
Implementación de TDA en base a archivos de entrada en formato XML bajo el concepto de POO.

Carnet 201904013 – Marlon Isaí Figueroa Farfán

Resumen

Para llevar el planteamiento integral que implemente tipos de datos abstractos (TDA) en base a archivos de entrada en formato XML bajo el concepto de programación orientada a objetos se empezó definiendo los conceptos básicos sobre los tipos de datos abstractos, que funcionalidad tienen, dando enfoque a las listas doblemente enlazadas y la lista ortogonal. También se definieron las partes elementales de la programación orientada a objetos y de qué manera estas se relacionan con los tipos de datos abstractos para así llegar a una buena optimización de memoria y código en las soluciones planteadas para la correcta manipulación de datos en la listas doblemente enlazadas y ortogonales.

Palabras clave

1. **TDA:** Tipo de Dato Abstracto.
2. **Nodo:** Unión de varios elementos.
3. **POO:** Programación Orientada a objetos.
4. **Matriz:** Matriz es un arreglo bidimensional de datos.
5. **Lista:** Estructura dinámica de datos.

Abstract

To carry out the integral approach that implements abstract data types (ADT) based on input files in XML format under the concept of object-oriented programming, we started by defining the basic concepts about abstract data types, their functionality, focusing on doubly linked lists and the orthogonal list.

We also defined the elementary parts of object oriented programming and how they are related to abstract data types in order to reach a good optimization of memory and code in the solutions proposed for the correct manipulation of data in the doubly linked and orthogonal lists.

Keywords

1. **TDA:** Abstract Data Type.
2. **Node:** Union of several elements.
3. **OOP:** Object Oriented Programming.
4. **Matrix:** matrix is a two-dimensional array of data.
5. **List:** Dynamic data structure.

INTRODUCCIÓN

Día con día se presentan diferentes tipos de problemas en el campo de la programación para los cuales se puede buscar la optimización de las soluciones planteando soluciones integrales que implementen tipos de datos abstractos para una mejor administración de memoria en el sistema y poder optimizar el desarrollo de la solución aplicando los conceptos de programación orientada a objetos y así poder llevar a cabo una mejor manipulación de datos dentro del sistema.

Desarrollo del tema

I. ¿QUÉ ES UN TDA?

Un Tipo de dato abstracto (*TDA*) es un conjunto de datos u objetos al cual se le asocian operaciones. El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en cómo estén implementadas dichas operaciones. Esto quiere decir que un mismo TDA puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad.

II. PROGRAMACIÓN ORIENTADA A OBJETOS

Para definir la funcionalidad de la programación orientada a objetos se debe mencionar que ésta se compone de:

- Clases
- Objeto
- Atributos
- Métodos

Objeto: Permite separar los diferentes componentes de un programa, simplificando su elaboración.

A los objetos se les otorga ciertas características en la vida real. Cada parte del programa que se desea realizar es tratado como objeto.

Clase: Consta de una serie de métodos y datos que resumen las características de este objeto. Definir clases permite trabajar con código reutilizable.

Atributos: Tiene unas características que le identifican como tal, así:

OBJETO	CARACTERÍSTICAS
rectángulo	lados
	ángulos
persona	nombre
	cédula
	fecha Nacimiento
libro	título
	autor
	año

Figura I. Ejemplo de Atributo

Fuente: elaboración propia

Métodos: Sobre los objetos es posible realizar acciones (operaciones, cálculos), para lo cual podría utilizarse la pregunta, ¿Qué se puede hacer sobre el objeto X? Las respuestas a esta interrogante facilitarán la identificación de las funcionalidades de los objetos.

OBJETO	FUNCIONALIDADES
rectángulo	CalcularArea
	CalcularPerimetro
detalleFactura	CalcularIVA
	CalcularDescuento
	CalcularTotal
transacción	Depositar
	Retirar
	CalcularIntereses

Figura II. Ejemplo de Métodos

Fuente: Elaboración propia

III. ¿QUÉ RELACIÓN TIENE CON LA POO?

El paradigma de orientación a objetos permite el encapsulamiento de los datos y las operaciones mediante la definición de clases e interfaces, lo cual permite ocultar la manera en cómo ha sido implementado el TDA y solo permite el acceso a los datos a través de las operaciones provistas por la interfaz.

IV. TIPOS DE DATOS ABSTRACTOS

- LISTA ENLAZADA
- PILA
- COLA

Para entender este tipo de estructuras es necesario tener claro algunos conceptos:

PUNTERO/APUNTADO

Un puntero es un espacio en la memoria que almacena una referencia o dirección de memoria de otra variable, conocida como su apuntado. El valor de un puntero puede ser **NULL**, lo que significa actualmente no se refiere a ningún apuntado.

REFERENCIA

La operación de referencia en un puntero sirve para tener acceso a su apuntado.

ASIGNACIÓN DE PUNTERO

Una operación de asignación entre dos punteros, como $p=q$, hace que los dos punteros se refieran (o apunten) al mismo apuntado.

No se copia dos veces en la memoria al apuntado, sino que los dos punteros almacenan la dirección de memoria del apuntado.

V. OPERADORES BASICOS

Los operadores básicos de una lista enlazada son:

Insertar: Inserta un nodo con dato x en la lista, pudiendo realizarse esta inserción al principio o final de la lista o bien en orden.

Eliminar: Elimina un nodo de la lista, puede ser según la posición o por el dato.

Buscar: Busca un elemento en la lista.

Localizar: Obtiene la posición del nodo en la lista.

Vaciar: Borra todos los elementos de la lista

VI. TIPOS DE LISTAS ENLAZADAS

1. Lista simple.
2. Lista doblemente enlazada.
3. Lista simple circular.
4. Listas enlazadas doblemente circulares.

Simplemente enlazadas:

Permiten recorrer la lista en un solo sentido y desde la cabeza hasta la cola.

Doblemente enlazadas:

Permiten el recorrido en dos direcciones, de la cabeza a la cola y de la cola hacia la cabeza.

Simplemente circulares:

Permiten el recorrido en una dirección, pero al llegar al último nodo (cola) este se encuentra comunicado o enlazado a la cabeza, haciendo un anillo o círculo si se representa gráficamente.

Doblemente circulares:

Permiten el recorrido en ambas direcciones y la cabeza y cola se encuentran conectadas en ambas direcciones.

LISTA DOBLEMENTE ENLAZADA

La característica principal de una lista doble lineal es que las ligas del último nodo y del primer nodo apuntan hacia el valor nulo.

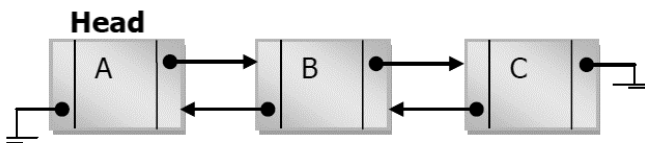


Figura IV. Ejemplo de Lista Doblemente Enlazada

Fuente: Elaboración propia

El nodo de una lista doble lineal debe contener como mínimo tres campos: uno para almacenar la información y otros dos para guardar la dirección de memoria del nodo antecesor y sucesor.

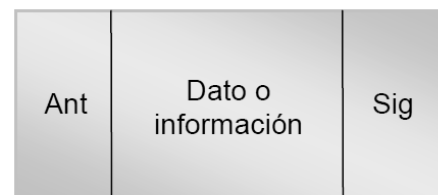


Figura IV. Estructura de nodo de lista doble enlazada

Fuente: Elaboración propia

LISTA ORTOGONAL

La característica principal de una lista ortogonal lineal es que las ligas de los últimos nodos apuntan hacia el valor nulo.

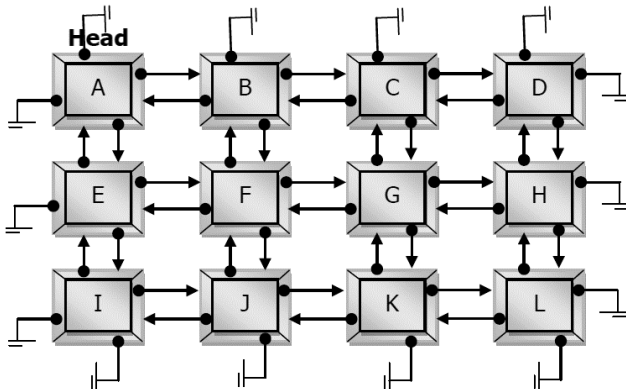


Figura IV. Estructura de nodo de lista doble enlazada

Fuente: Elaboración propia

El nodo de una lista ortogonal debe contener como mínimo cinco campos: uno para almacenar la información y cuatro para guardar la dirección de memoria hacia el siguiente, anterior, arriba y abajo nodo de la lista.

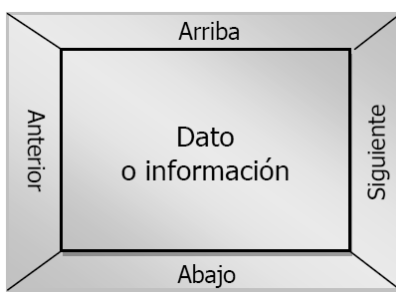


Figura IV. Estructura de nodo de lista ortogonal

Fuente: Elaboración propia

VII. ¿QUÉ ES UN ARCHIVO?

Para poder acceder a determinada información en cualquier momento, se necesitará que ella esté almacenada en forma permanente. Este es el caso de la memoria externa o auxiliar en las cuales la información permanece allí.

La forma de guardar los datos en estos dispositivos auxiliares es mediante unas estructuras llamadas **archivos** o ficheros.

ARCHIVO XML

XML son las siglas de *Extensible Markup Language*, los archivos XML contienen información de cualquier tipo, se componen de etiquetas que nos aportan datos e información que queremos procesar. Estas etiquetas pueden estar de forma **individual** o **anidadas**.

ESTRUCTURA

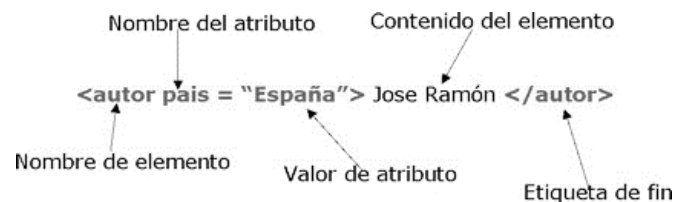
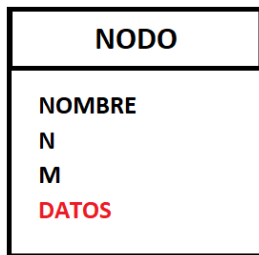


Figura V. Estructura de archivo XML

Fuente: Elaboración propia

VIII. IMPLEMENTACIÓN DE CONCEPTOS

Se aplicaron los conceptos definidos anteriormente para llevar a cabo la elaboración de un programa utilizando como objeto a los Nodos con la siguiente estructura.



Donde:

- **NOMBRE:** Nombre de la Matriz.
- **N:** Número de filas.
- **M:** Número de Columnas
- **DATOS:** Lista Ortogonal.

Para acceder a los datos de la lista ortogonal se debe recurrir al

Conclusiones

- La implementación de POO facilita el desarrollo de la solución para proyectos.
- La utilización de estructuras de programación secuenciales, cíclicas y condicionales permiten un mejor desarrollo y optimización.
- Para una mejor visualización de la correcta manipulación de datos con TDA's se utilizó un software externo (Graphviz).
- La utilización de archivos XML como insumos para la lógica y comportamiento contribuye a una óptima resolución.

Referencias bibliográficas

Chile, U., 2015. *CC3001 Algoritmos y Estructuras de Datos: Tipos de datos abstractos*. [online] Users.dcc.uchile.cl. Available at: <<https://users.dcc.uchile.cl/~bebustos/apuntes/cc3001/TDA/>> [Accessed 5 March 2021].

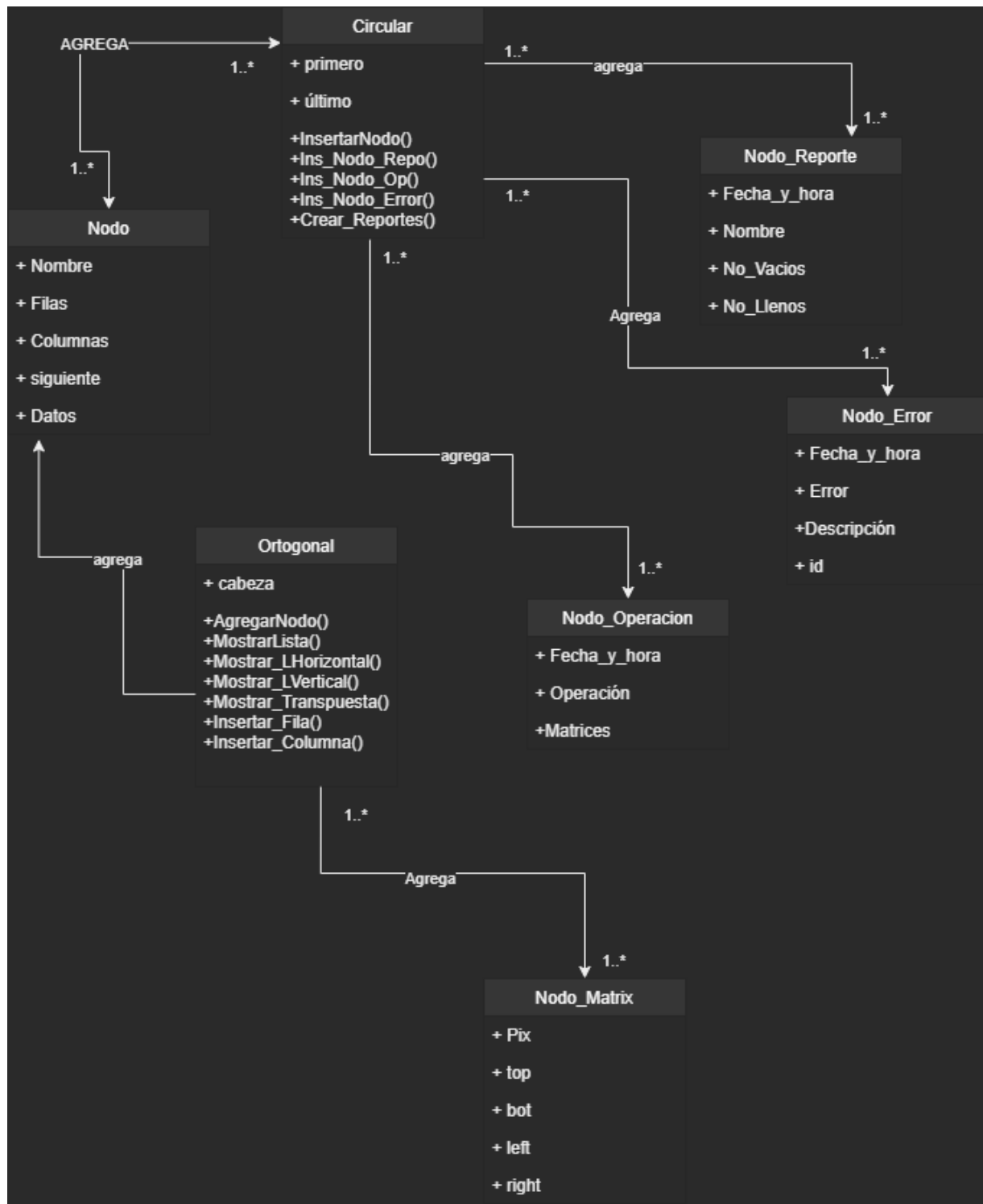
GraphEverywhere. 2021. *Grafos | Qué son, tipos, orden y herramientas de visualización*. [online] Available at: <<https://www.grapheverywhere.com/grafos-que-son-tipos-orden-y-herramientas-de-visualizacion/>> [Accessed 5 March 2021].

Morero, F., 2000. *Introducción a la OOP*. [online] Kataix.umag.cl. Available at: <<https://kataix.umag.cl/~ruribe/Utilidades/Introduccion%20a%20la%20Programacion%20Orientada%20a%20Objetos.pdf>> [Accessed 5 March 2021].

Moya, R., 2002. *Estructuras de datos: listas enlazadas, pilas y colas*. [online] Calcifer.org. Available at: <<https://calcifer.org/documentos/librognome/glib-lists-queues.html>> [Accessed 5 March 2021].

Upload.wikimedia.org. 2021. *Estructura de Datos*. [online] Available at: <<https://upload.wikimedia.org/wikipedia/commons/5/51/APUNTES.pdf>> [Accessed 26 March 2021].

IX. DIAGRAMA DE CLASE



X. DIAGRAMA DE ACTIVIDADES

