

---

## PROYECTO 1

---

202200252 – Néstor Enrique Villatoro Avendaño

### Resumen

El Centro de Investigaciones de la Facultad de Ingeniería está experimentando un método para lograr comprimir señales de audio, por lo que se han enfocado en dos parámetros propios de las ondas de sonido: Frecuencia y Amplitud, estos parámetros describen la señal de audio en función del tiempo. La frecuencia es la cantidad de ciclos que se realizan en un segundo y se mide en Hertz (Hz), mientras que la amplitud representa la altura de la onda y hace referencia a la intensidad del sonido, la amplitud se mide en decibelios (Db).

El Centro de Investigación de la Facultad de Ingeniería ha abordado el problema de diseño para la compresión de señales de audio como un problema combinatorio NP-Hard ya que analizó que algunas de las situaciones comunes observadas cuando se resuelven instancias muy grandes de un problema NP-Hard son: Fuerte requerimiento de tiempo y fuerte demanda de recursos, por lo que optó por un método para resolver este tipo de problemas aplicando una metodología de agrupamiento.

### Palabras clave

Nodos, tuplas, listas simplemente enlazadas, POO (programación orientada a objetos), TDAs (tipos de datos abstractos).

### Abstract

*The Research Center of the Faculty of Engineering is experimenting with a method to compress audio signals, so they have focused on two parameters of sound waves: Frequency and Amplitude, these parameters describe the audio signal based on the time. The frequency is the number of cycles that are carried out in a second and is measured in Hertz (Hz), while the amplitude represents the height of the wave and refers to the intensity of the sound, the amplitude is measured in decibels (Db). The Research Center of the Faculty of Engineering has addressed the design problem for the compression of audio signals as an NP-Hard combinatorial problem since analyzed that some of the common situations observed when resolving instances very big problems of an NP-Hard problem are: Strong time requirement and strong demand resources, so he opted for a method to solve this type of problem by applying a clustering methodology.*

### Keywords

*Nodes, Tuples, simply linked lists, OOP(Object-oriented programming), ADTs (abstract data types)*

## Introducción

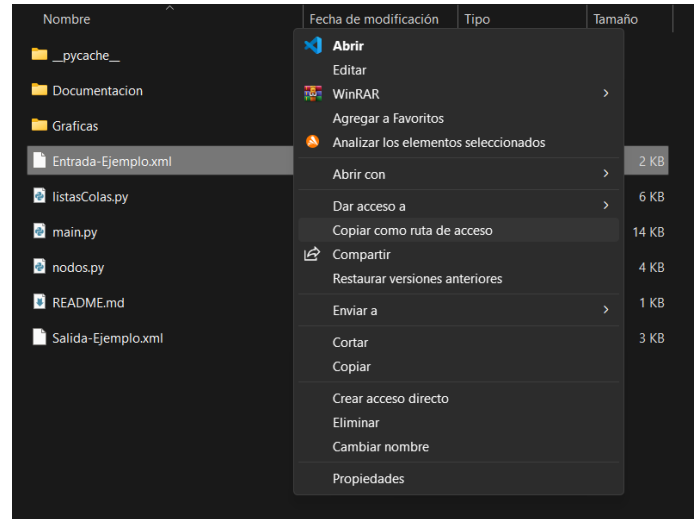
Para implementar esta metodología de agrupamiento, el Centro de Investigación ha definido una matriz de tiempo (t), amplitud (A) y frecuencias (f) para distintas señales de audio  $S[t][A]$ , transformarla en una matriz de patrones de frecuencia y agrupar las tuplas con el mismo patrón. La matriz de patrones de frecuencias para una matriz de frecuencias dada es la matriz binaria que indica en qué tiempos y amplitudes hay frecuencias en la señal de audio en estudio. Para ello se hará uso del lenguaje Python en el cual se implementarán listas simplemente enlazadas bajo el concepto de programación orientada a objetos (POO) y tipos de datos abstractos (TDAs). Y para el apartado grafico se hará uso de Graphviz.

## Desarrollo del tema

Para acceder a las diferentes funciones del programa se hará uso de un menú controlado por consola en el cual se podrán acceder a las siguientes funciones:

1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Mostrar datos del estudiante
5. Generar gráfica
6. Inicializar sistema
7. Salida

Para la primera opción del menú, cargar archivo, se le solicitará al usuario que ingrese la dirección del archivo con la extensión .xml la cual se puede encontrar dirigiéndose a la ubicación del archivo, dando click derecho y copiar como ruta de acceso, como se muestra en la siguiente imagen:



Si el proceso fue correcto, se guardará la ruta del archivo en una variable global y volverá al menú.

Para la segunda opción del menú, procesar archivo, se procederá a guardar los datos del archivo de entrada en una matriz hecha con listas simplemente enlazadas y nodos. Seguido de esto se convertirán a binarias, se compararán las filas y se sumarán las semejantes, generando así la matriz reducida. Todo este proceso se podrá apreciar en la terminal como se ve en las siguientes imágenes:

```
Ingrese una opción: 2
----- Procesar archivo -----
Mostrando datos originales...
Ejemplo1
| 0 | 59 | 84 | 0 |
| 5 | 80 | 97 | 2 |
| 5 | 5 | 0 | 50 |
| 17 | 48 | 33 | 72 |
| 26 | 37 | 0 | 0 |

Ejemplo2
| 17 | 81 | 39 | 82 |
| 21 | 66 | 0 | 59 |
| 88 | 98 | 20 | 68 |
| 56 | 50 | 44 | 0 |
| 55 | 14 | 90 | 0 |

Ejemplo3
| 28 | 85 | 0 | 0 | 0 |
| 22 | 0 | 81 | 15 | 5 |
| 66 | 49 | 88 | 82 | 83 |
| 57 | 76 | 11 | 31 | 41 |
```

Convirtiendo a binario...

Ejemplo1 en binario

0	1	1	0
1	1	1	1
1	1	0	1
1	1	1	1
1	1	0	0

Ejemplo2 en binario

1	1	1	1
1	1	0	1
1	1	1	1
1	1	1	0
1	1	1	0

Ejemplo3 en binario

1	1	0	0	0
1	0	1	1	1
1	1	1	1	1
1	1	1	1	1

Generando datos de salida...

Ejemplo1

0	59	84	0
22	128	130	74
5	5	0	50
26	37	0	0

Ejemplo2

105	179	59	150
21	66	0	59
111	64	134	0

Ejemplo3

28	85	0	0	0
22	0	81	15	5
123	125	99	113	124

Las matrices resultantes también guardarán el grupo de tiempos agrupados. Un ejemplo un poco mas ordenado y entendible de este proceso se puede ver en las siguientes imágenes:

ejemplo1				
0	59	84	0	
5	80	97	2	
5	5	0	50	
17	48	33	72	
26	37	0	0	
ejemplo 1 a binario				
0	1	1	0	
1	1	1	1	
1	1	0	1	
1	1	1	1	
1	1	0	0	
ejemplo 1 combinado				
0	59	84	0	G1 (1)
22	128	130	74	G2(2&4)
5	5	0	50	G3(3)
26	37	0	0	G4(5)
ejemplo2				
17	81	39	82	
21	66	0	59	
88	98	20	68	
56	50	44	0	
55	14	90	0	
ejemplo 2 a binario				
1	1	1	1	
1	1	0	1	
1	1	1	1	
1	1	1	0	
1	1	1	0	
ejemplo 2 combinado				
105	179	59	150	G1(1&3)
21	66	0	59	G2(2)
111	64	134	0	G3(4&5)
ejemplo3				
28	85	0	0	0
22	0	81	15	5
66	49	88	82	83
57	76	11	31	41
ejemplo 3 a binario				
1	1	0	0	0
1	0	1	1	1
1	1	1	1	1
1	1	1	1	1
ejemplo 3 combinado				
28	85	0	0	G1(1)
22	0	81	15	G2(2)
123	125	99	113	G3(3&4)

Para la opción 3, escribir archivo salida, se generará un archivo .xml con la siguiente estructura:

```
<senales>
  <senal nombre="Ejemplo1" t="4" A="4">
    <grupos G="1">
      <tiempo>1</tiempo>
      <datosGrupo>
        <dato t="1" A="1">0</dato>
        <dato t="1" A="2">59</dato>
        <dato t="1" A="3">84</dato>
        <dato t="1" A="4">0</dato>
      </datosGrupo>
    </grupos>
    <grupos G="2">
      <tiempo>2, 4</tiempo>
      <datosGrupo>
        <dato t="2" A="1">22</dato>
        <dato t="2" A="2">128</dato>
        <dato t="2" A="3">130</dato>
        <dato t="2" A="4">74</dato>
      </datosGrupo>
    </grupos>
  </senal>
</senales>
```

La cual contendrá los datos de la matriz reducida que a su vez tendrá un nombre, una amplitud (columnas), un tiempo (filas) y grupos con sus respectivos tiempos (filas) sumados.

Para la opción 4, mostrar datos del estudiante, simplemente se mostrarán los datos del creador del programa:

```
Ingrese una opción: 4
----- Mostrar datos del estudiante -----
Néstor Enrique Villatoro Avendaño
202200252
Introducción a la programación y computación 2 sección C
Ingeniería en ciencias y sistemas
4to semestre
```

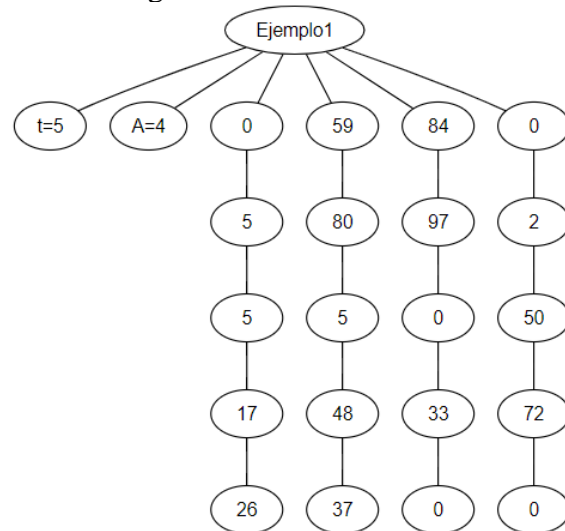
En la opción 5, Generar gráfica, se le solicitará al usuario el ingreso del nombre de una de las matrices para conseguir la grafica de su matriz original y matriz reducida. Esto por medio de la utilización de Graphviz y las los archivos .xml antes mencionados. Si todo se hizo correctamente saldrán los siguientes mensajes en la terminal y se generarán dos archivos en la carpeta Graficas, uno llamado Grafica-original.svg y Grafica-reducida.svg

```
Ingrese una opción: 5
----- Generar gráfica -----
Ingrese el nombre de la señal que desea graficar: Ejemplo1
Se generó la gráfica original exitosamente!
Se generó la gráfica reducida exitosamente!
```

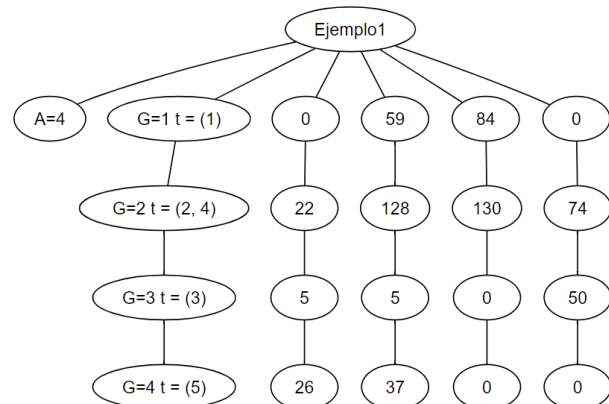
Nombre	Fecha de modificación	Tipo	Tamaño
Grafica-original.dot	6/09/2023 22:10	Word.Template.8	2 KB
Grafica-original.svg	6/09/2023 22:10	Chrome HTML Do...	11 KB
Grafica-reducida.dot	6/09/2023 22:10	Word.Template.8	2 KB
Grafica-reducida.svg	6/09/2023 22:10	Chrome HTML Do...	11 KB

Si abrimos los documentos en un navegador o un explorador de imágenes nos debería mostrar las gráficas anteriormente mencionadas de la siguiente forma y estructura:

Grafica original:



Grafica reducida:



En donde podemos apreciar que el nodo padre es el nombre de la matriz, los nodos hijos son, en el caso de la matriz original, el tiempo (cantidad de filas), la amplitud (cantidad de columnas) y la matriz ordenada. En el caso de la matriz reducida, la amplitud (cantidad de columnas), los grupos con sus respectivos tiempos y la matriz ordenada.

Para la opción 6, inicializar el programa, esta opción resetea todos los datos tanto de las listas enlazadas como de las matrices. Para la opción 7, Salir, finaliza el programa.

Datos a considerar: el programa no soportará una matriz de unos tiempos mayores a 3600 y una amplitud mayor a 130. Sin embargo, si existen mas matrices en un mismo archivo los leerá ignorando los que no cumplan con las reglas.

### Conclusiones

En conclusión, gracias al lenguaje Python y las facilidades que nos puede dar para hacer programación orientada a objetos (POO), se pueden facilitar muchos procesos como el de buscar datos en una matriz y mostrarlos.

La herramienta Graphviz es de una gran utilidad para visualizar tipos de datos abstractos (TDAs).

Los nodos y listas enlazadas por medio de estos son de gran utilidad a la hora de automatizar procesos como llenar una matriz de datos o cambiarlos, cabe resaltar que sin las estructuras de programación secuenciales, cíclicas y condicionales esto no sería posible.

Los archivos XML son de una gran utilidad a la hora de manejar datos en especial si son muchos y queremos mantenerlos ordenados e ingresarlos de forma limpia.

### Referencias bibliográficas

[Daniel Alfaro]. (2023, August 15). *IPC2 Sección C - Clase XML* [Video]. YouTube.

[https://youtu.be/ZTvRpX\\_3zAg?si=FrZuOocy1U7Hwh50](https://youtu.be/ZTvRpX_3zAg?si=FrZuOocy1U7Hwh50)

[Daniel Alfaro]. (2023, August 27). *IPC2 Sección C - Clase Graphviz* [Video]. YouTube. <https://youtu.be/tLXmvsDxi-o?si=K0TpG43igS6ryiPm>

Welcome to. (s/f). Python.org. Recuperado el 7 de

septiembre de 2023, de

<https://www.python.org/>

Graphviz. (s/f). Graphviz. Recuperado el 7 de septiembre

de 2023, de <https://graphviz.org/>

Extensions, L. M. A. (s/f). Visual Studio Code - code

editing. Redefined. Visualstudio.com.

Recuperado el 7 de septiembre de 2023, de

<https://code.visualstudio.com/>

## Referencias bibliográficas

