

---

## TDA'S Y POO – PROYECTO1

---

201602659 – María Cecilia Cotzajay López

### Resumen

Se implemento TDA'S específicamente Listas Enlazadas Simples que fueron 3 de este tipo y una Lista Simple Circular, cada una de las listas tiene su propio Nodo.

El proyecto hace básicamente el procesamiento de Archivos con formato (.xml) el cual contiene una serie de datos para formar las listas de listas o simplemente llamas matrices. Una vez hecha la carga de datos procede a traducirla a forma binaria para posteriormente verificar sus filas y si estas coinciden en un patrón se obtienen índices que indican la frecuencia que tienen dicha fila en la matriz.

Finalmente se produce un archivo con la misma extensión (.xml) que contiene datos de la matriz de frecuencia o reducida que fue generada a partir de dichos cálculos realizados

### Palabras clave

- i). **Nodo:** relación que incluye un puntero por cada lista así como información propia de la relación.
- ii). **Matriz:** en este proyecto significa que son listas simples dentro de listas simples creadas a mano, es decir sin librerías propias de Python.

- iii) **Tipo de Dato Abstracto (TDA):** Modelo formal de un ente junto con un conjunto de operaciones definidas sobre el modelo que nos permite procesarlo.

### Abstract

*TDA's were specifically implemented Simple Linked Lists, which were 3 of this type and a Simple Circular List, each of the lists has its own Node.*

*The project basically does the processing of files with format (.xml) which contains a series of data to form the lists of lists or simply arrays of flames. Once the data has been loaded, it proceeds to translate it into binary form to later verify its rows and if they coincide in a pattern, there are indexes that indicate the frequency that said row has in the matrix.*

*Finally, a file is produced with the same extension (.xml) that contains data from the frequency or*

*reduced matrix that was generated from said calculations.*

### **Keywords**

i). **Node:** *relationship that includes a pointer for each list as well as information about the relationship.*

ii). **Matrix:** *in this project it means that they are simple lists within simple lists created by hand, that is, without Python's own libraries.*

iii) **Type of Abstract Data (ADT):** Formal model of an entity together with a set of operations defined on the model that allows us to process it.

### **Introducción**

Es muy importante aprender sobre el tema de los TDA's así como su implementación en los diferentes proyectos. Cabe mencionar que la cualidad mas importante sobre estos tipos de estructuras de datos es que son parte de la memoria dinámica, es decir que se puede utilizar para una cantidad grande de datos de los cuales muy probablemente desconozcamos el tamaño, es decir la cantidad. Si bien podemos mencionar otra cualidad de mayor importancia es que estos tipos de estructuras se pueden crear, manipular y manejar de la manera que nosotros queramos o gustemos.

El proyecto esta basado en el uso, implementación y manejo de estas estructuras de datos, para tener un conocimiento profundo en el funcionamiento de las mismas, además de que nos dan un numero finito de datos pero que al mismo tiempo es desconocido o no tenemos un numero exacto de ellos.

En el siguiente apartado se describe lo que conlleva el proyecto, es decir cada parte del mismo.

### **Desarrollo del tema**

Se utilizo un Nodo que contiene los siguientes atributos una clase llamada Casilla, un apuntador con el nombre siguiente. Posteriormente se creo una lista S

### **Conclusiones**

Efectivamente los TDA's son implementadas a gusto de cada quien, pues hay formas diferentes de hacerlas.

El uso de los TDA's en el proyecto hizo que comprendiera el funcionamiento en si de una matriz, se puede entender todo el procedimiento que hace por ejemplo un ArrayList que son plantillas que trae por defecto cada lenguaje de programación.

El procedimiento de la carga de Archivos hizo comprender que el procedimiento con los TDA's es muy largo pero que a su vez es fácil encontrar posibles errores ya que al crearlas uno mismo es posible encontrarlos muy fácil y rápidamente

### **Referencias bibliográficas**

Máximo 5 referencias en orden alfabético.

C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.

MULTILISTAS (ugr.es) (Para consulta de algunas palabras clave.)