|  |
| --- |
| **PIXEL PRINT STUDIO**  FASE 2 |
| **202201947– PABLO ANDRES RODRIGUEZ LIMA** |

**Resumen**

Los árboles en estructuras de datos son un tipo de estructura jerárquica que consta de nodos interconectados. Cada nodo tiene un valor y puede tener cero o más nodos hijos, organizados en una estructura de árbol. Estos árboles se utilizan para representar relaciones jerárquicas entre elementos, como estructuras de archivos en sistemas operativos o relaciones de parentesco en genealogías. Ofrecen eficiencia en la búsqueda, inserción y eliminación de elementos, siendo esenciales en la implementación de algoritmos y aplicaciones informáticas complejas.

**Palabras clave**

1. Jerárquico
2. Nodos
3. Relaciones padre-hijo
4. Búsqueda
5. Eficiencia

***Abstract***

*Trees in data structures are hierarchical structures consisting of interconnected nodes. Each node holds a value and can have zero or more child nodes, organized in a tree-like structure. These trees are used to represent hierarchical relationships between elements, such as file structures in operating systems or parent-child relationships in genealogies. They offer efficiency in searching, inserting, and deleting elements, being essential in implementing complex algorithms and computer applications.*

*Principio del formulario*

***Keywords***

1. Hierarchical
2. Nodes
3. Parent-child Relationships
4. Search
5. Efficiency

**Introducción**

En el mundo de la programación, las Estructuras de Datos Abstractas (EDA) son fundamentales para organizar y manipular datos de manera eficiente. Una EDA es una colección de datos y un conjunto de operaciones definidas en esos datos, independientes de cualquier implementación concreta. Estas estructuras proporcionan un marco conceptual para entender cómo los datos están organizados y cómo se puede acceder a ellos y manipularlos.

**Desarrollo del tema**

En el ámbito de la programación, las Estructuras de Datos Abstractas (EDA) son herramientas esenciales para organizar y manipular datos de manera eficiente. Estas estructuras proporcionan un enfoque sistemático para almacenar y acceder a datos, independientemente de la implementación subyacente. Entre las EDA más comunes se encuentran las listas enlazadas simples, las listas enlazadas dobles, las listas circulares, las estructuras tipo pila y las estructuras tipo cola.

**Dato**: Un dato es una unidad de información básica que puede ser procesada por un computador. Los tipos de datos más comunes son números, letras y símbolos.

* Los datos pueden ser clasificados en diferentes tipos, como:
  + **Tipos básicos:** enteros, reales, booleanos, caracteres.
  + **Tipos compuestos:** cadenas de texto, arrays, estructuras.

**Estructura de Datos**: Una estructura de datos es una forma específica de organizar y almacenar datos en la memoria de un computador para que puedan ser utilizados de manera eficiente. Cada estructura de datos tiene sus propias características y ventajas para diferentes tipos de operaciones.

Algunas de las operaciones más comunes que se pueden realizar sobre las estructuras de datos son:

* **Búsqueda:** encontrar un elemento específico dentro de la estructura.
* **Inserción:** agregar un nuevo elemento a la estructura.
* **Eliminación:** remover un elemento de la estructura.
* **Recorrido:** acceder a todos los elementos de la estructura en un orden específico

**Árbol Binario:** Un árbol binario es una estructura de datos no lineal en la que cada nodo tiene como máximo dos hijos. Los nodos se pueden organizar de forma recursiva, creando una estructura con subárboles.

* Los árboles binarios se utilizan para representar relaciones jerárquicas entre datos.
* Un ejemplo de un árbol binario es un árbol genealógico.

**Árbol AVL**: Un árbol AVL es un tipo de árbol binario auto-balanceado, lo que significa que la altura de cada subárbol siempre es similar. Esta propiedad permite realizar operaciones de búsqueda, inserción y eliminación de forma eficiente.

* Los árboles AVL son más eficientes que los árboles binarios normales para operaciones de búsqueda, inserción y eliminación.
* Se utilizan en aplicaciones donde la velocidad de acceso a los datos es importante.

**Lista Doblemente Enlazada:** Una lista doblemente enlazada es una estructura de datos lineal en la que cada nodo tiene dos punteros: uno al nodo anterior y otro al siguiente. Esta estructura permite acceder a los datos en ambos sentidos, facilitando la eliminación e inserción de nodos en cualquier posición.

* Las listas doblemente enlazadas son más flexibles que las listas simples para realizar operaciones de inserción y eliminación de nodos.
* Se utilizan en aplicaciones donde es necesario modificar la lista con frecuencia.

**Árbol B:** Un árbol B es un árbol balanceado que puede tener más de dos hijos por nodo. Esta estructura se utiliza para almacenar grandes cantidades de datos de forma eficiente, ya que permite realizar búsquedas y operaciones de actualización en tiempo logarítmico.

* Los árboles B son más eficientes que los árboles binarios para almacenar grandes cantidades de datos.
* Se utilizan en bases de datos y sistemas de archivos.

**Lista Simple:** Una lista simple es una estructura de datos lineal en la que cada nodo tiene un solo puntero al siguiente nodo. Es la estructura de datos más simple y se utiliza para almacenar datos en orden secuencial.

* Las listas simples son la estructura de datos más simple y eficiente en términos de memoria.
* Se utilizan para almacenar datos en orden secuencial.

**FORTRAN FPM para la Lectura de JSON**: FPM (Flexible Parsing Model) es una extensión del lenguaje de programación FORTRAN que permite leer y escribir archivos JSON de forma sencilla. FPM proporciona una serie de herramientas para analizar la estructura del archivo JSON y acceder a sus datos.

* FPM es una herramienta poderosa para leer y escribir archivos JSON en FORTRAN.
* FPM facilita el acceso a los datos dentro del archivo JSON.

**Matriz Dispersa:** Una matriz dispersa es una matriz en la que la mayoría de los elementos son nulos. Se utiliza para almacenar datos que no están distribuidos uniformemente, lo que permite ahorrar espacio en memoria.

* Las matrices dispersas son una forma eficiente de almacenar matrices con muchos elementos nulos.
* Se utilizan en aplicaciones donde la mayoría de los elementos de la matriz son nulos.

**Conclusiones**

Las Estructuras de Datos Abstractas son herramientas esenciales en programación para organizar y manipular datos de manera eficiente.

Las listas enlazadas simples, dobles y circulares, junto con las estructuras tipo pila y cola, son ejemplos fundamentales de Estructuras de Datos Abstractas con aplicaciones variadas en el desarrollo de software.

Dominar estas estructuras es crucial para mejorar la eficiencia y la calidad del código, así como para resolver una amplia gama de problemas de programación de manera efectiva.

**Referencias bibliográficas**

*Programación II - Tipo de Dato Abstracto*. (n.d.). https://sites.google.com/site/programacioniiuno/temario/unidad-2---tipo-abstracto-de-dato/tipo-de-dato-abstracto

*Estructuras de datos: listas enlazadas, pilas y colas.* (n.d.). https://calcifer.org/documentos/librognome/glib-lists-queues.html