

Video15 (1)

Conceptos Principales

- 1. a
- 2. b
- 3. c

Notas

Listas Dobles – Inserción ordenada

Conceptos clave

- El algoritmo de **inserción ordenada** en listas dobles requiere más cuidado que la inserción al final.
 - Cada nodo tiene:
 - **valor**
 - **siguiente**
 - **anterior**
 - La idea: **colocar el nuevo nodo en la posición correcta** manteniendo el orden de los valores.
-

Proceso general

1. Crear el **nuevo nodo** con su valor y punteros iniciales en **NULL**.
 2. Si la lista está vacía → el nuevo nodo es el inicio.
 3. Si no está vacía:
 - Usar un puntero **actual** para recorrer desde el inicio.
 - Buscar la posición correcta: se avanza mientras **actual->siguiente->valor < nuevo->valor**.
 - Una vez encontrada la posición, puede usarse un puntero adicional **posterior** (opcional) para simplificar las conexiones.
-

Ajuste de punteros

Supongamos que se va a insertar entre **actual** y **posterior**:

- **nuevo->siguiente = posterior**
- **nuevo->anterior = actual**
- **posterior->anterior = nuevo**
- **actual->siguiente = nuevo**

El orden importa: usar un puntero **posterior** evita “perder” la referencia de la lista mientras se reasignan conexiones.

Ejemplos del video

- Lista inicial: **10 <-> 20 <-> 30 <-> 40**
 - Insertar **25**: queda **10 <-> 20 <-> 25 <-> 30 <-> 40**
 - Insertar **35**: queda **10 <-> 20 <-> 25 <-> 30 <-> 35 <-> 40**
 - Se demuestra que al final la lista puede recorrerse en ambos sentidos sin problemas.
-

Observaciones

- Si se usan **dos punteros (actual y posterior)**, es más seguro y sencillo manejar los enlaces.
- Se puede hacer solo con **actual**, pero el orden de asignaciones debe cuidarse más para no romper la lista.
- Una vez terminada la función, los punteros auxiliares desaparecen de la pila, pero la lista queda intacta.