



**UNIVERSIDAD  
DEL VALLE  
DE ORIZABA**

**UnivO®**

**Universidad del Valle de Orizaba**

**Funcionamiento de punteros**

**Miguel Angel Tress Ruiz**

**Docente. Flor Denisse Arenas Cortes**

**Licenciatura en Ingeniería en Sistemas  
computacionales**

**Asignatura. Estructura de Datos**





# Funcionamiento de Punteros (next, prev) en Listas Enlazadas

---

## Introducción

Las listas enlazadas son estructuras de datos fundamentales en la programación que permiten almacenar elementos de forma dinámica. A diferencia de los arreglos, las listas enlazadas no requieren un tamaño fijo, lo cual ofrece mayor flexibilidad para insertar y eliminar elementos. En este tipo de estructura, los elementos (llamados nodos) se conectan entre sí mediante punteros. Estos punteros son referencias que indican la posición del siguiente nodo (next) y, en el caso de listas doblemente enlazadas, también del nodo anterior (prev).

## Funcionamiento de los Punteros

### 1. Puntero next:

En las listas enlazadas simples, cada nodo contiene un dato y un puntero llamado 'next' que apunta al siguiente nodo de la lista. Si el puntero next de un nodo es null, indica que ese nodo es el último de la lista.

### 2. Puntero prev:

En las listas doblemente enlazadas, cada nodo incluye además un puntero llamado 'prev'. Este puntero apunta al nodo anterior en la secuencia, lo que permite recorrer la lista tanto hacia adelante como hacia atrás. El puntero prev del primer nodo es null, ya que no existe un nodo anterior a él.

El uso de estos punteros facilita operaciones como inserciones, eliminaciones y búsquedas de elementos en la lista. Manipulando adecuadamente los punteros next y prev, se pueden reorganizar nodos sin necesidad de mover físicamente los datos.

## Ejemplo Visual

Lista doblemente enlazada con tres nodos:

```
[null <- Nodo1 <-> Nodo2 <-> Nodo3 -> null]  
- Nodo1.prev = null, Nodo1.next = Nodo2  
- Nodo2.prev = Nodo1, Nodo2.next = Nodo3  
- Nodo3.prev = Nodo2, Nodo3.next = null
```



## Conclusión

Los punteros next y prev son componentes esenciales en las listas enlazadas. Permiten establecer conexiones entre los nodos, facilitando una gestión eficiente de los elementos almacenados. Comprender su funcionamiento es clave para implementar estructuras de datos dinámicas que optimicen el uso de memoria y mejoren el rendimiento de las operaciones en programas complejos.