
REPRODUCTOR DE MUSICA - TDA

201807201 – Juan Pablo Pérez Búc

Resumen

Se elaboró un reproductor de audio capaz de ordenar y clasificar los audios según el álbum y el artista al que pertenecen, para ello se implementaron estructuras de datos dinámicos como “Listas doblemente enlazadas” y su variante “Lista circular doblemente enlazada”.

Las estructuras de datos se basaron en la teoría de grafos, basándose en sus características de nodos (vértices) y apuntadores (aristas) que emulan un grafo dirigido.

Se implementaron los tipos de datos abstractos para poder manejar de manera eficiente y optimizar el rendimiento del software respecto a la utilización de la memoria de la máquina en la que se desplegará.

El manejo de las estructuras de datos en los objetos creados (a partir del Paradigma Orientado a Objetos o POO) fue vital para optimizar la organización de los datos dado que facilita el manejo de los datos al abstraer los objetos de la vida real a un ambiente informático.

Palabras clave

- Estructuras

- Abstracción
- Objetos
- Datos
- Memoria

Abstract

An audio player capable of sorting and classifying the audios according to the album and the artist to which they belong was developed, for this, dynamic data structures such as "Doubly linked lists" and its variant "Double linked circular list" were implemented.

The data structures were based on graph theory, based on their characteristics of nodes (vertices) and pointers (edges) that emulate a directed graph.

Abstract data types will be implemented to be able to efficiently handle and optimize the performance of the software with respect to the use of the memory of the machine on which it will be deployed.

The management of the data structures in the created objects (from the Object Oriented Paradigm or OOP) was vital to optimize the

organization of the data since it facilitates the management of the data by abstracting the objects from real life to an environment computer scientist.

Keywords

- Structures
- Abstraction
- Objects
- Data
- Memory

Introducción

La teoría de grafos, elemento de las matemáticas discretas, tiene, en su forma más básica, a los nodos como objeto de estudio, ahora bien, en las estructuras de datos implementaremos la abstracción de algunos tipos de grafos básicos para su creación, por ejemplo, los grafos dirigidos y circuitos, estos grafos representaran a la Lista doblemente enlazada y a la Lista doblemente enlazada circular, respectivamente.

La implementación de estas estructuras de datos fue para poder aprovechar la memoria del equipo de la manera más optima posible, haciendo uso de ella de una manera dinámica, es decir la petición y liberación de memoria en tiempo de ejecución del software, logrando así, evitar el acaparamiento de memoria que podría no llegar a utilizarse.

La abstracción, un pilar de la programación orientada a objetos, es necesaria para poder representar los grafos en líneas de código.

Desarrollo del tema

a) Memoria Dinámica

La importancia de la memoria dinámica es vital en los programas más recientes, dado que tienden a exigir cada vez más a exigir más recurso de los ordenadores, poder gestionar la memoria de forma en que el programa solo requiera la menor cantidad de recursos posibles permite un sistema más estable. Es aquí donde entran en juego las Estructuras de Datos dinámicas, es decir, son estructuras que no tienen un tamaño máximo, y cuyo mínimo es el nulo. El uso de este tipo de estructuras es necesario para optimizar el rendimiento del software dado que este solo solicita memoria cuando ha de utilizar un nuevo espacio y no solicita una cantidad predefinida como ocurre con el software basado en memoria estática, poder solicitar memoria en tiempo de ejecución también está ligado a su liberación en la misma línea de tiempo, haciendo posible que el ordenador pueda disponer de la memoria liberada para otras operaciones.

b) Teoría de Grafos:

Ahora, habiendo comprendido la necesidad de la implementación de la memoria dinámica, se explicará brevemente aspectos de la teoría de grafos:

Nodo:

Un nodo es el valor mínimo de un grafo, este puede existir unitariamente o estar enlazado a otros nodos por medio de aristas, puede estar incluso enlazado a sí mismo.

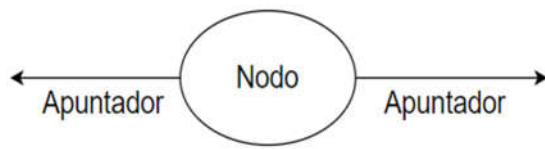


Ilustración 1 : Vértice con aristas dirigidas

Fuente: elaboración propia

Grafo Dirigido:

Un grafo dirigido es aquel cuyos vértices apuntan directamente hacia otro nodo o incluso hacia ningún lado, en nuestro caso, nuestra implementación será la lista doblemente enlazada.

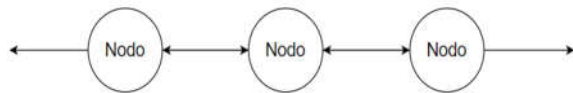


Ilustración 2 Grafo dirigido que representa una Lista Doble

Fuente: elaboración propia

Circuito:

Un circuito es un tipo de grafo cuyos nodos están enlazados de manera que hay una al menos una forma de recorrer todos los nodos y volver al inicio sin pasar dos veces por un mismo nodo.

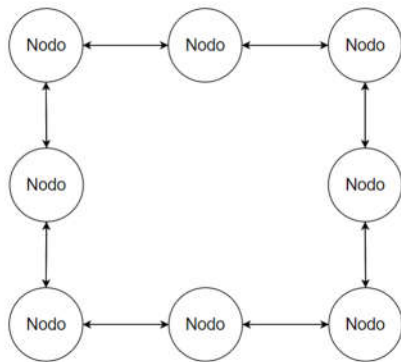


Ilustración 3: Grafo en forma de circuito

Fuente: Elaboración propia

c) Paradigma Orientada a Objetos

Este paradigma de la programación se basa en poder abstraer objetos de la vida real de manera en que podamos entender sus cualidades (atributos) y acciones (métodos o funciones) y poder llevar dichos elementos a un plano de código, por ejemplo, en nuestro caso se ha abstraído un *nodo* y *dos tipos de grafos*, como lo son el grafo dirigido y el grafo circuito.

d) Estructuras de datos:

Las estructuras de datos son la culminación de la abstracción realizada los grafos. Estos se basan en un tipo de dato específico, el *nodo*, en el caso de las estructuras de datos, cada nodo apunta hacia una dirección de memoria en la que se almacena algún valor, puede ser desde un entero, una cadena o un valor booleano hasta incluso otro objeto, podría ser incluso otra estructura de datos.

En el caso de una lista doblemente enlazada se basa en un nodo con dos apuntadores, uno que se dirige hacia un nodo siguiente y otro que se dirige hacia un nodo anterior, en caso de que no exista alguno de esos nodos, apunta hacia un valor nulo, una lista doblemente enlazada se compone también por dos nodos de referencia, el nodo de cabecera y el nodo final. Una lista doblemente enlazada puede no tener nodo alguno dentro de sí.

Una lista doblemente enlazada circular, es, en sí mismo, lo mismo que se ha explicado anteriormente, la única diferencia es que el nodo de cabecera y el nodo final, apuntan a sus direcciones de memoria, logrando así, que la lista se convierta en un circuito.

Ambos tipos de listas, aunque similares su implementación en este proyecto fue para casos específicos.

Conclusiones

La creación de software basado en la implementación de memoria dinámica es, en el mejor de los casos, la forma más optima de operar en el sistema y en el peor de los casos, la única manera, dada la magnitud y el alcance del software, haciendo necesaria la comprensión desde la base que sería la teoría de grafos hasta su implementación por medio del “Paradigma orientado a objetos” para realizar su abstracción e implementación y permitir que el software pueda hacer uso del principio de solicitud y liberación de memoria en tiempo de ejecución.

Referencias bibliográficas

(N.d.). Edu.Co. Consultado Diciembre 19, 2021,

desde

https://www.unipamplona.edu.co/unipamplona/portallIG/home_23/recursos/general/11072012/grafos3.pdf

