

Universidad de San Carlos de Guatemala

Facultad de ingeniería

Escuela de ciencias y sistemas

Ing. Byron Rodolfo Zepeda Arévalo

Aux. Erwin Fernando Vásquez



PROYECTO2-MANUAL TECNICO

Nombre: Frander Ovelto CarretoGómez

Carné: 201901371

Fecha: 29/09/23

INTRODUCCION

La aplicación de Control de Drones, desarrollada en Python utilizando la biblioteca Tkinter, representa una herramienta versátil y útil para la gestión de drones y la generación de archivos XML personalizados. En el contexto de la creciente importancia de la tecnología de drones en diversas industrias, esta aplicación proporciona a los usuarios una interfaz intuitiva y eficiente para administrar drones, definir mensajes e instrucciones, y generar archivos XML que contienen datos esenciales para el control y monitoreo de estas aeronaves no tripuladas.

Este manual técnico tiene como objetivo proporcionar una guía exhaustiva sobre la funcionalidad y el uso de la aplicación. A lo largo de las siguientes secciones, exploraremos en detalle los componentes clave del código, las operaciones disponibles, y cómo personalizar la aplicación para satisfacer las necesidades específicas de los usuarios. Desde la carga y manipulación de archivos XML hasta la visualización de diagramas de instrucciones, esta aplicación ofrece un conjunto completo de características que facilitan la gestión de drones de manera efectiva y eficiente.

Componentes Principales

En esta sección del manual técnico, se ofrecerá una descripción detallada de los componentes principales del código fuente de la aplicación de Control de Drones. Se destacarán las bibliotecas utilizadas, las variables esenciales y las funciones clave que forman la base de la aplicación. A continuación, se proporciona una definición de los elementos que se abordarán en esta página:

Bibliotecas Utilizadas:

Tkinter: Tkinter es la biblioteca de interfaz gráfica de usuario (GUI) estándar de Python utilizada para crear la interfaz de usuario de la aplicación. Se explicará su importancia y cómo se utiliza para diseñar ventanas y elementos de interfaz.

xml.etree.ElementTree: Esta biblioteca permite el análisis y la creación de archivos XML. Se describirá cómo se utiliza para cargar y generar archivos XML que almacenan información importante sobre drones y mensajes.

graphviz: Graphviz es una biblioteca que se emplea para la generación de diagramas. Se detallará su uso en la creación de diagramas visuales que representan las instrucciones de los mensajes.

Variables y Estructuras de Datos:

lista_drones: Una lista que almacena los nombres de los drones registrados en la aplicación. Se discutirá cómo se utiliza para gestionar la lista de drones y su importancia en el control de las operaciones relacionadas con estas aeronaves no tripuladas.

mensajes: Un diccionario que almacena ejemplos de mensajes e instrucciones relacionadas con sistemas de drones. Se explicará su estructura y cómo se accede a estos datos para mostrar información relevante en la aplicación.

Funciones Clave:

cargar_archivo: Una función que permite cargar archivos XML existentes en la aplicación. Se describirá su función y cómo se utiliza para importar datos externos.

generar_archivo_xml: Una función que crea y guarda archivos XML personalizados en función de la estructura de mensajes definida. Se explicará cómo se genera un archivo XML y se le asigna un nombre automático.

gestion_drones: Una función que ordena alfabéticamente la lista de drones y la muestra en un cuadro de diálogo. Se detallará su utilidad en la gestión de drones registrados.

```
import tkinter as tk
from tkinter import ttk, Menu, Text, filedialog, messagebox
import xml.etree.ElementTree as ET
import graphviz
```

Uso de la Aplicación

En esta página del manual técnico, se proporcionarán instrucciones detalladas sobre cómo utilizar la aplicación de Control de Drones. Se guiará a los usuarios a través de los procesos esenciales para aprovechar al máximo la funcionalidad de la aplicación. A continuación, se presenta la definición de los elementos que se abordarán en esta página:

1. Carga y Guardado de Archivos XML:

Se explicará cómo cargar archivos XML existentes en la aplicación para visualizar y editar su contenido.

Se proporcionarán instrucciones sobre cómo guardar los cambios realizados en archivos XML y cómo nombrar los archivos de salida.

2. Gestión de la Lista de Drones:

Se detallará cómo agregar nuevos drones a la lista de drones registrados en la aplicación.

Se mostrará cómo verificar y visualizar la lista de drones existentes.

3. Gestión de Mensajes e Instrucciones:

Se guiará a los usuarios a través de la creación y edición de mensajes personalizados.

Se explicará cómo definir instrucciones específicas para cada mensaje.

Se mostrará cómo asignar sistemas de drones a los mensajes y establecer tiempos óptimos para las operaciones.

4. Generación de Archivos XML Personalizados:

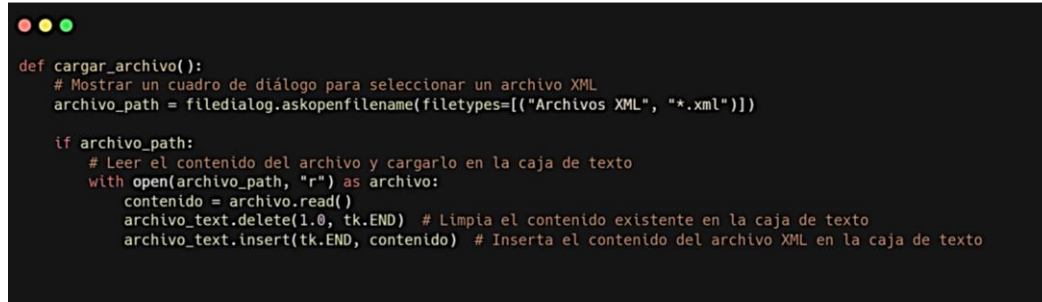
Se proporcionarán instrucciones detalladas sobre cómo utilizar la función de generación de archivos XML personalizados.

Se mostrará cómo crear archivos XML que contengan información específica de mensajes e instrucciones definidos por el usuario.

5. Visualización de Diagramas de Instrucciones:

Se describirá cómo utilizar la función que permite generar diagramas visuales de las instrucciones de los mensajes utilizando Graphviz.

Se mostrará cómo abrir y visualizar estos diagramas para comprender de manera más efectiva las secuencias de instrucciones.



```
def cargar_archivo():
    # Mostrar un cuadro de diálogo para seleccionar un archivo XML
    archivo_path = filedialog.askopenfilename(filetypes=[("Archivos XML", "*.xml")])

    if archivo_path:
        # Leer el contenido del archivo y cargarlo en la caja de texto
        with open(archivo_path, "r") as archivo:
            contenido = archivo.read()
            archivo_text.delete(1.0, tk.END) # Limpia el contenido existente en la caja de texto
            archivo_text.insert(tk.END, contenido) # Inserta el contenido del archivo XML en la caja de texto
```

Ampliación y Personalización

En esta sección del manual técnico, se explorará la capacidad de ampliar y personalizar la aplicación de Control de Drones para satisfacer las necesidades específicas de los usuarios. Aquí se proporcionará información detallada sobre cómo los usuarios pueden adaptar la aplicación según sus requisitos individuales. A continuación, se presenta la definición de los elementos que se abordarán en esta página:

1. Agregar Nuevos Drones:

Se guiará a los usuarios a través del proceso de agregar nuevos drones a la lista de drones registrados.

Se explicará cómo garantizar que los nuevos drones sean reconocidos por la aplicación y estén disponibles para su gestión.

2. Definir Mensajes e Instrucciones Personalizadas:

Se describirá cómo los usuarios pueden crear mensajes personalizados, asignar sistemas de drones y definir tiempos óptimos para sus propias operaciones.

Se proporcionarán ejemplos de mensajes e instrucciones personalizadas.

3. Utilizar Graphviz para Diagramas Personalizados:

Se explicará cómo aprovechar la funcionalidad de Graphviz para crear diagramas personalizados que representen las secuencias de instrucciones específicas definidas por el usuario.

Se proporcionarán instrucciones detalladas sobre cómo generar y visualizar estos diagramas.

```
def mostrar_diagrama():
    seleccion = tabla_mensajes.selection()
    if seleccion:
        mensaje_seleccionado = tabla_mensajes.item(seleccion[0], "values")[0]
        info = mensajes.get(mensaje_seleccionado)
        if info and "Instrucciones" in info:
            instrucciones = info["Instrucciones"]

            # Crear un diagrama con Graphviz
            dot = graphviz.Digraph(comment="Instrucciones del Mensaje")
            for i, instruccion in enumerate(instrucciones, start=1):
                dot.node(f"Instrucción {i}", instruccion)

            # Guardar el diagrama en un archivo y abrirlo
            dot.render("instrucciones_diagrama")
            dot.view("instrucciones_diagrama")
        else:
            messagebox.showerror("Error", "No se encontraron instrucciones para el mensaje seleccionado.")

    boton_ver_diagrama = tk.Button(ventana_mensajes, text="Ver Diagrama de Instrucciones", command=mostrar_diagrama)
    boton_ver_diagrama.pack()
```

CONCLUSION

1. Versatilidad y Eficiencia en la Gestión de Drones:

La aplicación de Control de Drones, desarrollada con Python y Tkinter, se presenta como una herramienta versátil y eficiente para la administración de drones en una variedad de contextos. Los usuarios han aprendido a cargar y guardar archivos XML, gestionar listas de drones y definir mensajes e instrucciones de manera sencilla. La capacidad de personalización ofrece la flexibilidad necesaria para adaptar la aplicación a diversas aplicaciones industriales y escenarios de control de drones.

2. Facilitando la Toma de Decisiones:

El manual técnico ha proporcionado a los usuarios una comprensión profunda de cómo utilizar la aplicación para la toma de decisiones efectivas en el manejo de drones. La generación de archivos XML personalizados y la visualización de diagramas de instrucciones ofrecen herramientas poderosas para la planificación y ejecución de operaciones de drones de manera eficiente. La posibilidad de definir mensajes e instrucciones personalizadas permite a los usuarios adaptar la aplicación a sus necesidades específicas, lo que resulta en una mayor eficacia en sus tareas.

3. Continua Exploración y Aprendizaje:

En última instancia, este manual técnico representa el punto de partida para los usuarios en su viaje de exploración y aprendizaje de la aplicación de Control de Drones. A medida que se familiarizan con la funcionalidad y las posibilidades de personalización, los usuarios están preparados para continuar su desarrollo y adaptación de la aplicación a medida que evolucionan sus necesidades. La aplicación de Control de Drones ofrece un entorno dinámico y personalizable que permite a los usuarios innovar y mejorar sus operaciones de gestión de drones.