

---

## PROYECTO 2

---

202200075 – Dominic Juan Pablo Ruano Pérez

### Resumen

Utilizando Tipos de Datos Abstractos (TDA), Tkinter para crear una interfaz gráfica de usuario amigable y Graphviz para generar gráficos, hemos desarrollado un sistema capaz de procesar archivos con extensión ".XML" y almacenar los datos en listas doblemente enlazadas. Estas listas han sido implementadas mediante una clase personalizada para listas dobles, creada por el alumno, y haciendo uso de una clase de datos.

La estructura de la lista doble se ha diseñado de manera secuencial para optimizar el acceso a los datos. Además, hemos creado una clase que importa la librería Graphviz y se enfoca en diversas funciones para crear nuevos nodos y generar gráficos de los diferentes sistemas de drones presentes en los archivos procesados.

Mediante la lectura y análisis de archivos con extensión ".XML", hemos recopilado datos que, gracias a la aplicación de lógica, bucles y estructuras de control, utilizamos para generar gráficos representativas de los distintos sistemas de drones contenidos en el archivo procesado.

### Palabras clave

1. Desarrollo
2. Procesamiento
3. Tkinter
4. Graphviz

### Abstract

*Using Abstract Data Types (ADT), Tkinter to create a friendly graphical user interface and Graphviz to generate graphs, we have developed a system capable of processing files with ".XML" extension and storing the data in doubly linked lists. These lists have been implemented using a custom class for double lists, created by the student, and making use of a data class.*

*The structure of the double list has been designed in a sequential way to optimize the access to the data. In addition, we have created a class that imports the Graphviz library and focuses on various functions to create new nodes and generate graphs of the different drone systems present in the processed files.*

*By reading and analyzing files with ".XML" extension, we have collected data that, thanks to the application of logic, loops and control structures, we use to generate representative graphs of the different drone systems contained in the processed file.*

### Keywords

1. Development
2. Processing
3. Tkinter
4. Graphviz

## Introducción

En este proyecto, se ha desarrollado un sistema que utiliza Tipos de Datos Abstractos (TDA), la biblioteca Tkinter para la creación de una interfaz gráfica amigable para el usuario, y Graphviz para generar gráficos. Este sistema tiene la capacidad de procesar archivos con extensión ".XML" y almacenar los datos en listas doblemente enlazadas. Estas listas se implementan mediante una clase personalizada para listas dobles y una clase de datos. El propósito principal de este sistema es procesar datos de sistemas de drones contenidos en archivos ".XML" y representar visualmente esta información a través de gráficos. Además, se ha tenido en cuenta la usabilidad, proporcionando una interfaz gráfica para facilitar su uso a usuarios sin experiencia en programación.

## Desarrollo del tema

Para el desarrollo de este sistema se implementaron algunos puntos importantes para la construcción del mismo, De esta forma mantener un flujo de trabajo mas ordenado.

Los puntos principales en este trabajo fueron los siguiente:

- Lectura del XML.
- Creación del GUI.
- Ordenamiento de los Datos de forma alfabética.
- Realización de las gráficas necesarias.

A continuación se mostrara la estructura de código del sistema donde se iniciara mostrando los TDA existente y sus respectivos métodos, además de que también pueden ser observados en el diagrama de clases que se encuentra en anexos. Seguido se mostrara y describirá en funcionamiento de la GUI realizada con tkinter, realizada para mayor comodidad del usuario.

## App.py

Es donde se encuentra la estructura base del sistema, se pueden observar los métodos generales que efectúan los que los botones implementados en la GUI del sistema requieren, además se observa toda la estructura del código necesario para generar la GUI con tkinter.

Cuenta con los siguientes métodos:

1. Inicializar
  - a. Inicializa el sistema borrando todos los datos existentes.
2. CargarXML
  - a. Esta función crea o inicializa el objeto con los datos del archivo seleccionado.
3. generarXML
  - a. generando un archivo de salida que contenga las instrucciones.
4. Drones
  - a. Este método obtiene los datos leídos y los muestra de una manera legible para el usuario y agrega la funcionalidad de agregar nuevos drones.
  - b. El método de addDron agregar un nuevo dron en orden alfabético al sistema.
5. Sistema
  - a. Genera una grafica con los sistemas de drones usando grapviz.
6. Mensajes
  - a. Muestra en pantalla de forma legible todos los sistemas de drones y sus atributos como nombre, sistema, etc...
7. Ayuda
  - a. Este método muestra en pantalla los datos del alumno para poder verificar el creador del sistema.

## XML.py

Esta clase abstracta es donde se encuentra todo lo necesario para este proyecto donde podemos leer, graficar ordenar y almacenar los datos que se leerán del archivo.

Se clasifica por las siguientes funciones:

1. `__init__`
  - a. Es el constructor de la clase donde debe enviar un path para poder obtener los datos.
  - b. Además de mostrar todas las características del objeto, como lo son: root, listadrones, listasistemas, mensajes.
2. `getDrones`
  - a. lee del archivo haciendo uso del root y el path para obtener todos los drones de ese archivo.
3. `Ordenar`
  - a. Ordena los drones obtenidos en orden alfabético.
4. `getSistemas`
  - a. obtiene todos los sistemas de drones del archivo.
5. `getMensajes`
  - a. obtiene todos los mensajes que existan en el archivo.
6. `ordenarMensajes`
  - a. ordena todos los mensajes de forma alfabética por medio del nombre del mensaje.
7. `graficar sistemas`
  - a. grafica utilizando graphviz los sistemas de drones.

## sistemaDrones.py

esta clase genera objetos los cuales deben ir listados en listas simples.

```
class SistemaDrones:
    def __init__(self, nombre, altura, cantidad, listadatos) -> None:
        self.nombre = nombre
        self.alturaMax = altura
        self.cantidadDrones = cantidad
        self.datos = listadatos

    def imprimir(self):
        print(f"Nombre: {self.nombre}, Altura: {self.alturaMax}, Cantidad de drones: {self.cantidadDrones}")
        temp = self.datos.primer
        while temp:
            print(f"Nombre del dron: {temp.getdato().NombreDron} | Letras: {temp.getdato().letrasSTR}")
            temp = temp.getsiguiente()
```

Donde vemos que este objeto contiene un nombre, alturaMax, cantidadDrones, datos.

En su estructura además de un método imprimir que fue utilizado para la verificación del ingreso de los datos.

Este objeto fue creado con la finalidad de ser contenido dentro de listas las cuales también formaran parte de un objeto a su vez.

## Contenido.py

Este TDA esta creado con la finalidad de ser almacenado en una lista doble conteniendo el nombre del dron y una lista con las distintas letras que puede ejecutar a distintas alturas.

```
class Contenido():
    def __init__(self, nombrDron, letrasSTR):
        self.NombreDron = nombrDron
        self.letrasSTR = letrasSTR
```

se puede observar los atributos de NombreDron y el de letras STR donde se guardaran los mismos.

## Instrucciones.py

Este TDA guarda una instrucción que debe realizar el dron, el cual debe ser implementado en una lista doble para conformar un mensaje.

```
class Instruccion:
    def __init__(self, dron, dato):
        self.dron = dron
        self.dato = dato
```

Se puede observar que tiene como atributos dron y dato guardando así que dato (instrucción/altura/letra) debe mostrar ese dron.

## Mensaje.py

Este mismo guarda o es la estructura principal para los mensajes guardando toda la información obtenida para el mensaje y conteniendo la lista de las instrucciones.

```
class Mensaje():  
    def __init__(self, nombre, sistema, lista  
    ):  
        self.nombre = nombre  
        self.sistema = sistema  
        self.instrucciones = lista
```

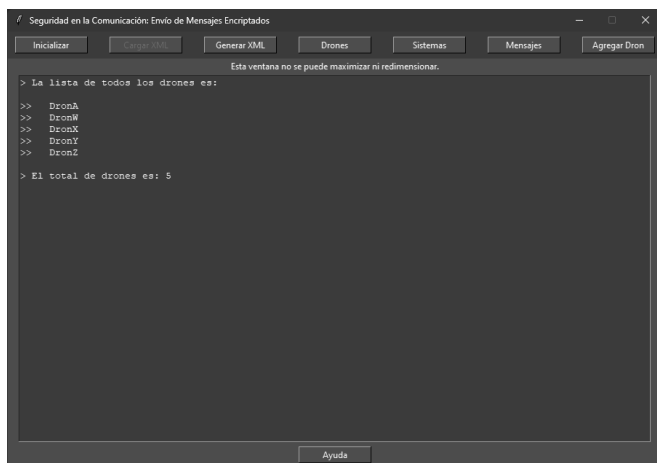
Se ve que los atributos son nombre del mensaje (con el cual mas adelante se ordenara), sistema de drones a utilizar y una lista doble con las instrucciones del mensajes.

### Creación de GUI

Para la creación de la GUI fue utilizado tkinter haciendo uso de los distintos objetos se creo una ventana con un alto, ancho predefinido y centrada en la pantalla del usuario.

Se estableció que para una mejor experiencia de usuario no se permitirá que esta misma pueda ser redimensionada o maximizada.

Se estableció una paleta de colores azul oscuro para ofrecer un tema oscuro y al mismo tiempo agradable y consistente.



### Lista de las funcionalidades en la GUI

1. Inicializar.
  - a. Esta inicializa el sistema borrando todos los datos guardados en el mismo, al iniciar el sistema no esta habilitado el botón.
2. Cargar XML.
  - a. Este botón abre una ventana de dialogo para seleccionar el archivo al cual se debe acceder, y por o tanto cual es el que contiene todas las instrucciones.
3. Generar XML.
  - a. Genera un archivo XML de salida con las instrucciones de los drones.
4. Drones.
  - a. Muestra en el cuadro de texto el cual esta deshabilitado para el usuario en todo momento, de una forma textual cuales son los drones exsistentes en el sistema, además hacia el lado derecho se muestra un botón para agregar nuevos drones.
5. Sistemas.
  - a. Genera una grafica con Graphviz y la abre para que el usuario la pueda visualizar.
6. Mensajes.
  - a. Muestra en el espacio de texto de forma textual todos los mensajes y su información.
7. Ayuda.
  - a. Muestra en el campo de texto la información de usuario y abre automáticamente la documentación del sistema.

### Conclusiones

El uso de listas doblemente enlazadas diseñadas de manera secuencial demuestra una consideración importante hacia la eficiencia en el acceso a los datos. Esto garantiza que la recuperación y manipulación de la información sea rápida y efectiva.

La incorporación de una interfaz gráfica de usuario (GUI) mediante la biblioteca Tkinter mejora significativamente la accesibilidad del software. Esto permite que usuarios con menos experiencia en programación puedan interactuar de manera más sencilla y efectiva con la aplicación.

El uso de la librería Graphviz para crear gráficos de los sistemas de drones es una característica valiosa. Proporciona una forma visual de comprender y analizar los datos, lo que puede ser esencial para la toma de decisiones.

En general, el proyecto combina eficiencia en el procesamiento de datos con una interfaz amigable, lo que lo convierte en una herramienta versátil con un alto potencial tanto en la industria como en la educación.

## **Referencias bibliográficas**

programmerclick.com (2023 – año de consulta)  
[python] Empezando con el módulo de python graphviz  
<https://programmerclick.com/article/32021840155/>

## Anexos

### Diagrama de clases:

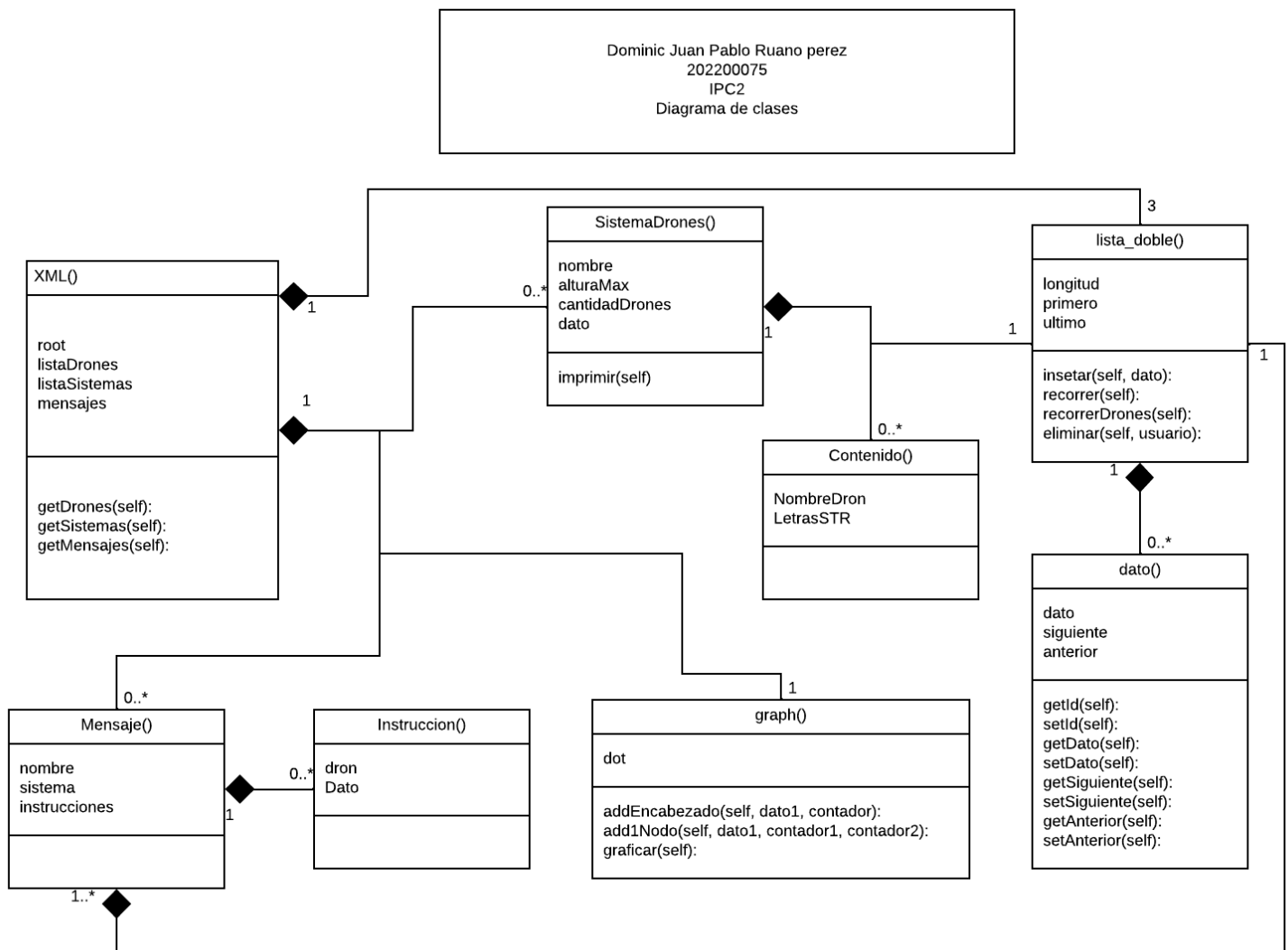


Figura 20. Diagrama de clases.

Fuente: elaboración propia.