
PROCESAMIENTO DE SEÑALES

2022201524 – Carlos Manuel Lima y Lima

Resumen

El informe presenta un proyecto implementado en Python que se centra en el procesamiento de señales a partir de archivos XML. Mediante el uso de Programación Orientada a Objetos, se diseñan y gestionan objetos señal obtenidos de la lectura del XML, dichos objetos contienen atributos que son de gran importancia para el objetivo del proyecto, como los datos señal que contiene las “matrices” que son manipuladas en el sistema. Estos objetos se almacenan eficientemente en una lista simple enlazada, lo que permite una manipulación efectiva y un acceso rápido. Para visualizar la estructura y relaciones de los objetos señal y todos sus atributos, se emplea la herramienta Graphviz para generar gráficas visuales. Este enfoque facilita la comprensión y análisis de las relaciones entre los datos. En conjunto, el proyecto demuestra cómo la combinación de POO, listas enlazadas y herramientas de visualización puede ser aplicada exitosamente para el procesamiento y representación de información a partir de archivos XML, ofreciendo una solución integral y eficaz.

Palabras Clave

1. Programación
2. Objetos
3. Señales
4. Abstracción
5. Sistema

Abstract

The report presents a Python-implemented project that focuses on processing signals from XML files. By utilizing Object-Oriented Programming, signal objects obtained from XML parsing are designed and managed. These objects contain attributes crucial to the project's objective, such as signal data stored in matrices manipulated within the system. These objects are efficiently stored in a singly linked list, enabling effective manipulation and swift access. To visualize the structure and relationships of the signal objects along with their attributes, the Graphviz tool is employed to generate visual graphs. This approach facilitates understanding and analysis of data relationships. Collectively, the project showcases how the combination of OOP, linked lists, and visualization tools can be successfully applied for processing and representing information from XML files, providing a comprehensive and effective solution.

Keywords

1. Programming
2. Objects
3. Signals
4. Abstraction
5. System

Introducción

En el mundo de la programación, la orientación a objetos se erige como un enfoque fundamental para modelar sistemas complejos de manera eficiente y organizada. Al adoptar esta metodología, se da vida a la noción de tipos de datos abstractos, cuya implementación abstrae la complejidad interna y facilita la interacción con componentes externos. Dentro de este contexto, las listas simples enlazadas emergen como estructuras cruciales, permitiendo la gestión dinámica de información de manera eficaz.

Este informe aborda la sinergia entre la programación orientada a objetos, tipos de datos abstractos y las listas enlazadas, trazando un recorrido por un proyecto desarrollado en Python. Este proyecto se enriquece con el uso de archivos XML, que actúan como vehículos de datos estructurados, y la herramienta Graphviz, que posibilita la visualización gráfica de relaciones complejas. El uso de estas tecnologías en la resolución de desafíos de programación y diseño es importante.

1. Programación Orientada a Objetos (POO)

La programación orientada a objetos representa un paradigma de programación en el cual la estructuración del software se fundamenta en torno a objetos o datos, en contraposición al uso de funciones y lógica. Su enfoque radica en los objetos que los programadores requieren manipular, en vez de centrarse en la lógica necesaria para llevar a cabo dicha manipulación. Un objeto se conceptualiza como un conjunto de atributos y comportamientos distintivos. (Universidad Europea, 2022)

Por lo tanto, la característica central de este enfoque reside en su capacidad para manejar objetos, cada uno de los cuales posee un tipo o clase específica. Estas clases pueden heredar atributos de clases superiores, o superclases. Esta particularidad impulsa su implementación en proyectos extensos y complejos que necesitan actualizaciones periódicas.

1.1. Clase

Una clase en la programación orientada a objetos funciona como una plantilla que establece las propiedades y acciones de una entidad. Se puede visualizar como un patrón a partir del cual es posible crear instancias con características y comportamientos similares. Básicamente, una clase define tanto los atributos como los comportamientos que caracterizan a una entidad.

1.2. Abstracción

La abstracción es cuando el usuario interactúa solo con los atributos y métodos seleccionados de un objeto, utilizando herramientas simplificadas de alto nivel para acceder a un objeto complejo.

En la programación orientada a objetos, los programas suelen ser muy grandes y los objetos se comunican mucho entre sí. El concepto de abstracción facilita el mantenimiento de un código de gran tamaño, donde a lo largo del tiempo pueden surgir diferentes cambios.

Así, la abstracción se basa en usar cosas simples para representar la complejidad. Los objetos y las clases representan código subyacente, ocultando los detalles complejos al usuario. Por consiguiente, supone una extensión de la encapsulación. Siguiendo con el ejemplo del coche, no es necesario que conozcas todos los detalles de cómo funciona el motor para poder conducirlo. (Martínez, 2020)

2. Python

Python es un lenguaje de programación de propósito general que es compatible con una variedad de arquitecturas de sistemas y tiene aplicaciones versátiles en diversos campos, desde la creación de sitios web hasta el desarrollo de soluciones de aprendizaje automático. Su capacidad para adaptarse a diferentes áreas y su facilidad de uso lo convierten en uno de los lenguajes de programación más extendidos.

Las personas pueden usar y distribuir el código fuente del mismo de forma gratuita, incluso con fines comerciales. Hoy en día, cualquier persona con una computadora y una fuerte voluntad puede aprender a programar en este lenguaje. (Weisheim, 2023)

Adicionalmente a su facilidad de aprendizaje, su popularidad también deriva de su gran versatilidad. Este lenguaje encuentra aplicaciones en una amplia gama de campos, que incluyen la ciencia de datos, el desarrollo web y el aprendizaje automático.

Además, su naturaleza multiplataforma permite que se ejecute en diversos sistemas operativos como Windows, Linux y macOS.

2.1. POO en Python

Cuando se habla de programación orientada a objetos, se define a un conjunto de conceptos y patrones que utilizan para resolver problemas con objetos.

Un objeto en Python es una colección única de atributos y métodos. Puedes pensar en los objetos como cosas reales que te rodean. (Díaz, 2023)

La POO permite crear software seguro y fiable. Muchos marcos y bibliotecas de Python utilizan este paradigma para construir su código base. Algunos ejemplos son Django, Kivy, pandas, NumPy y TensorFlow. (Díaz, 2023)

Una de las mejores cosas de Python es que permite a los programadores crear una gran variedad de software, desde un programa interfaz de comandos, hasta una compleja aplicación web.

2.2. Implementación de POO en Python

Python es un lenguaje multiparadigma: soporta la programación imperativa y funcional, pero también la programación orientada a objetos. En Python todo es un objeto. Cuando se crea una variable y se asigna un valor entero, ese valor es un objeto, lo mismo con las listas, conjuntos, tuplas, cadena de caracteres, etc.

Implementar POO en Python implica utilizar las características y sintaxis que ofrece dicho lenguaje.

- Utiliza la palabra reservada "class" para crear definiciones de clases, incluyendo atributos y métodos en su interior.
- Incorpora la herencia al establecer la clase base entre paréntesis durante la creación de una nueva clase.
- Aprovecha la característica de herencia múltiple y el concepto de polimorfismo de forma orgánica, dado que Python posibilita estas prácticas.
- Haz uso de propiedades y decoradores para llevar a cabo la encapsulación, permitiendo un control más preciso del acceso a los atributos de una clase.

3. Graphviz

Graphviz constituye un software de código abierto de distribución gratuita orientado a la creación de representaciones gráficas, las cuales permiten visualizar datos estructurales en forma de diagramas. Su aplicabilidad es amplia y abarca campos tan diversos como el análisis de redes, la bioinformática, la ingeniería de software, las bases de datos, el diseño de páginas web y el aprendizaje automático. Además, proporciona interfaces gráficas diseñadas para diferentes dominios. (Takeyas, 2012)

El conjunto de herramientas Graphviz engloba una diversidad de programas diseñados para la creación de diagramas. Entre estas opciones, el lenguaje "dot" sobresale como la herramienta principal destinada a elaborar representaciones de estructuras jerárquicas o de capas dirigidas. Este lenguaje puede ser ejecutado como un programa desde la línea de comandos, a través de un servicio de visualización web o mediante una interfaz gráfica compatible. (Takeyas, 2012)

4. Tipos de Datos Abstractos (TDA's)

Es un modelo que define valores y las operaciones que se pueden realizar sobre ellos. Y se denomina abstracto ya que la intención es que quien lo utiliza, no necesita conocer los detalles de la representación interna o bien el cómo están implementadas las operaciones.

Es por esto una práctica que nos provee un grado de abstracción que permite desacoplar al código que usa un TDA de aquel código que lo implementa.

4.1. Estructuras de datos

Las estructuras de datos son aquellas que nos permiten, como desarrolladores, organizar la información de manera eficiente, y en definitiva diseñar la solución correcta para un determinado problema. Una estructura de datos nos permite trabajar en un alto nivel de abstracción almacenando información para luego acceder a ella, modificarla y manipularla.

4.2. Estructura de Datos Lineales

Las estructuras de datos lineales son aquellas en las que los elementos ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y un único predecesor, es decir, sus elementos están ubicados uno al lado del otro relacionados en forma lineal. (SoyHenry, 2022)

Las estructuras lineales de datos se caracterizan porque sus elementos están en secuencia, relacionados en forma lineal, uno luego del otro. Cada elemento de la estructura puede estar conformado por uno o varios subelementos que pueden pertenecer a cualquier tipo de dato.

Existen tres tipos de estructuras de datos lineales:

- Listas enlazadas
- Pilas
- Colas

4.3. Listas Simples Enlazadas

La estructura de datos conocida como lista enlazada se presenta como un Tipo de Dato Abstracto que proporciona un método organizado de almacenamiento de información. En este sentido, guarda similitudes con los vectores, aunque se diferencia en su carácter dinámico. A diferencia de los vectores, la lista enlazada no requiere conocimiento previo de la cantidad de elementos a contener.

En el contexto de una lista enlazada simple, cada elemento señala al siguiente, a excepción del último que carece de sucesor y, en consecuencia, su enlace adopta el valor nulo. Los elementos adoptan la forma de registros que incorporan tanto los datos a almacenar como un enlace que conecta con el siguiente componente. En ocasiones, se les atribuye a estos elementos el nombre de "nodos" dentro de la lista.

En las estructuras de datos, las listas enlazadas se construyen con elementos que están ubicados en una secuencia. Aquí, cada elemento se conecta con el siguiente a través de un enlace que contiene la posición del siguiente elemento. De este modo, teniendo la referencia del principio de la lista podemos acceder a todos los elementos de la misma (SoyHenry, 2022)

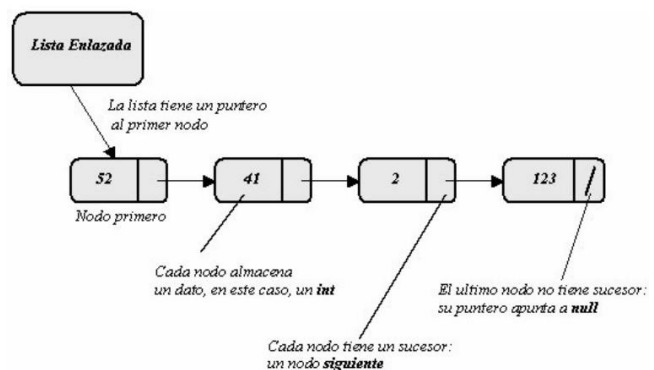


Figura 1. Estructura de una Lista Simple

Fuente: Facultad de Estadística e Informática, Universidad Veracruzana

5. Archivos XML

XML consiste en un lenguaje de marcado creado por el W3C con la finalidad de definir una sintaxis para la codificación de documentos, que tanto los usuarios como las propias máquinas en sí puedan ser capaces de leer.

Para ello, lo hace mediante la utilización de una serie de etiquetas que definen la estructura que posee el documento en cuestión, además de cómo debe ser transportado y almacenado. (Pérez, 2020)

Podemos establecer una comparación con otro lenguaje de marcado con el cual es probable que estemos más familiarizados: el lenguaje de marcado de hipertexto (HTML), ampliamente empleado en la codificación de páginas web. HTML emplea una serie de símbolos de marcado predefinidos que delinean el formato del contenido de una página web.

No obstante, una distinción fundamental entre XML y HTML radica en su extensibilidad. Mientras HTML opera con un conjunto predeterminado de símbolos de marcado, XML se distingue por su naturaleza extensible.

6. Señal Periódica

Las señales periódicas exhiben un patrón de repetición que les confiere su naturaleza distintiva. Tras un intervalo definido, los valores previos se reiteran en una secuencia constante. A esta recurrencia se le denomina ciclo de onda, cuya duración es denominada período y se mide en segundos.

La frecuencia de una señal, expresada en Hertz (Hz), refleja cuántos ciclos pueden desarrollarse en un segundo. De esta forma, la frecuencia captura la tasa de repetición de los patrones en la señal.

La amplitud de la señal, por su parte, se refiere al valor máximo que puede alcanzar. Representa la distancia entre el punto más alejado de la onda y su punto de equilibrio central, caracterizando la intensidad de los cambios en la señal.

Conclusiones

1. El proyecto demuestra una integración exitosa de múltiples tecnologías en el entorno de programación Python. Mediante la utilización de la Programación Orientada a Objetos (POO), se logró diseñar una estructura modular y reutilizable para la manipulación de objetos señal provenientes de un archivo XML. La implementación de Listas Simples Enlazadas como estructuras de datos para almacenar y gestionar estos objetos permitió un almacenamiento eficiente y un acceso rápido a los mismos.

2. La incorporación de la librería Graphviz añadió una dimensión visual a este proyecto, permitiendo la generación automática de gráficas que representan las relaciones entre los objetos señal almacenados en la lista enlazada. Esta característica no solo ofrece una representación visual de la estructura de datos, sino que también facilita la identificación de patrones y conexiones entre los objetos señal.

3. La utilización de XML como formato de entrada para la creación de objetos señal brindó flexibilidad y escalabilidad al proyecto. La arquitectura basada en POO y Listas Simples Enlazadas proporciona un sólido fundamento para incorporar nuevas funcionalidades sin comprometer la integridad del sistema.

Referencias Bibliográficas

Díaz, D. (15 de Junio de 2023). *Guía para Principiantes de la Programación Orientada a Objetos (POO) en Python*. Obtenido de Kinsta: <https://kinsta.com/es/blog/programacion-orientada-objetos-python/>

Martínez, M. (02 de Noviembre de 2020). *¿Qué es la Programación Orientada a Objetos?* Obtenido de Profile: <https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

Pérez, C. (18 de Diciembre de 2020). *Qué es y cómo abrir un archivo XML*. Obtenido de Muyinteresante: <https://www.muyinteresante.es/tecnologia/23571.html>

SoyHenry. (10 de Febrero de 2022). *Estructura de datos en programación*. Obtenido de SoyHenry: <https://blog.soyhenry.com/que-es-una-estructura-de-datos-en-programacion/>

Takeyas, B. L. (2012). *Estructura de Datos Orientados a Objetos: Pseudocódigo y Aplicaciones en C#. Net*. México: Alfa Omega.

Universidad Europea. (24 de Agosto de 2022). *Programación orientada a objetos*. Obtenido de Universidad Europea: <https://universidadeuropea.com/blog/programacion-orientada-objetos/>

Weisheim, R. (26 de Junio de 2023). *Qué es Python: conoce uno de los lenguajes de programación más populares*. Obtenido de Hostinger: <https://www.hostinger.es/tutoriales/que-es-python>

Anexos

1. Tabla de Actividades

Semana	Actividad
Primera Semana	Realización del menú principal, los submenús, y manipulación de la ruta inicial.
Segunda Semana	Leer archivo xml, almacenar las señales y los datos en la lista, cálculo de matriz reducida y almacenamiento.
Tercera Semana	Archivo salida, inicializar el sistema, y graficas.

Tabla 1. Cronograma de Actividades
Fuente: Elaboración Propia

2. Diagrama de Clases

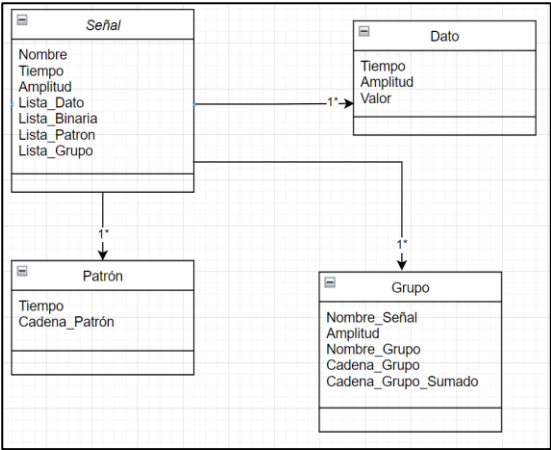


Figura 1. Diagrama de Clases
Fuente: Elaboración Propia

3. Diagrama de Listas

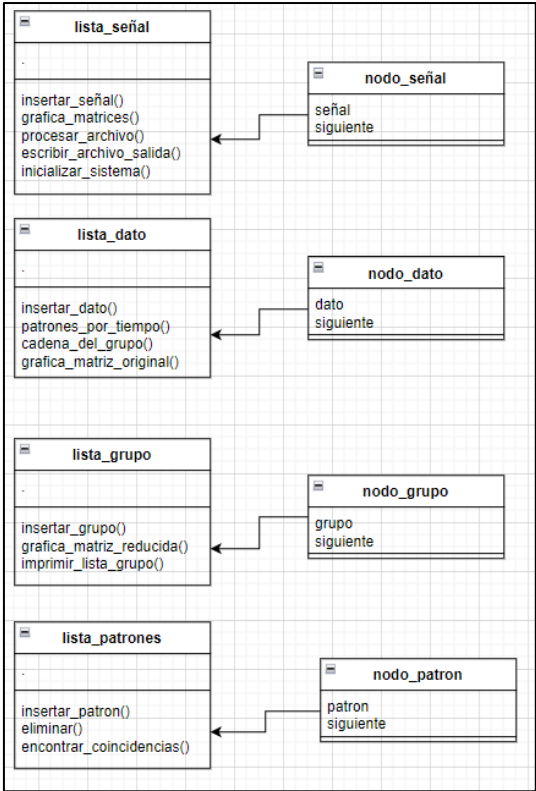


Figura 2. Diagrama de Listas
Fuente: Elaboración Propia