
Proyecto 1 - SISTEMA DE SEÑALES DE AUDIO

202201444 – José David Góngora Olmedo

Resumen

Para el proyecto de El Centro de Investigación de la Facultad de Ingeniería se realizó un sistema para comprimir y trabajar con señales de audio, se tomaron en cuenta los siguientes parámetros: Frecuencia y Amplitud. 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). Para realizar el sistema se utilizó el lenguaje de programación Python, se utilizaron listas simplemente enlazadas para el almacenamiento y agrupación de los datos.

Para trabajar con los datos se realizaron tres matrices, una para los datos normales, otra para los datos en binario y una última que era la reducida, las cuales pueden trabajar con cualquier cantidad de señales que se obtengan del archivo de entrada.

Así mismo, se realizó un sistema para graficar las matrices, esto utilizando la librería Graphviz.

Palabras clave

Sonido, Frecuencia, Amplitud, Python, Matrices.

Abstract

For the project of El Centro de Investigación de la Facultad de Ingeniería a system was developed to compress and work with audio signals, the following parameters were taken into account: Frequency and Amplitude. Frequency is the number of cycles in one second and is measured in Hertz (Hz), while amplitude represents the height of the wave and refers to the intensity of the sound, the amplitude is measured in decibels (Db).

Python programming language was used to create the system, and simply linked lists were used for data storage and grouping.

To work with the data, three matrices were created, one for the normal data, another for the data in binary and the last one was the reduced one, which can work with any number of signals obtained from the input file.

Likewise, a system for graphing the matrices was created using the Graphviz library.

Keywords

Sound, Frequency, Amplitude, Python, Arrays.

Introducción

En la era actual de la información, el manejo efectivo de datos es esencial en diversos campos. Este ensayo explora un destacado programa desarrollado en Python, diseñado para interpretar archivos XML y transformarlos en matrices utilizando como estructura de datos una lista simplemente enlazada. Al aprovechar la versatilidad de Python, el programa demuestra su capacidad para analizar archivos XML, extraer la información sobre los datos, como las señales, tiempo y amplitud para así construir matrices que facilitan el análisis posterior. Además, su funcionalidad se extiende a la generación de gráficos informativos y archivos XML de salida con sus respectivos datos, grupos y tiempos. Este programa ilustra cómo la combinación de programación y procesamiento de datos puede proporcionar una solución eficaz para el objetivo del Centro de Investigaciones de la Facultad de Ingeniería.

Desarrollo del tema

En este programa fue diseñado para el manejo y agrupamiento de datos de sonido. El Centro de Investigaciones necesitaba que se crearan tres matrices para trabajar con los datos. La primera, una matriz con los datos normales sin ningún cambio. Era una por cada señal.

		AMPLITUD (db)			
TIEMPO (seg)		1	2	3	4
	1	2	3	0	4
	2	0	0	6	3
	3	3	4	0	2
	4	1	0	1	5
	5	0	0	3	1

Figure 1 Ejemplo de cómo sería la primera matriz

Después de hacer o construir la primera matriz, era necesario construir la matriz de patrones, la cual era la matriz con los mismos tiempos y mismas amplitudes, pero los valores de los datos debían de estar en binario (1,0), es decir, si el valor era mayor a 0, el valor se convertía a un 1 y si era igual a 0 se quedaban de la misma manera.

	AMPLITUD (db)				
TIEMPO (seg)		1	2	3	4
	1	1	1	0	1
	2	0	0	1	1
	3	1	1	0	1
	4	1	0	1	1
	5	0	0	1	1

Figure 2 Ejemplo de como quedaría la matriz de patrones

Al tener ya la matriz o lista de patrones era necesario verificar en que filas de tiempo el patrón es el mismo, esto para agrupar los patrones idénticos y formar una tercera matriz, la cual es la matriz reducida. Pero para terminar de obtener la matriz reducida se necesita sumar las tuplas de los grupos identificados en la matriz de patrones.

		AMPLITUD (db)			
TIEMPO (seg)		1	2	3	4
	Grupo 1 (tiempo 1 y 3)	5	7	0	6
	Grupo 2 (tiempo 2 y 5)	0	0	9	4
	Grupo 3 (tiempo 4)	1	0	1	5

Figure 3 Ejemplo de como quedaría la matriz reducida

Graficas

Para poder ver lo comentado anteriormente de manera gráfica, fue necesario utilizar la herramienta o librería Graphviz para representar las matrices como un grafo y que la lectura de los datos sea más fácil.

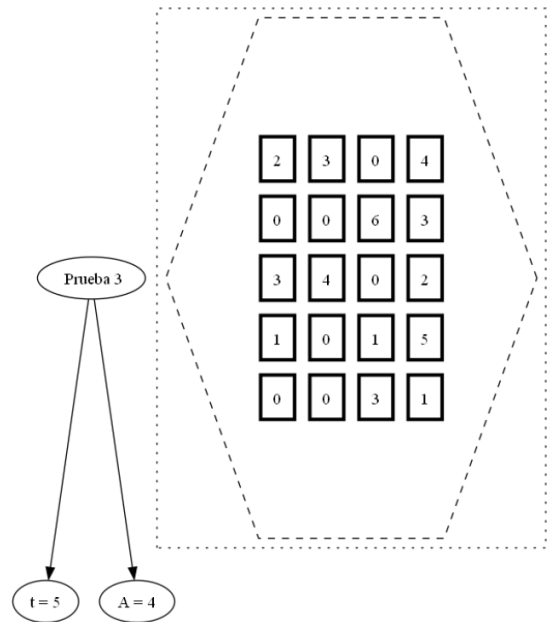


Figure 4 Ejemplo de grafica generada para la matriz original

Así mismo pasara con la gráfica de la matriz reducida.

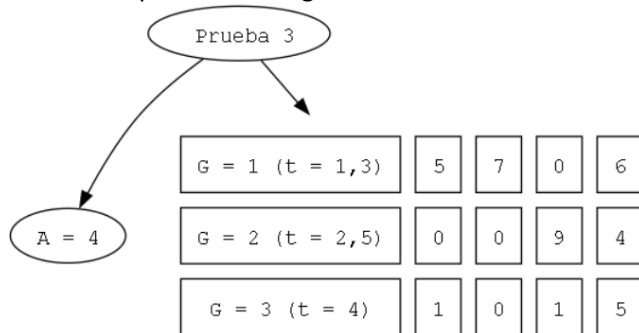


Figure 5 Ejemplo de grafica generada para la matriz reducida

Archivo de entrada

Para leer y procesar todos los datos, es necesario la lectura de un archivo con extensión y estructura XML. El cual tendrá una estructura definida para agregar señales y sus datos. Es necesario que el archivo tenga esa estructura para una correcta lectura de datos.

```
<?xml version="1.0"?>
<senales>
  <senal nombre="Prueba 3" t="5" A="4">
    <dato t="1" A="1">2</dato>
    <dato t="1" A="2">3</dato>
    <dato t="1" A="3">0</dato>
    <dato t="1" A="4">4</dato>
    <dato t="2" A="1">0</dato>
    <dato t="2" A="2">0</dato>
    <dato t="2" A="3">6</dato>
    <dato t="2" A="4">3</dato>
    <dato t="3" A="1">3</dato>
    <dato t="3" A="2">4</dato>
    <dato t="3" A="3">0</dato>
    <dato t="3" A="4">2</dato>
    <dato t="4" A="1">1</dato>
    <dato t="4" A="2">0</dato>
    <dato t="4" A="3">1</dato>
    <dato t="4" A="4">5</dato>
    <dato t="5" A="1">0</dato>
    <dato t="5" A="2">0</dato>
    <dato t="5" A="3">3</dato>
    <dato t="5" A="4">1</dato>
  </senal>
</senales>
```

Figure 6 Estructura para el archivo XML de entrada

Archivo de salida

Este al igual que el archivo de entrada, debe tener una extensión y estructura XML. Este debe de tener una estructura definida.

```
<senalesReducidas>
  <senal nombre="Prueba 3" A="4">
    <grupo g="1">
      <tiempos>1,3</tiempos>
      <datosGrupo>
        <dato A="1">5</dato>
        <dato A="2">7</dato>
        <dato A="3">0</dato>
        <dato A="4">6</dato>
      </datosGrupo>
    </grupo>
    <grupo g="2">
      <tiempos>2,5</tiempos>
      <datosGrupo>
        <dato A="1">0</dato>
        <dato A="2">0</dato>
        <dato A="3">9</dato>
        <dato A="4">4</dato>
      </datosGrupo>
    </grupo>
    <grupo g="3">
      <tiempos>4</tiempos>
      <datosGrupo>
        <dato A="1">1</dato>
        <dato A="2">0</dato>
        <dato A="3">1</dato>
        <dato A="4">5</dato>
      </datosGrupo>
    </grupo>
  </senal>
</senalesReducidas>
```

Figure 7 Estructura para el archivo XML de salida

Programa – Sistema de Señales de Audio

El programa cuenta con un menú inicial, el cual permite al usuario elegir una opción para realizar.

```
** PROYECTO 1 - INTRODUCCIÓN A LA PROGRAMACIÓN 2 **
-----
##### SISTEMA DE SEÑALES DE AUDIO #####
-----
| 1. CARGAR ARCHIVO |
| 2. PROCESAR ARCHIVO |
| 3. ESCRIBIR ARCHIVO SALIDA |
| 4. DATOS DEL ESTUDIANTE |
| 5. GENERAR GRÁFICA |
| 6. INICIALIZAR SISTEMA |
| 7. SALIR |
-----

## INGRESE UNA OPCION: ##
```

Figure 8 Ejemplo del menú

Opción 1 – Cargar archivo

Esta opción permite al usuario cargar un archivo XML antes de hacer la lectura de datos. En esta se le solicitará la ruta o nombre del archivo al usuario.

```
## INGRESE UNA OPCION: ##
1

INGRESE EL NOMBRE DEL ARCHIVO: prueba

## ARCHIVO CARGADO CON EXITO ##
```

Figure 9 Ejemplo de la opción del menú

Opción 2 – Procesar archivo

Esta opción le permitirá al usuario empezar a leer y procesar los datos del archivo cargado en la opción 1. Esta opción solo será válida si el usuario ya tiene un archivo cargado anteriormente.

```
## INGRESE UNA OPCION: ##
2

-> LEYENDO ARCHIVO...

-----
-> GENERANDO MATRIZ BINARIA DE SEÑAL: Prueba 3 ...
-> REALIZANDO SUMA DE TUPLAS DE SEÑAL: Prueba 3 ...
-> PROCESO TERMINADO
```

Figure 10 Ejemplo de opción 2 del menú

El programa mostrara mensajes de cómo va el proceso de los datos. Así mismo también mostrara errores en casos que los hayan.

```
=====
-> GENERANDO MATRIZ BINARIA DE SEÑAL: Prueba 2 ...
-> REALIZANDO SUMA DE TUPLAS DE SEÑAL: Prueba 2 ...
-> PROCESO TERMINADO

** DATOS NO VALIDOS, VALOR t = 4 o A = 1 PASAN EL RANGO **
** DATOS NO VALIDOS, VALOR t = 1 o A = 5 PASAN EL RANGO **
```

Figure 11 Ejemplo de errores al procesar el archivo

Opción 3 – Escribir archivo de salida

Esta opción le permitirá al usuario escribir el archivo XML de salida, con los datos ya procesados. Se le solicitará el nombre o ruta para guardar el archivo al usuario. Esta opción solo será válida si el usuario ya cargo y proceso el archivo.

```
## INGRESE UNA OPCION: ##
3

INGRESE EL NOMBRE DE PARA GUARDAR EL ARCHIVO: pruebaXML

-> PROCESO TERMINADO...

## SE CREO CORRECTAMENTE EL ARCHIVO pruebaXML##
```

Figure 12 Ejemplo de opción 3 del menú

Opción 4 – Datos del estudiante

Esta opción le mostrara al usuario los datos del programador (estudiante) que realizo el programa.

```
## INGRESE UNA OPCION: ##
4

##### DATOS DEL ESTUDIANTE #####
-----
-> José David Góngora Olmedo
-> 202201444
-> Introducción a la Programación y Computación 2 sección D
-> Ingeniería en Ciencias y Sistemas
-> 4to Semestre
```

Figure 13 Ejemplo de opción 4 del menú

Opción 5 – Generar grafica

Esta opción le mostrará las señales que hayan sido procedas, el usuario deberá de elegir la señal a graficar y después que tipo de grafica desea generar, Original o Reducida. Así mismo el usuario deberá de ingresar el nombre o ruta para guardar la gráfica generada.

```
## INGRESE UNA OPCION: ##
5

## SEÑALES DISPONIBLES ##
TOTAL DE SEÑALES: 5

Señal: Prueba 1
Señal: Prueba 4
Señal: Prueba 3
Señal: Prueba 2
Señal: Prueba 5

INGRESE EL NOMBRE DE LA SEÑAL A GRAFICAR: Prueba 1

## ¿QUÉ TIPO DE GRAFICA DESEA GRAFICAR? ##
## 1. ORIGINAL ##
## 2. REDUCIDA ##
INGRESE LA OPCIÓN A GRAFICAR: 1

## GENERANDO GRAFICA... ##

INGRESE EL NOMBRE DE LA GRAFICA: grafica1

## GRAFICA GENERADA ##
```

Figure 14 Ejemplo de opción 5 del menú

Opción 6 – Inicializar sistema

Esta opción le permitirá al usuario borrar todos los datos previamente cargados y procesados para que pueda ingresar y procesar otros archivos.

```
## INGRESE UNA OPCION: ##
6

-> LIMPIANDO DATOS...
-> PROCESO TERMINADO...
```

Figure 15 Ejemplo de opción 6 del menú

Opción 7 – Salida

Esta opción permite que el usuario pueda cerrar el programa y borrar todos los datos.

```
## INGRESE UNA OPCION: ##
7

## SALIENDO... ##
```

Figure 16 Ejemplo de opción 7 del menú

Las 7 opciones del menú, son todas las opciones que tiene el usuario para hacer con el programa, leyendo y procesando correctamente los datos ingresados.

Conclusiones

- El programa tiene la capacidad para leer y escribir datos en archivo XML.
- El programa es capaz de procesar datos y trabajar como una matriz dichos datos, pero utilizando listas simplemente enlazadas.
- El programa tiene la capacidad para mostrar de manera gráfica las matrices trabajadas con la implementación de Graphviz.

Referencias bibliográficas

Graphviz. (s. f.). Graphviz. <https://graphviz.org/>

Shaik, H. K. (2021). Comprender la implementación de listas vinculadas en Python. Geekflare.

<https://geekflare.com/es/python-linked-lists/>

xml.etree.ElementTree — la API XML de ElementTree. (s. f.). Python documentation.

<https://docs.python.org/es/3/library/xml.etree.elementtree.html>

Extensión: de cuatro a siete páginas como máximo

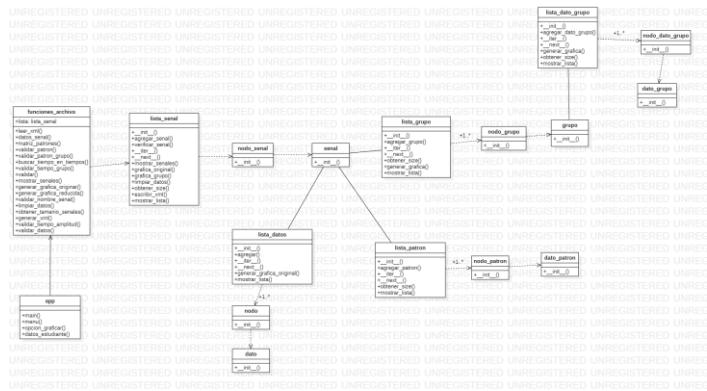


Figure 17 Diagrama de clases