

# Video17 (1)

---

## Conceptos Principales

---

1. a
2. b
3. c

## Notas

---

### Listas Dobles, eliminar elemento

---

#### Contexto

- Lista inicial: **10 <-> 20 <-> 25 <-> 30 <-> 35 <-> 40**
  - Queremos **eliminar el nodo con valor 25**.
  - Se usan **punteros auxiliares** (**predecesor** y **sucesor**) para simplificar.
- 

#### Pasos del algoritmo

1. Crear puntero **actual** e ir recorriendo la lista hasta encontrar el nodo a eliminar (**25**).
  2. Una vez encontrado:
    - **predecesor = actual->anterior**
    - **sucesor = actual->siguiente**
  3. Ajustar enlaces:
    - **sucesor->anterior = predecesor**
    - **predecesor->siguiente = sucesor**
  4. Liberar memoria:
    - **free(actual)**
    - Opcional: **actual = NULL** para evitar puntero colgante.
- 

#### Ejemplo visual

Antes de eliminar:

```
20 <-> 25 <-> 30
```

Después de los ajustes:

20 <-> 30

El nodo **25** queda fuera de la lista y luego se libera.

---

## Consideraciones especiales

- El algoritmo mostrado funciona bien cuando **el nodo no está en los extremos**.
  - Si el nodo a eliminar es:
    - **Primero de la lista** → hay que actualizar el puntero **inicio**.
    - **Último de la lista** → hay que actualizar el puntero **fin**.
- 

## Lección importante

- Usar punteros adicionales (**predecesor** y **sucesor**) hace el algoritmo **más seguro y sencillo**.
  - Intentar hacerlo solo con **actual** es posible, pero se vuelve confuso y propenso a errores.
  - Sacrificar **4 bytes de memoria extra** por puntero simplifica mucho la implementación.
- 

### Idea central:

Eliminar un nodo en listas dobles consiste en **reconectar el predecesor y el sucesor** alrededor del nodo a borrar, y luego liberar su memoria.