**UNINPAHU**

**LISTAS ENLAZADAS**

**INGENIERIA MULTIMEDIA**

**NATHALIA KATHERIN JUNCO MARTIN**

**PROFESOR:**

**LUIS ALBRTO REYES**

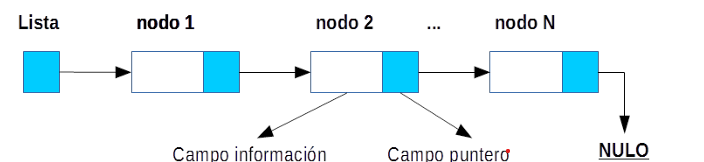
**BOGOTÁ D.C**

**23 DE ABRIL 2025**

**LINEAS ENTRELAZADAS**

Una lista entrelazada es una colección lineal de elementos llamados nodos. El orden entre ellos se establece mediante punteros; direcciones o referencias a otros nodos.

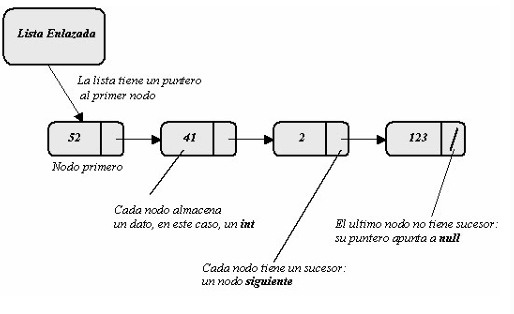
**IMAGEN 1**



**NODO:** Un nodo puede contener datos de cualquier tipo, incluyendo objetos de otras clases, también esta como:

* **CAMPO DE INFORMACIÓN:**  Tipo de datos que se quiere almacenar
* **CAPO LIGA TIPO PUNTERO:** se establece con una liga o con otro nodo de la misma lista.
* Podemos encontrar una diferencia ente los arreglos que en si las líneas enlazadas no almacenan elementos en una posición continua si no que el nodo a punta al siguiente como un sucesor formando una cadena.

**IMAGEN 2**

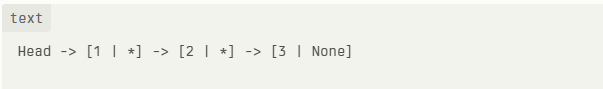


**TIPOS DE LINEAS ENLAZADAS:**

* **LISTA ENLAZADA SIMPLE:** Cada nodo indica el apuntador siguiente y cuando se llega a el ultimo nodo se apunta a NULL o NONE definiendo el final.
* Un ejemplo lo podemos ver en java, pero también se suele usar mucho en Python y otros lenguajes de programación se pueden adaptar perfectamente.

**IMAGEN 3 (LISTA ENLAZADA SIMPLE)**

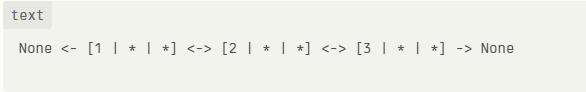
* **Cada** nodo tiene un dato y un enlace al siguiente nodo
* **El** ultimo nodo a punta a none

****

* **LISTA DOBLE ENLAZADA**: Esta permite que cada nodo tenga dos referencias una al siguiente nodo y la otra al anterior nodopermitiendo recorrer la lista en ambos sentidos.
* Se encuentra un ejemplo respecto al tipo usando lenguaje javo

**IMAGEN 4 (LISTA DE DOBLE ENLAZADA)**

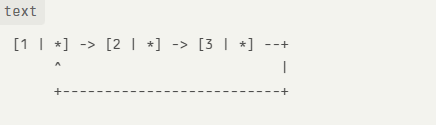
* Cada nodo tiene un dato, un enlace al siguiente nodo y otro al nodo anterior.
* Permite recorrer la lista en ambos sentidos.



* **LISTA CIRCULAR**: El ultimo nodo apunta al primero formando un ciclo puede ser simple o doblemente enlazado.

**IMAGEN 5 (LISTA CIRCULAR)**

* El último nodo apunta de nuevo al primer nodo, formando un ciclo.
* Se puede recorrer desde cualquier nodo y volver al inicio.



**VENTAJAS:**

* Una lista enlazada es apropiada cuando el número de elementos de datos que se van a representar en un momento dado es impredecible.
* Las listas enlazadas son dinámicas, por lo que la longitud de una lista puede incrementarse o reducirse, según sea necesario.
* Las listas enlazadas se llenan sólo cuando el sistema no tiene suficiente memoria para satisfacer las peticiones de asignación dinámica de almacenamiento.

**OPERACIONES BASICAS:**

* **USO TÍPICO:**
* Manejo de datos con tamaño variable e impredecible
* Implementación de pilas, colas, tablas hash, y otra estructura

**EJERCICIO DE APLICACIÓN**

**<?php**

**class Nodo{**

**public $dato;**

**public $siguiente;**

**public function \_\_construct($dato) {**

**$this->dato = $dato;**

**$this->siguiente = null;**

**}**

**}**

**class listasentrelazada{**

**public $comienzo;**

**public function \_\_construct(){**

**$this->comienzo = null;**

**}**

**public function insertar($dato){**

**$nuevoNodo = new Nodo($dato);**

**$nuevoNodo->siguiente = $this->comienzo;**

**$this->comienzo = $nuevoNodo;**

**}**

**public function imprimirHTML(){**

**$actual = $this->comienzo;**

**echo "<ul>";**

**while($actual != null){**

**echo "<li>" . $actual->dato . "</li>";**

**$actual = $actual->siguiente;**

**}**

**echo "</ul>";**

**}**

**}**

**// Corrección aquí: usar el nombre correcto de la clase**

**$lista = new listasentrelazadas();**

**$lista->insertar("fresas");**

**$lista->insertar("crema");**

**$lista->insertar("chispas de chocolate");**

**$lista->imprimirHTML();**

**?>**

**CONCLUSIÓN**

Las listas enlazadas ofrecen un equilibrio entre eficiencia y flexibilidad, siendo especialmente útiles cuando no se conoce de antemano la cantidad exacta de datos a manejar o cuando se requiere una estructura que se adapte fácilmente a cambios frecuentes en su tamaño.

También se genera un gran método a la de equilibrar una buena cadena sobre datos equivalida menté importantes con el fin de llevar un conteo o una estabilidad de datos dando un final a lo esperado en los nodos y sus sucesores.

**REFERENCIAS**

(perplexity, págs. 3,5,6) INVESTIGACIÓN

(estructura de datos, pág. 2) IMAGEN Y INVESTIGACIÓN

(estructura de datos, págs. 2,4,5) INVESTIGACIÓN

(PERPLEXITY, pág. 3) IMAGENES REPRESENTATIVA DE LOS TIPOS

(PERPLEXITY, pág. 4) IMAGEN REPRESENTATIVA DE LOS TIPOS