

---

## NP-HARD. TDA'S

---

201709088 – Leonel Antonio González García

### Resumen

La programación orientada a objetos (POO) es una metodología de desarrollo que se basa en la idea de que los programas se pueden organizar como una colección de objetos interconectados, cada uno con su propio conjunto de datos y funcionalidades que hacen que el programa se desarrolle de forma eficiente.

Un tipo de dato abstracto es un modelo que define valores y las operaciones que se pueden realizar sobre estos. Este concepto es bastante utilizado en Poo.

Con las listas enlazadas somos capaces de manejar información de manera ordenada y rápida. También son una herramienta muy efectiva para trabajar con las estructuras Xml y así administrar su información de manera eficaz.

Mediante graphviz podemos dar a conocer de forma gráfica lo que nuestro programa es capaz de realizar y así el usuario tenga una perspectiva diferente y sea más fácil la interacción y el entendimiento del programa.

### Palabras clave

- POO
- TDA
- Listas Enlazadas
- Xml
- Graphviz

### Abstract

*Object-oriented programming (OOP) is a development methodology based on the idea that programs can be organized as a collection of interconnected objects, each with its own set of data and functionalities that make the program run efficiently.*

*An abstract data type is a model that defines values and the operations that can be performed on them. This concept is widely used in Poo.*

*With linked lists we are able to handle information in an orderly and fast way. They are also a very effective tool to work with Xml structures and manage your information efficiently.*

*With graphviz we can show graphically what our program is capable of doing so that the user has a different perspective and it is easier to interact and understand the program.*

### Keywords

- POO
- TDA
- Linked List
- Xml
- Graphviz

## Introducción

Mediante la implementación de técnicas y herramientas de programación que a través de la práctica y el estudio nos ayudan en la creación de programas que sirven a dar soluciones a problemas de forma rápida. En este caso tenemos la necesidad de crear un método para poder comprimir señales de audio mediante la Frecuencia y Amplitud, estos parámetros describen la señal del audio en función del tiempo. Se requiere un programa que ayude a disminuir la demanda de tiempo y recursos en la solución de este programa basado en una metodología de agrupamiento. Para esto se ha definido una matriz con los parámetros de tiempo y amplitud y por medio de esto hacer la conversión a una matriz agrupada que contenga los patrones de las frecuencias.

## Desarrollo del tema

La idea central de la aplicación es manejar información y trabajar con está de manera rápida y eficaz para crear gráficas que ayuden al usuario con el entendimiento de los datos y a la solución del problema.

Debemos tener en cuenta la interacción del usuario con el programa, por lo que se necesita una interfaz con la que el usuario pueda interactuar de manera fácil y de forma intuitiva. Esto lo podemos lograr con la ayuda de un menú que nos ayude a manejar de forma ordenada las opciones que el usuario tiene para interactuar con el programa y sirva de guía para el usuario.

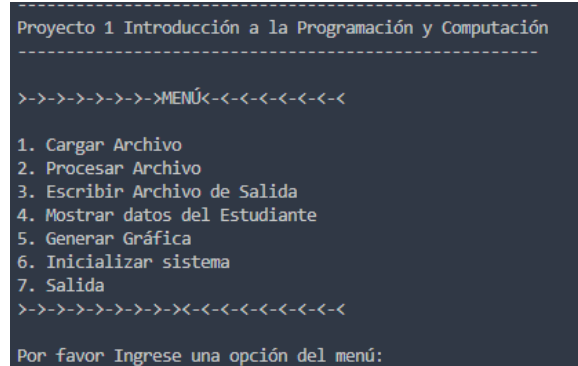


Figura 1. Menú principal.

Fuente: elaboración propia

Mediante la carga de archivos se realizarán la lectura y manejo de la información, la cual es proporcionada en archivo de tipo xml para lo que hay que conocer la manera en la que estos archivos están formados y sus características.

De esta manera nos adentramos en lo que es el manejo de objetos de forma jerárquica que es la forma en la que un archivo xml está formado. Necesitaremos conocer sobre listas enlazadas ya que serán las herramientas que nos ayuden a guardar y manejar la información que obtengamos de la lectura del archivo.

```
<INTERFACE>
  <NAME>"Nombre de la interfaz"</NAME>
  <DESCRIPCION>"Descripcion"</DESCRIPCION>
  <OBJECT>
    <TYPE>"tipo del objeto"</TYPE>
    <NAME>"nombre del objeto"</NAME>
    <TOP> coordenada X </TOP>
    <LEFT> coordenada Y </LEFT>
    ...
  </OBJECT>
  <OBJECT>
  </OBJECT>
</INTERFACE>
```

Figura 2. Estructura de Xml.

Fuente: Google imágenes

### **a. Carga de Archivo Xml**

Para realizar la lectura del archivo xml nos auxiliamos de librerías de Python que son útiles y facilitan la lectura y escritura de este tipo de archivos cómo lo son Element Tree o MiniDom.

Al cargar el archivo y leerlo con la ayuda de la librería utilizamos las listas para recorrer la información leída del archivo y agregarla a las listas enlazadas para poder manejar la información del archivo y trabajar de mejor forma.

Con la ayuda de las listas enlazadas nuestra información siempre estará conectada y nos facilitará el crear las matrices con los datos del tiempo y la amplitud.

### **b. Procesar Archivo y Escribir Archivo**

En esta parte se generarán las matrices y se agruparán para crear una matriz reducida.

Al tener nuestras listas enlazadas llenas con la información del archivo procedemos a crear nuestra matriz principal que contendrá la información inicial del archivo. Luego realizamos validaciones para el contenido de la frecuencia del archivo y así crear nuestra matriz de patrones que contendrá 0 si el dato es nulo o cero y 1 si el dato de la frecuencia es mayor a cero.

A partir de nuestra matriz de patrones podemos crear nuestra matriz reducida que contendrá la suma de las tuplas de la matriz de patrones de forma agrupada.

De esta manera usamos nuestra matriz reducida para escribir un archivo xml de salida que contendrá la información de los grupos y las frecuencias de las señales trabajadas.

### **c. Generar Gráficas**

Por medio de graphviz podemos obtener nuestras estructuras trabajadas de forma gráfica, lo cual nos es bastante útil para comprender de mejor forma el flujo de la aplicación. Con esta herramienta se pueden mostrar nuestras operaciones realizadas para obtener una perspectiva más clara y entendible de la solución que se le ha dado al problema.

Graphviz puede ser implementado mediante la librería “os” que ayuda a la fácil implementación de esta herramienta al proyecto.

### **Conclusiones**

-Es necesario la implementación de herramientas que ayuden al desarrollo del programa para facilitar la interacción del usuario y pueda tener una mejor comprensión de su funcionamiento y la solución que proporciona.

- Graphviz es una herramienta bastante útil para mostrar resultados de forma gráfica y su implementación es bastante accesible y eficaz con la gran mayoría de programas.

- Los archivos xml son de gran uso en el mundo de la programación debido a eso los lenguajes deben implementar librerías para así facilitar su manejo dentro de los programas.

- Las listas enlazadas son una forma bastante útil de trabajar ya que facilitan bastante el manejo de la información.

## Referencias bibliográficas

- M.E.S College of Engineering, (2006),  
Principles of data Structures using C and C++.
- School of Computer Science, University of  
Birmingham, UK, (2018), Data Structures and  
Algorithms.

## Anexos

### - Diagrama de clases:

Diagrama que muestra el flujo de la aplicación

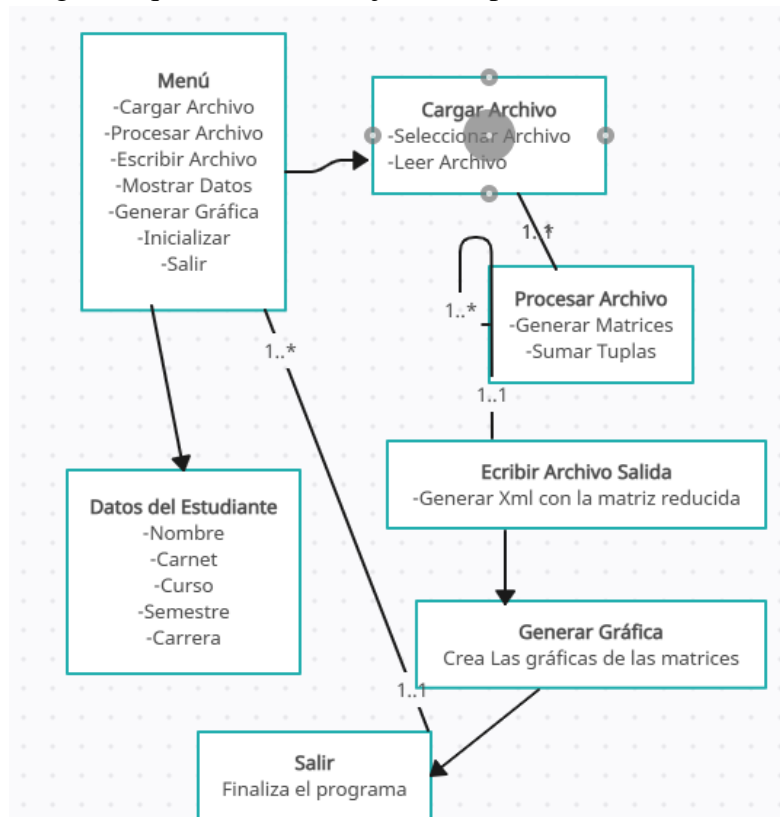


Figura 3 . Diagrama de Clases

Fuente: elaboración propia