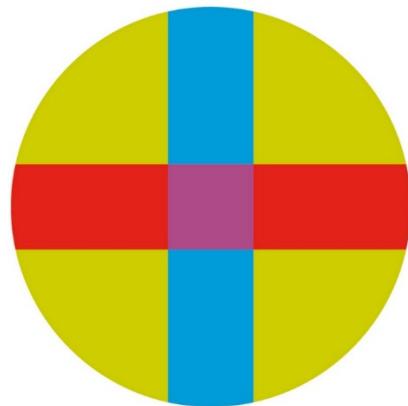


UNIVERSIDAD CEU - SAN PABLO  
ESCUELA POLITÉCNICA SUPERIOR  
GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

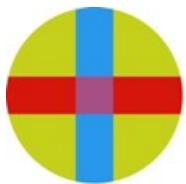
**DISEÑO E IMPLEMENTACIÓN DE UN  
DRON 5G Y SU INFRAESTRUCTURA**

**DESIGN AND IMPLEMENTATION OF A 5G  
DRONE AND ITS INFRASTRUCTURE**

Autor: Jaime Bermejo Torres  
Directores: Sergio Saugar García, Guillermo de la Calle  
Velasco

Junio 2024





UNIVERSIDAD SAN PABLO-CEU  
ESCUELA POLITÉCNICA SUPERIOR  
División de Ingeniería

Datos del alumno

NOMBRE:

Datos del Trabajo

TÍTULO DEL PROYECTO:

Tribunal calificador

PRESIDENTE:

FDO.:

SECRETARIO:

FDO.:

VOCAL:

FDO.:

Reunido este tribunal el \_\_\_\_ / \_\_\_\_ / \_\_\_\_ , acuerda otorgar al Trabajo Fin de Grado  
presentado por Don \_\_\_\_\_ la calificación de \_\_\_\_\_.



## **AGRADECIMIENTOS**

Este trabajo está dedicado a mi madre, quién ha estado durante estos años difíciles siempre ahí, dándome todo su cariño, tiempo y apoyo incondicionales. Y también a mi abuelo Julián, quien me enseñó la constancia del trabajo, el sacrificio y el amor por la naturaleza.

Quiero agradecer a toda mi familia por todo lo que me han dado, a mi tía Marta, mi tío Quique y Edu por su consejo y cariño, y al otro Teleco de la familia, Rubén, por escucharme siempre.

Quiero agradecer a mis tutores de este TFG todo el tiempo invertido en este trabajo y por ayudarme a plasmar mis ideas sobre papel. También quiero agradecer a mis dos grandes amistades de la carrera, Marina y Teresa, por todo lo que hemos vivido estos últimos años juntas.

## **ABSTRACT**

*This work covers the design, implementation, and manufacturing, as well as the operation and testing, of a 5G drone and its external infrastructure, with the goal of flying and data collection.*

*All the communication and control logic of the drone will be contained in an application within a smartphone, which will be responsible for collecting data from different fields: real-time photography, inertial data from various sensors, and the device's location coordinates. All this data will be packaged and sent through the Internet to a control and visualization system. Individual services for data collection and drone control will be executed on this system. A backend service will process the received data, a frontend service will display it in real-time, and a database will store it. The communication interface between the smartphone and the drone's electronics will be managed by a microcontroller (MCU).*

*The user will be able to control the drone with a remote control, whose control signals will be managed by a program. These signals will be sent to the drone through the Internet encrypted. Additionally, all communication will be protected under a Virtual Private Network (VPN).*

## **RESUMEN**

*En este trabajo se recoge el diseño, implementación y fabricación, así como el funcionamiento y testeo, de un dron 5G y su infraestructura externa, cuya finalidad última será el vuelo y la recolección de datos.*

*Toda la lógica de comunicación y control del dron estará recogida en una aplicación dentro de un "smartphone", el cual se encargará de recoger datos de diferentes ámbitos: fotografía en tiempo real, datos iniciales de diferentes sensores y coordenadas de ubicación del dispositivo. Todos estos datos serán empaquetados y enviados a través de Internet a un sistema de control y visualización. Servicios individuales para la recogida de datos y el control del dron serán ejecutados en dicho sistema. Un servicio de backend tratará los datos recibidos, un servicio frontend los representará en tiempo real y una base de datos los almacenará. La interfaz de comunicación entre el smartphone y la electrónica del dron estará gestionada por un microcontrolador (MCU).*

*El usuario podrá manejar el dron gracias a un mando de control cuyas señales de control serán gestionadas por un programa. Estas serán enviadas cifradas al dron a través de Internet. Además, toda la comunicación estará protegida bajo una Virtual Private Network (VPN).*

# ÍNDICE GENERAL

<b>1 INTRODUCCIÓN.....</b>	<b>1</b>
1.1 OBJETIVO.....	2
<b>2 ANÁLISIS Y REQUISITOS .....</b>	<b>3</b>
2.1 ESTADO DEL ARTE .....	3
2.2 ANÁLISIS .....	4
2.2.1 <i>Análisis de la ciberseguridad</i> .....	5
2.3 REQUISITOS .....	6
<b>3 ARQUITECTURA .....</b>	<b>8</b>
3.1 CONTEXTO.....	8
3.2 ARQUITECTURA GENERAL .....	8
<b>4 DISEÑO.....</b>	<b>10</b>
4.1 DRON.....	10
4.1.1 <i>Hardware y electrónica</i> .....	10
4.1.2 <i>Software</i> .....	11
4.2 SISTEMA DE CONTROL Y VISUALIZACIÓN.....	12
4.2.1 <i>Hardware</i> .....	12
4.2.2 <i>Software</i> .....	13
4.3 INFRAESTRUCTURA DE COMUNICACIÓN .....	16
4.3.1 <i>Red móvil</i> .....	16
4.3.2 <i>Seguridad</i> .....	16
<b>5 IMPLEMENTACIÓN.....</b>	<b>19</b>
5.1 DRON.....	19
5.1.1 <i>Hardware y electrónica</i> .....	19
5.1.2 <i>Software</i> .....	27
5.2 SISTEMA DE CONTROL Y VISUALIZACIÓN.....	30
5.2.1 <i>Hardware</i> .....	30
5.2.2 <i>Software</i> .....	31
5.3 INFRAESTRUCTURA DE COMUNICACIÓN .....	36
5.3.1 <i>Red móvil</i> .....	36
5.3.2 <i>Seguridad</i> .....	36
<b>6 EXPERIMENTACIÓN Y PRUEBAS .....</b>	<b>38</b>
6.1 PRUEBA DE VUELO CONTROLADA .....	38

6.2 EVALUACIÓN DEL SCV .....	38
6.2.1 <i>Protocolo de trasmisión de imágenes</i> .....	39
6.3 RENDIMIENTO Y EVALUACIÓN DE LA COMUNICACIÓN.....	40
6.4 EVALUACIÓN FINAL DE LA EXPERIMENTACIÓN .....	41
<b>7 CONCLUSIONES .....</b>	<b>42</b>
7.1 COSTES .....	42
7.2 EVALUACIÓN DE LOS REQUISITOS CUMPLIDOS .....	43
7.3 LÍNEAS FUTURAS DE TRABAJO .....	44
<b>8 REFERENCIAS .....</b>	<b>48</b>
<b>9 ANEXOS .....</b>	<b>50</b>
9.1 IMÁGENES ADICIONALES .....	50
9.2 CÓDIGO: .....	55
9.3 MANUAL DE USO .....	56

## ÍNDICE DE FIGURAS

FIGURA 1. ESQUEMA GENERAL DEL SISTEMA.	8
FIGURA 2. ARQUITECTURA GENERAL DEL SISTEMA.	9
FIGURA 3. MOVIMIENTOS DEL DRON.	13
FIGURA 4. ARQUITECTURA DE PROTOCOLOS DE COMUNICACIÓN DEFINIDOS EN EL PROYECTO.	14
FIGURA 5. DIAGRAMA DE PINES DE LA PLACA RASPBERRY PI PICO.	20
FIGURA 6. ROTORES 2212 920KV CW BRUSHLESS MOTOR	21
FIGURA 7. ESC DE 30A UTILIZADO.	21
FIGURA 8. ARMAZÓN F450 (A) Y HÉLICES (B).	22
FIGURA 9. BATERÍA ZEEE 3S LIPO BATTERY DE 3200MAH.	23
FIGURA 10. DIAGRAMA DE CONEXIONES ELÉCTRICAS EN EL DRON.	24
FIGURA 11. VISTA SUPERIOR DEL RESULTADO DEL DRON.	25
FIGURA 12. VISTA FRONTAL DELANTERA DEL DRON. BATERÍA Y MÓVIL.	25
FIGURA 13. VISTA FRONTAL TRASERA DRON. MÓVIL Y MCU.	26
FIGURA 14. VISTA INFERIOR DEL RESULTADO DEL DRON.	26
FIGURA 15. DIAGRAMA INTERNO DE LA APLICACIÓN KOTLIN.	28
FIGURA 16. MANDO DE CONTROL Y MAPEADO DE COMANDOS SOBREPUESTO.	31
FIGURA 17. DIAGRAMA DE LA ARQUITECTURA INTERNA DEL SCV.	32
FIGURA 18. FLUJO INTERNO DEL SISTEMA DE VISUALIZACIÓN.	33
FIGURA 19. ESTRUCTURA DEL MENSAJE DE CONTROL ANTES DE APLICAR CIFRADO AES-128.	37
FIGURA 20. CAPTURA DE PANTALLA DE LA APLICACIÓN MÓVIL TRASMITIENDO IMAGEN.	39
FIGURA 21. RESULTADO DE CABLEADO Y CONEXIONES FÍSICAS.	50
FIGURA 22. LATERAL DE DRON E INTERRUPTOR.	50
FIGURA 23. RESULTADO DE CONEXIÓN ROTOR-ESC.	51
FIGURA 24. VISUALIZACIÓN DE IMAGEN EN EL SCV.	51
FIGURA 25. SISTEMA DE VISUALIZACIÓN DE DATOS.	52
FIGURA 26. DATOS ORDENADOS EN LOCALIZACIÓN Y COMPONENTES DE ACELERÓMETROS EN <i>MONGODB</i> .	53
FIGURA 27. ACCESO A LOS DATOS DE LAS COMPONENTES DE LOS ACELERÓMETROS EN LA BASE DE DATOS.	53
FIGURA 28. COMPARACIÓN PESO DE LAS IMÁGENES DEPENDIENDO DEL COLOR.	54
FIGURA 29. APLICACIONES EN EL TELÉFONO MÓVIL DE ESTE PROYECTO.	54

## ÍNDICE DE TABLAS

TABLA 1. RESULTADO DE LA CAPTURA DE <i>WIRESHARK</i> .....	40
TABLA 2. COSTES FINALES DEL DRON. ....	42
TABLA 3. TABLA DE RESULTADOS DE LOS REQUISITOS DEL PROYECTO .....	44

## 1 INTRODUCCIÓN

En la actualidad, el uso de drones se ha extendido rápidamente, y son habitualmente utilizados en sectores como la agricultura, la vigilancia, la entrega de paquetes, emergencias, el cine.... Sin embargo, la escasez de drones con tecnología 5G ha limitado muchas veces estas aplicaciones.

La conectividad 5G puede ofrecer mayor velocidad de transmisión de datos y menor latencia, mejorando significativamente el rendimiento y control de los drones. Además, los drones operados con tecnologías de telefonía móvil solucionan el límite de rango de cobertura del radiocontrol de señal *Wi-Fi*, permitiendo que estos operen en áreas más amplias y remotas sin perder la conexión. Incluso, con el uso de tecnologías móviles un dron podría cambiar de tecnología 3G, 4G o 5G dependiendo de la calidad de la señal de cada una de estas tecnologías, creando un sistema de control resiliente y de cobertura de vuelo aún más amplia.

Sin embargo, el precio elevado de los drones equipados con 5G y la falta de infraestructura adecuada en muchas regiones dificultan su adopción masiva. Estos factores combinados ralentizan el avance y la integración completa de los drones en diversos sectores, lo que subraya la necesidad de mayores inversiones y desarrollo en esta combinación de tecnologías, drones y 5G.

Cabe destacar que Estados Unidos y China, como principales potencias tecnológicas mundiales, lideran el desarrollo de todas estas tecnologías avanzadas, siendo estos países aquellos que poseen más cantidad de drones 5G. Aunque, si bien 5G y 4G se caracterizan por ser estándares regidos por *3GPP (3rd Generation Partnership Project)* [1] y ser así protocolos de interoperabilidad y estandarización global, existe la disyuntiva de que la banda de frecuencias para las cuales ha sido fabricado un dron estadounidense o chino no esté siendo desplegada en España o viceversa. Para ilustrar esta casuística, en España para 5G se utilizan en banda baja 700 MHz, en banda media 3.5 GHz y en banda alta 26 GHz; y por otro lado en Estados Unidos se utiliza en banda baja 600 MHz, en banda media 2.5 GHz y en banda alta 24 GHz y 39 GHz, estas dos últimas bandas de frecuencias no están disponibles en España. Es por ello por lo que existe una necesidad de crear un dron independiente de su origen de fabricación y que este pueda volar en

cualquier parte del planeta utilizando 5G, siempre que exista esta tecnología, u otra (3G o 4G) en su defecto.

Como dato curioso, en el mercado tecnológico de cara al público, sobre todo en tiendas generalistas on-line que cubren todo el sector mercantil, no se pueden obtener drones con tecnología de telefonía 5G. Si en alguna tienda on-line se busca el término "Dron 5G" aparecerán drones y la palabra 5G acompañará su nombre, pero dicha tecnología se refiere a un malentendido comercial. Dicho 5G se refiere a 5 GHz, la banda de frecuencia de *Wi-Fi* que pueden utilizar citados drones.

## **1.1 Objetivo**

El objetivo central del proyecto es el diseño, implementación y creación de un dron 5G de bajo coste. Este objetivo principal, se desglosa en los siguientes subobjetivos relativos a la fabricación y control.

En cuanto a la fabricación, el dron deberá contar con componentes totalmente modulares, idóneos funcionalmente pero fácilmente configurables y escalables. El dron volará de manera autónoma, bajo el control remoto de una persona, ya sea en la proximidad del aparato o a varios, incluso cientos de kilómetros del propio dron. A su vez, para que esta última tarea pueda ser posible, se deberá diseñar y llevar a cabo una infraestructura de comunicación segura y resiliente entre el usuario y el dron, capaz de permitir una experiencia inmersiva en tiempo real. Para ello el dron deberá proporcionar a su operador la mayor información posible del entorno haciendo así más fácil la acción del manejo. El dron recopilará y enviará datos importantes de su vuelo, como puede ser: fotografía en tiempo real, datos procedentes de sensores y ubicación. Estos datos serán manejados, mostrados en tiempo real y almacenados para que el usuario disponga de toda la información necesaria para el control del aparato.

## 2 ANÁLISIS Y REQUISITOS

### 2.1 Estado del arte

Hoy en día la mayoría de los drones que se han implantado son aquellos controlados por una persona en las proximidades del dron mediante un mando de control que usa una conexión inalámbrica de radiofrecuencia, generalmente en las bandas de 2.4 GHz o 5 GHz.

En la actualidad existen pocas empresas con un producto comercial parecido al contemplado en este trabajo, un dron 5G. Pero incluso estas empresas distan en ciertos aspectos respecto a la implementación y el camino seguido para llevar a cabo la idea. Por ejemplo, dos empresas que colaboran para sacar dicha novedosa solución son *Qualcomm* [2] y *Modal-AI* [3]. En el caso de *Qualcomm*, su producto se llama "*Qualcomm Flight™ RB5 5G Platform*" y su precio está establecido en unos 4,199.99\$ [4] aunque esta cifra puede aumentar dependiendo de las características que se configuren. Este dron es un producto de código cerrado y el software de control está ligado con su plataforma empresarial bajo licencia comercial.

Por otro lado, estas dos empresas ofrecen a su vez un producto "más abierto" y configurable llamado "*VOXL® 2 AI & 5G Development Drone - Sentinel*" [5] cuya idea va dirigida para los integradores o desarrolladores que buscan crear su propio dron 5G. El dron parte de una cifra de coste de 5,129.99\$ [5], la transmisión de comandos de control al dron funciona mediante RF y la transmisión de datos del dron a control (vídeo y sensores) funciona mediante 5G, es decir es un dron de tecnología de comunicación híbrida, y no puramente 5G. Tanto este producto como el anterior están destinados a ser vendidos en Estados Unidos. Esto puede ser una problemática ya que las bandas de frecuencias destinadas a 5G pueden variar entre países. Las bandas en donde puede operar este dron pueden verse en la documentación técnica de dicho producto [6].

Otra empresa en donde se puede encontrar un dispositivo volador con tecnología 5G es "*Beyond Vision*" [7], aunque en este caso es un híbrido de aeroplano y dron llamado "*VTOne*" [8]. Esta empresa ofrece servicios de fotografía con drones. Además, posee en su página un enlace para su adquisición, pero, a día de hoy, dicho enlace no funciona.

Existen dos grandes gigantes en el sector comercial de los drones, *Dji* [9] y *Parrot* [10]. *Dji* utiliza protocolos propios auto-desarrollados llamados *OcuSync* [11] y *Lightbridge* [12], además de *Wi-Fi*, para las comunicaciones usuario-dron. Estos protocolos utilizan la banda de frecuencia *ISM (Industrial, Scientific and Medical)* de 5 y 2.4 GHz y están limitados por la cobertura del mando de control. *Dji* ofrece como módulo adicional un usb modem llamado “*Dji Cellular Dongle*” [13] capaz de conectarse mediante *LTE (Long Term Evolution o 4G)* a las antenas de telefonía. Este producto solo está disponible en China. *Parrot* dispone de un producto similar, aunque lo incorpora dentro del propio dron en este caso. Se utiliza también la tecnología *LTE*, pero de manera diferente. El dron funciona con tecnología *Wi-Fi*, pero si se habilita una opción de transmisión de vídeo mejorada todo el flujo de vídeo se enviará por 4G. Este producto se llama “*Anafi Ai*” [14], su precio es de 5.000\$ y solo es posible adquirirlo en Estados Unidos [15].

Retomando el dron “*Sentinel*” previamente mencionado, que es el que más se ajusta a la idea de este trabajo, cabe destacar que dicho dron opera en la banda n78, la más utilizada en 5G en España, por lo que no habría inconvenientes, montando así para dicha banda la tarjeta de comunicación “*5G RM50xQ series*” [16]. Pero ante esto surge un problema. Este dron no puede operar con otra tecnología, como 4G o 3G. Por lo tanto, fuera del rango de 5G, vuelve a ser un dron limitado por cobertura, similar a los drones controlados mediante *Wi-Fi*. Estos drones no pueden cambiar de tecnología y de bandas tan fácilmente como lo haría un teléfono móvil.

## **2.2 Análisis**

Dada la evolución y el estado actual de la tecnología respecto a drones 5G, se han identificado varias necesidades críticas que este proyecto debe abordar:

- Construcción de un dron capaz de volar sin límites de cobertura de radiocontrol usando tecnologías de telefonía móvil (a poder ser 5G). Gracias a ello se podrá garantizar vuelos sin restricciones de cobertura, independientemente del país o la región, siempre y cuando haya cobertura móvil.

- Modularidad: capacidad de mejorar o escalar componentes del dron o de la infraestructura sin afectar otras partes. Esto es esencial para la evolución continua y correcta del proyecto.
- Datos en tiempo real: el dron deberá ser capaz de transmitir datos en tiempo real de su entorno y posición, así como a la vez recibir comandos de control y ejecutar dichos comandos. A su vez este debe ser capaz de moverse en todas las direcciones del espacio como haría cualquier dron.
- Manejo de los datos: es necesario que tanto el control como la recepción de los datos sean gestionados en una plataforma software. Los datos se verán a partir de una interfaz gráfica y a su vez será necesario que sean almacenados para su posterior análisis.
- Coste asequible: es importante que el coste del proyecto no supere los 500€, haciendo que esta solución sea accesible y competitiva en el mercado.
- Seguridad en la comunicación y salvaguardas: garantizar que la comunicación del dron esté protegida contra ataques de agentes externos es crucial para su operación segura y confiable. Además, será necesario definir medidas de salvaguarda para el caso de pérdida de conexión o de control (ver siguiente apartado 2.2.1 Análisis de la ciberseguridad).

### 2.2.1 Análisis de la ciberseguridad

Este proyecto se caracteriza por manejar un activo peligroso, ya que un dron no deja de ser un objeto volador que vuela a grandes velocidades e incluso al disponer de batería puede llegar a explotar en caso de accidente. Por ello hay que analizar posibles vulnerabilidades que pueden hacer de este activo, el dron, un elemento peligroso tanto en manos ajenas como en sí mismo. Estas posibles vulnerabilidades se pueden clasificar en 2 categorías: ciberseguridad y seguridad física:

Ciberseguridad:

1. Intercepción de la comunicación (también conocido en terminología de ciberseguridad como “*Man in the Middle*” o “*MitM*” [17]): un atacante podría interceptar los paquetes de datos enviados entre el dron y el sistema controlador, comprometiendo la integridad de la comunicación.

2. Ataques de repetición (“*Replay Attacks*” [18]): un atacante podría capturar y reenviar paquetes cifrados, causando un comportamiento anómalo en el dron.
3. Conexión insegura a Internet: la comunicación entre el dron y los sistemas de control y visualización podría ser interceptada debido a la naturaleza insegura de Internet. Las comunicaciones más vulnerables son aquellas que no están cifradas.
4. Caída de los servicios o pérdida de conexión: podrían interrumpir la operación del dron, haciendo que este se comporte de manera incorrecta, incluso haciendo que pueda llegar a estrellarse.

Seguridad física:

1. Fallo Mecánico o pérdida de control: el dron podría experimentar fallos mecánicos o electrónicos, como puede ser la desconexión o perdida de algún componente. O situaciones de pérdida del control durante el vuelo por un ciberataque o caída de la conexión. Ante tales fallos es posible que el dron se estrelle y no pueda pararse de manera controlada.
2. Acceso no autorizado al dron: en caso de que el dron caiga en manos ajena, los datos y la operación podrían estar comprometidos.

### **2.3 Requisitos**

Para poder llevar a cabo el objetivo establecido en este trabajo, se han definido los siguientes requisitos que refinan las necesidades de coste, diseño, control, materiales, seguridad y tratamiento de los datos, que se han introducido en los apartados anteriores. Así, se establecen los siguientes requisitos:

- REQNAV-01: El dron debe ser capaz de volar a grandes distancias sin que el límite por cobertura sea un problema. Incluso debe ser capaz de volar en regiones alejadas del usuario.
- REQNAV-02: El dron debe ser capaz de volar y ser controlado en cualquier país del mundo independientemente de donde se haya fabricado.

- REQDAT-01: El dron debe ser capaz de recopilar la siguiente información: datos iniciales del movimiento del dron, imágenes del entorno del dron y localización de este.
- REQDAT-02: El usuario debe disponer en tiempo real de todos los datos recopilados por el dron.
- REQDAT-03: Toda la información recopilada por el dron deberá ser almacenada en una base de datos externa, para que pueda ser posteriormente analizada.
- REQCNT-01: El dron debe ser capaz de moverse en todas las direcciones del espacio.
- REQCNT-02: El usuario debe poder manejar el dron a través de una interfaz de control que interprete sus órdenes y las comunique al sistema de control del dron.
- REQCOM-01: La comunicación debe ser segura entre el dron y el usuario final, de tal manera que ningún agente externo, ya sea una persona o un evento inesperado, afecte al funcionamiento del dron.
- REQCOM-02: El dron debe incorporar salvaguardas de seguridad para el caso en que se pierda la conexión.
- REQCOM-03: El dron deberá ser capaz de cambiar dinámicamente entre diferentes tecnologías de comunicación (5G, 4G, 3G). De esta manera, en caso de que la conexión óptima no esté disponible, podrá cambiar a la siguiente mejor opción sin perder la comunicación, extendiendo así su área de vuelo
- REQDIS-01: El proyecto entero no debe superar de coste los 500€.
- REQDIS-02: El proyecto debe ser completamente modular, pudiendo mejorar o escalar cualquier parte de este sin que esto afecte a ninguna de las otras partes.

## 3 ARQUITECTURA

### Contexto

La Figura 1, una ilustración general y sin detalles técnicos, muestra lo que se persigue en este proyecto. En azul se muestra un entorno al aire libre y en verde un entorno doméstico, separados (a pocos o cientos de kilómetros) pero conectados a través de Internet.

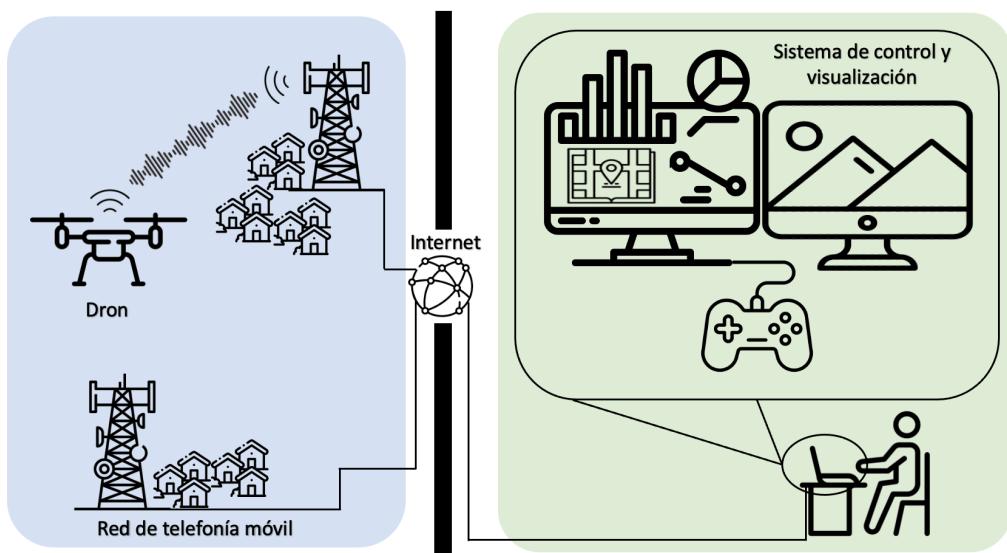


Figura 1. Esquema general del sistema.

Así, se puede distinguir los siguientes componentes clave: el dron, y el sistema de control y visualización (SCV). El dron y el SCV se comunican a través de internet y la red de telefonía de datos móviles.

### 3.1 Arquitectura general

Una arquitectura bien definida facilita el desarrollo y mantenimiento de un proyecto. En la Figura 2 se puede observar la arquitectura general del sistema, desde el dron, y sus componentes internos, hasta el SCV y sus partes diferenciadas. El establecimiento de una arquitectura robusta no solo ayudará a comprender el proyecto, sino que garantiza su posterior escalabilidad, mejora de rendimiento y solidez.

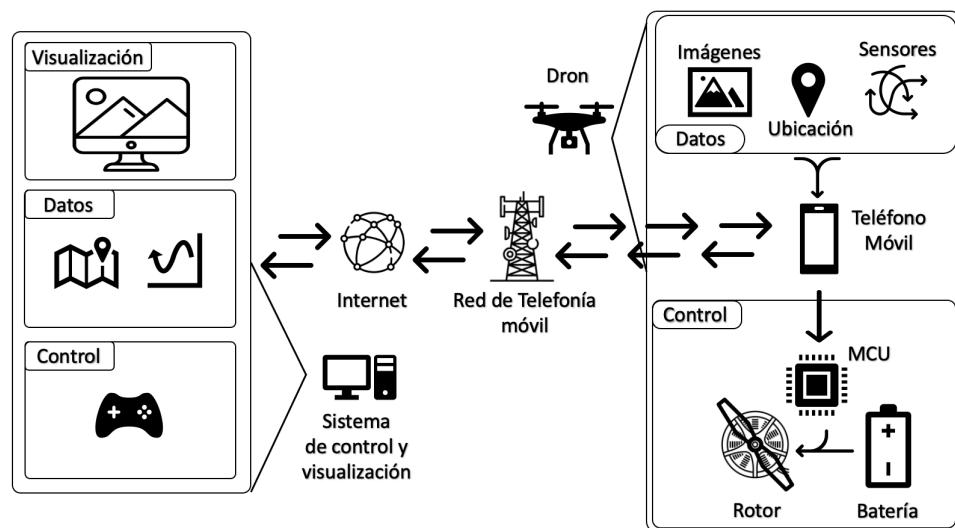


Figura 2. Arquitectura general del sistema.

## 4 DISEÑO

Para poder cumplir los requisitos anteriormente mencionados, primero se deben definir una serie de estrategias de diseño. Para ello se estructurará el proyecto y el documento en base a los tres componentes claves del mismo: dron, SCV e infraestructura de comunicación. De tal manera que se abordará y presentará cada una de ellas de manera independiente.

A lo largo de este apartado (sección 4) y el siguiente (sección 5) se abordará el diseño e implementación, respectivamente, de cada una de las estructuras mencionadas. Así mismo, cada una de ellas se ha dividido en subapartados siendo estos hardware y software para el caso de Dron y el caso de SCV, y red móvil y ciberseguridad para el caso de la infraestructura de comunicación.

### 4.1 Dron

#### 4.1.1 Hardware y electrónica

Para gestionar la parte lógica del dron es necesario definir un dispositivo compacto con conexión 5G, o 4G o 3G en su defecto (REQCOM-03), sensores internos, antena GPS incorporada, cámara fotográfica de alta resolución (REQDAT-01) y que dicho dispositivo sea fácilmente programable. En este sentido, dado su amplio uso y diferentes costes (REQDIS-01), se ha optado por escoger un teléfono móvil que, conectado a un microcontrolador (*MCU*), permita gestionar el hardware del dispositivo. Este último aportará los pines, y el control de señales y hardware, pudiendo ser controlado a través de código.

Para poder volar, el dron necesita de 4 rotores. Cada rotor necesita una hélice de un tamaño proporcional a su potencia, y un controlador electrónico de velocidad (*Electronic Speed Controller, ESC*) que será en encargado de administrar la energía que proviene de la batería para que el rotor gire a más o menos velocidad, dependiendo de las órdenes del microcontrolador. Además, para sustentar los rotores se necesita un armazón y para dotar de energía a estos se necesita una batería.

Para la elección de la batería hay que tener en cuenta varios factores: uno de ellos es su capacidad, cuanta más capacidad, más tiempo de vuelo puede ofrecer; otro es el peso, una batería demasiado pesada podría afectar al vuelo del dron; y otro es la tasa de descarga, que es la velocidad a la que la batería puede liberar su energía almacenada. Esta tasa de descarga se calcula con la siguiente fórmula (Ecuación 1):

$$\text{Tasa de descarga} \rightarrow C = \frac{n^o \text{ de } ESCs * I_{\max ESC}}{\text{Capacidad}_{Batería}}$$

**Ecuación 1. Cálculo de Tasa de descarga de batería.**

Adicionalmente, como medida de seguridad, se requerirá de un interruptor físico conectado a los cables de la batería para que, en caso de emergencia, cesar la entrega de energía a los motores y estos se paren de inmediato (REQCOM-02).

#### 4.1.2 Software

La elección de un teléfono móvil facilita el desarrollo del software necesario para el control del dron, ya que se implementará como una aplicación para dicho dispositivo. Esta será capaz de interactuar con los sensores internos del dispositivo, así como con sus cámaras o la antena *GPS* (*Global Positioning System*). Hay que tener en cuenta que este software deberá ser capaz también de establecer conexiones *UDP* (*User Datagram Protocol*) y *TCP* (*Transmission Control Protocol*) con servicios remotos. Además, se tratará de un software de ejecución paralela en lugar de secuencial, ya que muchas tareas deberán realizarse de manera simultánea. Por ejemplo, la obtención de los datos de sensores y ubicación, la captura de fotos, el envío de esta información y la recepción de comandos, todo ello de manera concurrente.

Este software deberá manejar grandes conjuntos de datos y enviarlos a través de Internet. Además, deberá ser capaz de enviar un flujo de imágenes continuo simulando un efecto de vídeo. Las imágenes recibidas deben ser de buena calidad, lo suficiente para distinguir claramente objetos y personas, sin necesidad de alcanzar una resolución extremadamente alta.

Con respecto al microcontrolador caben dos opciones en cuanto a su desarrollo software; la primera de ellas consiste en desarrollar un código interno en el microcontrolador, que se ejecutará continuamente esperando recibir señales de otro dispositivo (el teléfono móvil). Un ejemplo de señal sería “SUBIR” y el microcontrolador enviaría señales eléctricas a los controladores de velocidad (*ESCs*) de los rotores para que dejen pasar energía al rotor y este impulse el dron hacia arriba. Este método implica que el microcontrolador realizará tareas de monitoreo o espera activa, procesando las señales recibidas según se van presentando.

El segundo enfoque es la inyección dinámica de código desde otro el dispositivo (el teléfono móvil), en la cual el microcontrolador ejecuta las instrucciones a medida que son transmitidas. Este método permite una mayor flexibilidad y adaptabilidad, ya que el código puede modificarse o actualizarse en tiempo real sin necesidad de reprogramar el microcontrolador de forma manual.

En este proyecto se utilizará este último enfoque para tener un control más centralizado de todo el código.

## **4.2 Sistema de control y visualización**

### **4.2.1 Hardware**

En cuanto al hardware necesario para el control del dron y la recepción de la información que este recopila, se requiere un equipo con características específicas. Este equipo debe ser capaz de ejecutar varios programas simultáneamente. Además, necesita tener una interfaz serie para conectar un mando de control (REQCONT-02). Y también es esencial que disponga de conexión a Internet. Dicha conexión debe ser rápida y estable (REQCOM-01), gigabit ethernet, por ejemplo, evitando latencias o pérdidas (más frecuentes con *Wi-Fi*).

Los movimientos que debe realizar un dron en el espacio (REQCONT-01) pueden categorizarse en lineales y rotacionales. Los desplazamientos lineales se pueden hacer en los 3 ejes del espacio: eje X (movimiento hacia delante y hacia atrás), eje Y (movimiento izquierda y derecha) y el eje Z (movimiento hacia arriba y hacia abajo). El

movimiento rotacional que todos los drones incorporan es el conocido como “Yaw”, que consiste en una rotación sobre su eje vertical, cambiando la dirección de la ”nariz” del dron. Estos movimientos se pueden observar en la Figura 3, vista superior (a) y vista frontal (b).

Para el control del dron se necesitará un mando de control conectado al ordenador mediante cable serial. Este mando debe tener joysticks capaces de realizar todos los movimientos que el dron pueda hacer en todas las direcciones del espacio (REQCONT-01). Además, este mando deberá incorporar como mínimo 3 botones, uno para iniciar el dron, otro para pausarlo y otro adicional para cambiar de cámara.

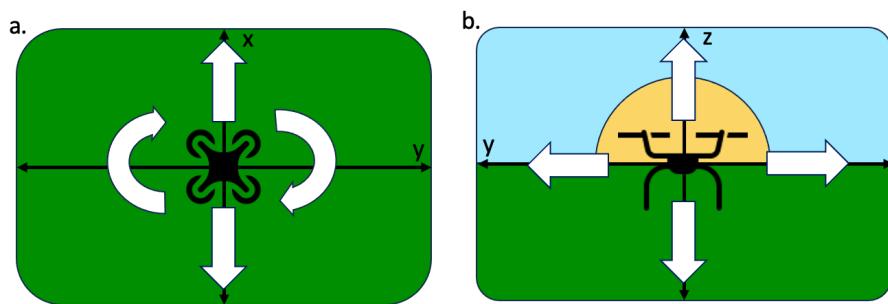


Figura 3. Movimientos del dron.

#### 4.2.2 Software

El proyecto contará con tres protocolos de comunicación. Según se muestra en la Figura 4 existirá un protocolo de control (gestionará el control del dron), un protocolo de transmisión de imágenes (gestionará el envío y recepción de imágenes del dron hacia el SCV) y un protocolo de transmisión de datos (gestionará el envío de datos de los sensores y la ubicación del dron hacia el SCV). Cada uno de ellos cumple con el requisito de modularidad del proyecto (REQDIS-02). Así mismo, cada uno de los protocolos de comunicación establecidos contará con un programa encargado de administrarlo con un servicio que estará escuchando en un puerto diferente dependiendo del protocolo de comunicación.

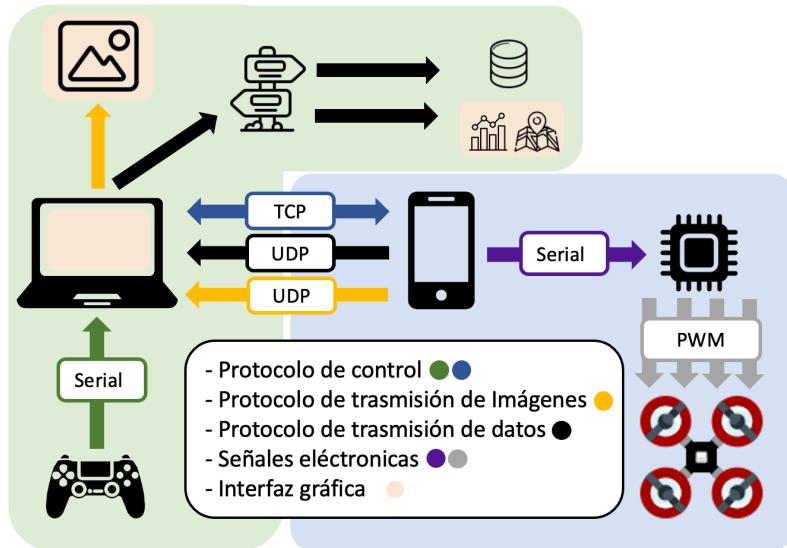


Figura 4. Arquitectura de protocolos de comunicación definidos en el proyecto.

**Protocolo de trasmisión de Datos:** se encargará de trasmisir todos los datos de los sensores, así como la localización del dron. Este protocolo de comunicación será unidireccional, del dron al SCV. Este tiene un papel importante en el requisito REQDAT-02 (datos en tiempo real) porque se necesita que su trasmisión sea lo más rápida posible. Para ello se ha usado *UDP*, ya que, al ser un protocolo no orientado a conexión, la rapidez de este es mayor. Además, la perdida de alguno de los paquetes en la recepción sería importante, pero no sumamente grave, por lo que el sistema puede afrontar ligeras pérdidas de paquetes. El programa encargado de este protocolo se servirá de un servidor *UDP* y cumplirá la función de recepción de datos, almacenamiento y representación de estos. Para ello, su arquitectura está basada en un modelo de 3 capas o niveles [19], con un nivel de presentación, un nivel de aplicación y un nivel de datos.

- *Nivel de Presentación:* será la interfaz gráfica en donde se muestren los datos en tiempo real. Deberá contener dos apartados principales, uno para mostrar los datos de los sensores en forma de gráficas y otro en donde mostrar la ubicación del dron en un mapa.
- *Nivel de Aplicación:* Es el núcleo de la aplicación, en donde se ejecuta el servidor y quien se encargará de etiquetar y manejar los datos recibidos.
- *Nivel de datos:* a veces denominado también nivel de base de datos, es donde se almacenarán los datos. Este puede ser un sistema de base de datos relacional

(*SQL*) o no relacional (*NoSQL*). Para este proyecto se usará un esquema no relacional ya que ofrece una mejor escalabilidad frente a un modelo *SQL* y una mejor flexibilidad en cuanto a esquema y estructura de los datos.

**Protocolo de trasmisión de imágenes:** Las imágenes trasmitidas deberán ser de buena calidad, lo suficiente para distinguir claramente objetos y personas, sin necesidad de alcanzar una resolución extremadamente alta. Este protocolo será también unidireccional, desde el dron hacia el SCV. El software del teléfono móvil se encargará de tomar fotografías continuamente. El tamaño de cada una de las fotografías es demasiado grande como para poder ser trasmitidas en un solo paquete a través de internet. Por ello, simultáneamente, cada fotografía será fragmentada y cada uno de los fragmentos será etiquetado con un identificador de fotografía y otro identificador de fragmento. Cada uno de los fragmentos será enviado al SCV a través de *UDP* por la misma razón comentada en el protocolo de comunicación de datos. El programa deberá contar con un servidor *UDP*. A su vez, este programa contará con mecanismos de ejecución paralela, así como tareas internas para manejar los paquetes recibidos. Cada uno de los paquetes estará etiquetado mediante identificadores de fragmento e identificadores de fotografía. Posteriormente se recompondrá cada una de las imágenes lo antes posible y se dispondrá de una interfaz gráfica para mostrarla al usuario en tiempo real (REQDAT-02). Para que una trasmisión de fotografías se pueda considerar un vídeo de calidad tendrían que aparecer alrededor de 30 fotografías por segundo (*fps*).

**Protocolo de control de dron:** Esta parte también será unidireccional, pero en este caso del SCV al dron. Se encargará de enviar todos los comandos recopilados a través de la interfaz del mando de control hacia el dron (REQCONT-02), pero con una ligera diferencia con respecto a los protocolos de bajo nivel usados en los anteriores protocolos, ya que en este caso se utilizará *TCP*. Si bien es verdad que los comandos deben llegar con la mayor rapidez posible al dron, es más importante asegurar que lleguen correctamente y en orden. Estas características son propias de *TCP* al ser un protocolo orientado a la conexión. Por tanto, este programa tendrá un servidor *TCP* y a su vez se encargará de escuchar las órdenes del mando de control para redirigirlas hacia el dron. Se añadirá una capa de seguridad adicional en la trasmisión consistente en el cifrado de los comandos. Esto aportará seguridad y privacidad entre las dos partes, dron y usuario, haciendo que

sea inaccesible para cualquier agente externo al sistema que no esté autorizado (REQCOM-01). El cifrado contribuye a que haya confidencialidad, integridad de los datos y autenticación [20].

### **4.3 Infraestructura de Comunicación**

En este apartado se describirá la infraestructura de comunicación subyacente utilizada para la conexión entre el dron y los sistemas de control y visualización de la información.

#### *4.3.1 Red móvil*

La comunicación entre el dron y los sistemas se realizará a través de Internet. El SCV estará conectado a Internet mediante una conexión *Ethernet* de alta velocidad (REQCOM-01) proporcionada por el proveedor de servicios de Internet (*Internet Service Provider, ISP*). El teléfono móvil estará conectado a Internet a través de una red de telefonía móvil a través de una antena de telefonía cercana gracias a su tarjeta *SIM* (*Subscriber Identity Module*). El punto positivo de usar una tecnología tan desarrollada y totalmente opaca para el usuario final es que no hay que preocuparse por el diseño de la red telefonía móvil o de la conexión a esta. La única necesidad es el alta del servicio y el operador móvil se encargará de gestionar todos los detalles de bajo nivel, inclusive dos grandes necesidades para el proyecto: el cambio de antena de telefonía para no depender de coberturas limitadas (REQNAV-01) y el cambio de tecnología (de 5G a 4G o 3G) en el caso de que una superior no esté disponible (REQCOM-03) y así no quedarse sin señal y, por consiguiente, sin conexión a Internet, cuyo resultado sería la pérdida del dron.

#### *4.3.2 Seguridad*

Este proyecto se caracteriza por manejar un activo peligroso, como se vio en el análisis de la ciberseguridad (sección 2.2.1), por ello hay que diseñar una serie de medidas de salvaguarda y otras reactivas para proteger tanto al dron como al entorno en donde se vuela (REQCOM-01 y REQCOM-02). Estas medidas de seguridad se pueden clasificar en 2 categorías: ciberseguridad y seguridad física. De cada una de ellas se explicará aquí la solución adoptada:

## Ciberseguridad:

1. *Intercepción de la comunicación* (también conocido en terminología de ciberseguridad como “*Man in the Middle*” o “*MitM*”): Para mitigar este problema se utilizará un cifrado en la comunicación. Se utilizará cifrado simétrico para el protocolo de control debido a la confianza entre emisor y receptor [17], proporcionando dos capas de protección: los datos serán indistinguibles y no maleables. Este tipo de cifrado es más rápido y eficiente comparado con el cifrado asimétrico, ideal para las necesidades de control en tiempo real.
2. *Ataques de repetición* (“*Replay Attacks*” [18]): Para resolver este inconveniente se introdujeron marcas temporales en los comandos. El emisor incorporará una marca temporal a cada comando enviado, y el receptor rechazará cualquier mensaje anterior a dicho tiempo, evitando la repetición de comandos.
3. *Conexión insegura a Internet*: Para solucionar este problema se utilizó una red privada virtual (*VPN*). Esto asegurará que toda la comunicación se realice a través de una conexión segura y cifrada, protegiendo las imágenes y los datos transmitidos.
4. *Caída de los servicios o pérdida de conexión*: Para ello se diseñaron servicios y clientes resilientes e independientes. Los servidores se reiniciarán automáticamente en caso de fallo, y los clientes intentarán reconectarse repetidamente hasta restablecer la conexión. El modularidad del sistema (REQDIS-02) asegura que fallos en una parte no afecten al resto del proyecto, permitiendo medidas reactivas para mantener el dron operativo.

## Seguridad física:

1. *Fallo Mecánico o pérdida de control*: siempre y cuando se tenga acceso físico al dron, se incorporará un interruptor de apagado de emergencia en este. Este interruptor estará ubicado lejos de las aspas y será fácilmente accesible en caso de emergencia.

2. *Acceso no autorizado al dron:* Debido a que el núcleo del dron es el teléfono móvil se optó por usar el bloqueo de este con clave de acceso. Esto garantiza que el dispositivo esté protegido contra accesos no autorizados.

Estas medidas combinadas garantizan un alto nivel de seguridad tanto en la comunicación como en el funcionamiento del dron, protegiendo el proyecto de una amplia gama de vulnerabilidades en varios ámbitos.

## 5 IMPLEMENTACIÓN

### 5.1 Dron

#### 5.1.1 Hardware y electrónica

Con respecto al núcleo lógico del dron, es decir, el teléfono móvil, se tuvieron en cuenta varios detalles para su elección. En primer lugar, el dispositivo tiene que proporcionar la tecnología *OTG (On-The-Go)* para poder conectar un microcontrolador. Esto significa que los móviles que lo posean pueden ejercer de anfitriones sobre dispositivos periféricos que se conecten en su puerto de entrada (normalmente el de carga). Esta característica es algo que suelen traer hoy en día muchos de ellos, pero no todos. En segundo lugar, el dispositivo debe incorporar un sistema operativo *Android*. Si bien *iOS* podría utilizarse, este último es un sistema operativo mucho más cerrado que *Android* a la hora del desarrollo de aplicaciones y su curva de aprendizaje es más lenta.

Las otras características consideradas para la elección del teléfono móvil fueron: precio, peso, dimensiones, características de la cámara interna y externa, procesador, RAM, batería y memoria de almacenamiento. Se priorizó encontrar un equilibrio entre precio y rendimiento, con un enfoque particular en la calidad de la cámara y la capacidad de procesamiento.

Primero, se realizó una selección de teléfonos en el rango de precio entre 200 y 300 € que ofrecieran las mejores prestaciones. Tras evaluar una lista de siete dispositivos, que incluía seis teléfonos de la marca *Samsung* y uno de la marca *OPPO*, finalmente se eligió el teléfono *Samsung Galaxy A33 5G*. Este teléfono móvil cuenta con las siguientes características:

- Precio: 241,86 €
- Peso: 186 gramos
- Dimensiones: 159,7 x 74 x 8,1 mm
- Dimensión de la pantalla: 6,4 pulgadas
- Cámara Externa: Principal: 48 MP f/1.8, OIS + Angular: 8 MP f/2.2 + Profundidad: 2 MP f/2.4 + Macro: 5 MP f/2.4

- Cámara Interna: 13 MP f/2.2
- Procesador: Exynos 1280 GPU Mali-G68
- RAM: 6 GB
- Memoria: 128 GB
- Batería: 5.000 mAh
- Sistema Operativo: Android 12

El microcontrolador escogido es la placa *Raspberry Pi Pico®* por ser la mejor opción hoy en día en el mercado en cuanto a precio y características técnicas (como pines e interfaces de comunicación). Esta tarjeta diseñada por *Raspberry Pi Foundation* [21] cuenta con un procesador *RP2040* (con un núcleo *ARM Cortex M0+* de doble núcleo de 133 MHz), una memoria de 264 KB de *SRAM* y 2 MB de memoria *flash*, y cuenta con 26 pines *GPIO* multifuncionales que admiten interfaces como *SPI*, *I2C* y *UART* (Figura 5). Aunque es programable en varios lenguajes de programación para este proyecto se ha elegido *Micropython* por su similitud con *Python* y su facilidad. La placa *Raspberry Pi Pico®* ha tenido un coste de 5,04 €.

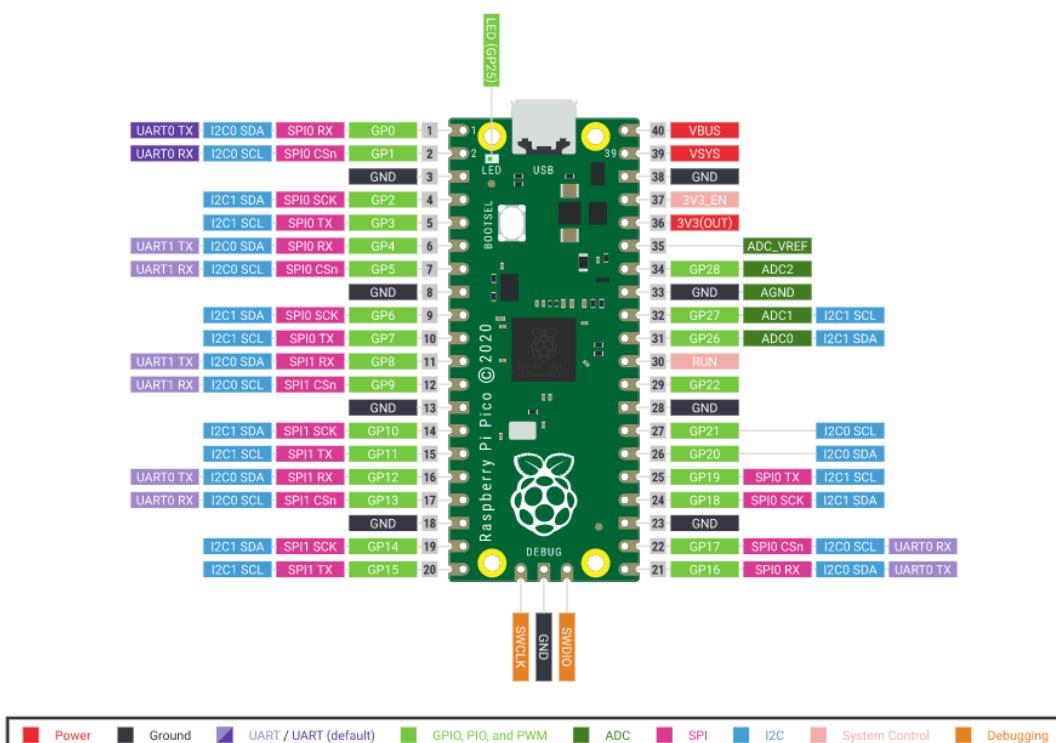


Figura 5. Diagrama de pines de la placa Raspberry Pi Pico

En cuanto a materiales físicos como rotores, controladores de rotores (*ESCs*), hélices y armazón se determinó utilizar un kit con los componentes preseleccionados para no cometer errores de incompatibilidades entre módulos.

El nombre de los rotores es *2212 920KV CW Brushless motor* (Figura 6). 2212 son las dimensiones del motor 22x12 mm. 920 KV significa que el motor está diseñado para girar a 920 RPM por cada voltio. “*CW*” significa “*Clockwise*” (en sentido horario), indica la dirección preferida de rotación del motor. “*Brushless Motor*” significa motor sin escobillas (“*Brushless*”). El color del tornillo indica su dirección de giro de cara a colocar las hélices.



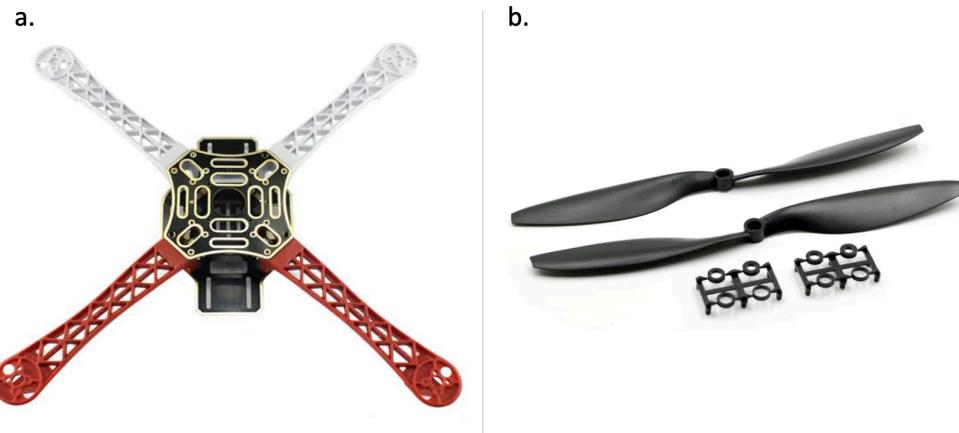
Figura 6. Rotores 2212 920KV CW Brushless motor

Los *ESC* (*Electronic Speed Controller*) son de 30A (Figura 7), diseñado para proporcionar una corriente continua de 30A, con una salida BEC de 2A continua y picos instantáneos de hasta 3A. Incorpora un control mediante *PWM* de 8 KHz y está diseñado para operar en un rango de tensión de 4V a 16V. Sus dimensiones son 45x24x11 mm y tiene un peso de 25 gramos, incluyendo cables. Tiene 2 cables gruesos que son para la alimentación proveniente de la batería y 3 cables finos para señales del microcontrolador.



Figura 7. ESC de 30A utilizado.

El armazón es conocido como *Frame F450* (parte izquierda de la Figura 8) y es ampliamente utilizado en el mundo del desarrollo de drones por componentes. Y las hélices (parte derecha de la Figura 8) miden 25 cm de largo y 2,5 cm máximo de anchura.



**Figura 8. Armazón F450 (a) y hélices (b).**

Para la batería se realizó el cálculo de su tasa de descarga siguiendo con la fórmula mencionada en la sección 4.1.1 de este trabajo (Ecuación 1) y teniendo en cuenta los componentes que ya se poseían, se determinó que se necesitaba para una batería de 3200mAh al menos una tasa de descarga superior o igual a 37,5C.

La batería escogida es la *3S Lipo Battery* de Zeee, de 3200mAh (Figura 9). La batería está fabricada con polímero de litio y posee un voltaje total de 11,1 V, con una configuración de *3S1P*, lo que significa que se compone de tres celdas en serie. Cada celda opera en un rango de voltaje de 3,2 a 4,2 V. La capacidad de la batería es de 3200 mAh, ofreciendo una tasa de descarga de 50C, superior a una tasa de 37,5C necesaria para este proyecto. Además, está equipada con un conector “*Deans T*” para conexiones seguras y eficientes.



Figura 9. Batería Zeee 3S Lipo Battery de 3200mAh.

Una parte muy importante del dron es la conexión entre componentes y cableado. El cableado consta de 4 principales conexiones:

- *ESC - Rotores:* En esta conexión el *ESC* proporciona energía eléctrica trifásica al motor dependiendo de la señal recibida proveniente del microcontrolador (Figura 23 del anexo). Cuanta más energía se deje pasar más rápido girará el rotor. El cambio de alguno de los cables provoca un cambio en la dirección de giro. (Apartado "a" de la Figura 10).
- *Batería - ESCs:* Esta conexión proporciona la energía necesaria a los *ESCs* para que estos la administren de la manera en que el microcontrolador ordene. La conexión debe hacerse en paralelo para que cada uno de los *ESC* reciba el mismo voltaje de la batería. (Apartado "b" de la Figura 10).
- *ESCs - MCU:* Se compone de tres cables: Señal (blanco), *VCC* (rojo) y *GND* (negro). El *ESC* gracias a señales del microcontrolador en forma de pulsos (Modulación de onda mediante pulsos, o "*Pulse Width Modulation*", *PWM*) a través del cable de señal se encarga de regular cuanta energía procedente de la batería debe dejar pasar y así se regula la velocidad del rotor. (Apartado "c" de la Figura 10). Los pines del *MCU* útiles para este proyecto son aquellos que pueden soportar *PWM* (verde claro en la Figura 5). Así como los pines *GND*, o tierra. Concretamente se usarán los pines GP6, GP7, GP14, GP15, GP16, GP17, GP26 y GP27. Y para *GND* los pines 8, 18, 23 y 33.

- Móvil - MCU: Esta es una conexión de interfaces *USB-C* (móvil) a *Micro-USB* (microcontrolador). El teléfono móvil a partir de comandos *MicroPython* ejecutará órdenes hacia el microcontrolador y a partir de estas órdenes, el microcontrolador enviará señales de diferentes pulsos en sus pines de salida hacia los *ESC*. A su vez, a partir de esta conexión, el móvil proporciona energía para el funcionamiento del *MCU*. (Apartado "d" de la Figura 10). El resultado de las conexiones físicas de estos tres últimos puntos se puede apreciar en la Figura 21 en el anexo.

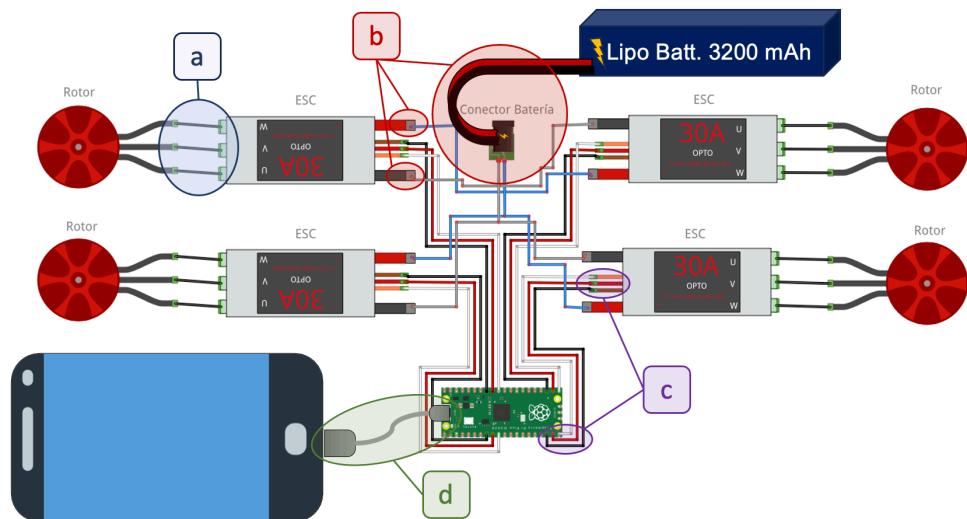


Figura 10. Diagrama de conexiones eléctricas en el dron.

#### **Resultado final de la implementación del hardware y electrónica:**

Una vez todo conectado y ensamblado, en la Figura 11 se puede observar el resultado final de la construcción del dron desde una vista superior. En la Figura 12 se puede apreciar como la batería encaja perfectamente en el interior del dron, protegiendo a ésta ante posibles golpes o accidentes. Los cables unidos de color amarillo, negro y rojo de igual tamaño (observable en la Figura 12) es un ramal utilizado para cargar el dron sin desmontar la batería de este. El teléfono móvil queda en la parte inferior del dron ya que sus cámaras deben dar al exterior.



Figura 11. Vista superior del resultado del dron

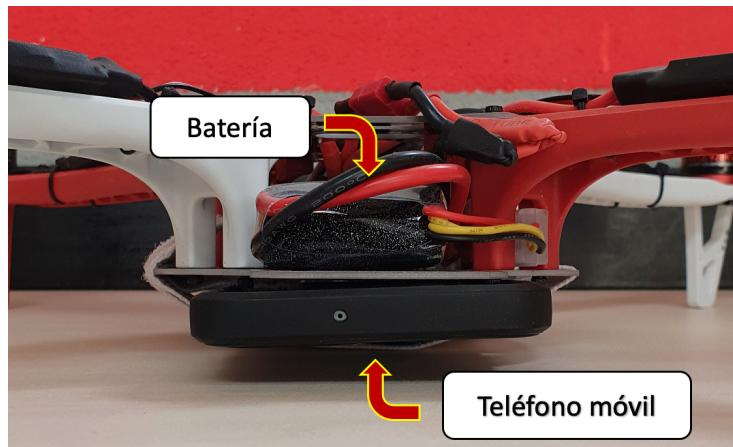
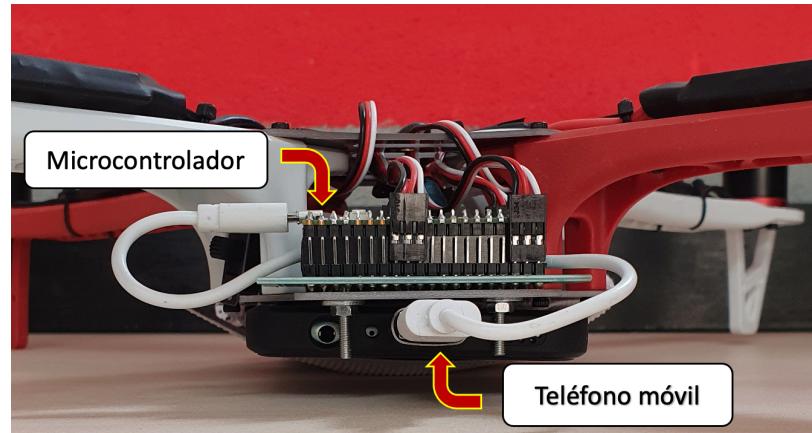


Figura 12. Vista frontal delantera del dron. Batería y móvil.

El microcontrolador, *Raspberry Pi Pico®*, queda en la parte trasera del dron conectado mediante un cable serial al teléfono móvil. A su vez a este llegan ramales de cables agrupados en 3, que se conectan a los pines del microcontrolador y que llegarán de cada uno de los *ESCs* (observable en la Figura 21 del anexo).



**Figura 13. Vista frontal trasera dron. Móvil y MCU.**

En la Figura 14 se puede ver la parte inferior del dron y se puede observar como el teléfono móvil ha sido sujetado al armazón del dron mediante una cincha de tejido de goma elástica. A la vez este amarre sujeta la batería en el interior del dron. Se puede observar además como las 4 cámaras del teléfono móvil quedan mirando hacia el exterior permitiendo así la captura de imágenes de todo lo que sobrevuela el dron. Por el lado contrario, la cámara frontal queda al descubierto dejando la posibilidad de que al cambiar de cámara haya otra vista posible para que el controlador pueda saber qué rodea al dron antes de despegar, por ejemplo.



**Figura 14. Vista inferior del resultado del dron.**

### 5.1.2 Software

Como se ha mencionado anteriormente en la sección 5.1.1 el dispositivo seleccionado debe ser *Android*, y se debe desarrollar una aplicación móvil para este sistema operativo. Para el desarrollo de la aplicación en *Android* se eligió *Kotlin* como lenguaje de programación. Esta elección se tomó frente a *Java*, debido a que es un lenguaje multiplataforma, sirve a la vez para *Android* como para *iOS*, tiene una compatibilidad total con *Java*, y por sus ventajas en términos de rendimiento y eficiencia.

Una aplicación *Kotlin* consta de varias partes:

- *Layouts*: Son los encargados de la interfaz gráfica del usuario. Están escritos en “*XML*”.
- Recursos: los recursos incluyen cadenas de texto, imágenes, colores y estilos. También están escritos en “*XML*”. En nuestro caso estos recursos son usados para almacenar la dirección *IP* del *SCV* y puertos de cada uno de los servicios a donde se tiene que enviar información (cadenas de texto), los logotipos de la aplicación *Android* (imágenes) y los colores y estilos de los botones de la interfaz gráfica.
- *Gradle*: es el lugar destinado a la construcción del código y la gestión de las dependencias.
- *Manifest*: es donde se declaran componentes esenciales de la aplicación como rutas a código (o imágenes) así como en donde se declaran los permisos.
- Archivos de código *Kotlin*: están organizados en paquetes. Dentro de estos archivos se encuentran las clases, funciones, y objetos que conforman la lógica de la aplicación. En adelante se tratará la implementación seguida con estos archivos de código *Kotlin*.

En la parte del diseño de software del dron (sección 4.1.2) se especificó que la aplicación debía ejecutar tareas en paralelo, para ello en *Kotlin* existen unas implementaciones llamadas “*Corrutinas*”, denominadas comúnmente en otros lenguajes de programación como “*Threads*”. Así, la aplicación se divide en varias *corrutinas* que se ejecutan en paralelo, gestionando varias tareas de manera simultánea. Concretamente

hay 6 *corrutinas*; la función principal (o “*Main*”) y 5 más (esquema representado en la Figura 15), que se detalla a continuación:

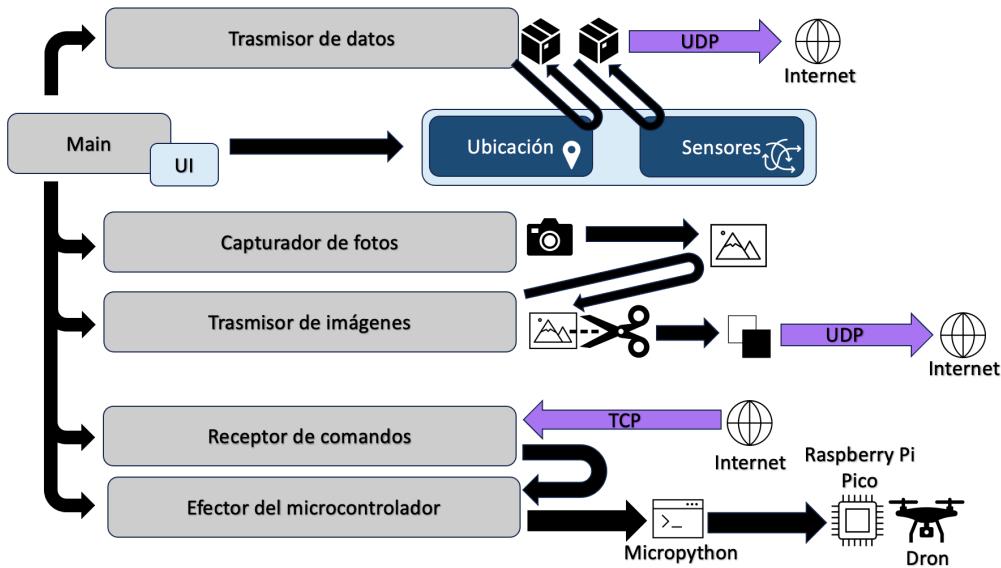


Figura 15. Diagrama interno de la aplicación Kotlin

**Main:** Esta corrutina ejecuta la interfaz gráfica. Dicha interfaz consta de un cuadrado que muestra la vista de la cámara y un botón para cambiar de cámara. Aunque esta funcionalidad está destinada principalmente para el desarrollador, resulta útil para visualizar qué cámara está en uso. A su vez también se ejecutan dos tareas críticas del proyecto que serán:

- La obtención de la ubicación en tiempo real. Se utiliza la ubicación más concreta que puede proporcionar el teléfono móvil. Esta se dará en dos componentes: latitud y longitud.
- La obtención de los datos de los sensores iniciales. Se cogerán 5 sensores de todos los que dispone el móvil:
  - Acelerómetro: Mide la aceleración del dispositivo en las tres direcciones del espacio (x, y, z), proporcionando datos sobre los movimientos lineales.
  - Giroscopio: Mide la velocidad angular del dispositivo, es decir, cómo esté rotando alrededor de los tres ejes.

- Sensor de gravedad: Proporciona una medida de la fuerza de gravedad actuando sobre el dispositivo, ayudando a distinguir entre movimientos intencionados y la aceleración debida a la gravedad.
- Vector de rotación de juego: Ofrece información sobre la orientación del dispositivo en el espacio, combinando datos de varios sensores para proporcionar una representación precisa de su rotación.
- Campo magnético: Mide la intensidad y dirección del campo magnético alrededor del dispositivo, útil para determinar su orientación con respecto al norte magnético.

Además, a partir de los datos del acelerómetro se calculará una representación suavizada de sus componentes mediante una operación matemática específica, obteniendo así un sexto tipo de datos iniciales. En total, se enviarán seis tipos de datos procedentes de los sensores iniciales para su análisis.

**Capturador de fotos:** Aquí se ejecuta otra *corrutina* encargada de realizar periódicamente fotografías con la cámara que se le indique y las guarda en una ubicación temporal del dispositivo.

**Trasmisor de imágenes:** Otra *corrutina* realiza una conexión *UDP* con un servidor remoto (*SCV*). Posteriormente toma cada una de las fotos que está realizando la corrutina anteriormente mencionada y las divide en fragmentos de 1024 bytes, establece un identificador de foto y otro identificador de fragmento y envía todo a través de la conexión *UDP*. En el otro extremo un servicio se encargará de recomponer dicha foto a partir del identificador de fotografía y de fragmento. El paquete tendrá la siguiente estructura: *Cabecera UDP + identificador de foto* (siempre un número de dos dígitos: 00) + *identificador de fragmento* (siempre una secuencia de dos letras: AA) + *Datos* (bytes de la foto) + *identificador de fragmento de nuevo* (siempre una secuencia de dos letras: AA) + [Si fuese la última foto identificador de final 0x45, 0x4F, 0x46 -> EOF].

**Receptor de comandos:** Otra *corrutina* ejecuta un cliente *TCP* que se conectará a un servicio remoto (*SCV*) el cual le enviará los comandos cifrados mediante *AES-128*. Las características de la implementación del cifrado se abordará en la sección 5.3.2. Los comandos una vez descifrados pasan a formar parte de una cola de comandos que esperan

a ser resueltos por otra corutina, concretamente la corutina que se comunica con el microcontrolador.

**Efecto del microcontrolador:** Esta corutina está continuamente comprobando si hay un comando en la cola de comandos y si lo hay, envía el código *Micropython* a ejecutar correspondiente a la acción que demanda hacer dicho comando.

**Trasmisor de datos:** Los datos anteriormente mencionados de ubicación y sensores iniciales son almacenados en variables globales, las cuales son accedidas por esta *corutina* que previamente ha establecido una conexión UDP con un servicio receptor de datos en el SCV. Estos datos no son encolados ya que la perdida de alguno no es tan importante como la de la pérdida de un comando. Se envía el dato que está en ese momento en la variable global cada 100 ms aunque esto podría modificarse dependiendo de las necesidades del usuario.

Un aspecto importante relacionado con la creación de aplicaciones *Android* es la gestión de los permisos. En este caso, esta aplicación trabaja con datos que, por su naturaleza, son sensibles y deben estar protegidos. Es por ello que, antes de usar esta aplicación, se deben conceder permisos de ubicación, almacenamiento interno, cámara, USB, procesamiento en segundo plano e internet. Estos permisos quedan recogidos en el archivo "*AndroidManifest.xml*" de la aplicación *Kotlin*.

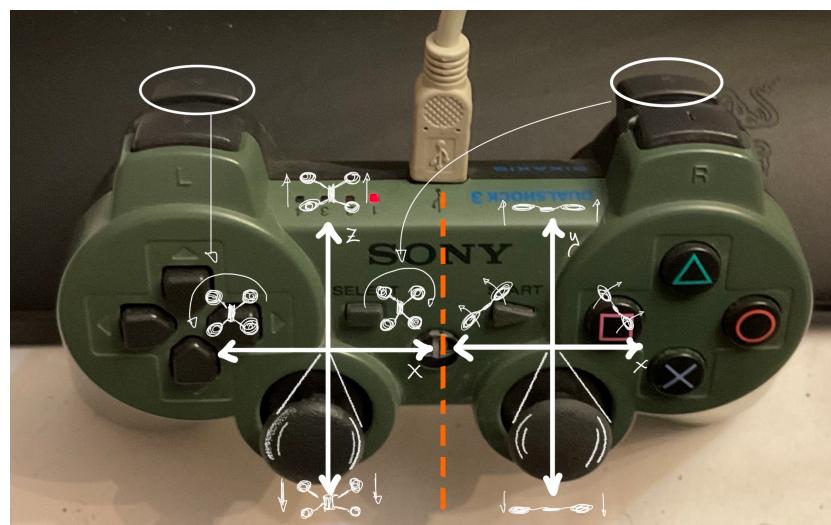
Con respecto al software del microcontrolador, inicialmente no contiene código preprogramado y se encuentra en un estado vacío. Una vez que la aplicación se activa en el móvil empieza a enviarle código que irá recibiendo y ejecutando al mismo tiempo. En un principio es así como se importan las librerías y se declaran las variables, luego a medida que se necesita alguna acción concreta dependiendo del momento la aplicación *Kotlin* enviará código *Micropython* relativo a dicha acción hacia el microcontrolador.

## 5.2 Sistema de control y visualización

### 5.2.1 Hardware

Los servicios del *SCV* se ejecutarán simultáneamente sobre un ordenador portátil.

Para registrar las señales del dron y que estas sean más fáciles de ejecutar, se decidió incorporar un mando con botones y joysticks, siguiendo con las especificaciones de movimiento descritas en la sección 4.2.1 y mostradas en la Figura 3. Con los joysticks se registran los comandos de movimientos y con los botones acciones específicas como iniciar o apagar el dron. Para este proyecto se ha utilizado específicamente un mando, de la marca “Sony”, conocido en el mercado por pertenecer a la consola “PlayStation3”. El mando se conecta al ordenador mediante un cable y transmite los comandos vía serie al ordenador, donde un programa se encargará de recoger dichos comandos y enviarlos al dron. En la Figura 16 se pueden observar los diferentes movimientos que se han asignado a la interfaz de control del dron.



**Figura 16. Mando de control y mapeado de comandos sobrepuerto.**

Los movimientos de *elevación - descenso* y *derecha - izquierda* corresponden al joystick de la derecha, y así mismo los de *rotación y avance - retroceso* corresponden al joystick izquierdo. Los botones traseros inferiores también pueden ser usados para rotación. Además, el botón “*equis*” inicia el dron, el botón “*círculo*” para el dron y el botón “*triángulo*” cambia la cámara del móvil de la frontal a la trasera y viceversa.

### 5.2.2 Software

El sistema de control y visualización (*SVC*) contará con tres servicios diferenciados ejecutándose al mismo tiempo y utilizando diferentes tecnologías como se observa en la Figura 17 y se describirá cada una de ellas en este apartado.

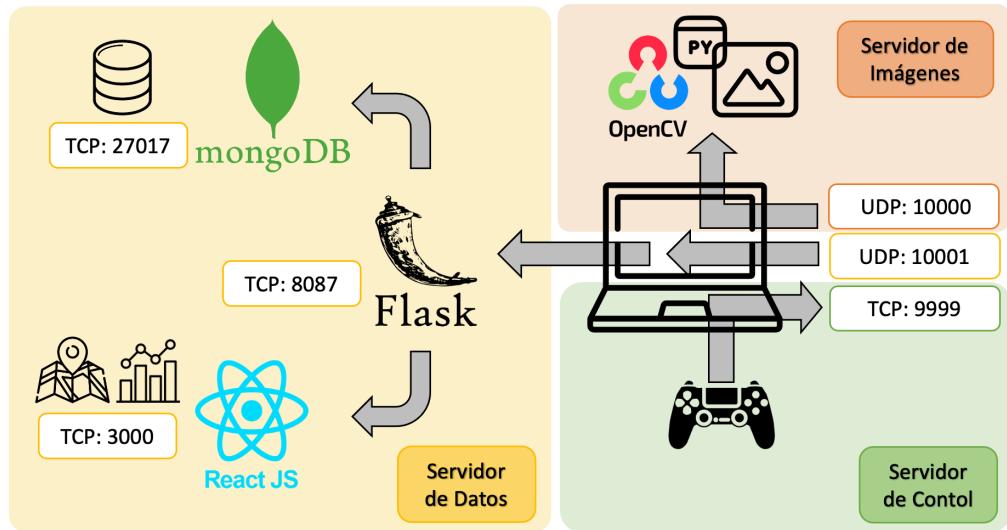


Figura 17. Diagrama de la arquitectura interna del *SCV*.

**Sistema de visualización de imágenes:** Este servicio es un programa *multithread* implementado en *Python*, cuyos *threads* se comunican mediante colas. Al principio se ejecutan 2 *threads*, uno será el socket *UDP* que escucha en el puerto 10000 y luego el *thread* principal o Main en donde estará la interfaz gráfica. Dicha interfaz gráfica está realizada con la librería *CV2* perteneciente al framework *OpenCV* quien se encarga de la representación final de las imágenes recomuestas.

Este servicio de imágenes implementa el protocolo de envío de imágenes descrito en la sección 4.2.2, así cuando llega un paquete al *SCV* siempre se mira su identificador de imagen y de fragmento, si el identificador de foto es nuevo se levanta un nuevo *thread* (para no bloquear el servicio) y este *thread* se encargará de manejar todos los fragmentos que lleguen con ese identificador de imagen. Al servicio se le pasa una cola específica para ese identificador de imagen, que será el medio por el cual se entregará a este *thread* todos los paquetes que tengan el identificador de imagen correspondiente asociado a este *thread*. Una vez que este *thread* ha recibido todos los fragmentos de una imagen, cuyo último fragmento está marcado por un delimitador, el *thread* se lo enviará a otro *thread* llamado "assembler" que será el encargado a partir de todos los fragmentos de recomponerla mirando sus identificadores de fragmento, una vez que este termina pasará la imagen a través de una cola a la interfaz gráfica, que se encargará de representarla. Al mismo tiempo existe un *thread* que manejará todos los *threads* de este programa. Lo que se persigue con esta implementación *multi-thread* es evitar el bloqueo

de los procesos y así conseguir la menor latencia posible en la representación de las imágenes. En la Figura 18 se puede observar el flujo que sigue el programa.

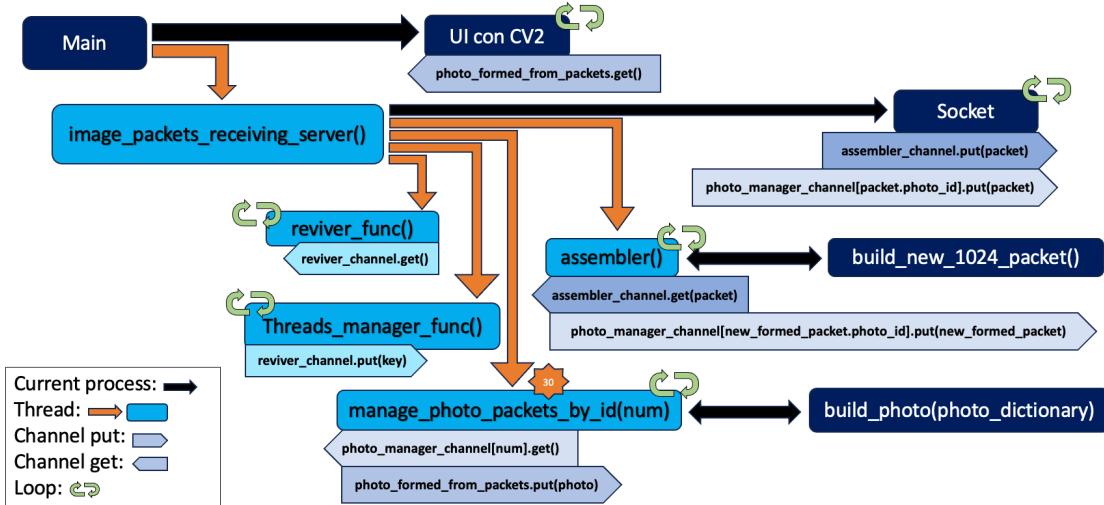


Figura 18. Flujo interno del sistema de visualización.

**Sistema de visualización de datos:** Para este programa se diseñó una arquitectura de 3 capas o modelo de 3 niveles como se comentó en el apartado de diseño (sección 4.2.2). Este modelo consta de un Nivel de Presentación, un Nivel de Aplicación y un Nivel de Datos.

- Nivel de Presentación o *frontend*: es la interfaz de usuario y de comunicación de la aplicación, donde la aplicación interactúa con el usuario final. Este nivel se ejecutará en un navegador web y se ha desarrollado usando *HTML*, *CSS* y *JavaScript*, y dentro de *JavaScript* se ha usado específicamente *React* como framework.
- Nivel de Aplicación o *backend*: también conocido como el nivel lógico, es el núcleo de la aplicación. Esta implementación ha sido desarrollada en *Python* y consta de tres partes principales:
  - Un socket *UDP* escuchando en el puerto *10001* por donde se reciben los datos procedentes del dron.
  - Un sistema de enrutamiento de datos que ofrece los datos a través de *endpoints*, estos están diferenciados dependiendo del tipo de dato, y serán accedidos mediante llamadas *API* desde la parte del *frontend*. Esta parte se

ha servido de un *framework* de *Python* llamado *Flask*. Dicha librería permite desarrollar y levantar microservicios web con facilidad. Toda información es enviada y manejada mediante *JSONs*. La dirección en donde expone los *enpoints* es "*http://localhost:8087*" y dichos *enpoints* son:

- GET: "*/get-components-data*" sirve los datos de los sensores. "*/get-location-data*" sirve las componentes de longitud y latitud de los datos de ubicación. "*/get-command-control*" sirve el comando que se ha enviado hacia el dron.
- POST: "*/comandoControl*" recibe el comando de control por parte del servicio de control. Es el medio de comunicación entre el servicio de control y el servicio de visualización de datos. Una vez que se recibe, se expone hacia el *frontend* mediante el GET "*/get-command-control*" anteriormente mencionado.
- Un cliente de la base de datos. Este cliente se sirve de la librería "*pymongo*" de *Python* para la comunicación con esta. Toda la información recibida es manejada, clasificada y posteriormente enviada a la base de datos dependiendo de su naturaleza. Esta es una interfaz de comunicación con el restante nivel del modelo de tres niveles, el nivel de datos.
- Nivel de datos: se eligió un modelo no relacional y concretamente para este proyecto se ha escogido *MongoDB* como modelo *NoSQL* por su facilidad de integración con *Python*. El servicio utiliza "*localhost:27017*".

**Sistema de Control:** este servicio utilizará el mando de control como interfaz de comunicación entre el usuario y el ordenador. Este programa también está desarrollado bajo una implementación *multi-thread*. Al comienzo del programa, se crean tres hilos. El hilo principal, denominado *Main*, es responsable de inicializar los otros dos hilos y de gestionar la lectura de comandos provenientes del control remoto; otro *thread* para el socket; y el restante *thread* funcionará de comunicación con la parte de servicio de visualización de datos, concretamente con la parte de "*Flask*". Para la interfaz entre el mando de control y el ordenador se ha utilizado la librería de *Python* "*pygame*" y se han mapeado los valores numéricos que devuelve el mando al pulsar cada uno de los comandos o direcciones de los joysticks con *strings* que representan la acción deseada.

Por ejemplo, un 1 corresponde con el botón de círculo, que a su vez esto es mapeado a "*STOP\_DRONE*". Es importante destacar en esta parte que los comandos no se envían como texto plano, sino que son cifrados mediante *AES-128*. El socket es *TCP* y escuchará en el puerto *9999*. La comunicación con el servicio de visualización de datos servirá para que en este servicio se muestre qué comando se está enviando al dron con cada una de las señales procedente del mando del control. Esta comunicación se hace mediante el protocolo *HTTP* con un envío de "*POST*" contra el servicio *Flask*, utilizando el puerto *8087*.

#### ***Resultado de la implementación del software en el SCV:***

Las imágenes que se estén enviando aparecerán ya en el servicio encargado de representarlas en el *SCV* (Figura 24 en el anexo). Los datos que se estén recibiendo también estarán siendo representados por el servicio encargado de representarlos en el *SCV* (Figura 25 en el anexo). En la captura de pantalla de la visualización de los datos se pueden observar 3 principales partes. La primera es un mapa de "*Google Maps*" que representa mediante un ícono de un dron la ubicación de este en tiempo real. Adicionalmente este mapa es interactivo y se puede acercar o alejar la vista, así como rotar, poner dicho mapa en pantalla completa e incluso cambiar el tipo de mapa a visualización de satélite o visualización de "*maps*".

La segunda parte, al lado del mapa, se puede ver el mismo logotipo del dron, pero rodeado de triángulos. Cada uno de esos triángulos indica una posición en el espacio y hace referencia a que orden de control se ha enviado hacia el dron. Dicha orden aparecerá en el recuadro en donde pone "*DOWN*" dependiendo de cuál sea el comando enviado.

Por último, la tercera parte es el resto de la interfaz, es decir, debajo del mapa, en donde se pueden apreciar 6 gráficos con representaciones de los diferentes sensores incorporados en el dron.

Parte de esta información será almacenada en la base de datos de "*MongoDB*" y gracias a la herramienta "*MongoDB Compass*" se puede visualizar lo guardado en la base de datos. En el anexo se muestra una captura de cada una de las ubicaciones de los 2 tipos

de datos (Figura 26) y un ejemplo de lo que contiene la ubicación en donde se guardan los datos de los acelerómetros (Figura 27).

### **5.3 Infraestructura de Comunicación**

En este apartado se detallarán los componentes clave de la infraestructura de comunicación implementados para este proyecto. Se abordará tanto la red de telefonía móvil y el operador contratado para dar soporte a este proyecto, así como la seguridad y salvaguardas implementadas.

#### *5.3.1 Red móvil*

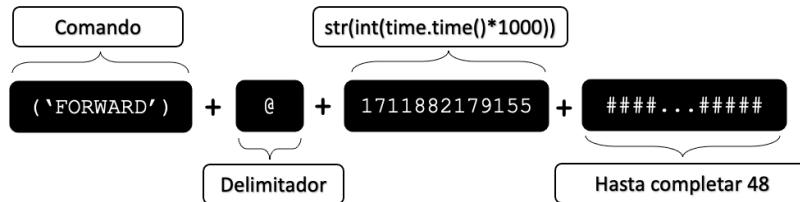
Para este proyecto se escogió la tarifa 5G del operador de telefonía *O2* de 12 GB con un coste de 7 euros mensuales [22]. Se ha seleccionado esta compañía por estar respaldada por la sólida y extendida infraestructura de red móvil de *Telefónica* [23], la más amplia de España. Además, se ha escogido este operador por ofrecer tarifas más económicas respecto a *Telefónica*. Esta elección garantiza una cobertura amplia y fiable (REQCOM-01), así como una velocidad de conexión adecuada para las necesidades del proyecto, asegurando así un rendimiento constante y eficiente, con un coste reducido (REQDIS-01).

#### *5.3.2 Seguridad*

En la sección 2.2.1 se mostraron los diferentes ataques que pueden afectar al dron y en la sección 4.3.2 la solución que se estableció para cada uno de ellos. Aquí se muestra cómo se ha llevado a cabo dicha solución:

1. Intercepción de la comunicación o “*Man in the Middle*”: La solución determinada fue aplicar cifrado *AES-128*. Esto asegura la integridad y confidencialidad de los comandos. Se eligió el Estándar de Cifrado Avanzado (*AES* [24]) de 128 bits. *AES-128* es un algoritmo de cifrado de clave simétrica por bloques que utiliza claves de cifrado de 128 bits. Ambas partes comparten la misma clave, la cual se protegerá en ambos sistemas, tanto en el emisor como en el receptor, evitando así que el atacante obtenga esta y pueda interceptar y/o modificar los mensajes.

2. Ataques de repetición o “*Replay Attacks*”: Para evitar ataques de repetición, se ha decidido incluir una marca temporal en cada mensaje antes de ser cifrado. Esta marca temporal se toma de la fecha en milisegundos del sistema. De esta manera, el mensaje cifrado se estructura de la siguiente manera: *COMANDO* + @ + *FechaEnMilis* + #####... hasta alcanzar un múltiplo de 16 bytes (Figura 19). El símbolo '@' actúa como delimitador, y los símbolos '#' rellenan el mensaje para asegurar que su longitud sea un múltiplo de 16 bytes, ya que *AES-128* cifra bloques de 16 bytes (128 bits). Este enfoque asegura que cada mensaje es único y no puede ser repetido sin ser detectado.



**Figura 19. Estructura del mensaje de control antes de aplicar cifrado AES-128.**

3. Conexión insegura a Internet: la solución adoptada fue la utilización de *VPN*. Cómo servicio *VPN* se escogió *OpenVPN* [25] por su facilidad, su amplia adopción y su popularidad. Además, ofrece la posibilidad de levantar un servicio *VPN* en la nube (este servicio se llama *Cloud Conexa*) y que los clientes se conecten a él. Una vez configurado dicho servicio *VPN* en la nube se instala en cada uno de los extremos, dron y sistema de control, un cliente *OpenVPN* (En la Figura 29 del anexo se puede observar como una de las aplicaciones es el cliente *OpenVPN*).
4. Caída de los servicios del *SCV* o pérdida de conexión: se han implementado medidas de resiliencia en el código de los servicios del *SCV*. Estas medidas incluyen la monitorización continua del estado de la conexión y la implementación de reglas para asegurar la persistencia de los servicios del *SCV* ante cualquier error. Se han configurado mecanismos de reconexión automática y políticas de reinicio automático para minimizar el tiempo de inactividad. Además, se ha gestionado la captura de errores en el código ante situaciones imprevistas.

## 6 EXPERIMENTACIÓN Y PRUEBAS

El dron no ha podido ser volado al aire libre para verificar su completo funcionamiento. Esto es debido a que dichos vuelos son ilegales, si no se posee un registro como operador, una formación como piloto y un seguro [26]. Así mismo al ser un dron de fabricación privada y un prototipo necesita ser previamente acreditado por un organismo notificado [26]. Inclusive, para operar el tipo de dron que se muestra en este proyecto, perteneciente a la categoría de drones sin visibilidad directa (*BVLOS*, por sus siglas en inglés) en España, se requiere una autorización especial de la *Agencia Estatal de Seguridad Aérea (AES)* [27]). Estas operaciones generalmente caen bajo una categoría específica, donde se necesitan condiciones y autorizaciones adicionales debido al mayor riesgo que implican.

Por lo tanto, teniendo esto último en cuenta, las pruebas seguidas para este proyecto han sido las contempladas en los siguientes apartados.

### 6.1 Prueba de vuelo controlada

El vuelo y funcionalidades del dron pudieron ser probados en una pequeña prueba en condiciones de laboratorio. Dicha prueba se realizó atando cada uno de los cuatro extremos del dron con cuerdas elásticas a una mesa pesada. Se inició el dron y se mandaron comandos de vuelo a este (REQCONT-02) y el dron levantó el vuelo (REQNAV-02) 50 cm de la mesa y se mantuvo estable. El dron respondió de manera efectiva ante dichos comandos, se pudieron recoger datos de vuelo en tiempo real (REQDAT-01), mostrándose estos en la interfaz gráfica de usuario (REQDAT-02) y almacenándose en la base de datos (REQDAT-03), dando por cumplimentados la mayoría de los requisitos y objetivos de este proyecto.

### 6.2 Evaluación del SCV

Los servicios del *SCV* se ejecutaron simultáneamente en un ordenador portátil equipado con un procesador *Intel Core i5* de 4 núcleos y 8 GB de *RAM*. En ningún momento el ordenador presentó signos de sobrecarga durante la ejecución de los distintos programas utilizados para monitorizar y controlar el dron.

### 6.2.1 Protocolo de trasmisión de imágenes

Al ser la trasmisión de imágenes una de las partes más complicadas de la comunicación, se realizaron pruebas y evaluaciones externas e independientes adicionales a la prueba de vuelo.

Con el objetivo de medir los tiempos de envío de las imágenes, se realizó una ejecución con código de depuración en la aplicación móvil. Esto permitió comprobar la cantidad de imágenes que la aplicación era capaz de capturar, gestionar, fragmentar y enviar, mostrando los tiempos en la consola de depuración.

Gracias a ello se comprobó que el software implementado en la captura de imágenes no supera las 3 fotografías por segundo (*fps*). Posteriormente se observó como a consecuencia de ello y de retrasos de procesamiento, en la recepción se obtiene una tasa de  $2,5 \sim 2,8 \text{ } fps$  (dependiendo del peso de la imagen procesada). Para una experiencia de vídeo la tasa de *fps* es de 30. Este número dista mucho del resultado, pero la experiencia es fluida y proporciona una visión adecuada del entorno (REQDAT-02).

Así mismo se realizó una prueba de la trasmisión de imágenes con el móvil sin anclar al dron y con objetos en una mesa. En la Figura 20 se puede observar una captura de la interfaz del teléfono móvil y en la Figura 24 (en el anexo) se observa una captura del servicio que muestra las imágenes del SCV (REQDAT-01 y RQDAT-02).



Figura 20. Captura de pantalla de la aplicación móvil trasmitiendo imagen

### 6.3 Rendimiento y evaluación de la comunicación

Mediante un software de monitorización de tráfico de redes, llamado “Wireshark”, se ha capturado el tráfico entre el dron y el *SCV*, tanto de origen como destino. *Wireshark* incorpora una opción que permite ver las estadísticas y mediciones de los flujos de información intercambiada. Gracias a ella se ha podido medir el tráfico entrante y saliente de los servicios específicos de este proyecto.

Los resultados obtenidos se muestran en la Tabla 1. Estos resultados muestran como el grueso de la comunicación reside en el flujo de las imágenes del dron al *SCV*, llegando a ser de alrededor de 1,6 Mbits por segundo.

Puerto	Paquetes	Bytes	Duración (s)	bits/s
10001	2210	489 K	113	34 Kbps
10000	20874	22,13 M	111	1,58 Mbps
9999	128	11 K	110	1,46 Kbps

Tabla 1. Resultado de la captura de *Wireshark*

Las imágenes realizadas por el teléfono móvil y recibidas en el *SCV* difieren mucho en peso dependiendo de las características de la imagen, llegando a ser aquellas en un color negro total las más pesadas con casi 80 KB de información y aquellas blancas completas una vigésima parte menor con 4KB de información (Figura 28 del anexo). Por ello dependiendo del tamaño de la imagen enviada la tasa de bits por segundo aumentará o disminuirá siendo de media una tasa de 1,5 Mbits por segundo. Esto es debido a que el cuello de botella no reside en la comunicación sino en la tasa de imágenes por segundo que se genera en el sistema emisor.

A su vez se ha podido realizar con este software una segunda prueba. Dicha prueba consistía en realizar una interceptación del tráfico del protocolo de control. Esta comunicación es importante debido a que es en donde se aplica cifrado *AES-128*. Se escogieron paquetes al azar enviados y se inspeccionó uno a uno su contenido. Se pudo observar como este era ininteligible y por lo tanto que el cifrado estaba funcionando (REQCOM-01).

Otra prueba realizada con respecto a la comunicación fue la pérdida de esta, estando el dron en funcionamiento. Para ello se simuló que se había perdido la comunicación no enviando comandos. Se procedió entonces a usar la medida de salvaguarda implementada; el botón de apagado físico. Se pudo acceder a él sin daños y se apagó el dron (REQCOM-02).

#### **6.4 Evaluación final de la experimentación**

Dejando la legislación a un lado, este prototipo no ha sido suficientemente testeado bajo condiciones de laboratorio para que pueda ser volado en un entorno hostil lleno de condiciones cambiantes.

Para ello deberán diseñarse una serie de experimentos y pruebas con diversos fines y objetivos, centrándose en módulos específicos del proyecto. Este proceso contempla una serie de periodos y tiempos que están fuera del alcance de este proyecto.

Como muestra del resultado del proyecto se ha realizado un corto vídeo explicativo y demostrativo. En dicho vídeo se abarca la estructura del proyecto así como una demostración de vuelo en tiempo real. El enlace al vídeo así como el código del proyecto se puede encontrar en el siguiente repositorio [28].

## 7 CONCLUSIONES

En este apartado se introducirán las distintas conclusiones obtenidas con el desarrollo del proyecto. Esta sección se estructurará analizando los objetivos propuestos en función a los costes y la evaluación de los requisitos. Además, para finalizar, se expondrán las líneas futuras de trabajo con cuestiones que quedan abiertas en este trabajo.

### 7.1 Costes

La Tabla 2 muestra el coste total de todo el dron. Este ha sido de 361,13 € (REQDIS-01), el cual es 10 veces menor que el precio de la opción más barata mencionada en el estado del arte (sección 2.1), el dron “*Qualcomm Flight™ RB5 5G Platform*”, cuyo precio es de 4199.99\$ [3] (3917,65 € al cambio en Junio de 2024).

Objeto	Modelo	Precio
Armazón	F450	16,50 €
Hélices, ESC y rotores	2212/920KV, ESC 30A, Hélice 1045	47,25 €
Teléfono móvil	Samsung M33 5G	241,86 €
Microcontrolador	Raspberry Pi Pico	5,04 €
Batería Dron	Zee 3S Lipo Battery	25,50 €
Funda Móvil	Keyision	0,99 €
Cableado y bridas	-	4,00 €
Cargador de batería	Haisito B3	19,99 €
<b>TOTAL</b>		<b>361,13 €</b>

Tabla 2. Costes finales del dron.

Adicionalmente a estos costes habría que añadir el coste mensual de la tarifa telefónica de O2. Teniendo en cuenta dicha tarifa contratada de 12 GB al mes (96 Gbits), y suponiendo un tráfico máximo de 1,75 Mbits por segundo (dejando margen suficiente) esto daría 57.313 segundos, es decir, aproximadamente se podrían realizar al menos 15 horas de vuelo al mes con esta tarifa. Si se necesitase realizar más horas de vuelo al mes los costes mensuales incrementarían dependiendo de las necesidades del usuario.

Los controladores de los drones mencionados en el estado del arte son específicos para el vuelo del dron. En este proyecto el núcleo del dron, al ser un teléfono móvil, puede ser

utilizado para otras funciones diferentes a la de un controlador de vuelo y comunicación del dron. El gasto del teléfono móvil no puede considerarse un gasto exclusivo de la construcción del producto, ya que un teléfono móvil también puede desempeñar otras funciones. Además, podría utilizarse cualquier otro teléfono capaz de ejecutar la misma aplicación *Kotlin*. Por lo tanto, el coste del dron se compondría de un coste fijo de 119,27€ y un coste variable, dependiendo del teléfono móvil empleado. No es necesario adquirir un teléfono móvil específicamente para el dron, ya que se puede utilizar el propio smartphone del usuario con la aplicación *Kotlin* ejecutándose en su interior.

## **7.2 Evaluación de los requisitos cumplidos**

Teniendo en mente los requisitos descritos en la sección 2.3, en este apartado se discutirá cual ha sido el alcance de cada uno de ellos en el proyecto, pudiendo diferenciarse dos clasificaciones concretas: en primer lugar, si se ha desarrollado o completado dicho requisito y, en segundo lugar, si se ha podido probar el resultado de éste.

Todos ellos han sido completados a excepción de uno de ellos que ha sido parcialmente completado. Este ha sido el requisito REQDAT-03, consistente en el almacenamiento de los datos recopilados por el dron en una base de datos. Si bien sí que se ha conseguido el almacenamiento de los datos de las componentes de movimiento y la ubicación, no se ha implementado el almacenamiento de las imágenes recibidas. La causa de ello ha sido la limitación temporal del proyecto y las restricciones de almacenamiento del equipo.

Por otro lado, el número de los requisitos probados ha sido menor que el de completados. De los 12 requisitos impuestos al proyecto sólo se pudieron probar 10 de ellos dando un resultado exitoso, de los cuales 2 fueron parcialmente probados.

Aquellos que no se pudieron probar fueron el requisito REQNAV-01 (consistente en vuelo sin límite de cobertura) y el requisito REQCOM-03 (consistente en el cambio de tecnologías entre 3G-4G-5G). Ambos no se pudieron probar al no poder realizar vuelos al aire libre por las razones comentadas en la sección 6.

Los dos requisitos probados parcialmente fueron el requisito REQNAV-02 (consistente en vuelo en cualquier país o región) y el requisito REQCONT-01 (consistente en movimientos en todas las direcciones del dron). El primero de ellos fue probado parcialmente puesto que, si bien no se ha probado en varios países, sí que se ha probado en este, España, con éxito. El segundo de ellos es marcado como parcial al haberse realizado las pruebas con el dron atado y estando privado de libre movimiento.

En la Tabla 3 se muestra la tabla de requisitos, indicando si estos han sido probados o no de forma resumida. Quedando como requisitos cumplidos todos aquellos que han sido completados y probados con éxito.

Código	Resumen de requisito	Resultado	Probado
REQNAV-01	Vuelo sin límite de cobertura	Completado	No
REQNAV-02	Vuelo en cualquier país o región	Completado	Parcialmente
REQDAT-01	Recopilación de datos	Completado	Si
REQDAT-02	Representación de los datos	Completado	Si
REQDAT-03	Datos almacenados base de datos	Parcialmente	Si
REQCONT-01	Movimientos del dron	Completado	Parcialmente
REQCONT-02	Interfaz de control	Completado	Si
REQCOM-01	Comunicación segura	Completado	Si
REQCOM-02	Incorporación de salvaguardas	Completado	Si
REQCOM-03	Cambio entre 3G - 4G - 5G	Completado	No
REQDIS-01	No superar coste de 500€	Completado	Si
REQDIS-02	Proyecto modular	Completado	Si

**Tabla 3. Tabla de resultados de los requisitos del proyecto**

### **7.3 Líneas futuras de trabajo**

A la vista de los resultados obtenidos, y una vez analizados estos con detenimiento, caben ciertas mejoras e incluso nuevas funcionalidades aplicables al proyecto. Con estas nuevas funcionalidades se dotará al conjunto de un mayor valor tanto tecnológico como económico. Entre las posibles mejoras detectadas y líneas futuras de trabajo destacan:

- Mejora en la calidad de envío usando de *RTP* (*Real Time Protocol*). Para el envío de los datos desde el dron hacia el *SCV* se usó *UDP* como protocolo. Este protocolo es válido para la tarea asignada pero no está suficientemente optimizado para garantizar la integridad y el orden de los paquetes en transmisión. Esto provoca que se deba hacer un desarrollo adicional en una capa superior cuya implementación es compleja. Añadiendo el requisito de evitar todo retraso posible (REQDAT-02), una mala implementación se traducirá en costes de latencia. Es por ello por lo que se debería introducir el protocolo *RTP*, ya que ofrece mecanismos integrados para manejar la sincronización y la entrega ordenada de los paquetes, lo que mejorará significativamente la calidad del servicio en la transmisión en tiempo real.
- Incorporación de audio al conjunto de datos: al igual que la transmisión de fotos, se podría incluir audio bidireccional, ya que el teléfono móvil lo soportaría, del mismo modo que si fuese una llamada telefónica. Este flujo debería incluirse a su vez de la mano de la anterior mejora usando el protocolo de *RTP* para trasmisir los paquetes de audio.
- Encapsulamiento de los servicios en contenedores: las mejoras que ofrece poder tener el código en contenedores diferenciados son varias, entre las que destacan:
  - Portabilidad: los servicios se pueden ejecutar de manera consistente en diferentes entornos.
  - Aislamiento: cada contenedor es independiente y contiene todas las dependencias necesarias, lo que evita conflictos entre bibliotecas y versiones. Además, los contenedores se pueden comunicar entre si a partir de redes internas, pudiendo controlar todo el tráfico que existe entre ellos, dotando a el entorno de una mejor seguridad y control.
  - Escalabilidad: es una característica relacionada con la anterior de aislamiento. Se puede escalar una determinada parte del proyecto sin que esta interfiera en las otras. Esto es gracias a un control de las versiones que componen el proyecto.
- Despliegue de los servicios del *SCV* en la nube: la posibilidad de poder tener estos servicios en la nube ofrecería:

- Alta disponibilidad y redundancia: los proveedores de nube ofrecen infraestructura distribuida globalmente, asegurando que los servicios estén siempre disponibles y que haya redundancia en caso de fallos.
- Reducción de costes operativos: mantener los servicios en un ordenador personal implica gastos significativos en hardware, electricidad, refrigeración y mantenimiento. La nube elimina estos costos, trasladándolos a un modelo de pago por uso.
- Acceso remoto y colaboración: la nube facilita el acceso a los servicios del SCV desde cualquier lugar con conexión a Internet, permitiendo un acceso más eficiente geográficamente. Además, abre las puertas a hacer de este un proyecto colaborativo.
- Salvaguardas adicionales ante perdida de conexión: si se pierde la conexión seguir un protocolo de descenso lento o de vuelta a origen. Adicionalmente el dron al aterrizar podría emitir sonidos para que pueda ser fácilmente localizable y recogido.
- Monitorizar el estado de la batería del dron: implementar un sistema de monitoreo continuo que alerte cuando los niveles de batería caen por debajo de un umbral crítico, asegurando así un retorno seguro o un aterrizaje de emergencia controlado.
- Nuevas funcionalidades como electroimán, servos, sensores de temperatura y de luz: al ser un proyecto tan escalable, se podría integrar estas funcionalidades para ampliar las capacidades del dron, permitiendo nuevas aplicaciones como el transporte de objetos, control preciso de movimientos y monitoreo del entorno. Con esta última acción, se recopilarían más datos potencialmente útiles para otras aplicaciones como entrenamiento de redes neuronales para modelos de inteligencia artificial.
- Respuestas de la aplicación del dron ante comandos de control para saber si se han recibido correctamente en origen. Se desarrollaría un sistema de confirmación de comandos en tiempo real, donde el dron envíe una señal de acuse de recibo para cada comando ejecutado, asegurando así una operación más fiable y precisa.

- Incorporación de placas solares al dron y cómo esto afecta a la relación peso-consumo-energía adicional. Si bien la incorporación de placas solares supondría un aumento en la autonomía de la batería, también supondría un peso adicional y por lo tanto un aumento de consumo. Bajo pruebas de laboratorio habría que determinar el beneficio de esta medida.
- Incrementar el conjunto de experimentos en laboratorio:
  - Pruebas de duración total de la batería dependiendo el tipo de vuelo.
  - Pruebas de carga máxima soportada por el dron.
  - Pruebas de ciberseguridad: Ataque de denegación de servicio al *SCV* y dron, acceso a claves *AES*, interceptación modificación de la comunicación, etc.
  - Pruebas de radiación solar y temperatura. Puede que los componentes usados no soporten bien temperaturas extremas (tanto altas como bajas).

## 8 REFERENCIAS

- [1] Página web oficial de 3GPP: <https://www.3gpp.org/> (Accedida: Abril 2024)
- [2] Página web oficial de Qualcomm: <https://www.qualcomm.com/> (Accedida: Abril 2024)
- [3] Página web oficial de Modal-AI: <https://www.modalai.com/> (Accedida: Abril 2024)
- [4] Página Web de RB5 de ModalAI: <https://www.modalai.com/collections/qualcomm/products/qualcomm-flight-rb5-5g-platform-reference-drone?variant=39517470326835> (Accedida: Abril 2024)
- [5] Página Web de Sentinel de ModalAI: <https://www.modalai.com/products/sentinel-development-drone?variant=47564491587888> (Accedida: Abril 2024)
- [6] Documentación de Sentinel Drone y sus especificaciones de radio: <https://beta-docs.modalai.com/sentinel-functional-description/#radio-specifications> (Accedida: Abril 2024)
- [7] Página web oficial de Beyond Vision: <https://beyond-vision.com/es/> (Accedida: Abril 2024)
- [8] Página Web de VTOne de Beyond Vision: <https://beyond-vision.com/vtone-ai-powered-vtol-drone/> (Accedida: Abril 2024)
- [9] Página web oficial de Dji: <https://www.dji.com/es> (Accedida: Abril 2024)
- [10] Página web oficial de Parrot: <https://www.parrot.com/en> (Accedida: Abril 2024)
- [11] Página web con la documentación oficial del protocolo OcuSync versión 3 de Dji: <https://www.dji.com/es/o3-air-unit> (Accedida: Abril 2024)
- [12] Página web con la documentación oficial del protocolo Ligghtbridge de Dji: <https://www.dji.com/es/dji-lightbridge> (Accedida: Abril 2024)
- [13] Documentación en pdf de DJI Celular Dongle: [https://dl.djicdn.com/downloads/DJI\\_Mavic\\_3/DJI\\_Cellular\\_Dongle\\_LTE\\_USB\\_Modem\\_User\\_Guide\\_v1.0.pdf](https://dl.djicdn.com/downloads/DJI_Mavic_3/DJI_Cellular_Dongle_LTE_USB_Modem_User_Guide_v1.0.pdf) (Accedida: Abril 2024)
- [14] Página Web de AnafiAI de Parrot: <https://www.parrot.com/en/drones/anafi-ai/technical-documentation/connectivity> (Accedida: Abril 2024)
- [15] Precio de AnafiAI de Parrot en DronVIVO: [https://www.dronivo.de/Parrot-Anafi-Ai\\_1?gad\\_source=1&gclid=CjwKCAiA3JCvBhA8EiwA4kujZqayW0DwmYd99F4jHm1oe-9LqrSk65BrosNeXBBwUBGopBmRnQXvhocQxcQAvD\\_BwE](https://www.dronivo.de/Parrot-Anafi-Ai_1?gad_source=1&gclid=CjwKCAiA3JCvBhA8EiwA4kujZqayW0DwmYd99F4jHm1oe-9LqrSk65BrosNeXBBwUBGopBmRnQXvhocQxcQAvD_BwE) (Accedida: Abril 2024)
- [16] Página Web de Quectel 5G RM50xQ series: <https://www.quectel.com/product/5g-rm50xq-series> (Accedida: Abril 2024)
- [17] William Stallings. Network Security Essentials: Applications and Standards. Sixth Edition. Pearson 2017. (ISBN: 978-0134527338)
- [18] William Stallings. Cryptography and Network Security, principles and practice. Seventh Edition 2017. Pearson. (ISBN: 9780134444284)
- [19] Len Bass, Paul Clements, Rick Kazman. Software Architecture in Practice. Third Edition. Addison Wesley 2012. (ISBN: 9780321815736)
- [20] Jean-Philippe Aumasson. Serious cryptography, a practical introduction to modern encryption. First Edition. No starch press 2017. (ISBN: 1593278268)
- [21] Página web oficial de Raspberry Pi Foundation: <https://www.raspberrypi.org/> (Accedida: Junio 2024)

- [22] Página web oficial con la tarifa de O2: <https://o2online.es/movil/> (Accedida: Mayo 2024)
- [23] Página web oficial de Telefónica: <https://www.telefonica.com/es/> (Accedida: Mayo 2024)
- [24] National Institute of Standards and Technology. Advanced Encryption Standard (AES), 2001, updated 2023. Federal Information Processing Standards FIPS PUB 197. <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf> (Accedido: Junio 2024).
- [25] Página web oficial de OpenVPN: <https://openvpn.net/> (Accedida: Mayo 2024)
- [26] Documento de European Union Aviation Safety Agency, "Easy Access Rules for Unmanned Aircraft Systems (Revision April 2024)": <https://easa.europa.eu/document-library/general-publications/easy-access-rules-unmanned-aircraft-systems> (Accedido: Junio 2024).
- [27] Página oficial de la AESA: <https://www.seguridadaerea.gob.es/> (Accedida: Junio 2024).
- [28] Enlace al repositorio del proyecto: [https://github.com/bermejo4/TFG\\_TELECO](https://github.com/bermejo4/TFG_TELECO) (Accedido: Junio 2024)

## 9 Anexos

### 9.1 Imágenes adicionales

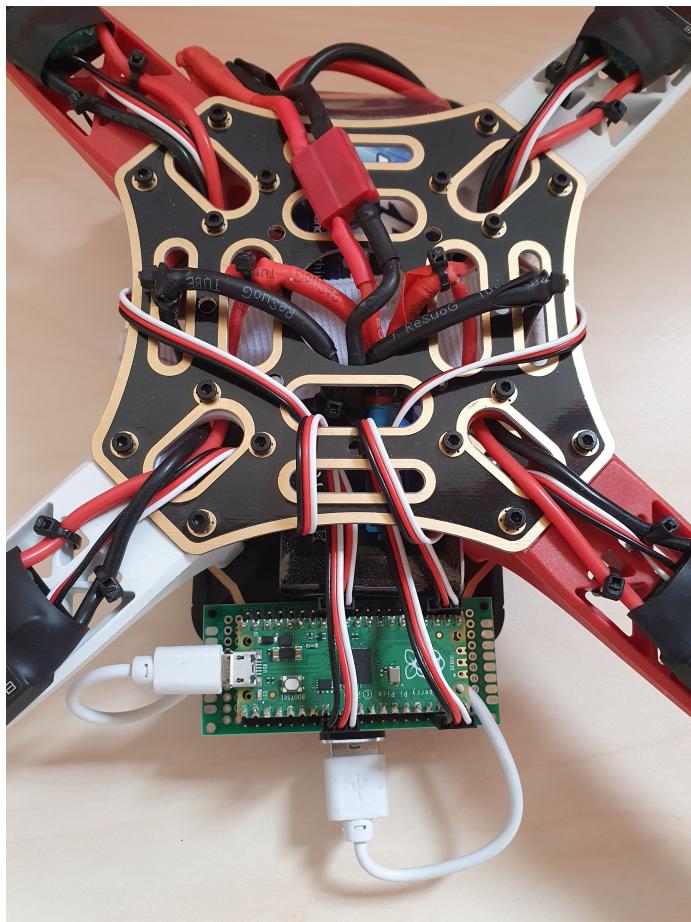


Figura 21. Resultado de cableado y conexiones físicas.

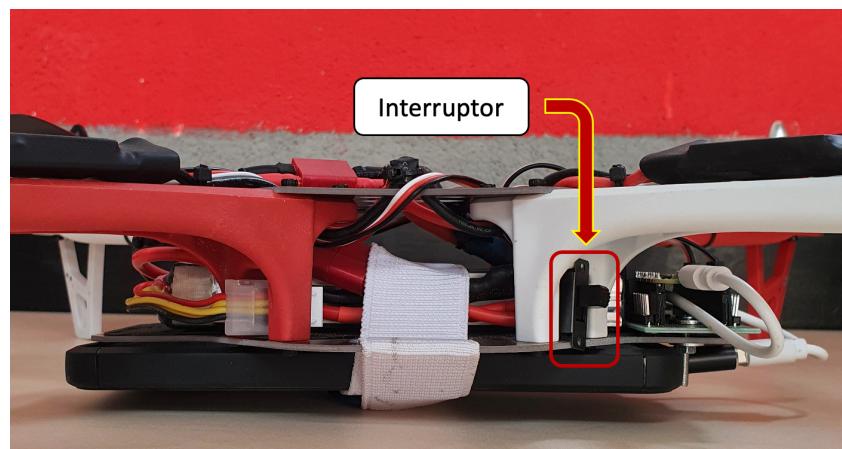
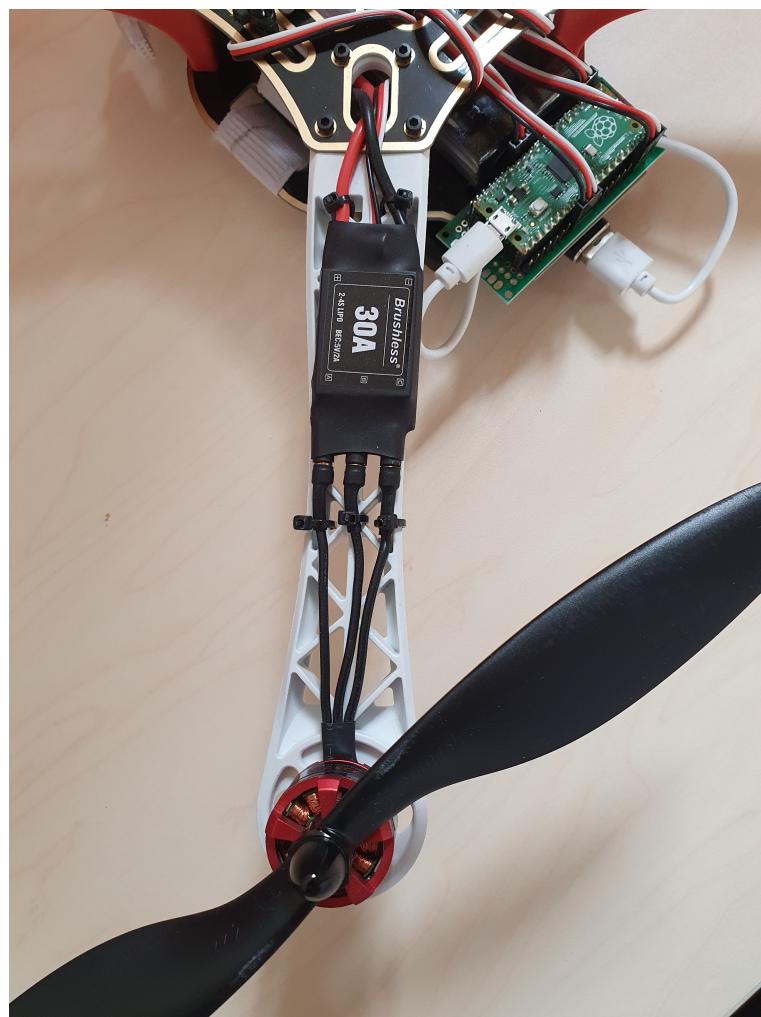


Figura 22. Lateral de dron e interruptor.



**Figura 23.** Resultado de conexión Rotor-ESC.



**Figura 24.** Visualización de imagen en el SCV

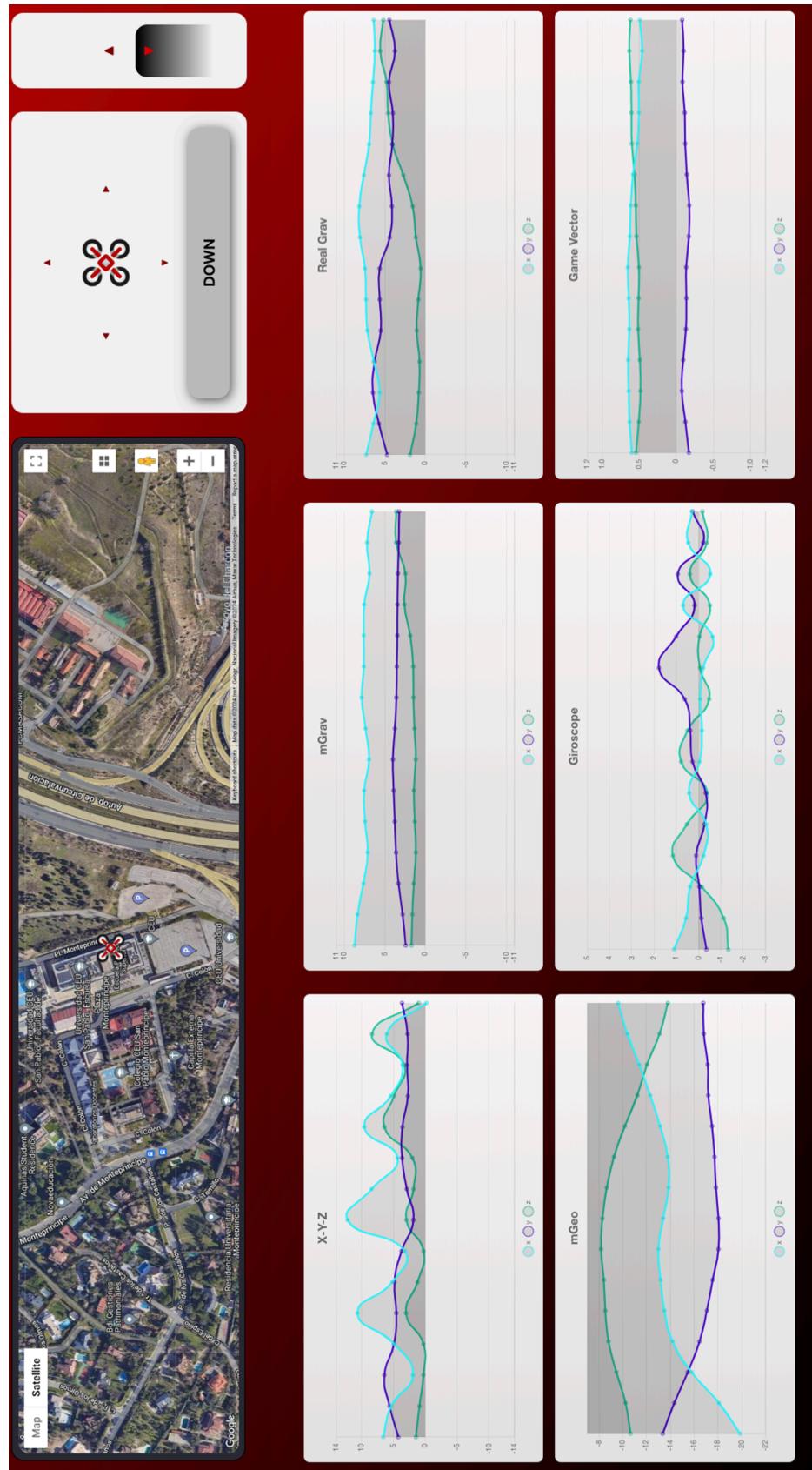


Figura 25. Sistema de visualización de datos.

The screenshot shows the MongoDB Compass interface connected to the database 'droneApp\_DB'. On the left sidebar, under 'Databases', 'droneApp\_DB' is selected. The main panel displays two sections: 'AccelerometerComponents' and 'Locations'. The 'AccelerometerComponents' section shows a storage size of 22.49 MB, 137 documents, an average document size of 380.00 B, 1 index, and a total index size of 2.89 MB. The 'Locations' section shows a storage size of 2.87 MB, 145 documents, an average document size of 107.00 B, 1 index, and a total index size of 3.04 MB.

**Figura 26. Datos ordenados en localización y componentes de acelerómetros en MongoDB.**

The screenshot shows the MongoDB Compass interface connected to the 'droneApp\_DB.AccelerometerComponents' collection. The top right corner indicates 136.8k DOCUMENTS and 1 INDEXES. The main panel has tabs for 'Documents', 'Aggregations', 'Schema', 'Indexes', and 'Validation'. Under the 'Documents' tab, there is a search bar, a filter dropdown, and a query input field. Below these are buttons for 'ADD DATA' and 'EXPORT DATA'. A status bar at the bottom shows '1 - 20 of 136795'. The main area displays five sample documents, each with an '\_id' field (ObjectId), a 'timestamp\_field' (e.g., 2023-12-30T19:50:01.783+00:00), and a 'data' field containing JSON objects representing sensor data.

```

_id: ObjectId('659066596f8d93255c0661d0')
timestamp_field: 2023-12-30T19:50:01.783+00:00
data: "[{"x":0.0,"y":0.0,"z":0.0}, {"mGrav":[0.0, 0.0, 0.0]}, {"mGeo":[0.0, 0.0, 0.0]}]"

_id: ObjectId('659066596f8d93255c0661d1')
timestamp_field: 2023-12-30T19:50:01.998+00:00
data: "[{"x":0.16998832,"y":0.079008654,"z":9.696517}, {"mGrav":[0.028143...]}]"

_id: ObjectId('6590665a6f8d93255c0661d4')
timestamp_field: 2023-12-30T19:50:02.001+00:00
data: "[{"x":0.15322891,"y":0.12210429,"z":9.694122}, {"mGrav":[0.0475274...]}]"

_id: ObjectId('6590665a6f8d93255c0661d6')
timestamp_field: 2023-12-30T19:50:02.044+00:00
data: "[{"x":0.1604115,"y":0.088585466,"z":9.725247}, {"mGrav":[0.0677024...]}]"

_id: ObjectId('6590665a6f8d93255c0661d8')
timestamp_field: 2023-12-30T19:50:02.196+00:00
data: "[{"x":0.19153613,"y":0.10534488,"z":9.718064}, {"mGrav":[0.0817963...]}]"

```

**Figura 27. Acceso a los datos de las componentes de los acelerómetros en la base de datos.**

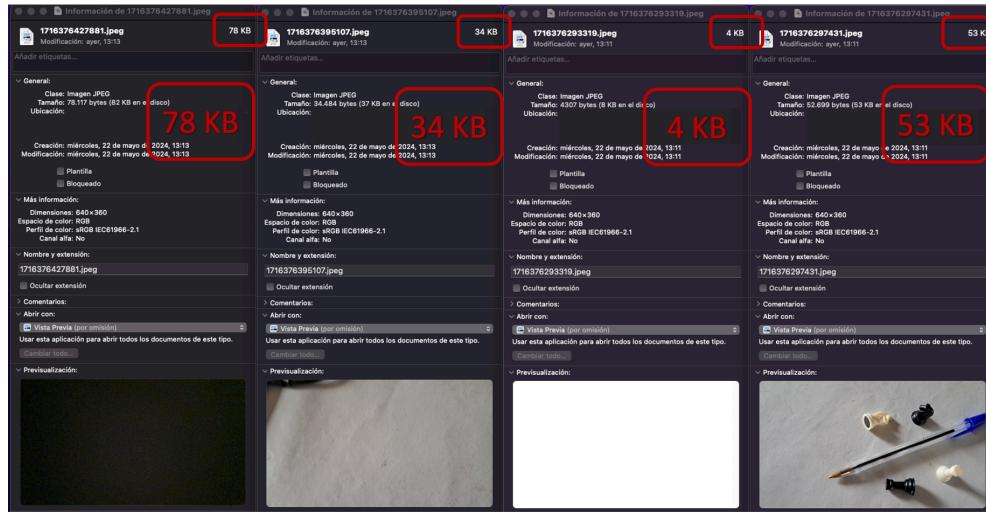


Figura 28. Comparación peso de las imágenes dependiendo del color.

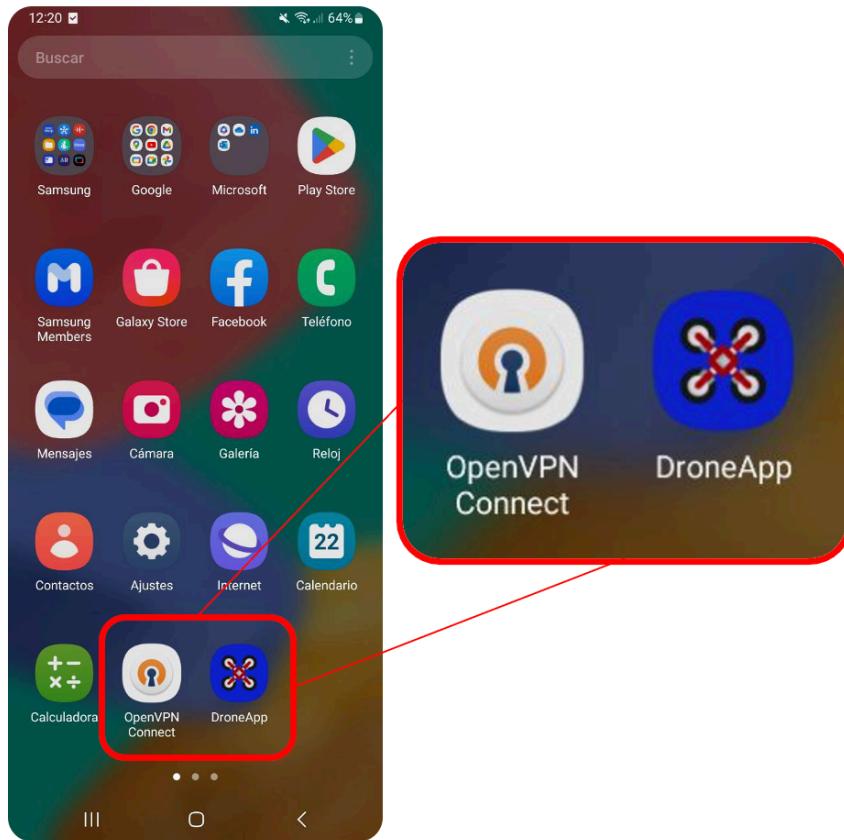


Figura 29. Aplicaciones en el teléfono móvil de este Proyecto.

## 9.2 Código:

```
#!/bin/bash
#Script de ejecución de la aplicación.
echo "----- EJECUTANDO APLICACIÓN DEL DRONE -----"

# Cargando variables de entorno
echo echo " ----- Cargando variables de entorno -----"
source .env

echo "----- Ejecutando el servidor de control -----"
PYTHON_COMMAND_CONTROL_SERVER="python3 $CONTROL_SERVER_PATH/main.py"
ENV_COMMAND="source $ABSOLUTE_PATH_TO_PROJECT/envDrone/bin/activate"
FULL_COMMAND_CONTROL_SERVER="$ENV_COMMAND; $PYTHON_COMMAND_CONTROL_SERVER"
# Creación del comando osascript
OSASCIPT_COMMAND_CONTROL_SERVER="tell application \"Terminal\" to do script
\"$FULL_COMMAND_CONTROL_SERVER\""
# Ejecución del comando osascript
osascript -e "$OSASCIPT_COMMAND_CONTROL_SERVER"

echo "----- Ejecutando la base de datos -----"
# Ejecución de la base de datos:
DATA_BASE_COMMAND="mongod --config $DATA_BASE_PATH/mongod.conf"
# Creación del comando osascript
OSASCIPT_COMMAND_DATA_BASE="tell application \"Terminal\" to do script
\"$DATA_BASE_COMMAND\""
# Ejecución del comando osascript
osascript -e "$OSASCIPT_COMMAND_DATA_BASE"

echo "----- Desplegando la base de datos -----"
sleep 2

# Ejecución de la app de flask.
echo "----- Ejecutando Backend -----"
# Creación de comandos para la nueva terminal
PYTHON_COMMAND="python3 $BACKEND_PATH/app.py"
ENV_COMMAND="source $ABSOLUTE_PATH_TO_PROJECT/envDrone/bin/activate"
FULL_COMMAND="$ENV_COMMAND; $PYTHON_COMMAND"
# Creación del comando osascript
OSASCIPT_COMMAND_BACKEND="tell application \"Terminal\" to do script
\"$FULL_COMMAND\""
# Ejecución del comando osascript
osascript -e "$OSASCIPT_COMMAND_BACKEND"

# Ejecución de la app de react.
echo " ----- Ejecutando Frontend -----"
# Creación de comandos para la nueva terminal
ENV_COMMAND="source $ABSOLUTE_PATH_TO_PROJECT/envDrone/bin/activate"
GO_TO_DIRECTORY="cd $FRONTEND_PATH"
```

```
NPM_COMMAND="npm start"
FULL_COMMAND="$ENV_COMMAND; $GO_TO_DIRECTORY; $NPM_COMMAND"
# Creación del comando osascript
OSASCIPT_COMMAND_FRONTEND="tell application \"Terminal\" to do script
\"$FULL_COMMAND\""
# Ejecución del comando osascript
osascript -e "$OSASCIPT_COMMAND_FRONTEND"

#Ejecución del Servidor de fotos
echo "----- Ejecutando el servidor de stream de fotos -----"
PYTHON_COMMAND_PHOTO_SERVER="python3 $PHOTO_STREAM_SERVER_PATH/main.py"
ENV_COMMAND="source $ABSOLUTE_PATH_TO_PROJECT/envDrone/bin/activate"
FULL_COMMAND_PHOTO_SERVER="$ENV_COMMAND; $PYTHON_COMMAND_PHOTO_SERVER"
# Creación del comando osascript
OSASCIPT_COMMAND_PHOTO_SERVER="tell application \"Terminal\" to do script
\"$FULL_COMMAND_PHOTO_SERVER\""
# Ejecución del comando osascript
osascript -e "$OSASCIPT_COMMAND_PHOTO_SERVER"
```

Código 1. “*launcher.sh*” Código para lanzar el SCV.

### 9.3 Manual de uso

Las dos aplicaciones que se usan en este proyecto dentro del teléfono móvil se pueden ver en la Figura 29. Una de ellas es la aplicación “*DroneApp*”, este es el resultado final de la aplicación *Kotlin* desarrollada para este proyecto. La otra es el cliente VPN de *OpenVPN*. Esta última se debe activar antes de usar la aplicación “*DroneApp*”. Es importante aclarar que se puede utilizar la aplicación “*DroneApp*” sin haber activado previamente la *VPN*, lo único que el tráfico no estará protegido y no sería seguro volar un activo tan peligroso.

Es necesario destacar que el *SCV* y sus servicios deben estar ejecutándose previamente a pulsar la aplicación de “*DroneApp*”. Para ello se ha creado un script en *bash* (Código 1 del anexo) con el cual a partir de variables de entorno de las rutas de cada servicio y de la creación de entornos virtuales con las dependencias de cada programa se ejecutan de manera ordenada cada uno de los servicios del *SCV*. Este programa se ejecuta en la terminal mediante el comando “*./launcher.sh*” y creará varias terminales, una para cada servicio, en donde se mostrarán mensajes de depuración de cada uno de los servicios.

Una vez que el *SCV* esta preparado se abre la aplicación “*DroneApp*” y el dron empezará su funcionamiento. Se abrirá una pantalla mostrando lo que está enfocando la

cámara (como por ejemplo lo visto en la Figura 20). Si se quiere cambiar la cámara antes del vuelo simplemente habría que pulsar sobre el botón “switch”.

En este momento el dron ya está enviando todos los datos que está capturando hacia el *SVC*. El segundo paso es amarrar el teléfono móvil funcionando al armazón del dron mediante la cincha blanca que se ve en la Figura 14. Posteriormente se encenderá el dron mediante el interruptor situado en la parte izquierda trasera del dron (Figura 22) y cada rotor aleatoriamente emitirá un pitido cada 1 segundo. Aquí el dron empieza su proceso de calibración y sincronización entre los 4 rotores. Este proceso termina cuando se escuchan 4 pitidos seguidos al unísono por los cuatro rotores. Una vez que esto concluya el dron ya estará preparado para recibir órdenes de vuelo.