

Sistema de videovigilancia en tiempo real a través de un dron programable.

Real-time video surveillance system through a programmable drone.

Hannia Mireles Moreno¹, María Guadalupe Espinosa Sánchez¹, Edgar Moreno Martínez¹, Igor Guryev¹

¹Universidad de Guanajuato. División de Ingenierías Campus Irapuato-Salamanca, Departamento de Estudios Multidisciplinarios. Av.

Universidad S/N. Col. Yacatitas. Yuriria, Gto.

h.mirelesmoreno@ugto.mx¹

mg.espinosasanchez@ugto.mx¹

e.morenomartinez@ugto.mx¹

guryev@ugto.mx¹

Resumen

Esta investigación se enfoca en el desarrollo e implementación de una aplicación del sistema Android destinado a lograr un control de vuelo preciso para drones, con el fin de realizar una ronda de vigilancia en el espacio abierto a partir de las coordenadas. El estudio aborda la necesidad de mejorar las capacidades para garantizar vuelos de drones precisos y seguros. Al aprovechar los avances tecnológicos y el potencial de las plataformas basadas en Android, se busca mejorar significativamente la autonomía y eficiencia operativa de los drones.

Para lograr estos objetivos, se explora el uso de Software Development Kits (SDK) específicos para drones en el entorno Android. Estos SDK ofrecen herramientas y recursos que permiten una integración más fluida de hardware y software, facilitando el desarrollo de funciones avanzadas de control y navegación. La investigación se enfoca en evaluar diferentes SDK disponibles en el mercado y seleccionar el más adecuado para la actualización del sistema Android.

Los resultados de esta investigación no solo contribuyen al avance de la tecnología de drones, sino que también abren nuevas oportunidades en diversas aplicaciones, como vigilancia, mapeo e inspección. La implementación exitosa de la actualización del sistema Android con el SDK seleccionado mejora la viabilidad y eficiencia de estas aplicaciones, buscando así el camino para una mayor integración de los drones en la actualidad.

Palabras clave: Android; drones; autonomía; videovigilancia; inspección.

Introducción

La década de los años ochenta marcó el inicio del estado del arte. El "estado del arte" se refiere al nivel más avanzado o actual de desarrollo, conocimiento, tecnología o investigación en un campo o área específica en un momento dado, tiene como objetivo inventariar y sistematizar la producción en un área del conocimiento, la cual servirá después como base para futuras investigaciones, desarrollos y avances [1].

El conocimiento sobre el estado del arte es esencial para la toma de decisiones de los avances más recientes en una disciplina y así poder identificar nuevas oportunidades, resolver problemas complejos y mejorar la eficiencia y efectividad en diversas áreas de estudio o aplicación.

En el contexto tecnológico, el estado del arte puede referirse a los dispositivos, métodos o sistemas más avanzados disponibles en un campo en particular. Por ejemplo, en el ámbito de los drones, el estado del arte implicaría los drones más sofisticados y avanzados tecnológicamente disponibles en el momento actual, donde se pueden llegar a incluir áreas muy donde se ha producido un progreso notable como: autonomía, carga útil, cámaras, sensores, tiempo de vuelo, alcance, seguridad, aplicaciones emergentes, por mencionar algunas [2].

Metodología

Para el desarrollo del sistema, se decidió utilizar Android Studio que es un entorno de desarrollo integrado (IDE) oficial para crear aplicaciones en el sistema operativo (SO) Android. Aunque existen varias plataformas disponibles para programar drones, se optó por trabajar con una de las más utilizadas, debido a que el SO Android proporciona un entorno de programación robusto y flexible para controlar y comunicarse con el dron, al utilizar dicha plataforma se puede aprovechar la potencia de Android Studio y sus características específicas para el desarrollo de aplicaciones Android.

Android Studio se basa en Java y se puede escribir, depurar y compilar código, además aprovecha las referencias del Android SDK, el cual proporciona las herramientas de desarrollo de software, documentación y librerías necesarias para programar el dron de manera efectiva [3]. Al utilizar las funciones internas y las librerías del SDK, se puede acceder a las características y capacidades del dron para controlar y realizar acciones específicas.

El dron utilizado para probar el sistema fue el modelo “DJI MAVIC AIR”. Se trata de un dron de cuadricóptero, lo que implica que está equipado con cuatro rotores, uno en cada esquina del dron. Entre sus características más destacadas se encuentran:

- Resolución máxima de la cámara de 4k.
- Tiempo máximo de vuelo de 21 minutos.
- Velocidad de 68.4 km/.
- Sensores de detección para evitar colisiones.

Implementación

Para implementar las nuevas actualizaciones del sistema, se siguió un proceso que comenzó con la configuración inicial. En esta etapa, se llevó a cabo la descarga e instalación de Android Studio desde el sitio oficial. Posteriormente, se procedió a adaptar Android Studio a las especificaciones del sistema con el que se iba a trabajar.

Una parte importante de esta adaptación consistió en la descarga del SDK adecuado para el desarrollo. Esto permitió acceder a las funciones internas del dron y utilizar sus capacidades específicas de manera adecuada.

Esta configuración inicial en Android Studio y la instalación del SDK establecieron las bases para el desarrollo de las nuevas actualizaciones del sistema, garantizando que se tuvieran todas las herramientas y recursos necesarios para llevar a cabo la implementación con éxito.

Antes de comenzar el desarrollo, se realizó el registro de la aplicación en DJI para obtener las credenciales y autorizaciones necesarias para la comunicación con el dron.

Después, se llevó a cabo la programación de la aplicación, poniendo énfasis en la implementación de la funcionalidad requerida. Es importante tener en cuenta que la plataforma Android Studio permite programar principalmente en dos lenguajes: Java y Kotlin. En este caso, el lenguaje de desarrollo del sistema en cuestión fue Java.

La principal actualización del sistema que se realizó consistió en la adición de un botón con la leyenda “FLY” en la interfaz principal del sistema.

En la interfaz principal del sistema se pueden observar varios elementos que facilitan la experiencia del usuario, los cuales se pueden observar en la parte inferior del dispositivo, los cuales son:

- Botón “CAPTURE”.
- Botón “START RECORD”.
- Botón “SHOOT PHOTO MODE”.
- Botón “RECORD VIDEO MODE”.

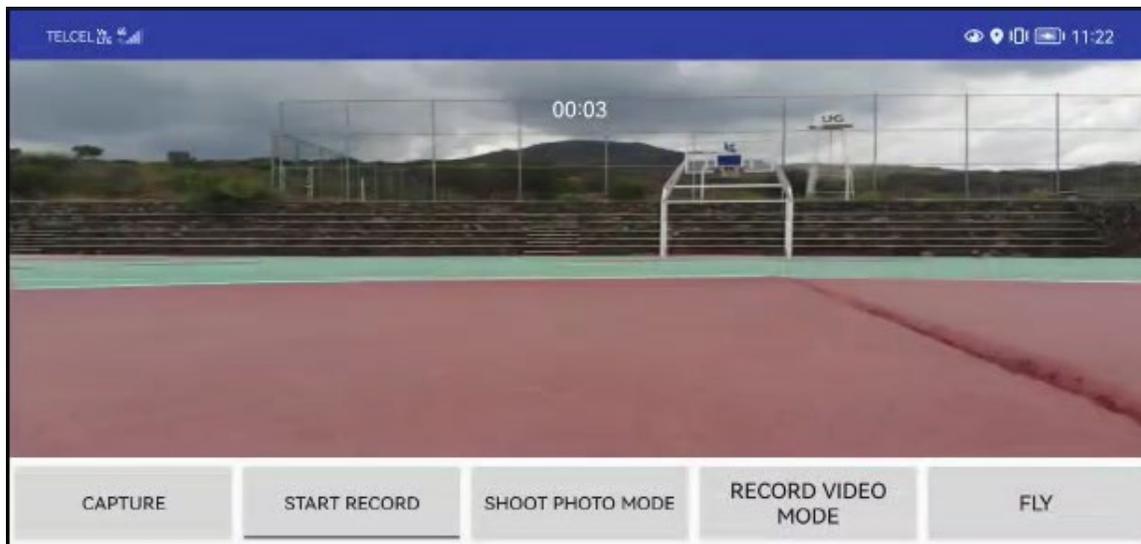


Figura 1. Interfaz principal modificada del sistema.

El nuevo botón agregado, permite al dron recibir instrucciones para establecerse en coordenadas iniciales y comenzar un recorrido hacia coordenadas objetivo. A continuación, se describen los pasos detallados del proceso:

1. Posicionamiento en las coordenadas iniciales: El dron se desplaza automáticamente hacia las coordenadas iniciales especificadas en el desarrollo del sistema. Además, en esta etapa, el dron ajusta la altitud deseada.
2. Inclinación automática de la cámara: Una vez posicionado el dron en las coordenadas iniciales, la cámara se inclina automáticamente después de realizar los cálculos necesarios, para obtener una mejor vista del entorno.
3. Posicionamiento en las coordenadas objetivo: El dron se dirige a las coordenadas objetivo específicas en el desarrollo del sistema. Durante este paso, el dron puede ajustar su altitud.
4. Vuelo en círculo: Una vez que el dron ha llegado a las coordenadas objetivo, realiza un vuelo circular alrededor del punto de interés.
5. Posicionamiento en las coordenadas de inicio: Finalmente, el dron regresa a las coordenadas iniciales, completando el recorrido establecido. Al llegar a ese punto, el dron aterriza.

Para realizar el posicionamiento de las coordenadas de inicio y objetivo, se empleó la clase "MissionControl", proporcionada por el SDK de DJI (DJI Mobile SDK), esta clase proporciona funcionalidades para controlar y administrar misiones y acciones relacionadas con vuelos de drones. Al ser una interfaz centralizada, facilita la planificación y ejecución de diversas tareas y secuencias de vuelo [4].

Es importante destacar que, para obtener las coordenadas necesarias, se utilizó Google Maps como herramienta de referencia. Se determinó que las coordenadas de inicio correspondieran a las canchas de básquetbol del Departamento de Estudios Multidisciplinarios (Sede Yuriria), mientras que las coordenadas objetivo se establecieron en una torre de agua ubicada a pocos metros de distancia de las canchas. Esta información geográfica permitió definir los puntos de partida y llegada para el vuelo del dron.

Para lograr la inclinación de la cámara, se empleó la clase "GimbalAttitudeAction". Esta clase representa una acción de movimiento de cardán de la cámara. Al crear una instancia de esta clase y agregarla a la línea de tiempo de Mission Control, el cardán de la cámara se moverá a la actitud (Attitude) o ángulo especificado cuando la línea de tiempo llegue a la acción correspondiente [5].

La actitud de la cámara no se agregó como un valor fijo, sino que se calculó de manera dinámica para adaptarse a posibles cambios de altitud y el radio, para esto se utilizó la siguiente ecuación:

Ecuación 1. Attitude de la cámara.

$$Attitude = \left(\tan^{-1} \left(\frac{altitude}{radius} \right) / \pi \right) * 180$$

Es esencial tener en cuenta que los resultados obtenidos pueden estar en grados o radianes, lo cual es crucial para la correcta inclinación de la cámara. Si los resultados no son interpretados correctamente, no observaran cambios en la posición de la cámara.

Este enfoque de cálculo dinámico permitió que la inclinación de la cámara se ajustara automáticamente. Al realizar el cálculo utilizando la ecuación mencionada anteriormente, se logró una adaptación flexible y precisa de la inclinación de la cámara.

Por último, se tiene la clase "HotPointAction", con esta clase el dron volará a lo largo de una parte de un círculo centrado, en una ubicación de interés cuando la línea de tiempo llegue a su acción [6].

Resultados

La implementación exitosa del sistema como una nueva aplicación en un celular con SO Android, permitió lograr vuelos precisos y seguros del dron, alcanzando altitudes de 25 metros en cada recorrido. Durante las pruebas realizadas, nos enfrentamos a diversos obstáculos que pusieron a prueba la capacidad del dron y el sistema.



Figura 3. Dron utilizado para las pruebas, modelo DJI MAVIC AIR.



Figura 2. Sistema implementado en un celular con SO Android.

Uno de los principales desafíos fue la malla ciclónica que cubre el perímetro de las canchas de básquetbol ubicadas cerca de las coordenadas de inicio, gracias al uso cuidadoso de las funcionalidades de la clase "MissionControl", el dron pudo evadir la malla sin dificultades y mantener su ruta de vuelo hacia la torre de agua.

Otro obstáculo fue la presencia de la torre de transmisión de la Universidad, para evitar cualquier daño al dron, se ajustó la trayectoria de vuelo, permitiendo al dron realizar el vuelo con precisión y seguridad.

Las condiciones climatológicas también fueron consideradas durante las pruebas. En situaciones de viento fuerte, el dron tuvo que compensar la fuerza de aire para mantener su estabilidad, manteniendo su trayectoria de vuelo y altura constante.

También se pudo observar que las imágenes mostradas en el sistema presentaban una considerable distorsión. Por tal motivo se planteó la posibilidad de que esta distorsión fuera causada por interferencias en la señal durante la realización del vuelo.

El tiempo de vuelo estimado para cada recorrido fue de 3 minutos y medio a 4 minutos, lo cual permitió al dron completar el vuelo circular y regresar al punto de inicio sin problema. La duración del vuelo fue adecuada para observar y capturar imágenes de la torre de agua con la inclinación automática de la cámara, proporcionando una vista panorámica del entorno desde diferentes ángulos.

Los resultados obtenidos demostraron que el sistema fue capaz de superar con éxito los objetivos del proyecto y realizar vuelos precisos a una altura de 25 metros, garantizando una observación efectiva de la torre de agua y una experiencia de vuelo segura y confiable.

A continuación, se presentan algunas de las evidencias graficas.

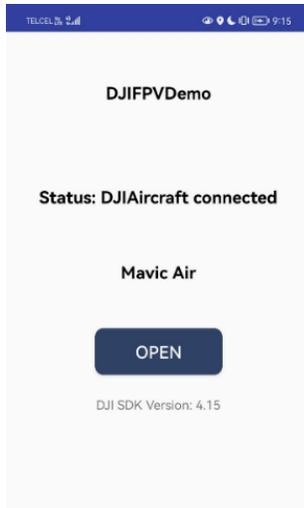


Figura 8. Registro del dron en el sistema.



Figura 7. Dron ubicado en las coordenadas de inicio.



Figura 6. Torre de agua del DEM Yuriria (Coordenadas objetivo).

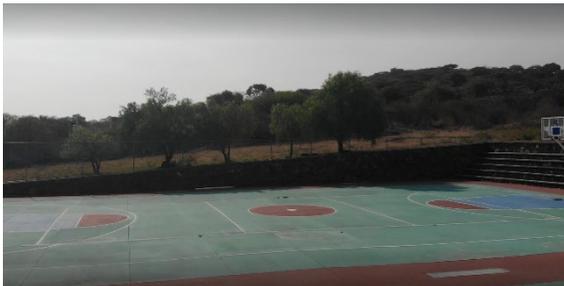


Figura 5. Imagen del entorno sin inclinación de la cámara.



Figura 4. Imagen del entorno con la inclinación automática de la cámara.



Figura 9. Trayectoria del dron.
Punto A: Coordenadas de inicio.
Punto B: Coordenadas objetivo

Conclusiones

El desarrollo del sistema, utilizando Android Studio y el SDK de DJI han sido la clave para lograr vuelos precisos y efectivos del dron. La elección de Android Studio como plataforma de desarrollo demostró ser acertada, ya que permitió aprovechar las capacidades y características del SO Android para el control y comunicación con el dron.

Es importante destacar que, durante las pruebas de implementación, nos encontramos con diversos tipos de obstáculos, que abarcaban desde las características del entorno donde se realizaron las pruebas hasta las condiciones climáticas presentes. Sin embargo, a pesar de cada uno de estos obstáculos, los resultados obtenidos fueron satisfactorios.

Y la incorporación de ciertas funcionalidades como la inclinación automática de la cámara durante el vuelo mejoro significativamente la experiencia de observación y proporcionando una perspectiva más amplia y detallada del entorno.

El uso de las clases del SDK de DJI facilito la administración de las misiones y acciones de vuelo del dron, permitiendo una planificación eficiente y una ejecución de los objetivos. La inclusión de un botón en la interfaz principal para enviar instrucciones al dron simplifico el proceso de control y mejoro significativamente la usabilidad del sistema.

Aunque se ha identificado una leve distorsión en las imágenes mostradas durante los vuelos, este aspecto no comprometi6 la funcionalidad general del sistema y puede ser abordado mediante un análisis más detallado y posibles ajustes en futuras versiones.

El sistema cumpli6 con los objetivos establecidos, permitiendo vuelos precisos, una visualización efectiva del entorno y de las coordenadas objetivo, adem6s de brindar una experiencia de vuelo segura. Con la optimizaci6n de la calidad de las imágenes y posibles mejoras futuras, el sistema tiene el potencial de ser una herramienta en diversas aplicaciones que requieren el control y monitoreo de drones de manera precisa y efectiva.

Bibliografía/Referencias

- [1] Molina Montoya, N. P. (2005). ¿Qué es el estado del arte? Ciencia y Tecnología para la salud Visual y Ocular, 3(5), 73-75.
- [2] Reuter, F., & Pedenovi, A. (2019). Los drones y sus aplicaciones a la ingeniería. Universidad Nacional de Santiago del Estero. Facultad de Ciencias Forestales. Serie Didactica N, 43.
- [3] Sol González, Jorge (2021). Diseño e implementaci6n de un SDK Android para facilitar la interacci6n de aplicaciones m6viles con una blockchain. Proyecto Fin de Carrera / Trabajo Fin de Grado, E.T.S. de Ingenieros Inform6ticos (UPM), Madrid, España.
- [4] DJI Mobile SDK Documentation. (s. f.). <https://developer.dji.com/api-reference/android-api/Components/Missions/DJIMissionControl.html>
- [5] DJI Mobile SDK Documentation. (s. f.-c). https://developer.dji.com/api-reference/android-api/Components/Missions/DJIGimbalAttitudeAction.html#djigimbalattitudeaction_initwithattitude_inline
- [6] DJI Mobile SDK Documentation. (s. f.-d). https://developer.dji.com/api-reference/android-api/Components/Missions/DJIHotpointAction.html#djihotpointaction_initwithmission_inline