

---

## IPC2 PROYECTO 1

---

202011059 – Kristel Jazmín Hernández Ricardo

### Resumen

El Centro de Investigaciones de la Facultad de Ingeniería se enfoca en un método innovador para comprimir señales de audio, centrándose en la frecuencia y la amplitud. Abordan el desafío NP-Hard, caracterizado por altos requisitos de tiempo y recursos, mediante una metodología de agrupamiento. Esta implica crear una matriz que relaciona tiempo, amplitud y frecuencia para diferentes señales de audio, transformándola en una matriz de patrones de frecuencia y agrupando tuplas similares. Estos patrones binarios indican cuándo y a qué amplitudes se encuentran frecuencias en las señales de audio.

Este enfoque busca mejorar la compresión de señales de audio, promoviendo la eficiencia en la transmisión y almacenamiento de datos de audio. La frecuencia (medida en Hz) y la amplitud (medida en dB) son elementos clave de estudio. El proyecto reconoce la complejidad del problema combinatorio NP-Hard, optando por una metodología que busca simplificar la representación de las señales de audio. Esto tiene el potencial de revolucionar la forma en que se gestionan y transmiten las señales de audio, contribuyendo a una mayor eficiencia en su uso y almacenamiento.

### Abstract

*The Research Center of the Faculty of Engineering is focusing on an innovative method for compressing audio signals, with a focus on frequency and amplitude. They address the NP-Hard challenge, characterized by high time and resource requirements, through a clustering methodology. This involves creating a matrix that relates time, amplitude, and frequency for different audio signals, transforming it into a matrix of frequency patterns and grouping similar tuples. These binary patterns indicate when and at what amplitudes frequencies are found in the audio signals.*

*This approach aims to enhance audio signal compression, promoting efficiency in audio data transmission and storage. Frequency (measured in Hz) and amplitude (measured in dB) are key elements of study. The project acknowledges the complexity of the NP-Hard combinatorial problem, opting for a methodology that seeks to simplify the representation of audio signals. This has the potential to revolutionize how audio signals are managed and transmitted, contributing to greater efficiency in their use and storage.*

## Introducción

En este proyecto, se busca desarrollar una solución integral para la compresión de señales de audio utilizando conceptos de programación orientada a objetos (POO), tipos de datos abstractos (TDA) y visualización de datos con Graphviz. Abordaremos el desafío de NP-Hard en la compresión de señales de audio, con el propósito de mejorar la eficiencia en la transmisión y almacenamiento de datos de audio. Este enfoque promete una solución innovadora y eficaz en este campo tecnológico en constante evolución.

## Desarrollo del tema

El desarrollo del proyecto se realizó con listas enlazadas que es una estructura de datos que se utiliza para organizar y almacenar una colección de elementos. A diferencia de las listas en Python que son arrays dinámicos, las listas enlazadas están formadas por nodos, donde cada nodo contiene un elemento de datos y una referencia (o puntero) al siguiente nodo en la secuencia. La estructura básica de una lista enlazada consiste en un nodo inicial llamado "nodo cabeza" (head) que marca el inicio de la lista y un nodo final que generalmente apunta a None para indicar el final de la lista.

El proyecto cuenta con el `NodoMatriz` que está diseñada para ser utilizada en la implementación de estructuras de datos que involucra matrices o listas enlazadas. A continuación, una explicación de cómo se conforman los datos de mi `NodoMatriz`:

`fila`: Almacena el número de fila en el que se encuentra este nodo.

`columna`: Almacena el número de columna en el que se encuentra este nodo.

`dato`: Almacena algún dato asociado con este nodo.

`datoBinario`: Almacena otro dato (presumiblemente en formato binario) asociado con este nodo.

`siguiente_fila`: Un puntero que apunta al siguiente nodo en la misma fila.

`siguiente_columna`: Un puntero que apunta al siguiente nodo en la misma columna.

En el proyecto se usaron las librerías: `xml.etree.ElementTree`: es un módulo en la biblioteca estándar de Python que se utiliza para procesar documentos XML de manera eficiente y sencilla. Permite analizar, crear y manipular documentos XML de una manera orientada a objetos. El módulo `ElementTree` proporciona una forma Pythonic de trabajar con XML, lo que facilita la lectura y escritura de datos en formato XML.

`Graphviz`: es una potente herramienta y una librería utilizada para visualizar grafos y estructuras de datos en forma de diagramas gráficos. Puede generar gráficos dirigidos y no dirigidos a partir de datos estructurados. `Graphviz` es ampliamente utilizado en diversas disciplinas, como la informática, la visualización de datos, la teoría de grafos y más.

El programa cuenta con 3 clases en total y son: `NodoMatriz`, `MatrizEnlazada` y `menu_prin`

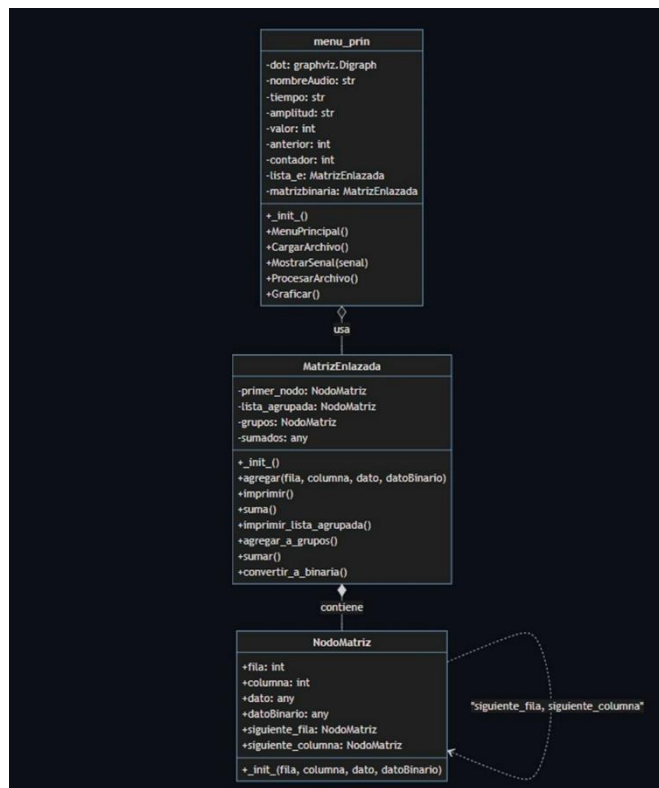
```
class NodoMatriz:
    def __init__(self, fila, columna, dato, datoBinario):
        self.fila = fila
        self.columna = columna
        self.dato = dato
        self.datoBinario = datoBinario
        self.siguiente_fila = None
        self.siguiente_columna = None
```

Figura 1. Nodo matriz

Fuente: elaboración propia.

[The lxml.etree Tutorial](#): La biblioteca lxml es ampliamente utilizada en Python para trabajar con XML debido a su eficiencia y facilidad de uso. Este tutorial te proporcionará información detallada sobre cómo leer y manipular archivos XML en Python utilizando lxml.

## Extensión: de cuatro a siete páginas como máximo



## E-grafía

### Visualización de Datos (Graphviz):

[Graphviz](#): El sitio oficial de Graphviz contiene documentación y ejemplos detallados.

[Data Structures - GeeksforGeeks](#): Este sitio web proporciona una amplia gama de tutoriales y ejemplos sobre tipos de datos abstractos y algoritmos.