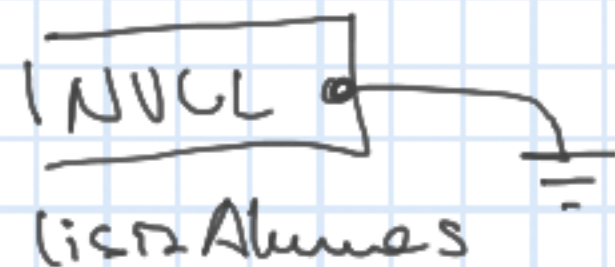


## Listas de alumnos

29/9

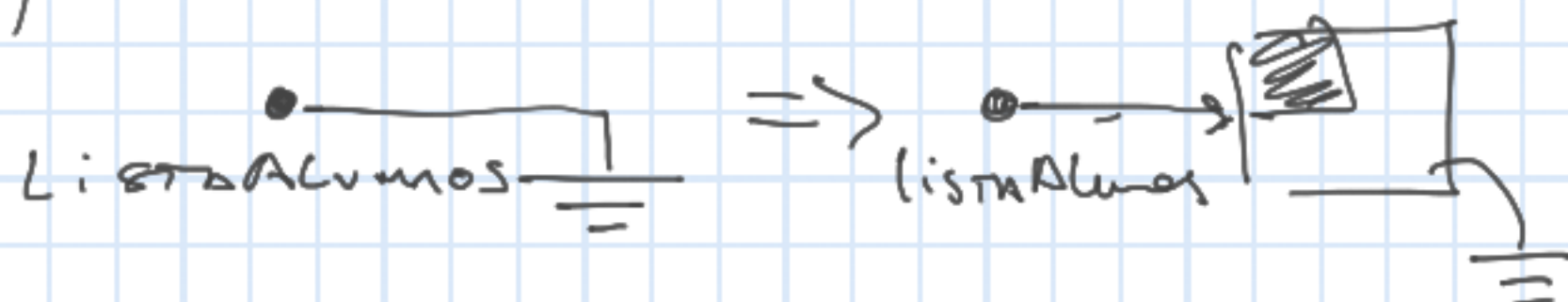
- Insertar Primero
- Insertar Ultimo
- Quitar Primero
- Quitar Ultimo



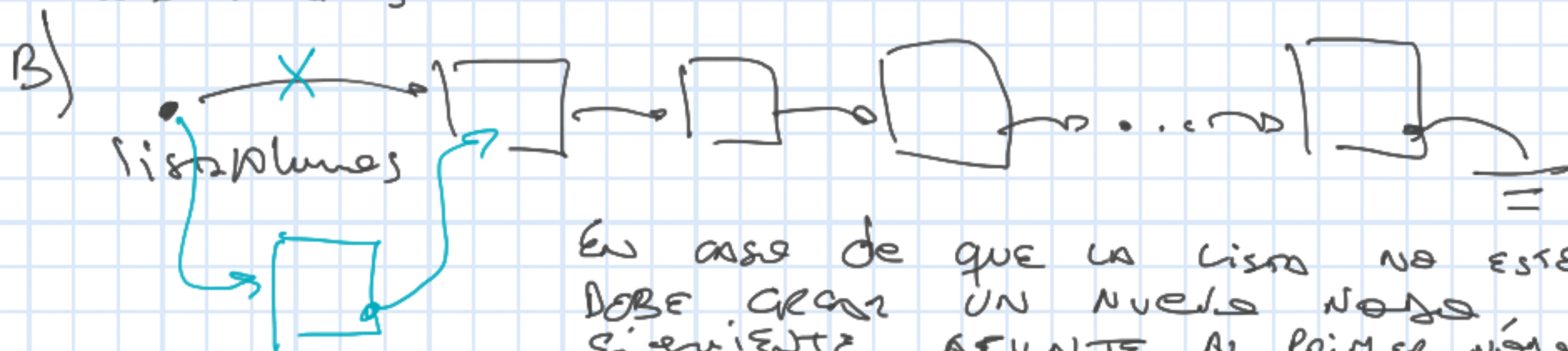
### 1) Insertar Primero

2 casos a analizar:

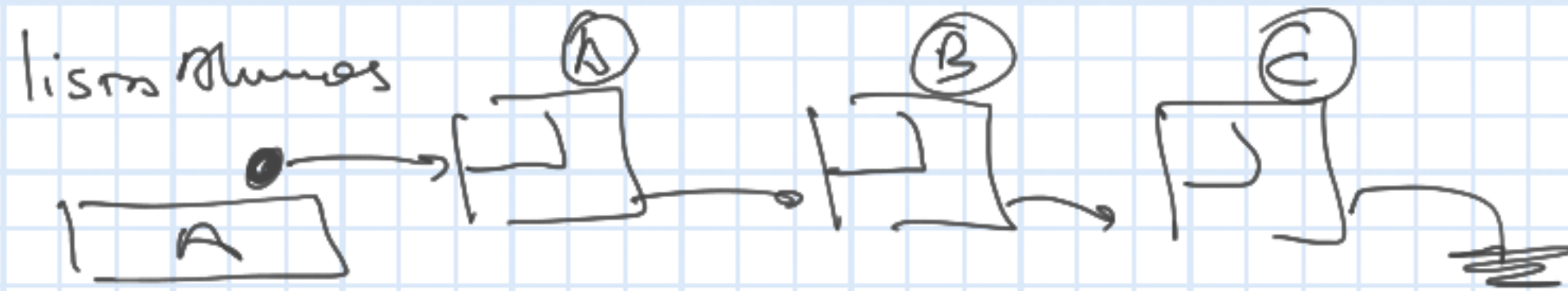
A) Lista vacía



En este caso debemos hacer `listaAlumnos = new Node()` y poner los valores

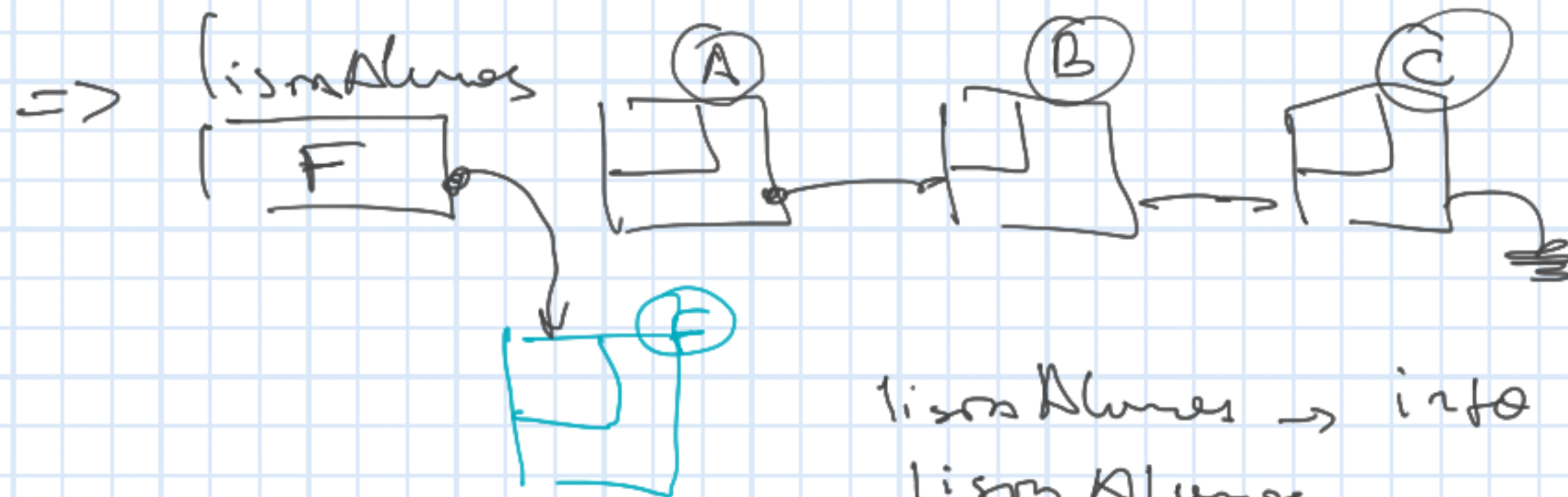


En caso de que la lista no este vacía se debe crear un nuevo nodo, hacer que su siguiente apunte al primer nodo de la lista y finalmente apuntar el inicio al nuevo nodo

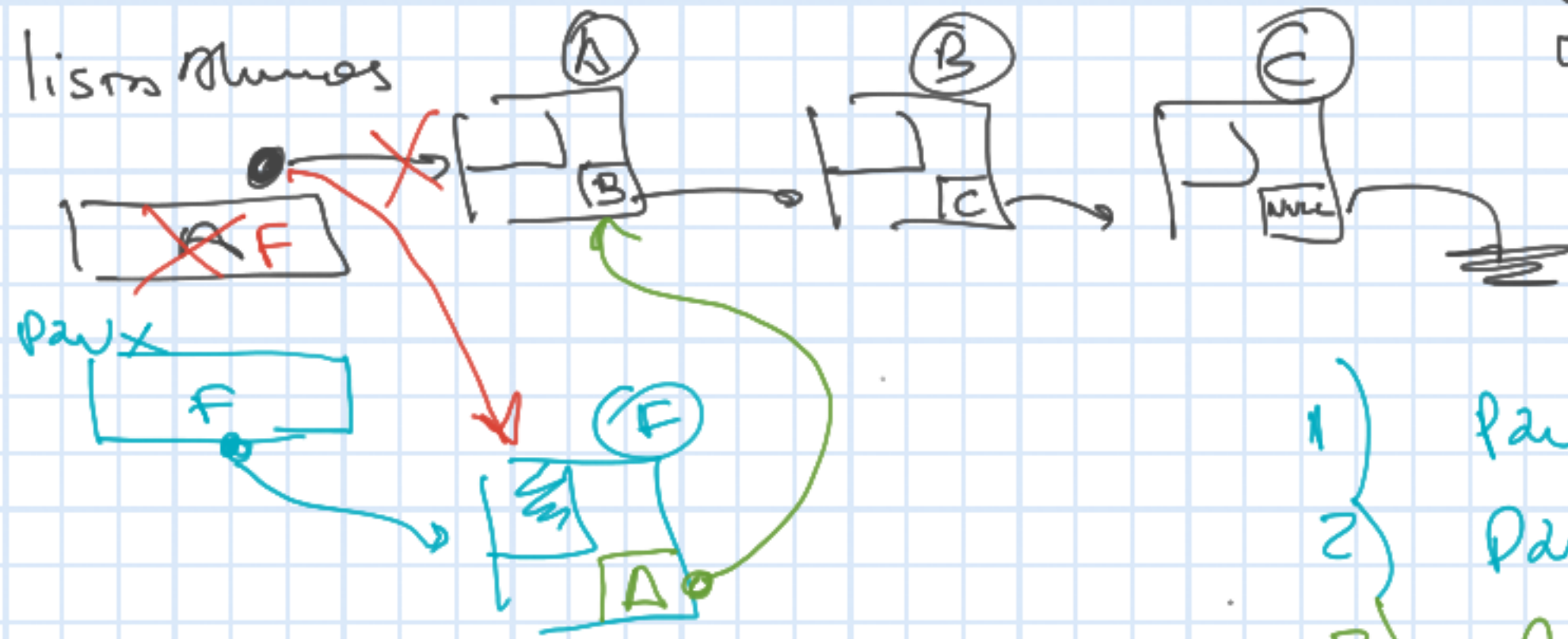


listaAlunos = new NodeListaSEAluno();

} Si hacemos esto perdemos toda la lista.



listaAlunos → info = unAluno;  
listaAlunos → sgte

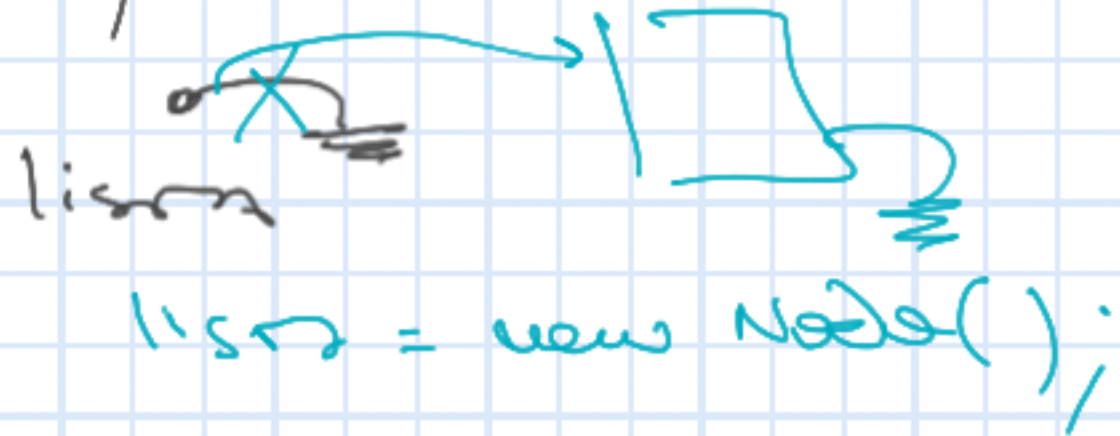


UNA MANERA DE HACER  
LA OPERACIÓN DE FORMAR  
SERIES SERÍA:

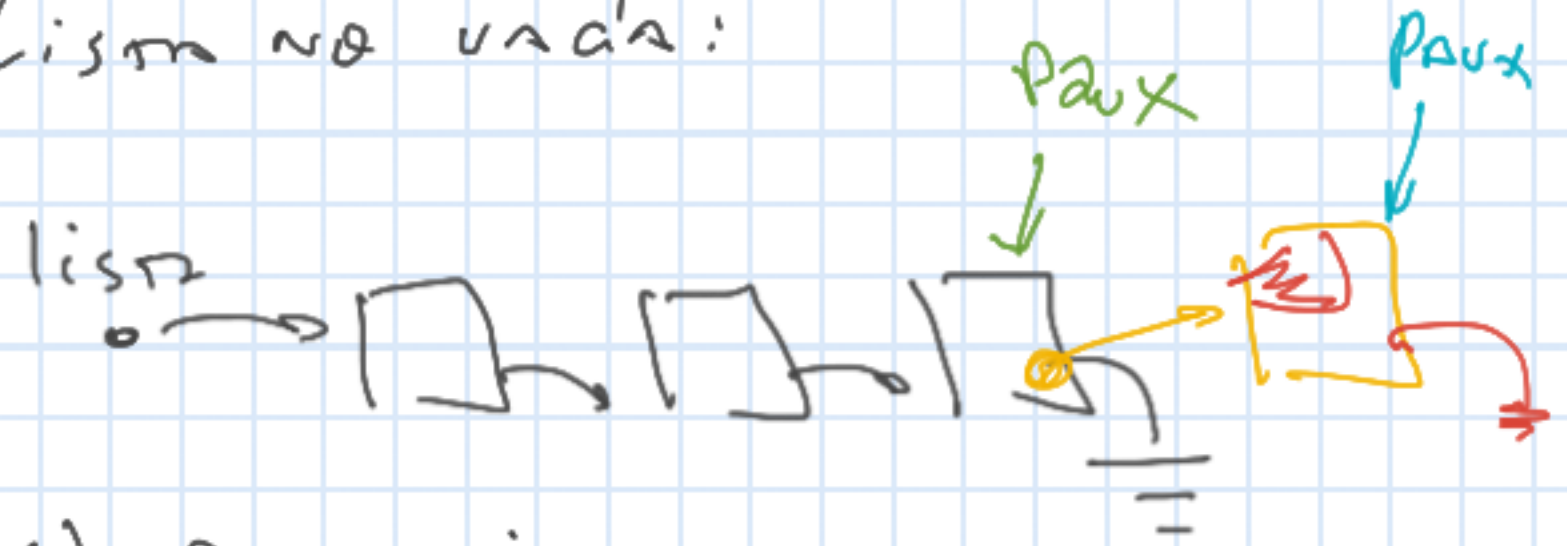
- 1)  $pAux = \text{new NodolistaAlumno()};$
- 2)  $pAux \rightarrow info = \text{Alumno};$
- 3)  $pAux \rightarrow sgte = \text{listasAlumnos};$
- 4)  $\text{listasAlumnos} = pAux;$

## • Insert at front


Case 1) List is empty





Case 2) List is not empty:




A)  $pAux = list$

B)  $\text{while} (pAux \rightarrow \text{next})$    
 $pAux = pAux \rightarrow \text{next};$

C)  $pAux \rightarrow \text{next} = \text{new Node}();$  

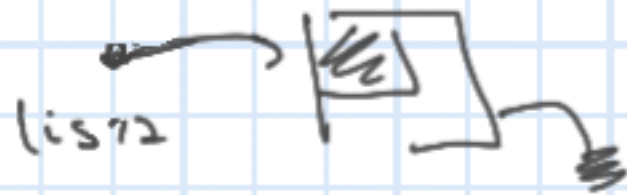
D)  $pAux = pAux \rightarrow \text{next};$  

E)  $pAux \rightarrow \text{next} = \text{NULL};$   
 $pAux \rightarrow \text{info} = \text{data};$  



Quitar Ultimo:

Caso 1) Único nodo caso 2) Lista con más de un nodo



ultimo = lista -> sigte

delete lista;

lista = NULL;



aux = lista;

auxant = NULL;

while (aux -> sigte)

{

auxant = aux;

aux = aux -> sigte;

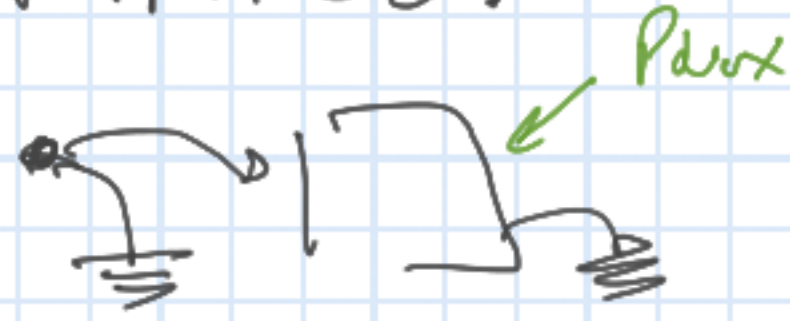
}

ultimo = aux -> sigte

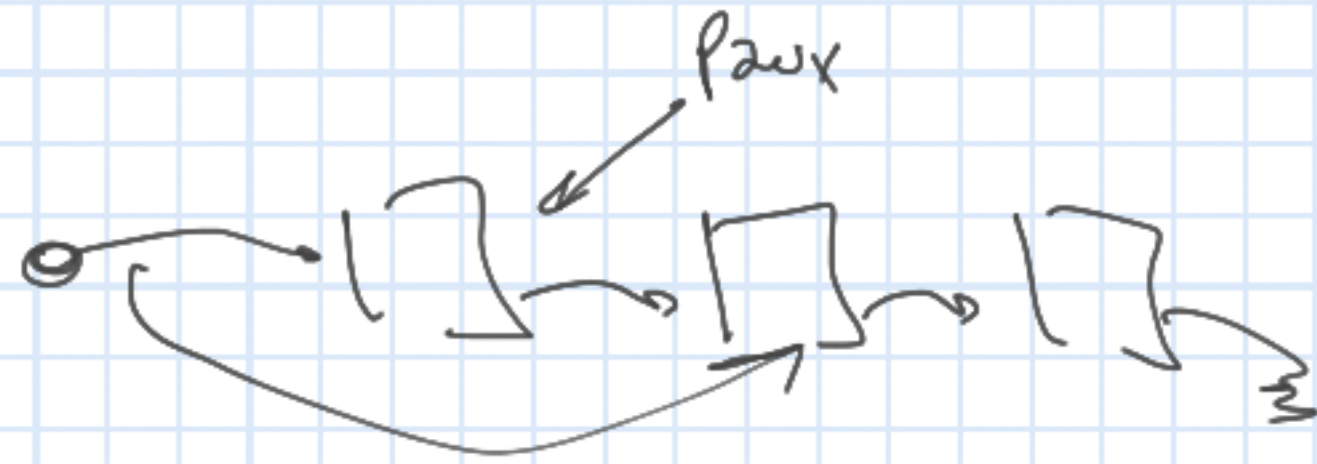
delete aux;

auxant -> sigte = NULL;

Quitar el número:



$2 = \text{lista} \rightarrow \text{info};$   
 $\text{pAux} = \text{lista};$   
 $\text{lista} = \text{lista} \rightarrow \text{sigte}$   
 $\text{delete pAux}$



PARA PODER HACER ESTO  
DEBO VERIFICAR QUE LA LISTA  
NO ESTE VACÍA.

T.D.A.:

→ Pila: Se puede implementar con una lista S.E.,  
solo que debe respetarse un comportamiento  
L.i.F.O.: LAST IN FIRST OUT.  
Se deben implementar dos operaciones:

- Push: Agregar a la Pila
- Pop: Quitar de la Pila

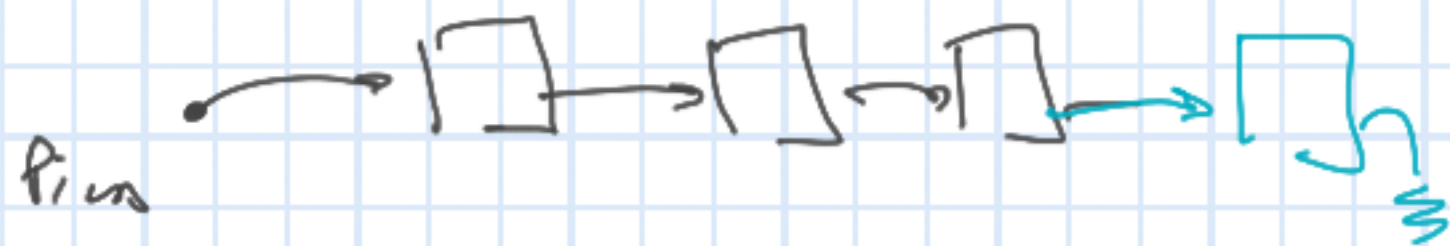
→ Cola: También se puede implementar con una  
lista S.E. pero su comportamiento debe ser

F.i.F.O.: FIRST IN FIRST OUT

Sus operaciones son

- Queue: Poner en la Cola
- Dequeue: Quitar de la Cola

## • Pila



Si cuando Push es **insertarUltimo**  
 $\Rightarrow$  Pop debería ser como  
el QuitarUltimo



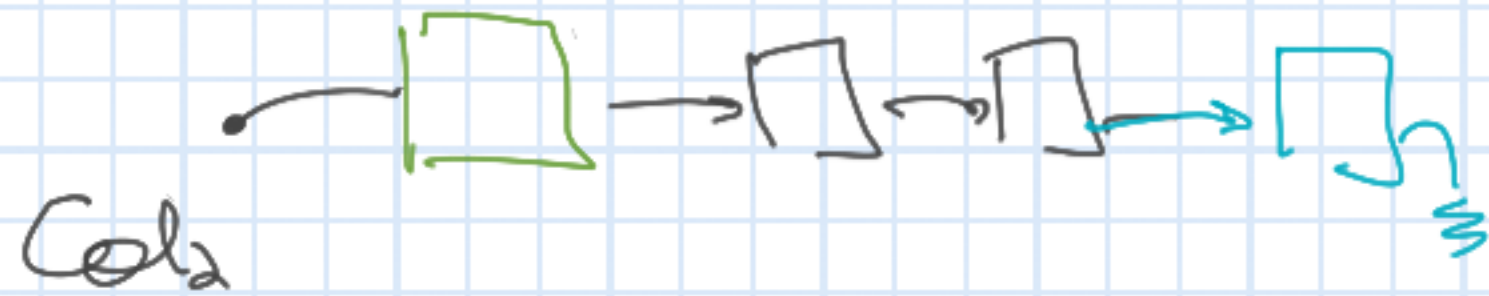
Si cuando hago Push se  
agrega el elemento al principio  
(insertarPrimero)

$\Rightarrow$  Pop debe ser QuitarPrimero

$\Rightarrow$  En las Pilas implementadas con listas S.E.:  
Tanto Push como Pop deben trabajar en el mismo  
extremo



## • Cola



Si cuando Queue es insertarUltimo  
 $\Rightarrow$  dequeue debería ser  
Quitar Primero



Si cuando hago Queue es  
insertar Primero  
 $\Rightarrow$  dequeue debería ser  
Quitar ULTIMO

$\Rightarrow$  En las Cola implementadas con listas S.E.:

Queue y dequeue deben trabajar a extremos  
opuestos.