
PROYECTO 1 - SEÑALES DE AUDIO

SMART_TEC

202200031 – Edison Mauricio García Rodríguez

Resumen

La realización de este proyecto tiene como objetivo diseñar un programa que procesa señales de audio representadas como matrices de frecuencia, genera matrices reducidas y crea gráficos a partir de estos datos.

El programa toma señales de audio en formato XML, las procesa para crear matrices reducidas y genera gráficos para representar visualmente los datos. También proporciona una interfaz de usuario con opciones para realizar diversas acciones relacionadas con el procesamiento y la visualización de las señales de audio.

Así mismo, se desarrollan temas ampliamente relacionados con los TDA'S (Tipo de Dato Abstracto) como lo es, la implementación de listas, pilas, listas enlazadas, entre otros. De igual manera se hace uso del software de Graphviz. Estructuras crucialmente importantes para el buen funcionamiento del software del programa de señales de audio.

Palabras clave

Software, gráficos, datos, estructuras y matrices.

Abstract

The realization of this project has the objective of designing a program that processes audio signals represented as frequency matrices, generates reduced matrices, and creates graphs from these data.

The program takes audio signals in XML format, processes them to create reduced matrices, and generates graphs to visually represent the data. It also provides a user interface with options to perform various things related to the processing and display of audio signals.

Likewise, topics widely related to ADD's (Abstract Data Type) are developed, such as the implementation of lists, stacks, linked lists, among others. In the same way, the Graphviz software is used. Structures of crucial importance for the correct functioning of the software of the audio signals program.

Keywords

Software, graphics, data, structures, and matrices.

Introducción

La importancia de analizar los TDA'S (Tipo de Datos Abstracto) es que tendremos una mejor percepción de cual es su estructura y como podemos implementarlo en nuestro código y que tareas específicas pueden hacer cada uno de ellos y como debemos de asignarle tareas. De igual manera poder observar como funciona internamente una lista nativa de Python y la transición que se genera al desarrollar este proyecto con listas propias, excluyendo el uso de las listas nativas que nos brinda el lenguaje de Programación Python.

Es importante mencionar, que existen varios tipos de datos abstractos, los cuales son listas, listas enlazadas, listas circulares, pilas, colas, entre otros. En el desarrollo de este proyecto se optó por utilizar listas y listas enlazadas para un mejor funcionamiento de las estructuras de los datos y la eficiencia que estas generan al momento de ejecutar el software del programa de las Señales de Audio.

Desarrollo del tema

En el siempre evolutivo mundo de la tecnología y la programación, un paradigma se ha destacado en las últimas décadas, transformando la forma en que los desarrolladores diseñan y crean software: la programación orientada a objetos (POO). Este enfoque ha tenido un impacto profundo en la eficiencia, la reutilización de código y la capacidad de mantenimiento de las aplicaciones. Al igual que todos los conceptos que engloba, como las estructuras y manejos de datos, entre otros. Por ende, en el desarrollo de este proyecto se desarrollaron temas como los TDA'S y POO que se explican en el transcurso de este informe.

El **TDA** (Tipo de Dato Abstracto) en programación es un concepto fundamental que se refiere a una abstracción que define un conjunto de datos y las operaciones que pueden realizarse en ellos, sin revelar los detalles internos de cómo se implementa esa estructura de datos. Así mismo, se presentan temas que se relaciona con un TDA.

- ✓ **Abstracción:** El TDA permite a los programadores crear estructuras de datos abstractas que encapsulan datos y operaciones relacionadas. Esto oculta los detalles de implementación y facilita el uso de la estructura sin preocuparse por su funcionamiento interno.
- ✓ **Datos y operaciones:** Un TDA consta de dos partes principales. Primero, define un conjunto de datos que representa la información que se desea almacenar y manipular. Segundo, especifica las operaciones que se pueden realizar en esos datos, como agregar, eliminar, buscar o modificar.
- ✓ **Ejemplos de TDA'S:** Ejemplos comunes de TDA'S incluyen listas, pilas, colas, árboles y grafos. Cada uno de estos TDA'S tiene su propio conjunto de operaciones y reglas específicas.
- ✓ **Encapsulación:** El TDA promueve el principio de encapsulación, que significa que los datos y las operaciones se agrupan en una sola entidad. Esto evita el acceso directo a los

datos internos y garantiza que las operaciones se realicen de manera controlada y segura.

- ✓ Implementación: Aunque el TDA define la interfaz y las operaciones, no especifica cómo deben implementarse. Los programadores pueden elegir la estructura de datos adecuada (arreglos, listas enlazadas, árboles, etc.) y desarrollar los algoritmos necesarios para realizar estas operaciones de manera eficiente.
- ✓ Modularidad: El uso de TDA'S fomenta la emisión de diseño de programas, ya que las estructuras de datos y sus operaciones se pueden reutilizar en diferentes partes del código, lo que facilita el mantenimiento y la escalabilidad.

De igual manera se realiza la implementación del software de **Graphviz**, es un software de código abierto ampliamente utilizado para visualizar estructuras de gráficos y redes. Fue desarrollado originalmente por AT&T Labs y es ampliamente utilizada en la industria y la comunidad de desarrollo para generar diagramas y representaciones visuales de datos estructurados, como grafos, diagramas de flujo, árboles jerárquicos y otros tipos de estructuras relacionales.

Las características más destacadas de Graphviz incluyen:

- ✓ Dibujo Automático: Graphviz tiene la capacidad de generar automáticamente diagramas a partir de descripciones de datos

estructurados. Los usuarios proporcionan una descripción textual de la estructura del grafo utilizando lenguajes como DOT (Graphviz DOT), y la herramienta se encarga de generar la representación visual del grafo.

- ✓ Soporte para Diferentes Tipos de Gráficos: Graphviz es versátil y puede utilizarse para dibujar varios tipos de gráficos, como grafos dirigidos y no dirigidos, árboles, diagramas de flujo, organigramas, entre otros.
- ✓ Personalización: Aunque Graphviz genera automáticamente la mayoría de los aspectos de los diagramas, ofrece opciones de personalización para ajustar la apariencia de los gráficos, como colores, tamaños de nodos y bordes, y estilos de línea.
- ✓ Salida en Diversos Formatos: Puede generar diagramas en varios formatos de salida, incluyendo imágenes rasterizadas (por ejemplo, PNG, JPEG) y formatos vectoriales (por ejemplo, SVG, PDF).
- ✓ Integración con Diversos Lenguajes de Programación: Graphviz tiene bibliotecas y enlaces disponibles para varios lenguajes de programación, lo que facilita su integración en aplicaciones y flujos de trabajo de desarrollo.
- ✓ Amplia Comunidad: Debido a su popularidad, Graphviz cuenta con una comunidad activa de usuarios y desarrolladores que comparten recursos, ejemplos y soluciones en línea.

Graphviz es especialmente útil en campos como la visualización de datos, la representación de redes, la

generación de diagramas de software y la documentación técnica. Su capacidad para generar diagramas automáticamente a partir de datos estructurados ahorra tiempo y ayuda a comprender y comunicar información de manera más efectiva.

El **proyecto** se centra en el desarrollo de un programa informático que procesa señales de audio expresadas como matrices de frecuencia. El programa realiza las siguientes tareas:

- ✓ **Entrada de Datos:** Acepta múltiples señales de audio en formato XML, cada una con su nombre, dimensiones de tiempo y amplitud, y valores de frecuencia en función de la matriz de frecuencia.
- ✓ **Procesamiento:** El programa analiza las señales para identificar patrones de frecuencia y agrupa las señales con patrones idénticos. Luego, genera matrices reducidas sumando las frecuencias en cada grupo.
- ✓ **Salida de Datos:** Genera archivos XML de salida que contienen información sobre las matrices reducidas y la estructura de grupos.
- ✓ **Generación de Gráficos:** Utiliza Graphviz para crear gráficos que visualizan las señales originales y las matrices reducidas, mostrando detalles como el nombre de la señal, dimensiones (tiempo y amplitud), y las frecuencias.
- ✓ **Interfaz de Usuario:** Ofrece un menú interactivo en la consola que permite a los usuarios cargar archivos, procesar datos, escribir archivos de salida, mostrar información del estudiante y generar gráficos.

- ✓ **Validaciones:** Realiza validaciones en los archivos de entrada para garantizar la coherencia de los datos, como la existencia de señales con el mismo nombre y el cumplimiento de las reglas de tiempo y amplitud.

En resumen, el proyecto se enfoca en el procesamiento y análisis de señales de audio, la generación de matrices reducidas y la creación de representaciones gráficas de las señales y sus resultados. Proporciona una interfaz de usuario amigable para realizar estas tareas de manera efectiva.

Conclusiones

Los Tipos de Datos Abstractos (TDA) desempeñan un papel esencial en la programación al permitir la encapsulación de datos y operaciones en una abstracción coherente. Esto facilita la administración de datos, la reutilización de código, y contribuye a la claridad en el diseño de programas. Los TDA ofrecen interfaces definidas claramente, ocultando los detalles internos, lo que simplifica el trabajo en equipo y reduce la complejidad. En la programación orientada a objetos, son fundamentales para modelar tipos de datos abstractos mediante clases y objetos, fomentando la encapsulación y la abstracción.

Graphviz es una herramienta versátil y potente para visualizar datos y crear gráficos. Ofrece muchas opciones de personalización y puede utilizarse en una variedad de aplicaciones, desde representar datos complejos hasta crear diagramas conceptuales y de flujo. Su capacidad para generar gráficos automáticamente a partir de datos brutos lo hace valioso para analizar y comunicar información

eficazmente. Sin embargo, su aprendizaje puede ser algo complejo debido a su conjunto de comandos y sintaxis específica.

En conclusión, el proyecto se centra en el desarrollo de un programa informático que permite el procesamiento y análisis de señales de audio expresadas en matrices de frecuencia. A través de una interfaz de usuario interactiva, el programa acepta múltiples señales de audio en formato XML, las procesa para identificar patrones de frecuencia y agruparlas, generando matrices reducidas y representaciones visuales de las señales y sus resultados utilizando Graphviz. Además, realiza validaciones para garantizar la coherencia de los datos de entrada. En resumen, este proyecto proporciona una herramienta eficaz para el análisis y visualización de señales de audio.

Referencias bibliográficas

Severance, C. R. (02-Abr-2020 Traducción al Español completa de Python 3.0). Python para todos: Explorando la información con Python 3.

Apéndices

```
-----
                          Proyecto 1 - IPC2 A
-----
# - - - - - MENU PRINCIPAL - - - - - #
1. Cargar archivo
2. Procesar archivo
3. Escribir archivo de salida
4. Mostrar datos del Estudiante
5. Generar Gráfica
6. Inicializar Sistema
7. Salir
Ingresa una opción: 1
```

Imagen 1: Ventana Principal
Fuente: Elaboración Propia

```
Ingresa una opción: 1

Ingresa la ruta donde se encuentra el archivo (.xml)
Ruta: |
```

Imagen 2: Interfaz – Opción 1
Fuente: Elaboración Propia

```
Ingresa una opción: 2

Para utilizar esta función, primero debes cargar un archivo (.xml) o procesarlo.

Presione ENTER para continuar...|
```

Imagen 3: Interfaz – Opción 2
Fuente: Elaboración Propia

```
Ingresa una opción: 3

Ingresa una ruta específica para guardar el archivo (.xml): |
```

Imagen 4: Interfaz – Opción 3
Fuente: Elaboración Propia

```
Ingresa una opción: 4

-----
Datos del Estudiante:
Edison Mauricio García Rodríguez
202200031
Introducción a la Programación y Computación 2 Sección A
Ingeniería en Ciencias y Sistemas
4to. Semestre
-----
```

Imagen 5: Interfaz – Opción 4
Fuente: Elaboración Propia

```
----- Genera una gráfica -----
1. Matriz Regular
2. Matriz Reducida
Ingresa una opción: |
```

Imagen 6: Interfaz – Opción 5
Fuente: Elaboración Propia

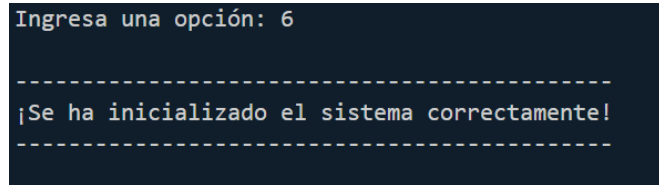


Imagen 7: Interfaz – Opción 6
Fuente: Elaboración Propia

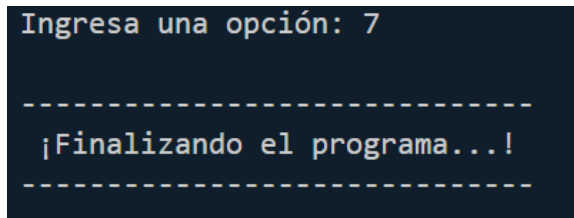


Imagen 8: Interfaz – Opción 7
Fuente: Elaboración Propia

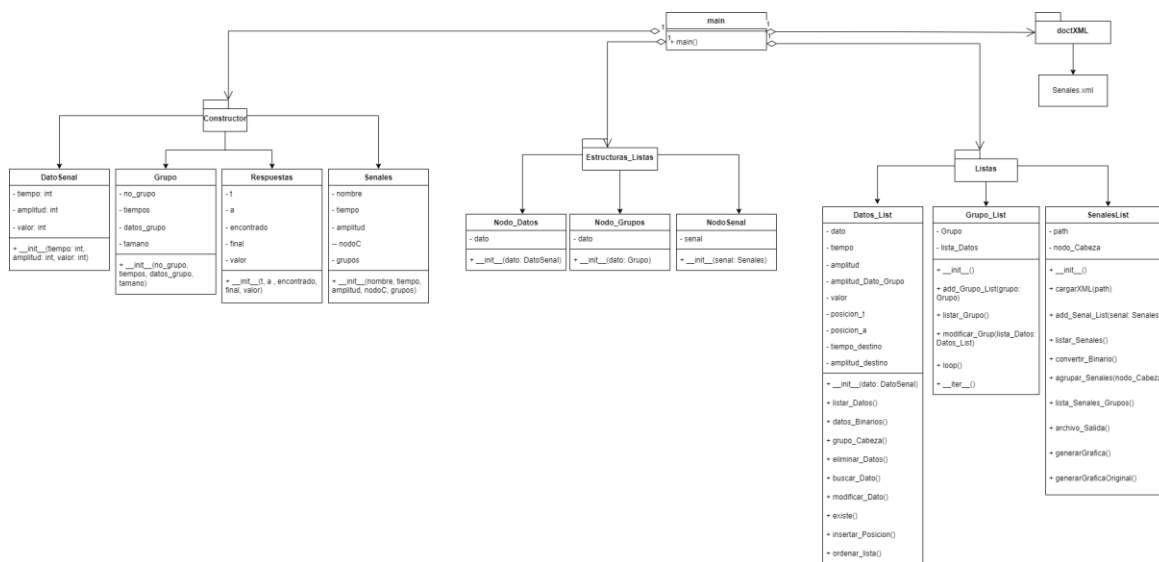


Imagen 9: Diagrama de Clases
Fuente: Elaboración Propia