Listas Enlazadas

Jessica Natalia Cortes Bolívar

Introducción a la Programación

Bogotá D.C, 21 de abril 2025

Introducción

En este informe presentaremos una investigación del concepto listas enlazadas en programación, dando a entender su significado, algunos ejemplos, para que sirve y algunos tipos de listas enlazadas, que son los nodos, para que sirven, algunos ejemplos de nodos en diferentes conceptos, de penúltimo daré a conocer mi conclusión final y para el final la infografía que fueron los sitios web de donde saque la información y las imágenes

Desarrollo

¿Qué es Lista Enlazada?

Una lista enlazada es una estructura de datos que desempeña un papel crucial en la organización y gestión de datos. Contiene una serie de nodos que se almacenan en ubicaciones aleatorias de la memoria, lo que permite una gestión eficaz de la memoria. Cada nodo de una lista enlazada contiene dos componentes principales: la parte de datos y una referencia al siguiente nodo de la secuencia. Nos permite almacenar datos de una forma organizada, al igual que los vectores pero, a diferencia de estos, esta estructura es dinámica, por lo que no tenemos que saber "a priori" los elementos que puede contener. En una lista enlazada, cada elemento apunta al siguiente excepto el último que no tiene sucesor y el valor del enlace es null. Por ello los elementos son registros que contienen el dato a almacenar y un enlace al siguiente elemento. Los elementos de una lista, suelen recibir también el nombre de nodos de la lista.

¿Para que sirve?

Las listas enlazadas sirven para almacenar datos de forma dinámica, donde la longitud de la lista puede cambiar en tiempo de ejecución. Permiten insertar y eliminar elementos fácilmente en cualquier posición, lo que las hace eficientes para operaciones que involucran mucho movimiento de datos. Adicionalmente las listas enlazadas se crearon para superar varios inconvenientes asociados al almacenamiento de datos en listas y matrices normales, como se indica a continuación:

Ventajas de las listas enlazadas:

* **Tamaño dinámico:**

La lista puede crecer o disminuir según sea necesario, lo que las hace útiles cuando no se conoce el tamaño de los datos de antemano.

* **Inserción y eliminación eficientes:**

Insertar o eliminar un elemento en cualquier punto de la lista no requiere mover otros elementos, como en los arreglos.

* **No hay necesidad de memoria contigua:**

Los elementos de la lista no tienen que estar almacenados en memoria consecutiva, lo que permite una mayor flexibilidad en el uso de la memoria.

* **Son útiles para implementar otras estructuras de datos:**

Las listas enlazadas se utilizan para crear pilas, colas y tablas hash.

Desventajas de las listas enlazadas:

* **Acceso aleatorio no eficiente:**

Para acceder a un elemento específico en la lista, se necesita recorrerla desde el principio, lo que puede ser lento si el elemento está al final de la lista.

* **Requieren más memoria que los arreglos:**

Cada nodo de la lista enlazada necesita espacio para almacenar el dato y el puntero al siguiente nodo.

Aplicaciones de las listas enlazadas:

* **Implementación de pilas y colas:**

[Según Skilled.dev](https://skilled.dev.en2es.search.translate.goog/course/linked-lists), las listas enlazadas son una forma común de implementar estas estructuras de datos.

* **Gestión de memoria:**

[DataCamp](https://www.datacamp.com/es/tutorial/python-linked-lists) muestra cómo las listas enlazadas se pueden usar para gestionar la asignación y liberación de memoria en tiempo de ejecución.

* **Tablas hash:**

Pueden utilizarse para manejar colisiones en tablas hash.

* **Representación de gráficos:**

Las listas enlazadas pueden usarse para representar nodos y aristas en gráficos.

**¿Cuándo debes utilizar listas enlazadas?**

Aunque las listas enlazadas ofrecen ciertas ventajas sobre las listas y matrices normales, como el tamaño dinámico y la eficiencia en memoria, también tienen sus limitaciones. Como hay que almacenar punteros para cada elemento para referenciar el siguiente nodo, el uso de memoria por elemento es mayor cuando se utilizan listas enlazadas. Además, esta estructura de datos no permite el acceso directo a los datos. Acceder a un elemento requiere un recorrido secuencial desde el principio de la lista, lo que da lugar a una complejidad de tiempo de búsqueda O(n).

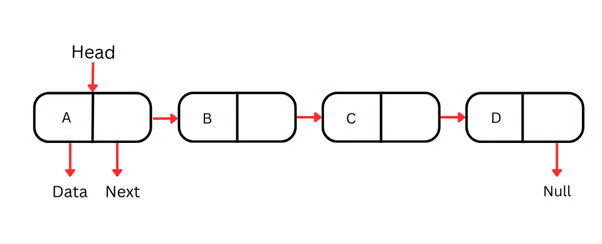
La elección entre utilizar una lista enlazada o una matriz depende de las necesidades específicas de la aplicación. Las listas enlazadas son más útiles cuando

* Necesitas insertar y eliminar con frecuencia muchos elementos
* El tamaño de los datos es impredecible o puede cambiar con frecuencia
* El acceso directo a los elementos no es un requisito
* El conjunto de datos contiene elementos o estructuras grandes

Tipos de listas enlazadas

Hay tres tipos de listas enlazadas, cada una de las cuales ofrece ventajas únicas para distintos escenarios. Estos tipos son:

**Listas simples**



Fuente:

<https://www.datacamp.com/es/tutorial/python-linked-lists>

Una lista enlazada simple es el tipo más sencillo de lista enlazada, en la que cada nodo contiene algunos datos y una referencia al siguiente nodo de la secuencia. Sólo pueden recorrerse en una única dirección: de la cabeza (el primer nodo) a la cola (el último nodo).

Cada nodo de una lista con un solo enlace suele constar de dos partes:

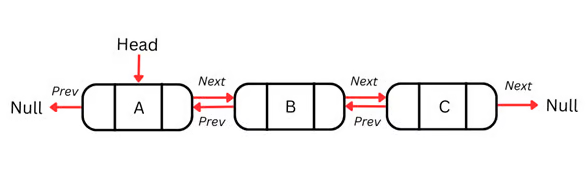
* **Data**: La información real almacenada en el nodo.
* **Siguiente Puntero**: Una referencia al siguiente nodo. El siguiente puntero del último nodo suele ser nulo.

Como estas estructuras de datos sólo pueden recorrerse en una única dirección, para acceder a un elemento concreto por valor o índice hay que empezar por la cabecera y recorrer secuencialmente los nodos hasta encontrar el nodo deseado. Esta operación tiene una complejidad temporal de O(n), por lo que es menos eficaz para listas grandes.

Insertar y eliminar un nodo al principio de una lista de un solo enlace es muy eficaz, con una complejidad temporal de O(1). Sin embargo, la inserción y la eliminación en el medio o al final requieren recorrer la lista hasta ese punto, lo que conlleva una complejidad temporal O(n).

### El diseño de las listas simples las convierte en una estructura de datos útil cuando se realizan operaciones que tienen lugar al principio de la lista.

### **Listas doblemente enlazadas**

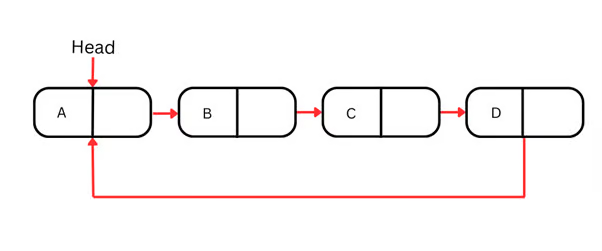


Fuente:

<https://www.datacamp.com/es/tutorial/python-linked-lists>

Una desventaja de las listas con un solo enlace es que sólo podemos recorrerlas en una dirección y no podemos iterar hacia el nodo anterior si es necesario. Esta restricción limita nuestra capacidad para realizar operaciones que requieran una navegación bidireccional.

Las listas doblemente enlazadas resuelven este problema incorporando un puntero adicional dentro de cada nodo, lo que garantiza que la lista pueda recorrerse en ambas direcciones. Cada nodo de una lista doblemente enlazada contiene tres elementos: los datos, un puntero al nodo siguiente y un puntero al nodo anterior.

**Listas enlazadas circulares**

Fuente:

<https://www.datacamp.com/es/tutorial/python-linked-lists>

Las listas enlazadas circulares son una forma especializada de lista enlazada en la que el último nodo apunta de nuevo al primer nodo, creando una estructura circular. Esto significa que, a diferencia de las listas simple y doblemente enlazadas que hemos visto hasta ahora, la lista enlazada circular no termina, sino que hace un bucle.

La naturaleza cíclica de las listas enlazadas circulares las hace ideales para situaciones en las que hay que hacer un bucle continuo, como los juegos de mesa que hacen un bucle desde el último jugador hasta el primero, o en algoritmos informáticos como la programación por turnos.

¿Qué son los nodos y para qué sirven?

En informática y redes, un nodo es un punto de conexión o un dispositivo que forma parte de una red. Estos nodos pueden ser computadoras, servidores, enrutadores, conmutadores, o incluso dispositivos IoT. Su función principal es permitir la comunicación entre diferentes elementos de la red, ya sea enviando, recibiendo, o reenviando información.

¿Cómo funcionan los NODOS?

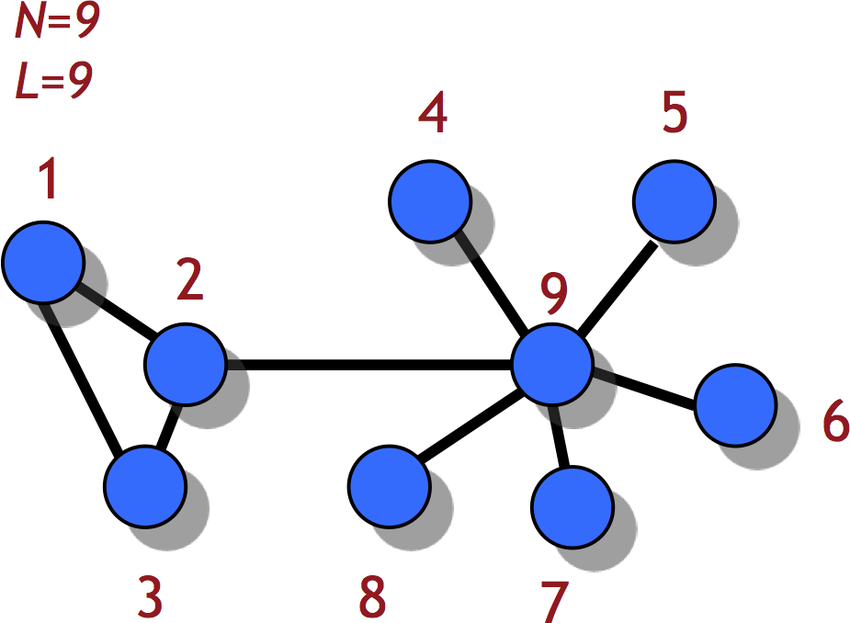
Cuando se trata de una conexión a Internet desde tu navegador, en la que se establece un enlace a un servidor web, este se convierte en un NODO que ofrece un servicio, en este caso una página web, el cual se conecta con nosotros gracias a una dirección IP. Igualmente son NODOS aquellos elementos que te permiten realizar la conexión como un router o un switch, por nombrar algunos, incluso tu PC o móvil con el que accedes a la página web, es un NODO dentro de esta conexión.

Los NODOS no son solo equipos físicos, por lo tanto, su funcionamiento no está condicionado al hardware, sino que también, puede ser una aplicación que permita al equipo establecer la conexión. Es decir, los NODOS serán todas aquellas herramientas tanto físicas como virtuales que lleven a los elementos a comunicarse entre sí, considerando al primero como un emisor de la información, el cual cuenta con la capacidad de procesar, y remitirla a los demás NODOS para continuar la trasmisión de información.

Y el segundo actuará como NODO receptor, captando toda la información que el NODO emisor envió, procesando, y distribuyendo a su vez la información a otros puntos, un proceso que se hará consecutivamente, creando infinitas conexiones, o aquellas que puedan ser necesarias.

De esta manera los NODOS cumplirán con la función de orientar el tráfico de red, de enviar y recibir la información, en un sinfín de redes a nivel mundial.

En Alai Secure, [operador M2M/IOT en seguridad TELCO](https://alaisecure.co/m2m-iot/), los sistemas han sido ideados para que estén 100% operativos ante cualquier tipo de incidencias o contingencias en alguna de sus sedes, por lo que contamos con [doble conexión NODO](https://alaisecure.co/red-inteligente/conexion-de-doble-nodo/). Es decir, toda la información referente a enrutamientos, desvíos, locuciones, esperas y números de abonado se equilibran de manera natural entre ambos NODOS con lo que, ante cualquier contingencia, podemos cambiar de NODO inmediatamente sin sufrir ni alteraciones, ni cortes.



Fuente:

<https://www.researchgate.net/figure/Figura-31-Ejemplo-de-una-red-con-9-nodos-y-9-conexiones-o-enlaces-De-esta-forma-se_fig4_299978643>

Ejemplos de nodos:

* Dispositivos personales: Computadoras (portátiles, de escritorio), teléfonos móviles, tabletas.
* Puntos de acceso a internet: Routers, módems, puntos de acceso WiFi.
* Servidores: Servidores en centros de datos, servidores web, servidores de base de datos.
* Dispositivos IoT: Sensores, dispositivos inteligentes, cámaras de seguridad.
* Conmutadores y enrutadores: Dispositivos que gestionan el tráfico de red y conectan diferentes nodos.
* Impresoras: Dispositivos que se conectan a la red y pueden ser utilizados por múltiples nodos.
* En física, un nodo: Puede ser un punto de máxima o mínima amplitud en una onda estacionaria.

Ejemplos en diferentes contextos:

* Redes informáticas:

Cada dispositivo que se conecta a la red (ordenador, servidor, teléfono móvil) es un nodo, según la explicación de Euroinnova.

* Circuitos eléctricos:

Un nodo es un punto de conexión común entre dos o más elementos de un circuito.

* Geografía:

Un nodo puede ser un punto de interés o cruce en un sistema de transporte o en una red.

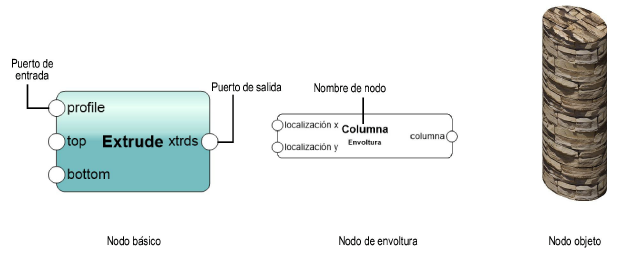
* Órbita celeste:

Los nodos ascendente y descendente son puntos donde una órbita cruza el plano del Ecuador celestial.

En resumen, un nodo es un elemento central en redes y sistemas, que puede ser un dispositivo físico o un punto de conexión en un circuito o sistema.

## Tipos de nodos

Nodos son los bloques de construcción básicos para la herramienta **Marionette**; cada nodo contiene secuencias de comandos que contribuye hacia la secuencia de comandos general o red. Existen tres tipos de nodos: nodo básicos, nodos de envoltura, y nodos objeto. Nodos básicos y de envoltura son combinados en la creación de redes.



Fuente:

<https://app-help.vectorworks.net/2017/spa/VW2017_Guide/Marionette/Tipos_de_nodos.htm#:~:text=Existen%20tres%20tipos%20de%20nodos,en%20la%20creaci%C3%B3n%20de%20redes>.

**Nodos básicos**

Nodos básicos realizan un comando o ejecutan una operación, o proporcionan valores relevantes de parámetros.

**Nodos de envoltura**

Nodos de envoltura consolidan una red de nodos en un único nodo para la organización y compartición. Entradas pueden ser accedidos directamente desde la paleta información de objeto; puertos de entrada y salida son usados al colocar el nodo de envoltura dentro de redes más grandes.

**Nodos objeto**

Nodos objeto son nodos de envoltura que toman la forma del resultado de su secuencia de comandos ejecutada, lo cuál significa que cambios en la secuencia de comandos pueden ser vistos inmediatamente y anotadas sin tener que ejecutar la secuencia de comandos. Mientras que un nodo de envoltura es una forma consolidada de una red que puede ser ejecutada para producir un objeto, un nodo objeto es una representación inmediata de lo que la secuencia de comandos produce.

Por ejemplo, un nodo de envoltura que contiene una secuencia de comandos para crear una esfera puede ser convertido en un nodo objeto; cada vez que un cambio es realizado a la secuencia de comandos subyacente, la esfera se ajustará de manera concordante en lugar de que la secuencia de comandos sea ejecutada múltiples veces.

**Crear y editar nodos**

Por defecto, la herramienta **Marionette**proporciona una amplia variedad de nodos que están listos para ser usados. Es posible crear nodos personalizados editando un nodo pre-existente y cambiando la secuencia de comandos; sin embargo, esto requiere una familiaridad con Python y VectorScript y no es recomendable para usuarios que no están familiarizados con las secuencias de comandos.

Conclusión

Las listas enlazadas se erigen como una estructura de datos fundamental en la informática, ofreciendo una alternativa flexible y dinámica a los arreglos tradicionales. A diferencia de estos últimos, donde los elementos se almacenan contiguamente en la memoria, las listas enlazadas emplean nodos dispersos, cada uno conteniendo un dato y un puntero (o enlace) al siguiente nodo de la secuencia.