

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
moisés sierra	1 / 1	Carlos Pichardo	19 / 3 / 2025

Title: Listas Enlazadas en C

Keyword

- nodo.
- Puntero.
- Recorrido.
- memorias dinámicas.

Topic: Listas enlazadas

Notes:

Cada nodo contiene dos partes: un dato y un puntero al siguiente nodo.

Las listas enlazadas permiten crecer dinámicamente sin necesidad de definir un tamaño fijo.

No se puede acceder a los elementos por índice, solo recorriendo la lista desde el inicio.

Questions

- ¿Cuáles son las ventajas de una lista enlazada sobre un arreglo?

Se debe liberar la memoria con `free()` para evitar fugas de memoria.

El último nodo siempre apunta a `NULL`, indicando el final de la lista.

Summary: Las listas enlazadas en C son estructuras dinámicas donde cada nodo apunta al siguiente, permitiendo la inserción y eliminación eficiente en comparación con los arreglos. Sin embargo, requieren más memoria y el acceso a elementos es secuencial, no directo.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
moisés Sierra	1 / 1	Carlos Pichardo	19 / 3 / 2025

Title: Listas doblemente enlazadas en C

<b>Keyword</b> - nodo. - Anterior. - Siguiente. - Punteros. - memoria dinámica.	<b>Topic:</b> Listas doblemente enlazadas
<b>Questions</b> - ¿En qué casos una lista doblemente enlazada es menos eficiente que una simple?	<b>Notes:</b> Cada nodo tiene tres partes: dato, puntero al siguiente y puntero al anterior. Permite recorrer la lista en ambas direcciones, facilitando la búsqueda de datos. Ocupa más memoria que una lista simple, ya que cada nodo almacena dos punteros. Es más eficiente para eliminar nodos, ya que no requiere recorrer toda la lista. Si no se manejan bien los punteros, pueden ocurrir errores de segmentación.

**Summary:** Dice que cada nodo contiene un puntero al siguiente y otro al anterior, permitiendo recorrerlos en ambas direcciones. Esto facilita la eliminación y modificación de nodos, sin recorrer toda la lista, pero ocupa más memoria y requiere un manejo cuidadoso de punteros.