VOLUMEN 48
XXXI Verano De la Ciencia
ISSN 2395-9797
www.jovenesenlaciencia.ugto.mx

Sistema de información para monitoreo y ajustes de trayectoria de un dron programable

Information System for Monitoring and Trajectory Adjustment of a Programmable Drone

Emiliano Ortega García, Dr. Igor Guryev.
Universidad de Guanajuato
e.ortegagarcia@ugto.mx, guryev@ugto.mx

Resumen

Este trabajo describe el desarrollo de un sistema para monitorear y ajustar la trayectoria de un dron DJI Mavic Air programable. El sistema está formado por tres partes que trabajan de manera conjunta: una aplicación móvil en Android Studio, un servidor backend encargado de procesar y transmitir la información en tiempo real, y una aplicación web para su visualización remota. Desde el teléfono es posible comunicarse con el dron para despegar, aterrizar y revisar datos como el estado de vuelo o el nivel de batería, mientras que la plataforma web recibe esos mismos datos y los muestra al usuario a distancia. Las pruebas realizadas confirman que esta arquitectura distribuida es una opción viable para supervisar y controlar vehículos aéreos no tripulados.

Palabras clave: dron; telemetría; aplicación móvil; monitoreo en tiempo real; arquitectura cliente-servidor.

Introducción

Los drones, o vehículos aéreos no tripulados (UAV), se usan hoy en día en inspección de infraestructura, vigilancia, agricultura de precisión, cartografía y monitoreo ambiental. Para que esas tareas se realicen de forma segura, se necesitan sistemas capaces de mostrar en tiempo real el estado del vehículo y de controlar sus funciones sin poner en riesgo la misión.

En este trabajo se construyó un sistema de este tipo para un dron DJI Mavic Air, compuesto por una aplicación móvil hecha en Android Studio y una aplicación web, comunicadas entre sí a través de un servidor backend. Desde el celular se controlan las funciones esenciales del dron, y en la página web se sigue el vuelo en tiempo real.

El objetivo es integrar estos componentes en una sola plataforma que permita ver el estado operativo del dron y enviarle comandos básicos de navegación, comprobando así que el celular, el UAV y el servidor pueden comunicarse entre sí sin problemas.

Revisión de literatura

En la última década, los vehículos aéreos no tripulados (UAV) han crecido de forma notable gracias a su versatilidad: hoy se emplean en inspección de infraestructura, agricultura de precisión, cartografía, vigilancia, búsqueda y rescate, monitoreo ambiental e incluso cinematografía aérea. Buena parte de ese crecimiento se explica por el software de control y monitoreo desarrollado para estos vehículos, que ha hecho posible automatizar misiones completas y analizar la información que generan casi al momento (Austin, 2010).

La telemetría es uno de los pilares de cualquier sistema basado en UAV: se trata de adquirir, transmitir y mostrar la información que el vehículo genera mientras vuela. Posición GPS, altitud, velocidad, orientación, nivel de batería, estado de los motores, intensidad de señal y modo de vuelo son algunos de los datos más relevantes. Tenerlos disponibles en todo momento hace posible seguir el comportamiento del dron y detectar a tiempo cualquier situación que pueda poner en riesgo la misión (Doherty & Rudol, 2007).

DJI ofrece a los desarrolladores el DJI Mobile SDK, un conjunto de bibliotecas para acceder desde Android o iOS a las funciones de sus drones. A través de esta interfaz de programación, una aplicación puede leer la telemetría, controlar la cámara, enviar comandos de vuelo y revisar el estado de los sensores del vehículo, todo sin tocar el firmware original (DJI, s. f.).

Las aplicaciones Android que controlan UAV suelen construirse con una arquitectura orientada a eventos: la información del dron llega mediante callbacks que avisan cada cambio de estado del sistema. Esto permite refrescar la interfaz al instante y enviar comandos con muy poca demora, lo que hace que el operador sienta el vuelo casi en tiempo real (Android Developers, s. f.).

Cuando varios usuarios o dispositivos necesitan consultar la misma información, conviene recurrir a una arquitectura cliente-servidor. Aquí la aplicación móvil actúa como cliente que obtiene los datos del dron, y un servidor central se encarga de procesarlos, guardarlos y repartirlos a otros clientes, como una aplicación web o un sistema de monitoreo. Este esquema facilita escalar el sistema y sumar nuevas funciones sin tocar la comunicación directa con el UAV (Richards, 2015).

Python se ha vuelto una de las opciones más comunes para construir estos backends, por lo fácil que resulta programar en él, la cantidad de bibliotecas disponibles y su compatibilidad con distintos protocolos de comunicación. Frameworks como FastAPI, en particular, permiten montar servicios REST rápidos y con tiempos de respuesta bajos, algo que importa bastante cuando se trabaja con telemetría que se actualiza constantemente (Ramírez, s. f.).

Del lado de la aplicación web, tecnologías como WebSocket permiten mostrar la información en tiempo real manteniendo una conexión abierta entre el navegador y el servidor. A diferencia de las solicitudes HTTP periódicas, con WebSocket es el servidor quien envía las actualizaciones apenas llegan nuevos datos del dron, lo que reduce el tráfico de red y hace la experiencia más fluida para quien la usa (Fette & Melnikov, 2011).

Combinar aplicaciones móviles, backend y plataformas web de esta manera es, hoy en día, uno de los esquemas más comunes para supervisar UAV, porque separa con claridad quién adquiere los datos, quién los procesa y quién los muestra, lo que en la práctica se traduce en sistemas más robustos, escalables y fáciles de mantener. El proyecto que aquí se presenta sigue justamente ese esquema: DJI Mobile SDK y Android Studio del lado del dron, un backend en Python en el medio, y una interfaz web para la supervisión remota.

Metodología

La Figura 1 muestra la arquitectura del sistema de monitoreo y ajuste de trayectoria construido para este proyecto. Se organizó en cuatro niveles: hardware, aplicación Android, backend e interfaz web, de modo que la adquisición de datos, el procesamiento, la comunicación y la visualización quedaran separados entre sí.

El primer nivel es el hardware: el dron DJI Mavic Air y su control remoto. El dron trae sensores que miden posición GPS, orientación, velocidad, altitud, nivel de batería y estado general del vuelo, y toda esa información viaja de forma continua hacia la aplicación móvil a través de la conexión que el control remoto establece con el dispositivo Android mediante el protocolo RC-OcuSync.

El segundo nivel es la aplicación Android, desarrollada en Android Studio con el DJI Mobile SDK. Es la pieza central del sistema: es la que habla directamente con el dron y la que administra tanto la información que llega como los comandos que se envían.

Al arrancar, el módulo DroneApplication registra el SDK y comprueba que exista conexión con el dron. Una vez establecida esa comunicación, entra en juego TelemetryCollector, que recibe de forma continua la telemetría entregada por el SDK: estado de vuelo, nivel de batería, posición GPS y coordenadas del joystick virtual.

Esos datos pasan luego al servicio WebSocketRelayService, que corre en segundo plano y mantiene abierta una conexión permanente con el backend. Desde ahí se retransmiten sin pausa, de modo que cualquier otro cliente conectado los recibe casi al mismo tiempo que ocurren.

La aplicación también incluye el módulo CommandDispatcher, que toma las órdenes que llegan del servidor o del propio usuario y las traduce a comandos que el SDK de DJI entiende. Con él quedaron implementadas funciones como el despegue automático (Take Off), el aterrizaje automático (Landing), el retorno al punto de origen (Return To Home) y el envío de puntos de trayectoria (Waypoints).

ISSN 2395-9797
www.jovenesenlaciencia.ugto.mx

El tercer nivel es el backend, construido con FastAPI. A través del servicio REST /ws/drone recibe sin parar la información que envía la aplicación Android; ahí los datos de telemetría se guardan de forma temporal y pasan por el Concentrador de Telemetría, que hace de intermediario entre la aplicación móvil y el resto de los clientes conectados.

Para la visualización en tiempo real está el servicio /ws/dashboard, hecho con WebSocket, que envía las actualizaciones de telemetría a los navegadores conectados sin que estos tengan que pedirlas. El backend también expone /api/command, que recibe las órdenes de control que salen de la interfaz web y las reenvía a la aplicación Android mediante peticiones HTTP.

El cuarto y último nivel es la interfaz web, hecha en HTML, CSS y JavaScript. Muestra en tiempo real los parámetros de vuelo del dron (posición GPS, altitud, estado de vuelo, nivel de batería) en un conjunto de paneles, y desde ahí mismo se pueden lanzar comandos de despegue, aterrizaje, retorno al punto de origen o envío de waypoints, que el backend recibe y reenvía a la aplicación Android.

En conjunto, el flujo arranca con la conexión entre el dron y la aplicación Android; de ahí la telemetría viaja sin pausa hacia el backend, que la distribuye a la interfaz web. Y en sentido contrario, cuando el usuario lanza un comando desde el navegador, el backend lo recibe y lo reenvía a la aplicación Android, que usa el DJI Mobile SDK para transmitirlo al dron. Así queda una comunicación de ida y vuelta entre todos los componentes, que es lo que finalmente permite monitorear y controlar el UAV en tiempo real.

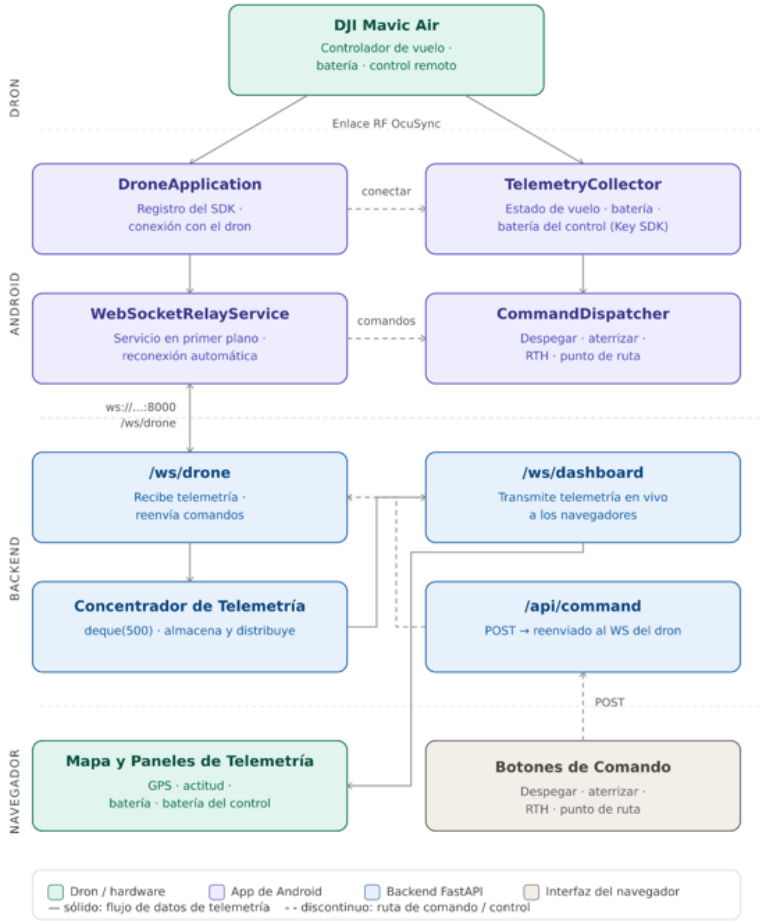


Figura 1. Esquemático general del sistema.

Resultados

Las pruebas confirmaron que tanto la aplicación Android como la plataforma web funcionan correctamente al comunicarse con el dron DJI Mavic Air. La aplicación logró conectarse sin problemas, habilitar las funciones básicas de control y mostrar la información de vuelo que llega a través del DJI Mobile SDK.



Figura 2. Fotografía del dron DJI Mavic Air.



Figura 3. Control remoto utilizado durante las pruebas.

Después se probó la aplicación móvil ya en funcionamiento: desde ahí fue posible ver el estado del dron en pantalla y ejecutar los comandos de operación sin inconvenientes.

VOLUMEN 48
XXXI Verano De la Ciencia
ISSN 2395-9797
www.jovenesenlaciencia.ugto.mx



Figura 4. Aplicación Android en funcionamiento.

Por último, la aplicación web recibió sin errores los datos que envía el backend y los mostró en tiempo real mientras el dron permanecía conectado.

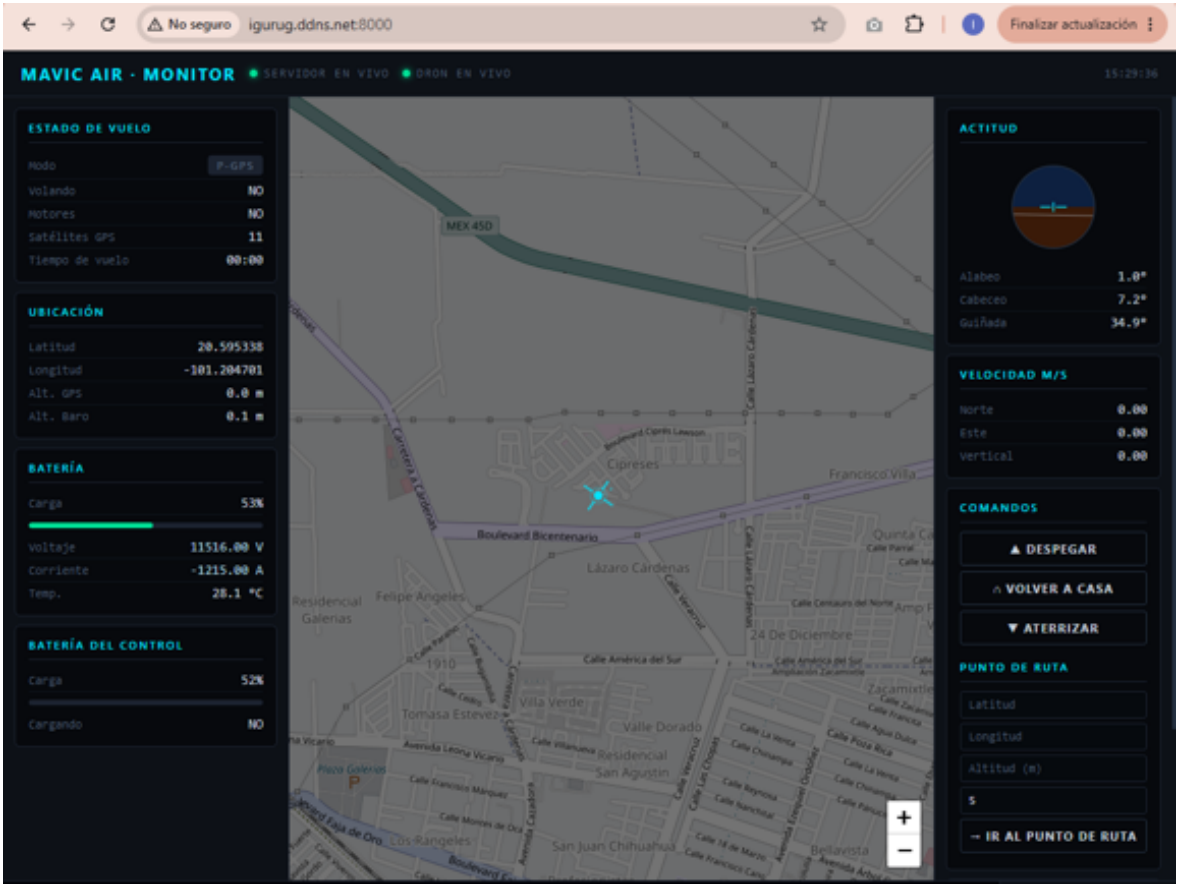


Figura 5. Interfaz web mostrando la información del dron en tiempo real.

VOLUMEN 48
XXXI Verano De la Ciencia
ISSN 2395-9797
www.jovenesenlaciencia.ugto.mx

Conclusiones

Se logró construir un sistema que integra un dron DJI Mavic Air, una aplicación móvil en Android Studio, un backend en Python y una aplicación web, todo trabajando junto para monitorear la información en tiempo real.

Con esta solución fue posible ejecutar funciones esenciales como el despegue y el aterrizaje, y también visualizar los parámetros del estado de vuelo y de la batería. El intercambio de información entre todos los componentes funcionó correctamente, lo que valida la arquitectura propuesta.

Como trabajo futuro queda la posibilidad de ampliar el sistema con navegación autónoma, seguimiento de trayectorias, un historial de vuelos y algoritmos de planificación y control que ajusten la trayectoria de forma automática.

Bibliografía/Referencias

Android Developers. (s. f.). *Android Studio*. https://developer.android.com/studio

Austin, R. (2010). *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*. John Wiley & Sons.

DJI. (s. f.). *DJI Mobile SDK Documentation*. https://developer.dji.com/mobile-sdk/

Doherty, P., & Rudol, P. (2007). A UAV search and rescue scenario with human body detection and geolocalization. En M. A. Orgun & J. Thornton (Eds.), *AI 2007: Advances in Artificial Intelligence* (pp. 1–13). Springer.

Fette, I., & Melnikov, A. (2011). *The WebSocket Protocol (RFC 6455)*. Internet Engineering Task Force. https://doi.org/10.17487/RFC6455

Ramírez, S. (s. f.). *FastAPI Documentation*. https://fastapi.tiangolo.com/

Richards, M. (2015). *Software Architecture Patterns*. O'Reilly Media.