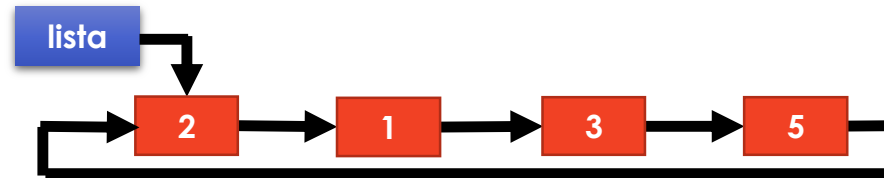


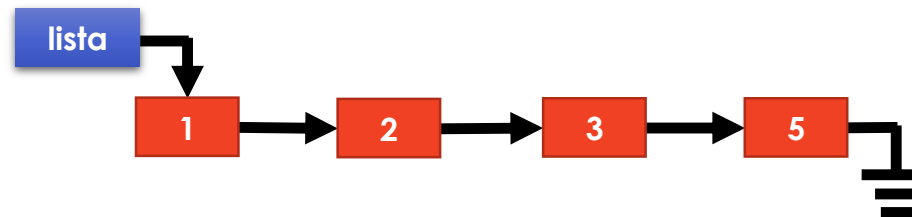
# Casos especiales de lista simple

Hay casos espaciales de lista simple que también son utilizados en la materia:

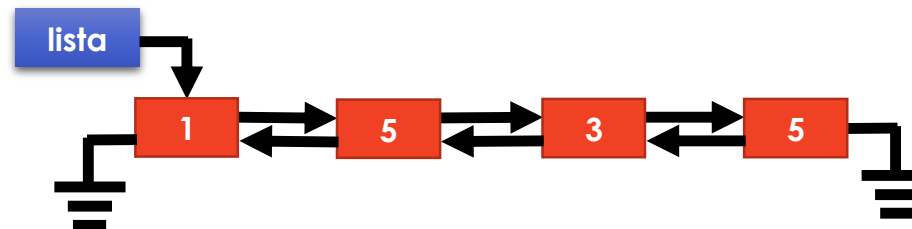
- **Lista simple circular:** el puntero siguiente del último nodo apunta al primero de la lista.



- **Lista simple ordenada:** los elementos mantienen un criterio de orden, que debe mantenerse mientras se realizan las distintas operaciones.



- **Lista doblemente vinculada:** los nodos mantienen un puntero al siguiente y al anterior.



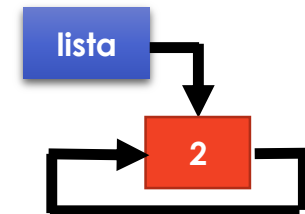
# Lista simple circular

Durante las búsquedas o recorridos no se puede basar en la existencia de NIL. Así, hay que considerar a veces casos que no están o difieren de la lista simple común.

Por ejemplo, para eliminar el primer nodo aparecen estos casos.

**If (lista <> nil) and (lista^.dato = dato) and (lista^.ste = lista) then...**

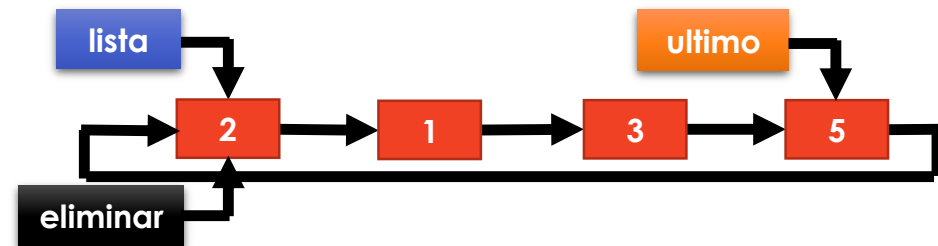
Directamente se elimina **lista** y se le asigna NIL (no requiere un puntero auxiliar).



**else If (lista <> nil) and (lista^.dato = dato) and (lista^.ste <> lista) then...**

Hay que buscar el puntero al **último** nodo.

```
...  
enlace:=lista;  
while (enlace^.ste <> lista) do  
    enlace:=enlace^.ste;  
...
```

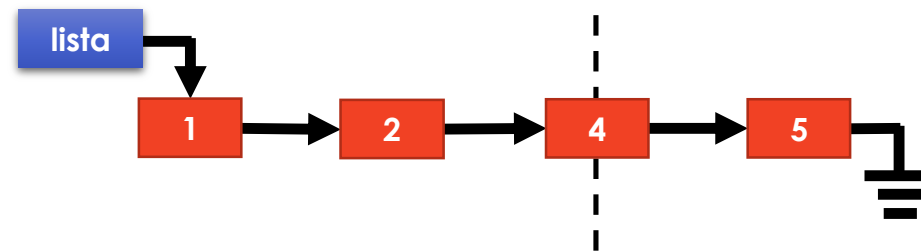


Luego se actualiza el puntero lista, se actualiza el puntero ultimo, y finalmente utilizar un puntero auxiliar para **eliminar** el nodo.

# Lista simple ordenada

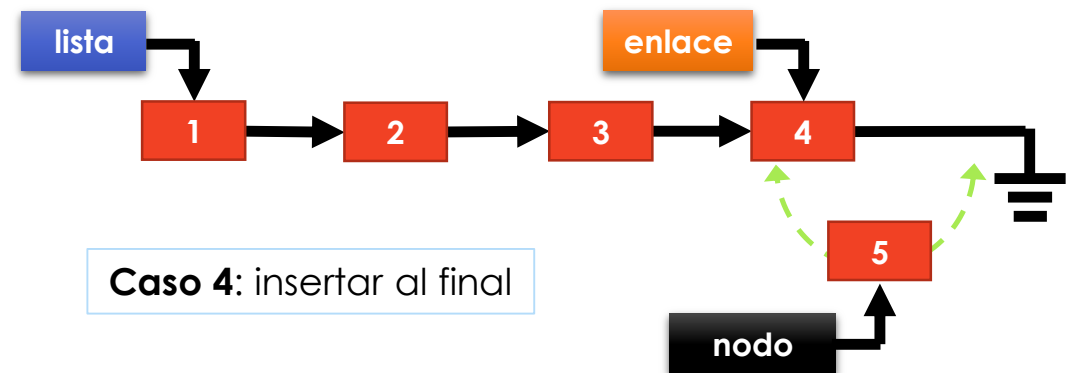
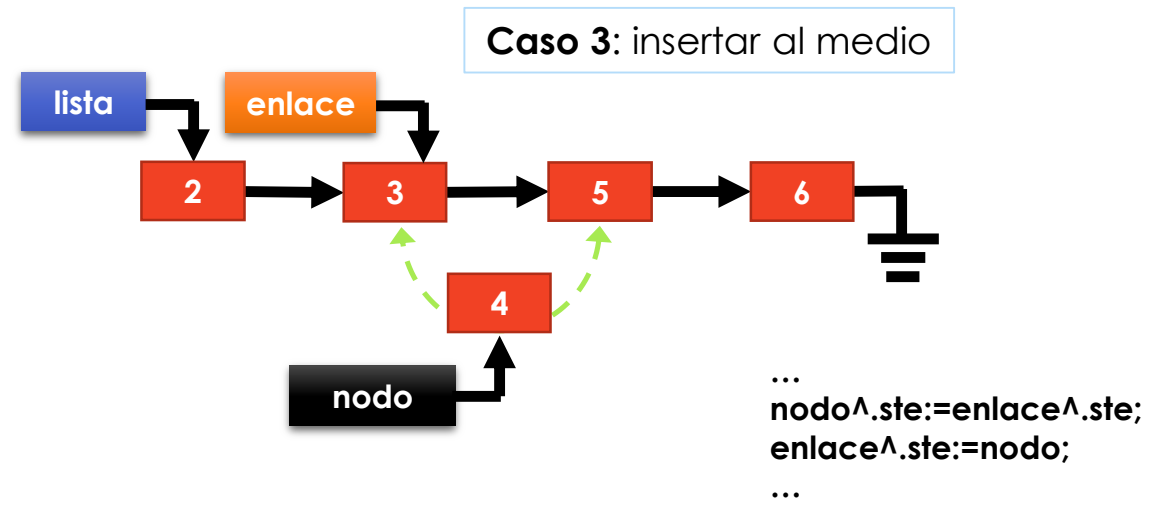
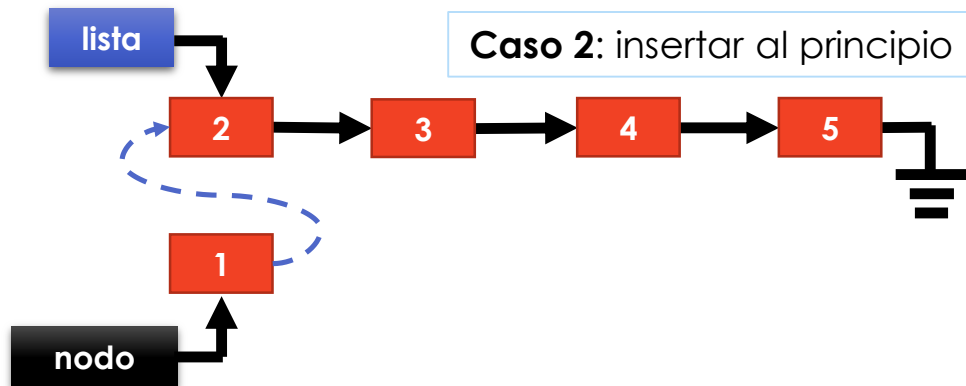
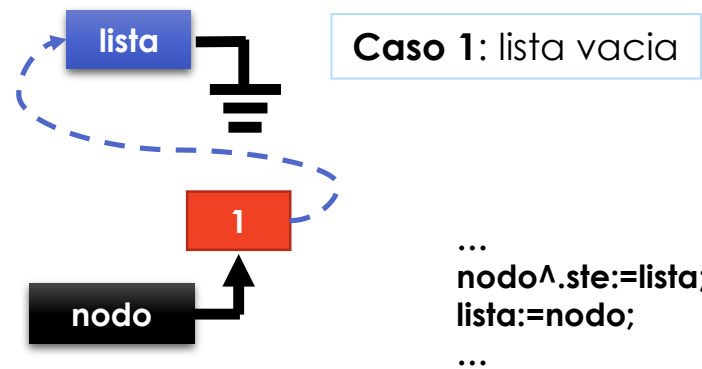
Para este tipo de lista también hay casos especiales que no están en la lista simple común.

No realizar una búsqueda innecesaria si por el criterio de orden el elemento no existe. Ejemplo, buscar o eliminar el nodo con contenido 3.



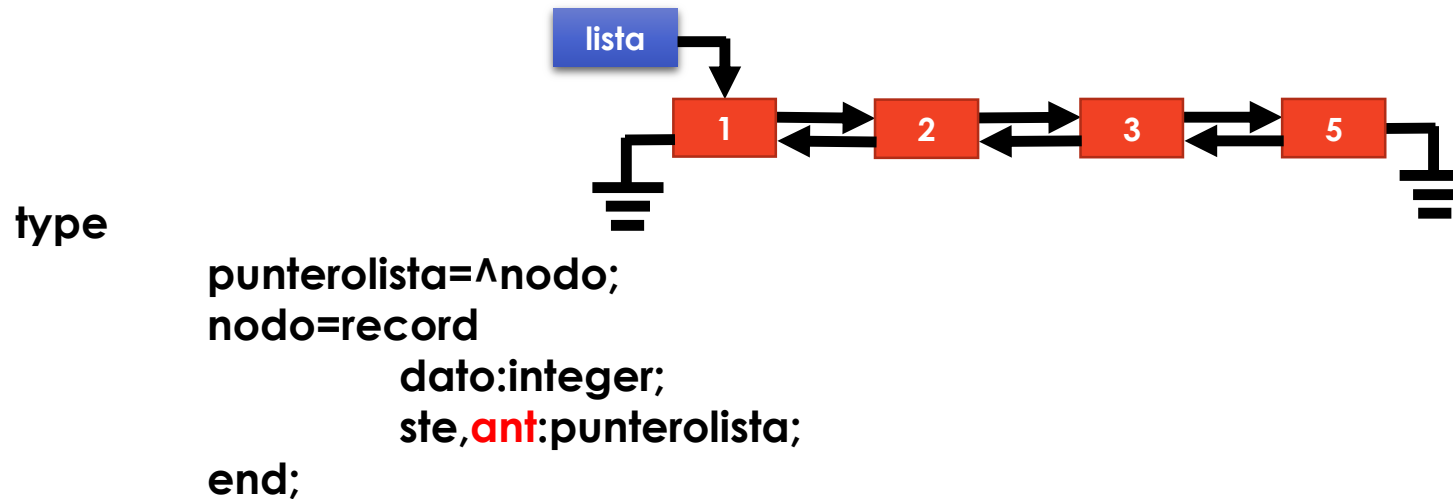
Cuando se agrega un elemento se debe insertar ordenado. Agregarlo al principio o al final y luego reordenar está **MAL**.

# Insertar nodo en lista simple ordenada



# Lista doblemente vinculada

Este tipo de lista mantiene por cada nodo un puntero al siguiente y al anterior nodo.



## Observaciones:

- Las operaciones de recorrido para eliminar o insertar no requieren ubicar el lugar desde el nodo anterior.
- Se requieren de más pasos ante una modificación para mantener la propiedad de doblemente vinculada.
- Según el problema puede indicar la presencia de un puntero al último nodo.

# Tipos de listas descriptas hasta ahora

Para cada tipo de lista mencionada se deberá tener en cuenta los siguientes métodos (procedimientos y funciones).

	Búsqueda, recorrido, etc.	Agregar nodo	Insertar nodo	Eliminar nodo	Eliminar lista	Ordenar lista
<b>Lista simple</b>	X	X		X	X	X
<b>Lista simple ordenada</b>	X		X	X	X	
<b>Lista circular</b>	X	X		X	X	X
<b>Lista circular ordenada</b>	X		X	X	X	
<b>Lista doblemente vinculada</b>	X	X		X	X	X
<b>Listas doblemente vinculada ordenada</b>	X		X	X	X	