UNIVERSIDAD PONTIFICIA CATÓLICA DEL ECUADOR SEDE MANABÍ

Estructura de Datos II

TAD lineales: Listas doblemente enlazadas

Andrea Valentina Campaña Intriago

Hernández Ménendez Axel Jhostin

Yakov David Seni Barcia

Mg. Josselyn Tatiana Gómez

Martes, 15 de octubre de 2024

**Contenido**

[Introducción 3](#_Toc179924874)

[Tipos de datos abstractos 3](#_Toc179924875)

[Concepto general de lista enlazada 4](#_Toc179924876)

[Listas doblemente enlazadas 4](#_Toc179924877)

[Implementación de las listas doblemente enlazadas 5](#_Toc179924878)

[Pros 6](#_Toc179924879)

[Contras 6](#_Toc179924880)

[Fundamentación matemática 7](#_Toc179924881)

[Agregar nodo al inicio 7](#_Toc179924882)

[Agregar nodo al final 7](#_Toc179924883)

[Mostrar lista impresa desde adelante 7](#_Toc179924884)

[Mostrar lista impresa desde atrás 7](#_Toc179924885)

[Buscar por índice 8](#_Toc179924886)

[Eliminar por índice 8](#_Toc179924887)

[Modificar por índice 8](#_Toc179924888)

[Implementación y ejemplo en Java 9](#_Toc179924889)

[Implementación y ejemplo en Python 12](#_Toc179924890)

[Bibliografía 17](#_Toc179924891)

# Introducción

Uno de los propósitos fundamentales en las ciencias de la computación es el manejo, representación y recuperación de la información, lo que implica diseñar estructuras que permitan almacenar y procesar una gran cantidad de datos de manera eficiente. Debido a esto, el empleo de estructuras de datos que optimicen el uso de la memoria son relevantes, tanto como su búsqueda, modificación y que minimice los tiempos de ejecución.

Es importante tener en cuenta que las estructuras de datos adecuadas poseen algoritmos que soporten operaciones rápidas, como la inserción, eliminación, búsqueda y ordenamiento.

# Tipos de datos abstractos

Un tipo de dato abstracto (TDA) es una forma de organizar datos que se enfoca en qué se puede hacer con ellos, sin preocuparse por cómo están implementados internamente. En otras palabras, los detalles de su funcionamiento interno se abstraen, y lo que importa son las operaciones que la estructura de datos soporta. Esto permite que el usuario se concentre en utilizarlo sin necesidad de conocer los detalles técnicos por medio de la serie de operaciones definidas sobre el conjunto de datos a manejar, en cambio, al pensar en la forma que la computadora se almacenará la información que queremos, se emplea el término estructura de datos.

# Concepto general de lista enlazada

Una lista enlazada es una estructura de datos dinámica que organiza sus elementos de mismo tipo en una secuencia, donde cada elemento, conocido como nodo, está conectado al siguiente mediante un "enlace". Según Zohonero & Joyanes (2008), esta conexión permite que los nodos se dispongan uno detrás de otro, formando una cadena que facilita la inserción o eliminación de elementos en cualquier parte de la lista, sin necesidad de reorganizar toda la estructura.

Una de las características más destacadas de las listas enlazadas es que el número de nodos puede variar rápidamente a lo largo del tiempo, aumentando debido a las inserciones o disminuyendo por la eliminación de nodos. Esta flexibilidad en la gestión de memoria hace que las listas enlazadas sean muy útiles.

## Listas doblemente enlazadas

Una lista doblemente enlazada es una estructura lineal la cual, en sus nodos, posee de tres campos: dos enlaces, referencias o punteros (frecuentemente llamados anterior y siguiente o antecesor y sucesor) y otro para el dato. Las listas doblemente enlazadas no necesitan de un nodo especial para acceder a ellas, logrando recorrer en ambos sentidos a partir de cualquier nodo debido que siempre es posible alcanzar cualquiera de la lista hasta llegar a uno de los extremos, siendo un extremo la cabeza (inicial) o cola (final). (Chalita Pérez, 2020).

Diagrama

Descripción generada automáticamente

Figura 1. Representación gráfica de una lista enlazada, Zohonero & Joyanes, 2008.

Hay que tomar en cuenta las siguientes consideraciones:

* El primer nodo (frente) de una lista es el nodo apuntado por cabeza.
* Todos los nodos tendrán una referencia hacia adelante o atrás.
* Los nodos inicial o frente y final o cola tienen una referencia a *null*.
* Una lista vacía (sin nodos), se representa con el puntero cabeza con *null*.

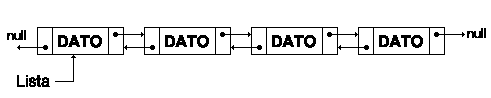


Figura 2. Lista doblemente enlazada, Salvador Poso, 2001.

## Implementación de las listas doblemente enlazadas

Para la creación de una lista doblemente enlazada se requiere declarar la clase nodo con dos punteros o referencias, para enlazar el nodo siguiente, anterior o nulo si es el caso, sumado a la parte correspondiente al campo dato.

Texto

Descripción generada automáticamente

Figura 3. Creación de nodo. Creación propia.

### Pros

* **Acceso bidireccional:** A diferencia de una lista enlazada, en la lista doblemente ligada se puede navegar y ser analizada en ambas direcciones. La referencia al siguiente nodo ayuda a que se pueda mover hacia adelante y la referencia al nodo anterior ayuda a mover en dirección contraria.
* **Remover o insertar nodos de forma eficiente:** Las operaciones básicas como agregar elemento y eliminar son fáciles de implementar y el poder navegar en dos direcciones facilita este hecho.

### Contras

* **Alta ocupación en memoria:** Cada nodo requiere de dos punteros (anterior y successor) utilizando más memoria en comparación que otras estructuras debido al almacenamiento utilizado.
* **Mayor complejidad:** Debido al manejo de dos punteros por nodo, su implementación posee de una dificultad mayor durante sus algoritmos, aumentando su probabilidad de error.
* **Fragmentación de memoria:** Con respecto al manejo dinámico de asignar un nodo en la memoria, cada que se agrega o elimina uno de forma frecuente y en diferentes partes, conlleva a una distribución no uniforme de la memoria, llegando a fragmentarse sin poder usarlo para nuevas asignaciones.

# Fundamentación matemática

### Agregar nodo al inicio

1. Crea un nuevo nodo con el dato proporcionado.
2. Si la lista está vacía, es decir, igual a *null*, el nuevo se convierte en la cabeza de la lista
3. Si ya hay nodos en la lista, el nuevo nodo se conecta al nodo actual de la cabeza y se actualiza la cabeza para que apunte al nuevo nodo

### Agregar nodo al final

1. Crea un nuevo nodo.
2. Si la lista está vacía, el nuevo nodo se convierte en la cabeza.
3. Si no está vacía, recorre la lista hasta llegar al último nodo y luego lo conecta al nuevo nodo, actualizando las referencias.

### Mostrar lista impresa desde adelante

1. Comienza desde la cabeza y recorre cada nodo mientras imprime su dato.
2. Termina cuando llega a un nodo que no tiene un nodo siguiente, es decir, que direcciona a *null.*

### Mostrar lista impresa desde atrás

1. Primero, recorre la lista hasta llegar al último nodo
2. Luego, desde el último nodo, recorre hacia atrás utilizando las referencias “anterior” para imprimir los datos.

### Buscar por índice

1. Comienza desde la cabeza y cuenta los nodos
2. Cuando el contador coincide con el índice proporcionado, devuelve el nodo correspondiente
3. Si no se encuentra el nodo, retorna *null*.

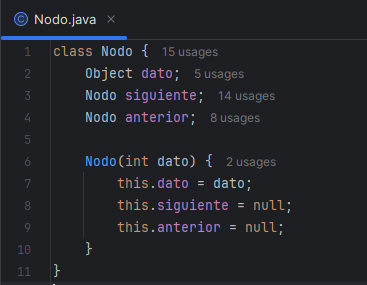
### Eliminar por índice

1. Busca el nodo utilizando el método de buscar por índice.
2. Si el nodo es la cabeza, actualiza la cabeza para que apunte al siguiente nodo.
3. Si no es la cabeza, ajusta las referencias del nodo anterior y del siguiente para eliminar el nodo sin romper la lista.

### Modificar por índice

1. Busca el nodo usando buscar por índice.
2. Si encuentra el nodo, actualiza su dato con el nuevo valor proporcionado.

# Implementación y ejemplo en Java

**Clase de los nodos**

**Clase de listas dobles**

**Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza mediaMétodo agregar nodo al inicio**

**Texto

Descripción generada automáticamenteMétodo agregar nodo al final**

**Texto

Descripción generada automáticamenteMétodo imprimir lista en orden y orden invertido**

**Texto

Descripción generada automáticamenteMétodo buscar en la lista por índice**

**Texto

Descripción generada automáticamenteMétodo eliminar por índice**

**Texto

Descripción generada automáticamenteMétodo modificar por índice**

**Demostración del código**

Texto

Descripción generada automáticamente

# Implementación y ejemplo en Python

Texto

Descripción generada automáticamenteClase de los nodos

Texto

Descripción generada automáticamenteMétodo de agregar nodo al final

Método de mostrar lista

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteMétodo mostrar lista nvertida

Texto

Descripción generada automáticamenteMeéodo agregar nodo al inicio

Texto

Descripción generada automáticamenteMétodo eliminar nodo

Método buscar nodo

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteMétodo contar nodos actuales

Texto

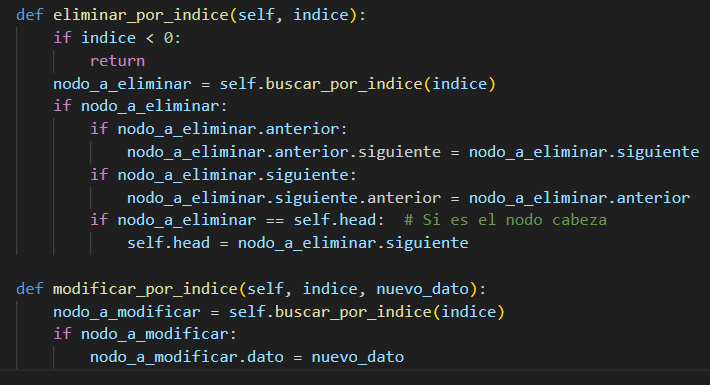
Descripción generada automáticamenteMétodo insertar después

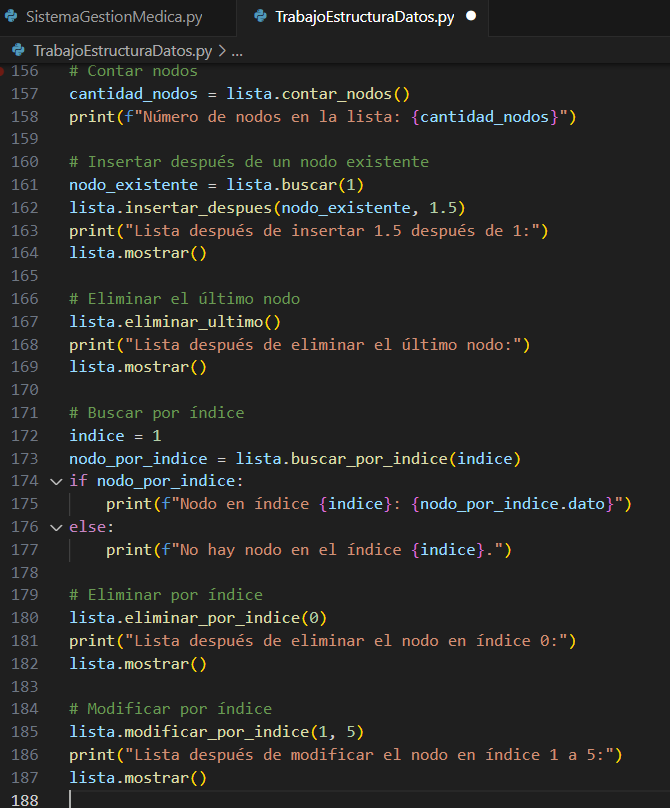
Pantalla de computadora con letras

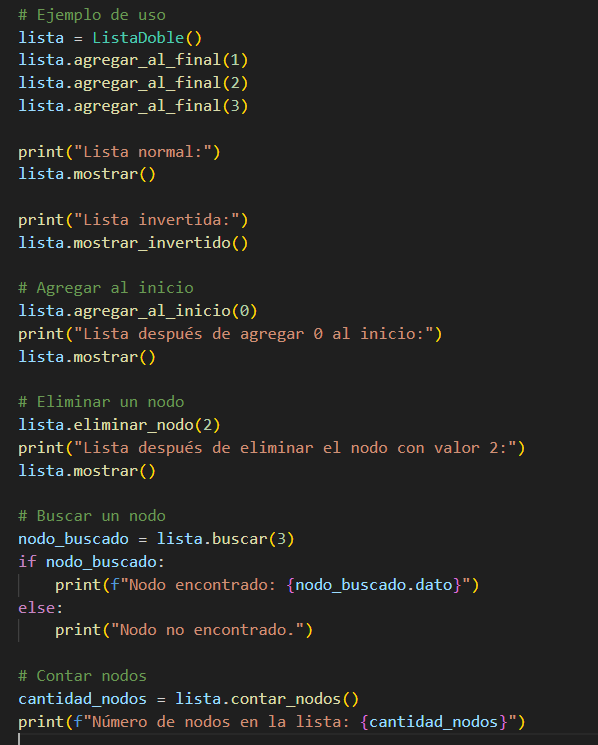
Descripción generada automáticamente con confianza mediaMétodo eliminar ultimo nodo

Texto

Descripción generada automáticamenteMétodo buscar por el indice

Método eliminar nodo por el indice

Creación de objetos e implementación



Texto

Descripción generada automáticamenteDemostración del código

# Bibliografía

Chalita Pérez, J. R. (21 de noviembre de 2020). *Implementando una lista doblemente*

*ligada en Python.* Medium. <https://www.rastreator.com/hipotecas/consejos/pedir-una-hipoteca.aspx>

Pozo, S. (2001). Capítulo 5 Listas doblemente enlazadas. *Con Clase*

*.*<https://conclase.net/c/edd/cap5>

Zohonero Martínez, I. & Joyanes Aguilar, L. (2008). *Estructuras de datos en Java*: ( ed.). McGraw-Hill España. <https://elibro.puce.elogim.com/es/lc/puce/titulos/50117>