

---

## TDA, COMPRIMIR SEÑALES DE AUDIO

---

202212209 – Juan José Almengor Tizol

### Resumen

Este programa se dedica a la implementación de un sistema de encriptación de mensajes en Python, y busca garantizar la privacidad de las comunicaciones mediante el uso de técnicas avanzadas de programación. Para lograrlo, emplearemos estructuras de datos como listas enlazadas con nodos, permitiéndonos almacenar y procesar información vital procedente de archivos XML, esencial en el proceso de encriptación y desencriptación.

La visualización efectiva del flujo de ejecución del código se logrará gracias a la herramienta Graphviz. Además, se aplicarán estructuras de programación secuenciales, cíclicas y condicionales, lo que optimizará el rendimiento y la eficiencia del programa.

Una parte clave de nuestra estrategia consiste en adoptar los principios de la programación orientada a objetos, lo que facilitará la regularización y organización del código de manera eficaz.

En resumen, este programa integra habilidades avanzadas de programación con técnicas de encriptación para crear una solución sólida que garantiza la seguridad de las comunicaciones mediante la manipulación de datos XML .

### Palabras clave

- Listas enlazadas
- Nodos
- POO
- Xml
- Encriptado

### Abstract

*This program is dedicated to implementing a message encryption system in Python, aiming to ensure communication privacy through advanced programming techniques. To achieve this, we will employ data structures like linked lists with nodes, enabling us to store and process vital information from XML files, crucial in the encryption and decryption process.*

*Effective visualization of code execution flow will be accomplished using the Graphviz tool. Additionally, sequential, cyclic, and conditional programming structures will be applied, enhancing program performance and efficiency*

*A key component of our strategy is embracing the principles of object-oriented programming, which will facilitate effective code modularization and organization.*

*In summary, this program integrates advanced programming skills with encryption techniques to create a robust solution that guarantees communication security by manipulating XML data*

### Keywords

- linked lists
- Nodes
- POO
- Xml
- Encryption

## Introducción

Este proyecto surge de una colaboración entre el Ministerio de Defensa de Guatemala y la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. El objetivo es desarrollar una tecnología que permita enviar mensajes encriptados que no puedan ser interceptados ni descifrados por personas no autorizadas.

El sistema se compone de dos partes: un emisor de mensajes y un receptor de mensajes. La idea es usar drones para enviar señales de luz a diferentes alturas, representando letras del alfabeto. El receptor puede detectar las alturas y determinar qué dron envió la señal, decodificar las letras y así obtener el mensaje.

Cada día, el Ministerio de Defensa proporcionará una tabla de instrucciones, ya que un mensaje puede ser codificado de varias maneras debido a las alturas y drones disponibles. El desafío es desarrollar un software que gestione eficientemente este sistema de drones, con la capacidad de manejar hasta 200 drones y alturas de 1 a 100 metros.

Este proyecto tiene un propósito importante: garantizar la seguridad de las comunicaciones en Guatemala.

## Desarrollo del tema

Empezamos a buscar alguna manera para lograr realizar el proyecto, como sabemos que nuestros drones y las instrucciones serán en un archivo xml, tenemos analizarlo, entonces empezamos con nuestro proyecto creando un archivo que se puede llamar LeerXml que nos va a ayudar a leer el primer xml, bueno ya después vamos agregando que vamos guardando, en este caso tiene varias cosas como dron, sistema del dron, las alturas, el mensaje, las instrucciones y el contenido, entonces nos enfocaremos en eso.

En mi proyecto, he desarrollado un sistema con un menú integral que abarca diversas funcionalidades clave. Este menú, diseñado para facilitar la gestión del sistema de drones, se compone de varias opciones esenciales que desempeñan un papel fundamental en el funcionamiento general del proyecto.

La primera opción del menú es la "Inicialización del Sistema". Aquí, se establecen las bases del sistema, preparándolo para su funcionamiento. Esto implica la creación de una estructura inicial y la configuración de parámetros cruciales que guiarán el comportamiento de los drones durante la transmisión de mensajes encriptados, bueno si usamos este botón todos los drones guardados se eliminan, se eliminan los mensajes y el xml que nos va a generar, para empezar de nuevo un nuevo mensaje.



Figura 1. Menú principal.

Fuente: elaboración propia.

La segunda opción, denominada "Cargar Archivo de Entrada", es un componente vital de nuestro proyecto. Hemos incorporado una función que utiliza una herramienta llamada "Leerxml" para analizar archivos de entrada. Esto es esencial ya que estos archivos contienen información crítica para el proceso de encriptación y desencriptación de mensajes. La capacidad de cargar y procesar estos archivos de manera eficiente es un aspecto central de nuestro proyecto.

Los elementos XML pueden contener las siguientes características: 1. Texto 2. Atributos 3. Otros. Además, hemos implementado una función que permite crear archivos XML. Esto brinda a los usuarios la flexibilidad de generar archivos personalizados que contengan instrucciones específicas para la transmisión de mensajes. Esto es especialmente útil para adaptar el sistema a diversas situaciones y requisitos.

#### a. Generar archivo salida

Utilizamos este botón para que nos genere un xml, este archivo contiene el tiempo óptimo para poder decifrar el mensaje, ya que, y claro también cuenta con el mensaje decifrado, y todo estructurado para que nosotros como personas se nos sea fácil entender cual fue el mensaje generado.

```
<?xml version='1.0' encoding='UTF-8'?>
<respuesta>
  <listaMensajes>
    <mensaje nombre="msg">
      <sistemaDrones>
        <tiempoOptimo>6</tiempoOptimo>
        <mensajeRecibido>IPC2</mensajeRecibido>
      </sistemaDrones>
    </mensaje>
    <mensaje nombre="msg2">
      <sistemaDrones>
        <tiempoOptimo>5</tiempoOptimo>
        <mensajeRecibido>IPC2</mensajeRecibido>
      </sistemaDrones>
    </mensaje>
  </listaMensajes>
</respuesta>
```

Figura 2. xml salida .

Fuente: elaboración propia.

Una vez que hayamos procesado el archivo XML, el siguiente paso consiste en poner el contenido en nuestras listas simples. Como mencioné anteriormente, vamos a organizar nuestros datos utilizando el tiempo como una dimensión de filas y la amplitud como una dimensión de columnas. Esta estructura nos permitirá crear nuestros mensajes

Este archivo XML tiene una estructura simple. Comienza con una línea que indica que es un archivo XML con codificación UTF-8. Luego, hay un elemento principal llamado <respuesta>.

Dentro de <respuesta>, encuentras <listaMensajes>, que es como una lista de mensajes. En este caso, hay dos mensajes dentro de esa lista: uno llamado "msg" y otro llamado "msg2".

Cada mensaje tiene información adicional dentro de <sistemaDrones>. Esto incluye el "tiempoOptimo", que parece ser el tiempo óptimo para algo, establecido en 6 para el primer mensaje y 5 para el segundo. Además, hay un elemento llamado "mensajeRecibido" que contiene el texto "IPC2" en ambos mensajes.

#### B. Listas para leer xml

Primero, está la listaDrones, que es como mi lista de drones. Cada dron tiene un montón de detalles como su nombre, y utilizo la clase Dron para organizar y almacenar toda esa información.

Luego, está la listaSistemasDrones, donde guardo información sobre los sistemas de drones. Aquí, me interesa saber cosas como el tiempo óptimo y el mensaje recibido. La clase SistemasDrones me ayuda a estructurar esos datos.

Para seguir, utilizo la listaAlturas. En esta lista, apunto las alturas de los drones, y esto es fundamental para calcular el tiempo y los mensajes encriptados de manera correcta. La clase Alturas me ayuda a mantener esto en orden.

En cuanto al contenido de los mensajes y las instrucciones que necesito transmitir, utilizo la listaContenido. Aquí, me aseguro de que todo esté bien organizado y listo para ser enviado. La clase Contenido es clave para esto.

Y, finalmente, tengo las listaInstrucciones y listaGenerarInstrucciones. Estas listas son para almacenar las instrucciones específicas relacionadas con la transmisión de mensajes. Las clases Instrucciones y GenerarInstrucciones me permiten darle la estructura adecuada a toda esta información.

En resumen, estas listas y clases son como las herramientas que utilizo para mantener todo en orden y hacer que mi proyecto de drones funcione sin problemas

### C. Graphviz para los mensajes eh instrucciones

SD1			
Dron	DronX	DronY	DronZ
	A	2	B
	I	C	C
	D	P	E
	F	G	H
	I	L	O
	J	M	P
	K	N	Q

SDF				
Dron	DronW	DronX	DronY	DronZ
	I	P	C	2

Figura 3. Sistema instrucciones .

Fuente: elaboración propia.

Con esta herramienta, podemos crear representaciones visuales extensas en este caso estamos graficando las instrucciones para el sistema de drones, que puede cambiar dependiendo el mensaje, drones o sistemas, en este ejemplo tenemos uno con varias alturas y otro con solo 1 altura.

también mas adelante generamos otra grafica que nos ayudara a saber cuando se mueven los drones y el mensaje que dio, pero basicamente para eso usamos graphviz, solo lo tenemos que importar, o usar un comando para que nos cree el dot.

ahora miraremos como ver los drones ya gregarlos:

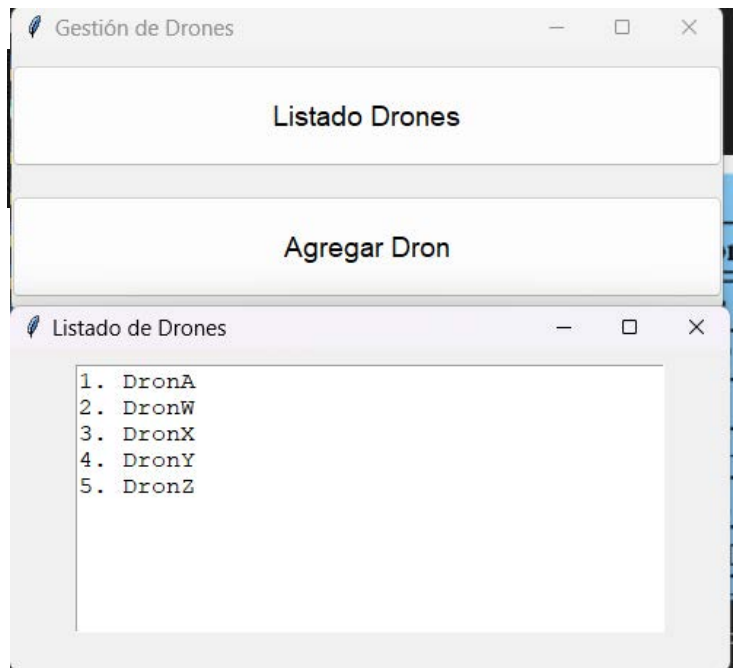


Figura 4. lista drones y agregar

Fuente: elaboración propia, 2023.

### D. Ayuda

Mostramos los datos del estudiante que realizo el proyecto y el link para este ensayo.

### E. Inicializar el sistema

El programa estará en ejecución mientras lo hayamos iniciado, pero que pasa cuando queremos volver a empezar o cargar nuevos datos a nuestro programa, para eso es que tenemos la opción de inicializar sistema, esta opción, no solo nos limpia la memoria de nuestro programa, sino que nos borra lo que teníamos antes para crear nuevas gráficas y nuevos archivos xml de salida, esto lo logramos fácilmente colocando las listas como vacías, y también los mensajes.

## **F. Ventajas y desventajas de las listas enlazadas**

Para este proyecto no se utilizan listas nativas de Python, todas las listas son creadas con nodos, entonces conocemos como es que se utilizan y errores o inconvenientes que esto contiene, claro no significa que si usamos otro tipo de listas no vamos a tener errores, mas bien es de conocer cómo funcionan.

Una de las principales ventajas es que podemos acceder a datos de nuestra de manera fácil, y recorrer datos usando ciclos que lo podemos implementar en cualquier lado, también podemos eliminar datos de nuestra lista, y no son tan complicadas de implementar, y en este proyecto las usamos y nos sirvieron para guardar los datos.

Hoy en día hay muchos métodos para hacer listas, todo depende del objetivo que tengamos en un proyecto

Ordenar una pila o una cola suele ser más sencillo que ordenar una lista enlazada, y esto se debe a cómo están estructurados los elementos en estas estructuras de datos.

## **Conclusiones**

- Con el uso de las listas enlazadas en nuestro proyecto, hemos logrado una forma eficiente de almacenar información de archivos XML que se asemeja a una matriz, permitiéndonos acceder a datos como si fueran filas y columnas. Esto demuestra la versatilidad de las listas enlazadas, que pueden ser una herramienta valiosa en una amplia variedad de proyectos.

- Los archivos xml son una buena manera de mostrar datos, ya que, aunque no este familiarizado alguien con este archivo, no le constara comprender, o en caso sea realizar, y entonces los programas también usando listas podemos utilizarlos para diferentes métodos.

- Además, los archivos XML se destacan como una excelente forma de representar datos. A pesar de que alguien pueda no estar familiarizado con este tipo de archivo, su estructura legible facilita su comprensión y, en caso necesario, su manipulación. Esta capacidad de trabajar con archivos XML, junto con las listas enlazadas, amplía las posibilidades de nuestros programas y los hace adaptables a diversos métodos y aplicaciones en futuros proyectos.

## **Referencias bibliográficas**

- Summerfield, M. (2010). Programming in Python 3: A Complete Introduction to the Python Language. Addison-Wesley Professional.
- Snoeyink, J., Lu, P., Su, K., & Wang, L. (2012). Frontiers in algorithmics and algorithmic aspects in information and management: Joint International Conference, FAW-AAIM 2012, Beijing, China, May 14-16, 2012, Proceedings. Springer.
- Sumathi, S., Rajappa, S., Kumar, L. A., & Paneerselvam, S. (2022). Machine learning for decision sciences with case studies in Python. CRC Press.

Anexos

- Diagrama de clases sobre la aplicación y el flujo que tiene.

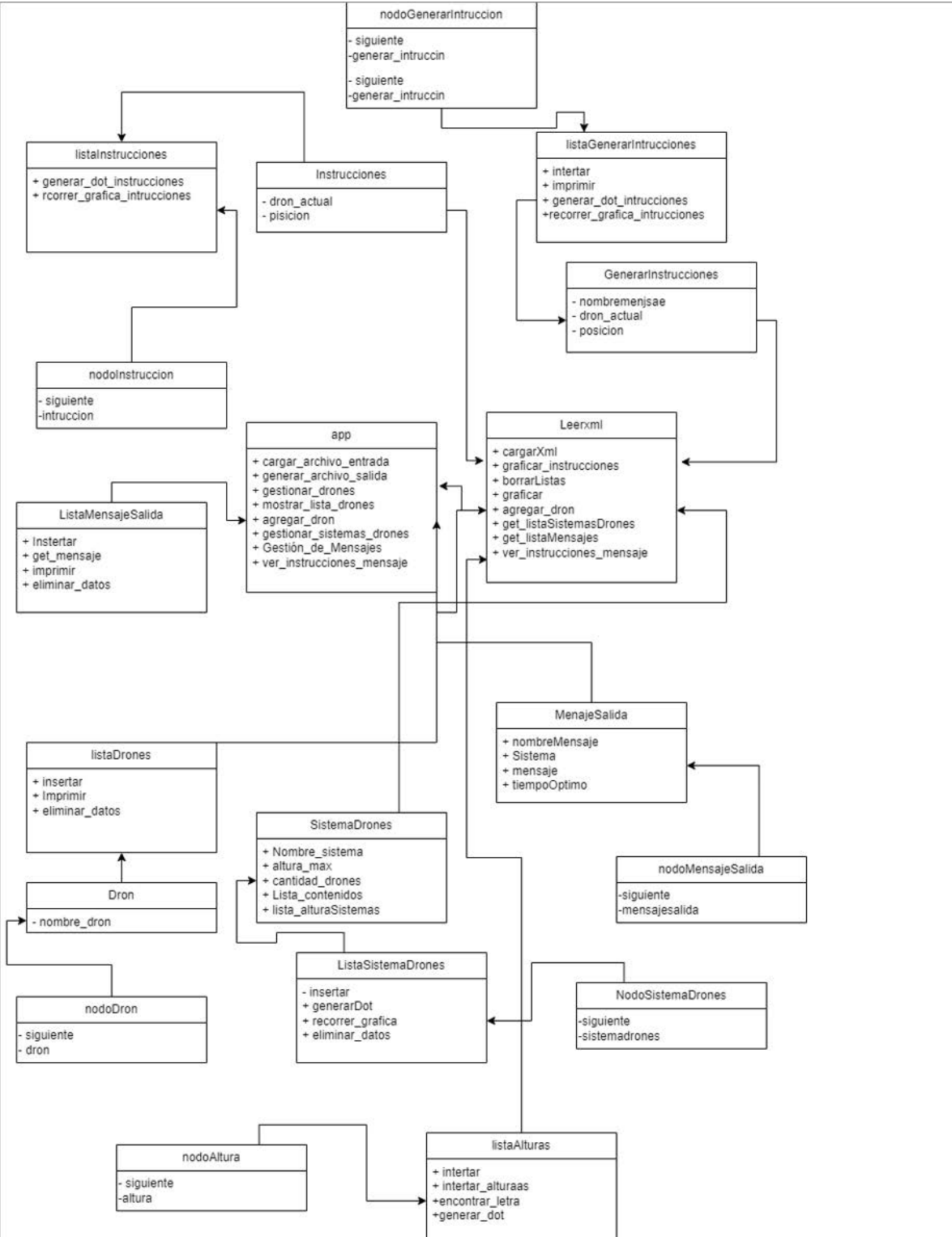


Figura 5. Diagrama de clases.

Fuente: elaboración propia, 2023.