
PROYECTO 1 IPC2 “DECIFRAMIENTO DE MENSAJES”

201901844 – Oswaldo Antonio Choc Cuteres

Resumen

El ministerio de la Defensa de Guatemala ha creado un acuerdo de colaboración con la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. El objetivo de este acuerdo es crear una nueva tecnología que permita el envío de mensajes encriptados de tal forma que no puedan ser interceptados y descifrados por personal o instituciones no autorizadas.

Para lograr este fin, se ha creado un sistema con 2 componentes. Un componente emisor del mensaje y un componente receptor del mensaje.

La Facultad de Ingeniería propone un diseño que funcionará con “n” drones, los cuales podrán subir una cantidad de metros y emitir una luz led de alta emisión de tal forma que dependiendo del dron que emita la luz y la altura a la que la emita, representará una letra del alfabeto, de esta forma, el componente receptor podrá medir alturas y determinar qué dron emitió el haz de luz, decodificar cada letra y finalmente obtener el mensaje que se desea transmitir.

Palabras clave

Dron, emitir luz, mensaje, decodificar.

Abstract

The Ministry of Defense of Guatemala has created a collaboration agreement with the Faculty of Engineering of the University of San Carlos of Guatemala. The objective of this agreement is to create a new technology that allows the sending of encrypted messages in such a way that they cannot be intercepted and decrypted by unauthorized personnel or institutions.

To achieve this end, a system with 2 components has been created. A message sending component and a message receiving component.

The Faculty of Engineering proposes a design that will work with “n” drones, which will be able to climb a number of meters and emit a high-emission LED light in such a way that depending on the drone that emits the light and the height at which it emits it, will represent a letter of the alphabet, in this way, the receiving component will be able to measure heights and determine which drone emitted the light beam, decode each letter and finally obtain the message that you want to transmit.

Keywords

Drone, emit light, message, decode.

Introducción

Los drones tienen misiones como identificar las posiciones enemigas, marcar el lugar que debe ser bombardeado y bombardear o corregir la posición en caso de que el bombardeo falle.

Para el diseño de este programa, con el objetivo principal de poder utilizar como referencia la posición de cada dron al momento de interceptar los mensajes es importante tener en cuenta que el dron puede desplazarse hacia arriba, hacia abajo y esperar en esa posición dependiendo de la instrucción que tenga, toda esta información deberá de venir previamente en un archivo de entrada en formato XML, para poder procesar cada instrucción y descifrar los mensajes que hay en el mismo.

Este archivo de entrada es muy importante para el funcionamiento del programa, ya que si no hay instrucciones no se podrá ejecutar ningún proceso.

Desarrollo del programa

Para desarrollar el programa se utilizó Visual Studio Code como IDE, utilizando Python como lenguaje único de programación y utilizando Tkinter para un interfaz gráfica que facilita la interacción con el usuario.

Al desarrollar el programa para poder descifrar los mensajes ingresados, se empezó con el análisis del archivo de entrada, tomando en cuenta que es un archivo en formato XML, es importante separar cada dato ingresado por sus etiquetas correspondiente.

Un archivo XML tiene la característica de tener el concepto de herencia, iniciando con una

jerarquía de padre e hijo, así se va estructurando cada una de sus etiquetas como los atributos de dron y altura de cada sistema que se ingresa en el archivo. Las etiquetas que se ingresan son las siguientes:

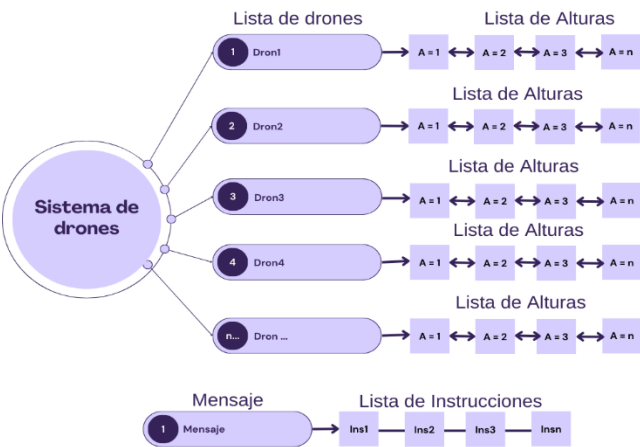
Archivo de entrada:

- I. **ListaDrones:** Esta etiqueta viene un inventario de los drones que se podrán utilizar en los sistemas.
- II. **listaSistemadrones:** esta etiqueta será la que indica los sistemas que se podrán utilizar al momento de descifrar los mensajes
- III. **SistemaDron** puede tener los siguientes atributos:
 - **nombre:** este contendrá el identificador del sistema de drones (se deberá validar la existencia de matrices con el mismo nombre, para mantener la consistencia de los datos).
 - **alturamaxima:** representa la altura máxima al cual el dron podrá llegar. Regla: $A > 0$ y $A \leq 100$.
 - **CantidadDrones:** representa la cantidad de drones que estará usando el sistema. Regla: $Drones > 0$ y $Drones \leq 200$.
 - **Contenido:** Esta etiqueta contendrá todo la información del sistema de drones desde la información de la letra en cada altura hasta que drones se va a utilizar del inventario previamente ingresado, no se podrá utilizar un dron que no esté en dicho inventario.
 - **Dron:** será el nombre del dron al cual se le almacenará la información.
 - **Altura:** Dada la altura se asignará un valor alfanúmero para poder saber a que letra o numero hacer referencia el dron al momento de estar en esa posición

Como se logra observar en la figura 1, la lógica que se utilizó fue una lista de listas básicamente, donde cada Sistema tendría una serie de listas de drones según el dato ingresado en el archivo y así mismo cada dron tendría una lista de alturas.

Si quisiéramos ver más detalladamente la funcionalidad del programa, veamos el siguiente diagrama de clases.

Partiendo de conceptos de Programación Orientada a Objetos (POO) y además de trabajar con un Tipo de Dato Abstracto (TDA) se implementaron listas enlazadas y listas doblemente enlazadas con las cuales se almacena toda la información ingresada desde el archivo de entrada, creando básicamente un conjunto de listas y utilizando colas para poder acceder a los datos de una mejor forma.

[illegible]

En la figura 2, se logra observar el diagrama de clases del programa que está diseñado para poder procesar cualquier mensaje encriptado, primeramente, la clase interfaz contiene la información de todos los datos y así mismo puede realizar cualquier operación que se solicite en la interfaz gráfica puede acceder a cualquier clase como leer, calcular, graficar y crear archivo, dependiendo de la opción que desee el usuario, así mismo la clase leer permite crear la lista de sistema de

drones con su respectiva lista de drones y lista de alturas por cada dron.

Inventario

El inventario es una lista tipo cola que nos permite poder almacenar todos los drones que se ingresen desde la interfaz o desde el archivo de entrada.

Lista de Sistemas

La lista de señales como su nombre lo indica estará almacenando todas los sistemas ingresados en el documento.

Un sistema tendrá los siguientes atributos, nombre, altura máxima y cantidad de drones. Partiendo de este dato de cada sistema se puede crear la lista de drones ya que se conoce cuantos drones hay en el sistema, de igual forma para cada dron se puede crear de una vez cada lista de alturas ya que se conoce la altura máxima que tendrá cada dron. Todo esto se trabajo con listas enlazadas simples lo cual permitía obtener la información de cada nodo y el nodo siguiente hasta recorrer cada una de las listas que tuviéramos, únicamente la lista de alturas es una lista doblemente enlazada que permite moverse de mejor forma con un nodo anterior o un nodo siguiente.

Para la lista de drones, cada dron tiene un atributo de altura máxima, nombre, lista de instrucciones, lista de alturas, lista de movimientos lo cual nos servirá para el desarrollo de cada función, como condiciones el programa solo puede recibir hasta un máximo de 200 drones y 100 alturas, si en todo caso viene mas serán ignoradas.

Para la lista de alturas, cada altura tiene únicamente como atributo un valor alfanumérico que nos indicará cual es el mensaje encriptado.

Altura (mts)	DronX	DronY	DronZ
1	A	2	B
2	I	C	C
3	D	P	E
4	F	G	H
5	I	L	O
6	J	M	P
7	K	N	Q

Figura 3, Sistema ejemplo

Fuente: Enunciado proyecto

Lista de mensajes

La lista de mensajes aguarda los mensajes que se van a descifrar juntamente con las instrucciones que va a realizar cada dron del sistema.

Instrucción
DronX,2
DronY,3
DronZ,2
DronY,1

Figura 4, Instrucción de un mensaje

Fuente: Enunciado proyecto

Procesar mensaje

La parte más importante de este programa es poder descifrar el mensaje con el mejor tiempo posible, para poder realizar este proceso la lógica para poder realizarlo fue la siguiente:

Sabiendo que existe una lista de instrucciones que se deben de ejecutar paso a paso y en orden para que le mensaje sea detectado correctamente, se implemento un ciclo para que se recorrieran cada una de esas instrucciones, posteriormente se tuvo que

haber asignado a cada dron la lista de instrucciones que debería de seguir, por tanto hasta que no se cumpliera la instrucción cada dron ya tiene su debida instrucción que seguir y puede irse moviendo en cada tiempo para no perder y optimizar de mejor forma la ejecución, si la instrucción que tiene cada dron es mayor a la posición en la que se encuentre debe de subir, así mismo si la posición es menor que la instrucción debe de bajar y si ya llego a esa posición de la instrucción que tiene pero no es su turno debe de esperar hasta que se vaya realizando en orden cada una de las ejecuciones, hasta llegar a un resultado como el siguiente.

Tiempo (seg)	DronX	DronY	DronZ
1	Subir	Subir	Subir
2	Subir	Subir	Subir
3	Emitir luz	Subir	Esperar
4	Esperar	Emitir luz	Esperar
5	Esperar	bajar	Emitir luz
6	Esperar	bajar	Esperar
7	Esperar	Emitir luz	Esperar

Figura 5, Tabla de resultados mensaje
Fuente: Enunciado proyecto

Construcción del grafico

Para representar de una forma más fácil de ver al usuario sobre los sistemas recibidos y los mensajes procesados, se realiza una método para poder graficarlo por medio de la herramienta de Graphviz, esta debe mostrar la gráfica de los sistemas y mensajes recibidos.

En el diagrama únicamente se muestra la información principal de cada sistema, como lo

es su nombre, cantidad de drones y alturas. Lo principal del diagrama es mostrar los valores en cada dron y altura que se ingresaron. Como se muestra a continuación

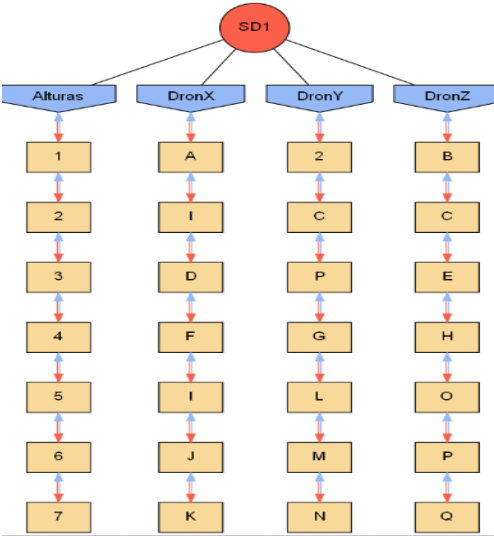


Figura 6, Gráfica sistema
Fuente: Elaboración propia

Igualmente, cuando se realice la gráfica de los mensajes procesados.

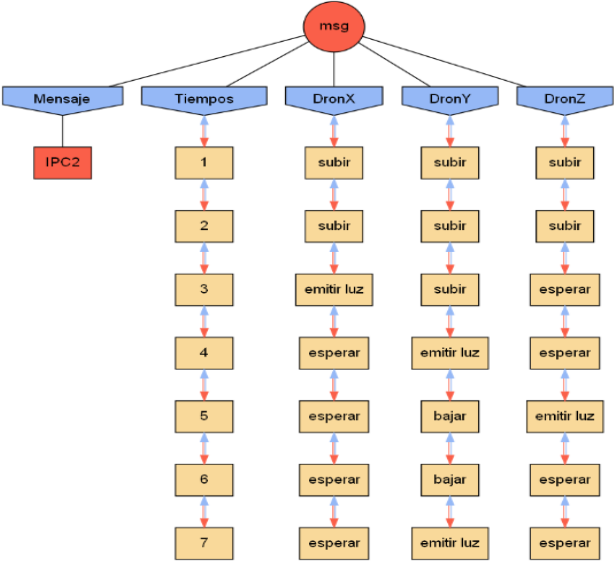


Figura 7, Gráfica mensaje
Fuente: Elaboración propia

Archivo de salida

El archivo de salida consiste en un archivo en formato XML que muestra los datos de la matriz reducida obtenida de cada señal. Estos se detallan a continuación:

- I. **respuesta:** Esta etiqueta muestra las respuestas obtenidas de los mensajes procesados.
- II. **ListaMensajes:** Esta etiqueta muestra todos los mensajes que se procesaron.
- III. **Mensaje:** esta etiqueta definirá información del mensaje que se procesó y contiene los siguientes atributos:
 - **SistemaDrones:** Indica el sistema que se utilizó
 - **TiempoOptimo:** Indica el tiempo que se tardo en interceptar el mensaje.
 - **MensajeRecibido:** Muestra el mensaje final que se recibió.
- IV. **Instrucciones:** Esta etiqueta siempre pertenece al mensaje que se está describiendo y muestra la lista de instrucciones que se realizaron y contiene los siguientes atributos.
 - **Tiempo:** Hace referencia a la operación que se realizó en este tiempo
 - **Acciones:** Esta etiqueta muestra el movimiento que realizo cada dron del sistema, ya sea que haya subido, bajado o esperado en dicho tiempo.

Funcionamiento de programa

El programa está diseñado con una interfaz gráfica para una interacción con el usuario de forma amigable, con menús y opciones detalladas para evitar todo tipo de error, además cada vez que se realice un evento se mostrar un mensaje dependiendo si todo salió correctamente o hubo algún tipo de error.

Inicialmente se mostrará una ventana principal la cual muestra la siguiente información:



Figura 8, Ventana principal

Fuente: Elaboración propia

En la parte superior derecha se logra observar una barra de menú, con cuatro opciones.

- Archivo
- Drones
- Sistema
- Mensaje

Al ingresar submenú “**Archivo**” se mostrará tres opciones:

- Inicializar Sistema
- Cargar
- Generar
- Salir

La opción Inicializar Sistema, permite volver a empezar nuevamente el sistema por se desea borrar todo el progreso.

La opción Cargar, permite abrir el archivo XML y se mostrará una ventana emergente donde podrás buscar en tu ordenador el documento en formato XML que desees ingresar con los datos de las señales. Al estar todo correctamente estructurado como se mostró

anteriormente se mostrará un mensaje de “Ingresado Correctamente” de lo contrario se mostrará un mensaje “Hubo un error, vuelva a verificar”.

En la opción Generar, una vez se haya procesado algún mensaje se podrá generar el archivo de salida el cual mostrará la información descrita anteriormente.

En la opción salir, finaliza el programa.

Al ingresar submenú “**Drones**” se mostrará dos opciones:

- Inventario
- Agregar

En la opción Inventario, se podrá visualizar los drones que existan o se hayan generado, si en todo caso no hay ningún dron, se mostrará un mensaje que indicara que no hay drones.

En la opción Agregar, se podrá agregar cualquier dron que el usuario desee siempre y cuando no exista en el inventario.

Al ingresar submenú “**Sistema**” se mostrará una opción:

- Ver gráfica

En la opción Ver gráfica, se mostrará los nombres de los sistemas que hay en memoria y se mostrará una barra de texto donde podrá escribir el nombre del sistema que desea graficar, una vez escrito siempre y cuando lo haya escrito correctamente se mostrará la gráfica.

Al ingresar submenú “**Mensaje**” se mostrará dos opciones:

- Mostrar mensajes
- Procesar mensajes

En la opción mostrar mensajes, se mostrará la información de cada mensaje desde su nombre hasta las instrucciones que tiene.

En la opción procesar mensaje, se mostrará una barra de desplazamiento donde se mostrará los nombres de los mensajes existentes y puede escoger cual desea procesar, una vez elegido el mensaje hay un botón donde podrá realizar el proceso y se mostrará de forma automática el tiempo optimo y el mensaje recibido.

Conclusiones

El utilizar listas enlazadas simples y listas enlazadas dobles es una herramienta muy importante al trabajar con matrices o cualquier otro tipo de datos que nos permitan manejar datos de forma ordenada y movernos en cualquier posición que uno desea.

Al darle solución a la interceptación de mensajes a través de listas doble mente enlazadas se logró alcanzar grandes objetivos propuestos para la elaboración de este programa, desde la lectura de un XML hasta la elaboración de diagramas utilizando Graphviz es una gran practica haber logrado culminar y hallar soluciones a este tipos de problemas, el uso de tipo de datos abstractos es una gran oportunidad para ir mejorando las habilidades para darle soluciones a problemas de una forma diferente, al poner en practica la abstracción para resolver problemas, es una gran habilidad que un programador tiene que tener.

Referencias bibliográficas

Bradlye Miller, & Ranum, D. L. (2017-2018). *Problem Solving with Algorithms and Data Structures using Python — Problem Solving with Algorithms and Data Structures*.

<https://runestone.academy/ns/books/published/pythonds/index.html>

Rosita Wachenchauzer, Margarita Manterola, Maximiliano Curia, Marcos Medrano, & Nicolás Paez. (2006). *Algoritmos de Programación con Python*. Diseño y programación web (libros, tutoriales y vídeos sobre HTML, CSS, JavaScript, PHP). <https://uniwebsidad.com/libros/algoritmos-python>

Anexos

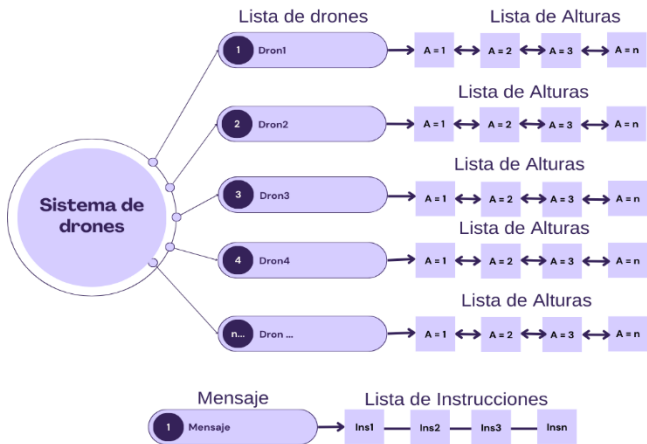


Figura 9, Lógica del programa

Fuente: Elaboración propia

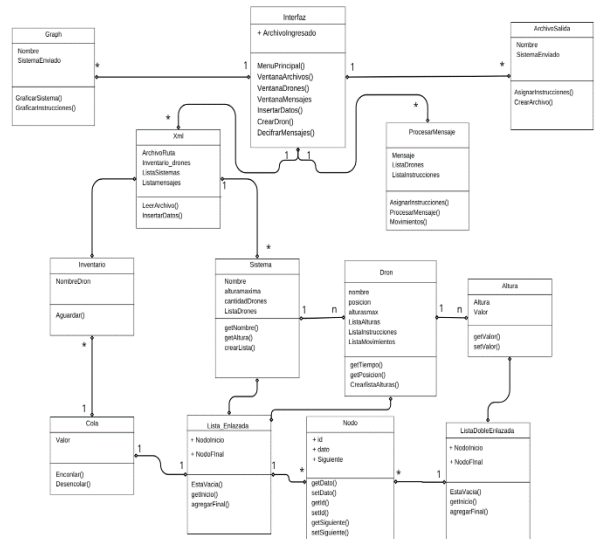


Figura 10, Diagrama de clases

Fuente: Elaboración propia



Figura 11, Menú principal

Fuente: Elaboración propia

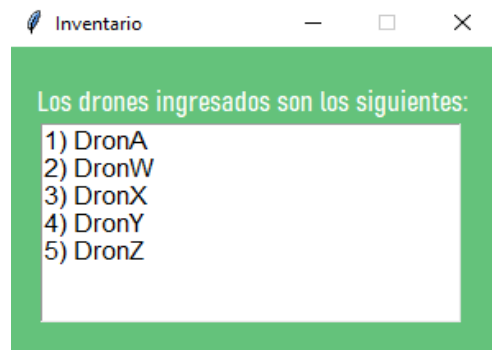


Figura 12, Inventario

Fuente: Elaboración propia

Figura 13, Crear Dron

Fuente: Elaboración propia

Figura 16, Procesar mensaje

Fuente: Elaboración propia

Figura 14, Graficar sistema

Fuente: Elaboración propia

Figura 15, Mensajes ingresados

Fuente: Elaboración propia

```
<?xml version='1.0' encoding='utf-8' ?>
<respuesta>
  <listaMensajes>
    <mensaje nombre="msg">
      <sistemaDrones>SD1</sistemaDrones>
      <tiempoOptimo>7</tiempoOptimo>
      <mensajeRecibido>IPC2</mensajeRecibido>
      <instrucciones>
        <tiempo valor="1">
          <acciones>
            <dron nombre="DronX">Subir</dron>
            <dron nombre="DronY">Subir</dron>
            <dron nombre="DronZ">Subir</dron>
          </acciones>
        </tiempo>
        <tiempo valor="2">
          <acciones>
            <dron nombre="DronX">Subir</dron>
            <dron nombre="DronY">Subir</dron>
            <dron nombre="DronZ">Subir</dron>
          </acciones>
        </tiempo>
        <tiempo valor="3">
          <acciones>
            <dron nombre="DronX">Emitir luz</dron>
            <dron nombre="DronY">Subir</dron>
            <dron nombre="DronZ">Esperar</dron>
          </acciones>
        </tiempo>
      </instrucciones>
    </mensaje>
  </listaMensajes>
</respuesta>
```

Figura 17, Archivo de salida

Fuente: Elaboración propia