

---

## Proyecto 1 de programación y computación 2

---

201900716 – Kelly Mischel Herrera Espino

### Resumen

Con la creación del proyecto se busca el almacenamiento de matrices de forma dinámica. Para ello se implementa el uso de listas simplemente enlazada siendo esta una estructura de datos. Para logra obtener la reducción de las matrices ingresadas por medio del archivo XML se realiza una matriz binaria, se verifica el patrón que tiene cada fila y se hace una comparación si estos coinciden se suman lo números de la matriz original. La fila que no se repite se copia de la forma original. Al finalizar esta reducción se procede a escribir el archivo de salida con la matriz reducida. Por ultimo se genera un grafica con graphviz con la matriz de entrada que el usuario tenga a disposición escoger. El cual se abrirá una vez creado y se mostrará en pantalla.

### Palabras clave

- Lista
- TDA
- Estructura de datos
- Base de datos.
- Nodo

### Abstract

*With the creation of the project, the storage of matrices is sought dynamically. For this, the use of simply linked lists is implemented, this being a data structure. In order to obtain the reduction of the matrices entered by means of the XML file, a binary matrix is made, the pattern of each row is verified and a comparison is made if they coincide, the numbers of the original matrix are added. The non-repeating row is copied from the original shape. At the end of this reduction, the output file is written with the reduced matrix. Finally, a graph is generated with graphviz with the input matrix that the user has a choice arrangement. Which will open once created and will be displayed on the screen.*

### Keywords

*Traducción al idioma inglés de las palabras clave.*

- List
- TDA
- Data structure

- Database.
- Node

## Introducción

En el siguiente ensayo se abordan los temas que forman parte de una estructura de datos, lo beneficios que se obtienen al implementarla en el proyecto como una forma mas eficaz de almacenar datos ajustándose al tamaño de memoria que se necesita según la cantidad de datos que se guardaran. expandiéndose o reduciéndose en tiempo de ejecución sin tener que recurrir a definir un tamaño de forma estática el cual puede que no se llegue a utilizar por completo o llegue a faltar espacio. Almacena datos de forma dinámica es una de las principales funciones de una lista enlazada ya sea simple, doblemente enlaza o circular. Se puede tener acceso a los datos por medio de sus nodos y de los enlaces que estos tienen. Es por ello que se almacenaron las matrices que el usuario ingresaba por medio de la carga de un archivo XML y utilizando el modelo Elementtree se pudo obtener los dato y almacenarlos en un una lista. Para posterior a ello crear una matriz copia pero de compuesta por 0 y uno. Si las filas coincidían en el patrón que tuviesen se sumaban y sino cumplían se colocaban de la forma original. Por ultimo se almacenaba la matriz reducida en una nueva lista.

Debe contener un máximo de 150 palabras.

## Desarrollo del tema

Para almacenar datos de forma dinámica es necesario adéntranos en el tema de tipo de datos abstractos conocida por su acrónimo TDA o TAD. El almacenar datos de forma estática se vuelve poco práctico y ya no es recomendado al momento de almacenar datos de gran tamaño tomando el TDA la cantidad de memoria según sea necesario.

### a. TDA

Este es un tipo de dato definido por el programador.

Un TDA o TAD se compone de estructuras de dato y los procedimientos o funciones que manipulan esas estructuras de datos.

Los algoritmos utilizados para implementar cada una de las operaciones de los TAD esta encapsulados dentro de los propios TAD. Dicha característica de ocultamiento de la información significa que los objetos tienen interfaces publicas. Sin embargo , las representaciones e implementaciones de esas interfaces son privadas.

### b. Ventajas de utilizar TDA

- Por medio de un TDA se puede modelar el mundo real, mejorando la forma en que se representa y la compresión de este.
- Se encarga de mejorar la robustez de un programa. Estos permiten la comprobación de tipos para evitar errores en tiempo de ejecución.
- Al utilizar un TDA se mejora el rendimiento. Para sistemas tipificados, el conocimiento de los objetos permite la optimización de tiempo de complicación.
- Los TDA agrupan o localizan operaciones y la represión de atributos.

### c. Listas enlazadas

Una lista enlazada es una colección de objetos de una clase los cuales van en secuencia, a estos se les conoce como nodos por la autoreferencia que los objetos se hacen. Para acceder a una lista se hace por medio de la referencia del primer nodo llamado cabeza o inicio. El programa puede acceder a cada nodo subsiguiente a través de la referencia de enlace almacenada en el nodo anterior.

Un nodo puede contener datos de cualquier tipo, ya sea de tipo booleano, entero, decimal entre otros. Incluso incluyendo referencias a otras clases.

Una ventaja del uso de listas enlazadas es que son dinámicas, por lo que la longitud de dichas listas puede incrementarse o disminuir, adaptándose al tamaño que se necesite.

#### **d. Listas Circulares**

En una lista lineal simple o en una doblemente enlazada siempre hay un primer nodo llamado cabeza o inicio y un último nodo llamado cola. Una lista circular, por propia naturaleza o como su nombre lo dice funciona de forma circular apuntando al inicio. Es decir que no tiene ni principio ni fin. Sin embargo, resulta útil restablecer un nodo a partir del cual se accede a la lista y así poder acceder a sus nodos.

Las operaciones que se realizan sobre las listas circulares son similares a las que se realizan en una lista simplemente enlazada, teniendo en cuenta que no hay primero ni último nodo, aunque cabe mencionar que si hay un nodo de acceso.

Para la construcción de una lista circular se puede hacer con enlace simple o enlace doble.

#### **e. XML**

Es un lenguaje de marcado sus siglas significan Extensible Markup Language, este define un conjunto de reglas para la codificación de un documento.

Un archivo XML se divide en prolog y body. La parte de prolog consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios. La parte de body se compone en estructural y comentarios.

#### **f. ElementTree**

El `xml.etree.ElementTree` módulo implementa una API simple para la creación y el análisis de datos XML.

#### **g. Árbol y elementos XML**

Este es un formato de datos de forma jerárquica y la forma más natural de representarlo es con un árbol. El `ElementTree` representa todo el documento XML como un árbol y `Element` representa un solo nodo en este árbol. Las interacciones con un solo elemento XML y sus subelementos se realiza con `Element`.

#### **h. Almacenamiento de bases de datos en Listas:**

Para este proyecto se necesita almacenar matrices con  $n$  y  $m$  para poder guardar estos datos se necesita la creación de una lista simple con su clase nodo, en la cual se le va a colocar los atributos de la posición  $x$ ,  $y$  y el número que corresponde en esa posición. Esta lista se guarda en otra lista en este caso una lista circular junto con su clase nodo la cual tendrá como atributos el tamaño de la matriz  $m$  y  $n$ , así como una lista.

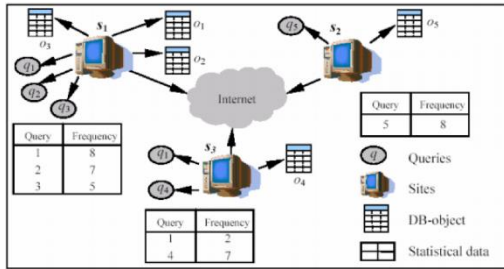


Figura 1. Problema de diseño de distribución en una base de datos.

Fuente: elaboración propia, o citar al autor, año y página

## i. Suma de filas

El objetivo del proyecto es que se obtengan los datos de un archivo XML por medio de elementTree y estos datos se guarden en las listas mencionada anteriormente. Con las matrices se hace una copia de esta pero con un patrón, si los numero que contiene la matriz son diferentes de cero se coloca el numero uno de no ser así se coloca el cero. Al tener esto se forma una matriz binaria.

Con la matriz binaria se verifica que filas tienen el mismo patrón para luego sumar estas filas con sus números originales. Las filas que no se repitan se colocan tal y como estaban en la primera lista.

Al obtener que filas se repiten y la suma de estas se almacena en otra lista para poder poder hacer uso de estos datos al momento de generar el archivo de salida.

## j. Generar grafica

Para generar la grafica de la matriz de entrada que el usuario escoja es necesario el uso de Graphviz para ellos se tiene que instala previamente dicho programa e importar en la clase para poder tener acceso a sus

métodos. Por medio de nodos se va haciendo referencia en el orden en que se va a mostrar la grafica obteniendo los datos de la lista circular. Al seleccionar el usuario la matriz se genera esta grafica de forma automática y posterior a ello se abre el archivo para poder visualizarla.

## Conclusiones

La creación de una estructura de datos suele ser difícil en el sentido de poner la referencia de los datos a los cuales se quiere acceder para poder obtener su información. Sin embargo es de mucha utilidad a la hora de manejar datos de gran cantidad en los cuales no se sabe el tamaño fijo que estos tendrán. Si fuese el caso de que la cantidad de datos es poco la lista se acoplan y reduce su tamaño al que se necesita. Pero si fuese el tamaño mayor se expande para acoplar a esta cantidad.

## Referencias bibliográficas

- Bibliotheca Python (2021), docs.python.org/3/library/xml.etree.elementtree.html
- D. Pual y H. Deitel, (2008). *Como programar en Java septima edition*. Pearson Education, Mexico.
- D. Pual y H. Deitel, (2012). *Como programar en Java novena edition*. Pearson Education, Mexico.
- L. Joyanes y I. Zahonero, (2008). *Estructura de datos en Java*. Mcgraw-Hill/Internamericana de España, S.A.U.
- C. R. Severance y I. Zahonero, (2009). *Python para todos*. Elliot Huaser, Sue Blumenberg .