
Sistema de Drones – Mensajes Encriptados

2022201524 – Carlos Manuel Lima y Lima

Resumen

El presente proyecto desarrollado en Python se enfoca en procesar archivos XML que contienen información sobre sistemas de drones, sus características y mensajes cifrados. Utiliza listas doblemente enlazadas para almacenar estos datos de manera organizada. El objetivo principal es descifrar mensajes utilizando instrucciones específicas con la ayuda de sistemas de drones. Cada sistema de drones tiene nombres, alturas máximas y una lista de drones, donde cada dron tiene un nombre, altura y símbolo asociado. La herramienta Graphviz se utiliza para crear gráficas que representan visualmente los sistemas de drones, mostrando los drones con sus alturas y símbolos correspondientes. También se generan gráficas para las listas de instrucciones necesarias para descifrar los mensajes de manera eficiente. Además este proyecto utiliza listas doblemente enlazadas para organizar información sobre sistemas de drones y mensajes cifrados, y utiliza Graphviz para visualizar gráficamente los sistemas de drones y las instrucciones necesarias para descifrar mensajes secretos.

Palabras Clave

1. Sistema
2. Drones
3. Mensaje
4. Descifrado
5. Listas Enlazadas

Abstract

The present project, developed in Python, focuses on processing XML files containing information about drone systems, their characteristics, and encrypted messages. It uses doubly linked lists to store this data in an organized manner. The main goal is to decipher messages using specific instructions with the assistance of drone systems. Each drone system has names, maximum altitudes, and a list of drones, where each drone has a name, altitude, and associated symbol. The Graphviz tool is utilized to create graphs that visually represent the drone systems, displaying the drones with their respective altitudes and symbols. Graphs are also generated for the instruction lists necessary to decipher messages efficiently. Furthermore, this project employs doubly linked lists to organize information about drone systems and encrypted messages, and it utilizes Graphviz to graphically visualize the drone systems and the instructions required to decipher secret messages.

Keywords

1. System
2. Drones
3. Message
4. Decryption
5. Linked Lists

Introducción

Python y la programación orientada a objetos son elementos esenciales en este proyecto. Python, por su legibilidad y versatilidad, se revela como la elección ideal para manejar archivos XML y estructurar datos complejos, como sistemas de drones y mensajes cifrados.

La programación orientada a objetos ofrece una metodología de desarrollo que encaja perfectamente con la naturaleza modular y jerárquica de los componentes del proyecto. Esto permite representar eficazmente sistemas de drones como objetos con propiedades y comportamientos definidos, simplificando la manipulación de datos y la interacción entre elementos del sistema.

La combinación de Python y la programación orientada a objetos potencia la capacidad del proyecto para procesar archivos XML, gestionar sistemas de drones y descifrar mensajes cifrados. La utilización de Graphviz incluye un valor agregado ya que se pueden obtener informes gráficos de la información más relevante del proyecto.

1. 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.

1.1 Estructuras De Programación Secuencial

La programación secuencial es una metodología que se basa en la ejecución de acciones o instrucciones de manera secuencial. En este enfoque, se realizan una serie de operaciones desde el inicio hasta el final, incluyendo tareas de asignación y cálculo, entre otras.

Una característica destacada de la programación secuencial es que las tareas se ejecutan de manera que la salida de una actividad se convierte en la entrada de la siguiente. Este proceso se repite de manera sucesiva hasta que se completa el proceso en el sistema. (KeepCoding Team, 2023)

1.2 Estructuras De Programación Cíclicas

Los ciclos, también conocidos como bucles o estructuras de control repetitivas, desempeñan un papel fundamental en el proceso de desarrollo de programas. En Python, un ciclo permite repetir una o varias instrucciones tantas veces como sea necesario. Por ejemplo, si deseamos mostrar los números del uno al cien, sería poco práctico escribir cien líneas de código para mostrar un número en cada una de ellas. En cambio, un ciclo resulta muy útil en esta situación. Los ciclos nos permiten llevar a cabo tareas repetitivas de manera eficiente y prácticamente automática, lo que acelera el proceso. (González, 2020)

Algunos de los bucles que Python ofrece son los siguientes, cada uno tiene distintas características pero su finalidad es la misma, iterar y recorrer:

- While
- For

1.2.1 Ciclo While

Los ciclos while son una estructura cíclica que permite ejecutar una o varias líneas de código de manera repetitiva sin la necesidad de tener un valor inicial y, en ocasiones, sin siquiera conocer cuándo se alcanzará el valor final deseado. (González, 2020)

Con el ciclo while, se establece la condición bajo la cual se detendrá el ciclo, pero no se tiene certeza de cuántas iteraciones serán necesarias ni cuánto tiempo tomará alcanzar ese punto. Por esta razón, se les denomina ciclos indeterminados. (González, 2020)

1.2.1 Ciclo For

Los ciclos for son una estructura de control cíclica que permite ejecutar una o varias líneas de código de manera iterativa o repetitiva, con un control y conocimiento precisos sobre las iteraciones. En el ciclo for de Python, se requiere especificar un valor de inicio y un valor final, y opcionalmente se puede definir el tamaño del "paso" entre cada iteración del ciclo. (González, 2020)

En resumen, un ciclo for en Python es una estructura iterativa que ejecuta un segmento de código un número específico de veces, con los valores de inicio, paso y final previamente definidos. Esto proporciona un alto nivel de control sobre el proceso iterativo. (González, 2020)

1.3 Estructuras De Programación Condicionales

Las sentencias condicionales if, else y elif en Python son herramientas fundamentales que permiten a un programa ejecutar código específico cuando se cumplen ciertas condiciones. Este sistema de toma de decisiones desempeña un papel esencial en la programación y resulta especialmente relevante en campos como la Ciencia de Datos. En lenguaje Python, la toma de decisiones se basa en sentencias condicionales. (DataScientist, 2023)

En programación de software, la mayoría de los grandes proyectos requieren controlar el flujo de ejecución del programa y ejecutar determinados conjuntos de sentencias solo cuando se cumple una condición. Si no se cumple una condición, debe ejecutarse otro conjunto de sentencias. Las sentencias condicionales también se denominan sentencias de decisión. Se utilizan para ejecutar un bloque específico de código si las condiciones dadas son verdaderas o falsas. (DataScientist, 2023)

2. 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)

3. 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 delinear el formato del contenido de una página web.

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)

4.3. Listas Doblemente Enlazadas

Una lista doblemente enlazada es una estructura de datos que consta de nodos, donde cada nodo contiene un elemento de datos y dos enlaces, uno al nodo anterior y otro al nodo siguiente en la secuencia. A diferencia de las listas enlazadas simples, las listas doblemente enlazadas permiten la navegación en ambas direcciones: hacia adelante y hacia atrás.

4.3.1 Características Principales de las Listas Doblemente Enlazadas

- **Navegación Bidireccional:** Cada nodo en una lista doblemente enlazada tiene dos enlaces, lo que facilita la navegación tanto hacia adelante como hacia atrás en la lista.
- **Inserción y Eliminación Eficiente:** La inserción y eliminación de elementos en una lista doblemente enlazada es más eficiente que en una lista enlazada simple, ya que no es necesario recorrer toda la lista para llegar al nodo deseado.
- **Uso de Memoria:** Las listas doblemente enlazadas consumen más memoria que las listas enlazadas simples debido a la necesidad de mantener dos enlaces por nodo.

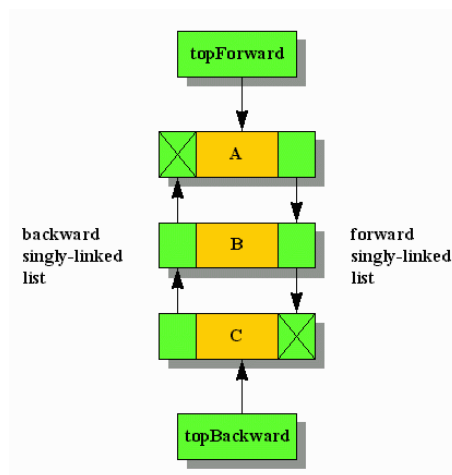


Figura 1. Estructura de una Lista Doblemente Enlazada

5. 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)

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)

2.2. 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.

Conclusiones

1. El uso de listas doblemente enlazadas para representar la información de sistemas de drones, drones, mensajes e instrucciones indica una estructura de datos compleja y eficiente para gestionar y acceder a la información. Esta elección de estructura de datos permite una fácil manipulación de la información relacionada y facilita la implementación de las operaciones necesarias para descifrar los mensajes.

2. El proyecto combina elementos de programación, procesamiento de datos, representación gráfica y criptografía para lograr su objetivo principal de descifrar mensajes. La utilización de Graphviz para visualizar los sistemas de drones y las instrucciones agrega un componente gráfico al proyecto, lo que sugiere una combinación de habilidades técnicas y creativas en su desarrollo.

3. El proyecto demuestra un enfoque interesante para la criptografía mediante el uso de sistemas de drones y alturas para descifrar mensajes. La generación de gráficas de instrucciones óptimas también indica un esfuerzo por optimizar la decodificación de mensajes. Este enfoque podría ser aplicado en contextos más amplios de seguridad y criptografía, lo que subraya la utilidad de la programación en la resolución de problemas complejos.

4. La realización de este proyecto en Python ha demostrado la capacidad de aplicar la programación orientada a objetos para abordar problemas complejos. La lectura y escritura de archivos XML permitieron la manipulación de datos estructurados de manera efectiva. La utilización de Graphviz agregó un componente visual valioso al generar informes gráficos que facilitan la comprensión de los sistemas de drones y las instrucciones de descifrado. Además, el uso de listas doblemente enlazadas

proporcionó una estructura de datos eficiente para almacenar y organizar la información extraída de los archivos XML. En conjunto, estos elementos han contribuido al éxito de este proyecto al abordar de manera efectiva la manipulación y visualización de datos relacionados con sistemas de drones y mensajes cifrados.

Referencias Bibliográficas

- DataScientist. (7 de Febrero de 2023). *Python If, Else: todo sobre las sentencias condicionales*.
Obtenido de DataScientist:
<https://datascientest.com/es/python-if-else>
- González, J. D. (2020). *Los ciclos o bucles en Python. Uso declaración y sintaxis de ciclos en Python*.
Obtenido de ProgramarYa:
<https://www.programarya.com/Cursos/Python/Ciclos>
- KeepCoding Team. (16 de Mayo de 2023). *¿Qué es la estructura secuencial en programación?*
Obtenido de KeepCoding:
<https://keepcoding.io/blog/que-es-estructura-secuencial-programacion/>
- 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/>
- 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 de la interfaz gráfica. Además realización de objetos, nodos y listas enlazadas.
Segunda Semana	Lectura del archivo XML, insertar en las listas enlazadas, realización de la gráfica sistema
Tercera Semana	Descifrar mensaje, obtener tiempo optimo e instrucciones por dron.
Cuarta Semana	Llenado de lista mensaje recibido, escritura de XML y grafica mensaje recibido.

Tabla 1. Cronograma de Actividades
Fuente: Elaboración Propia

2. Diagrama de Clases

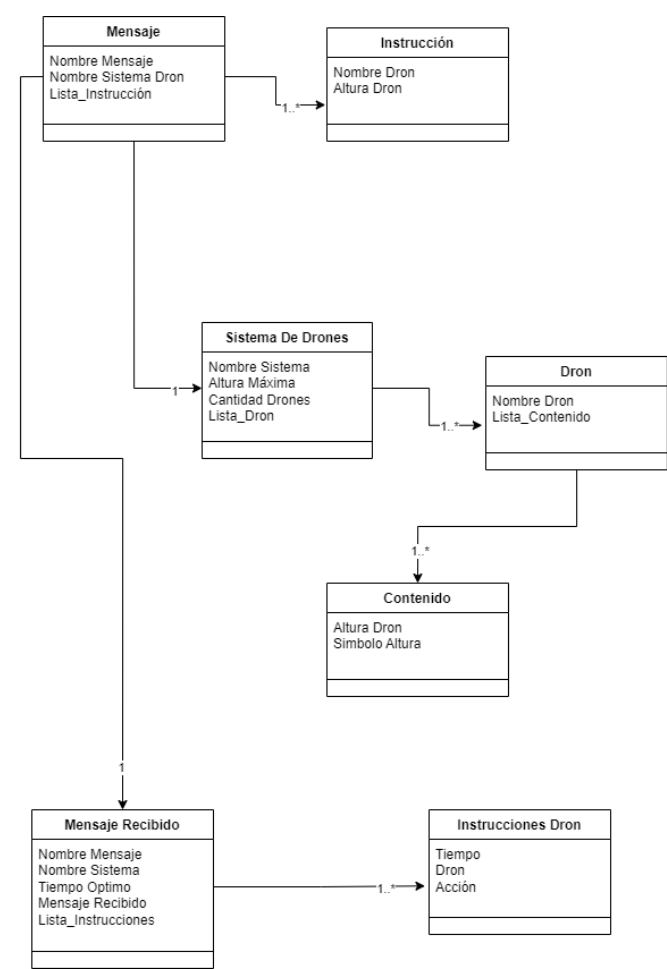


Figura 2. Diagrama de Clases
Fuente: Elaboración Propia