

**UNIVERSIDAD NACIONAL DE CHIMBORAZO**



**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES**

**“Proyecto de investigación previo a la obtención del Título de  
Ingeniero en Electrónica y Telecomunicaciones”**

**TRABAJO DE TITULACIÓN**

**DISEÑO CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN  
CUADRICÓPTERO AUTÓNOMO DE VIDEOVIGILANCIA EN LA CDLA  
SIXTO DURÁN DE LA CIUDAD DE RIOBAMBA.**

**AUTOR:**

**RAÚL ARTURO SAMANIEGO TELLO**

**DIRECTOR:**

**Mgs. GIOVANNY CUZCO**

**Riobamba - Ecuador**

**AÑO 2018**

## **INFORME DEL TUTOR**

Yo, Ing. Edgar Giovanni Cuzco Silva, en mi calidad de tutor del trabajo de graduación titulado: **“DISEÑO CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN CUADRICÓPTERO AUTÓNOMO DE VIDEOVIGILANCIA EN LA CDLA SIXTO DURÁN DE LA CIUDAD DE RIOBAMBA”**, tengo a bien informar que el mencionado trabajo cumple con los requisitos exigidos para ser expuesto al público, luego de ser evaluado por el tribunal designado.

Riobamba, febrero del 2018

Atentamente



Ing. Giovanni Cuzco.

## CERTIFICACIÓN

Los miembros del Tribunal de Graduación del proyecto de investigación de título:

**“DISEÑO CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN CUADRICÓPTERO AUTÓNOMO DE VIDEOVIGILANCIA EN LA CDLA SIXTO DURÁN DE LA CIUDAD DE RIOBAMBA”**

Presentado por:

Raúl Arturo Samaniego Tello.

Y dirigido por:

Mgs. Edgar Giovanni Cuzco Silva.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en el cual se ha conestado el cumplimiento de las observaciones realizadas, remite el presente para uso y custodia en la biblioteca de la facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Giovanni Cuzco  
**Director del Proyecto**



Firma

Ing. Juan Carlos Cepeda  
**Miembro del Tribunal**



Firma

Ing. Cristian Márquez  
**Miembro del Tribunal**



Firma

## AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, corresponde exclusivamente a: Raúl Arturo Samaniego Tello y Mgs Giovanni Cuzco Silva; y el patrimonio intelectual del mismo a la Universidad Nacional de Chimborazo”.

Autor:



Raúl Samaniego Tello

0604264937

## **Agradecimiento**

*Quiero agradecer en primer lugar a Dios, por guiarme en el camino y fortalecerme espiritualmente para lograr culminar de manera exitosa esta etapa de mi vida profesional.*

*A mi amada esposa Anais y mis hijos André y Gael, gracias a su sacrificio y apoyo fueron el motor que me dio fuerzas para continuar todos los días y alcanzar esta meta.*

*A mis padres Raúl y Mariana que gracias a su constante ayuda fueron un pilar fundamental para la realización de este proyecto.*

*Al Mgs Giovanni Cuzco quien con su conocimiento y guía fue una pieza clave para la elaboración de cada etapa del presente trabajo.*

## **Dedicatoria**

*Es mi deseo como sencillo gesto de agradecimiento, dedicarle el presente Proyecto de Grado, a mi Esposa Anais por su amor, permanente apoyo y comprensión.*

*A mis hijos André y Gael, a mis padres Raúl y Mariana, a mis suegros Julio y Marta, quienes permanentemente me apoyaron con espíritu alentador, contribuyendo incondicionalmente a lograr las metas y objetivos propuestos.*

# ÍNDICE

ÍNDICE.....	vi
ÍNDICE DE FIGURAS .....	viii
ÍNDICE DE TABLAS.....	ix
RESUMEN .....	x
CAPÍTULO I.....	1
1. INTRODUCCIÓN.....	1
1.1. ESTADO DEL ARTE .....	2
1.2. PROBLEMA.....	3
CAPITULO II.....	5
2. OBJETIVOS.....	5
CAPITULO III .....	6
3. FUNDAMENTACIÓN TEÓRICA.....	6
3.1. Tipos de drones según los métodos de generación de sustentación .....	6
3.2 Tipos de Drones según el tipo de ala.....	6
3.2.1 Drones de Ala Fija.....	7
3.2.3 Drone de Ala Rotatoria.....	8
3.4. Cuadricópteros.....	12
3.5 Componentes de un Cuadricóptero.....	13
3.5.1 Pixhawk .....	13
3.5.2 Características Pixhawk.....	13
3.5.3. Motores brushless .....	14
3.5.4. Controlador de velocidad. ....	14
3.6 Protocolo MAVlink. ....	15
3.6.1 Paquete MAVlink .....	15
3.6.2 Mensajes MAVLINK .....	16
CAPÍTULO IV .....	17
4. METODOLOGÍA.....	17
4.1 TIPO DE ESTUDIO.....	17
4.2 Técnicas.....	17
4.3 Hipótesis .....	17
4.3.1 Identificación de variables.....	17
4.3.1.1 Variable independiente .....	17
4.3.1.2 Variable dependiente .....	17
4.4 Operacionalización De Las Variables .....	18

4.5 Población .....	18
4.6 Arquitectura del sistema .....	18
4.6 Modelamiento y simulación del chasis.....	2
4.6.1 Cinemática de traslación.....	4
4.6.2 Cinemática rotacional.....	5
4.7. Diseño del cuadricóptero .....	5
4.7.2. Componentes del chasis .....	6
4.8 Estación Base.....	7
Capítulo V .....	12
5. Resultados.....	12
5.1 Pruebas de autonomía.....	14
5.2 Pruebas de Mantenimiento de Posición.....	15
5.3 Pruebas de Transmisión de Telemetría y Video .....	17
5.4 Misiones de patrullaje.....	18
5.4.1 Protocolo Waypoint.....	19
5.4.2 Envío de waypoints al cuadricóptero.....	19
5.4.3 Resultados misiones de patrullaje .....	20
Capítulo VI.....	23
6.1. Conclusiones.....	23
6.2. Recomendaciones.....	24
Capítulo VI.....	25
Bibliografía.....	25
ANEXOS .....	27
ANEXO 1 .....	28
ANEXO 2 .....	34
ANEXO 3 .....	38

## ÍNDICE DE FIGURAS

Figura 1. Clasificación de Drones según la sustentación .....	6
Figura 2. Drone Ala alta .....	7
Figura 3. Drone Ala media .....	7
Figura 4. Drone Ala baja .....	8
Figura 5. Drone Ala volante .....	8
Figura 6. CAMCOPTER® S-100.....	9
Figura 7. Drone con un único rotor .....	9
Figura 8. Drone coaxial .....	10
Figura 9. DP-12 Rhino UAS .....	10
Figura 10. Configuración multirrotores.....	11
Figura 11. Fuerzas que actúan sobre en cuadricóptero.....	12
Figura 12. PIXHAWK.....	13
Figura 13. DJI MOTOR 920KV.....	14
Figura 14. Esc EMAX 4 in 1 .....	14
Figura 15. Componentes de un mensaje Mavlink .....	15
Figura 16. Estructura XML de un mensaje Mavlink.....	16
Figura 17. Arquitectura del sistema.....	18
Figura 18. Sistemas de referencia considerados en el cuadricóptero. ....	3
Figura 19. Esquema de conexiones eléctricas .....	6
Figura 20. Brazo Cuadricóptero .....	6
Figura 21. Diseño placa central .....	7
Figura 22. Prototipo terminado.....	7
Figura 23. Diagrama de flujo de la estación base.....	8
Figura 24. Botón de pánico en Android .....	10
Figura 25. Interfaz gráfica de usuario de la estación base.....	10
Figura 26. Secuencia de despegue.....	11
Figura 27. Estabilización eje X .....	12
Figura 28. Estabilización eje Y .....	12
Figura 29. Estabilización eje Z.....	13
Figura 30. Respuesta Roll.....	13
Figura 31. Respuesta Pitch .....	13
Figura 32. Respuesta Yaw .....	14
Figura 33. Pruebas de autonomía de la batería.....	15
Figura 34. Variación de la posición respecto a home.....	16
Figura 35. Comparación mantenimiento de altura .....	16
Figura 36. Distancia máxima enlace de telemetría.....	17
Figura 37. distancia máxima enlace de video.....	17
Figura 38. Proceso de carga de misión .....	19
Figura 39. Trayectoria de patrullaje .....	20

## ÍNDICE DE TABLAS

Tabla 1. Parámetros MAVLink .....	15
Tabla 2. Operacionalización De Las Variables .....	18
Tabla 3. Variables del chasis de cuadricóptero .....	4
Tabla 4. Componentes a bordo del cuadricóptero .....	5
Tabla 5. Características enlace de telemetría.....	8
Tabla 6. Trama enviada desde el Smarthphone .....	9
Tabla 7. Parámetros del chasis .....	14
Tabla 8. Parámetros Pruebas de autonomía.....	15
Tabla 9. Características del transmisor.....	18
Tabla 10. Calidad de imagen vs distancia .....	18
Tabla 11. Parámetros misión de patrullaje .....	20
Tabla 12. Resumen de misiones realizadas .....	20
Tabla 13. Tabla de frecuencias observadas .....	21
Tabla 14. Parámetros de estimación.....	21
Tabla 15. Calculo del valor empírico .....	22
Tabla 16. Resultados de la prueba Chi para Bondad de Ajuste.....	22

## **RESUMEN**

El presente proyecto muestra el modelado e implementación de un sistema de vigilancia montado sobre un cuadricóptero, que de manera autónoma realiza patrullajes programados utilizando coordenadas UTM, y a su vez está en capacidad de acudir a puntos específicos al ser llamado desde una aplicación instalada en un teléfono móvil, siempre que el vehículo se encuentre dentro del radio de cobertura de la estación base. El procesamiento de la información y la toma de decisiones se realizan en la estación base mientras que el cuadricóptero se encarga de recoger datos y enviarlos utilizando el protocolo Mavlink, además se encarga de transmitir video en tiempo real y toma decisiones cuando ha perdido la comunicación con la estación base. Producto del trabajo propuesto, se logró desarrollar un cuadricóptero que tiene una capacidad de carga de 2,5 kg., el análisis de la cinemática de traslación muestra que el chasis diseñado logra estabilizarse en tiempos menores a 3s lo que denota un alto nivel de control y estabilidad durante el vuelo, en relación con la precisión con respecto a la trayectoria ideal, el cuadricóptero presenta un margen de error menor al 2% que expresado en distancia corresponde a un rango entre 1 a 2 m respecto a la trayectoria ideal.

**Palabras Clave:** Cuadricóptero, vigilancia, autónomo, protocolo Mavlink.

## ABSTRACT

This project shows the modeling and implementation of a surveillance system mounted on a quadcopter, which autonomously performs programmed patrols using UTM coordinates, and it can go to specific points when called from an application installed on a smartphone, as long as it is within the radius of coverage of the base station. Information processing and decision making are carried out at the base station, while the quadcopter is responsible for collecting data and sending them using the Mavlink protocol. The quadcopter is also responsible for transmitting real time video and decision making when they have lost communication with the base station. Product of the proposed work, it was possible to develop a quadcopter that has 2.5 kg. of payload, the analysis of the kinematics of translation shows that the designed chassis manages to stabilize in times less than 3s which denotes a high level of control and stability during the flight, in relation to the precision with respect to the ideal trajectory, the quadcopter has a margin of error of less than 2% that expressed in distance corresponds to a range between 1 to 2 m with respect to the ideal trajectory.

**Keywords:** Quadcopter, surveillance, autonomous, Mavlink.

Translation reviewed by: Fuertes, Narcisca  
Language Center Teacher



# CAPÍTULO I

## 1. INTRODUCCIÓN.

La investigación y el desarrollo de vehículos aéreos no tripulados y micro vehículos aéreos se encuentran en auge, debido a su amplio campo de aplicación, que va desde campos de radio aficionados hasta los que se emplean en operaciones de seguridad y vigilancia, todos los avances en el desarrollo de los cuadricópteros son resultado de la evolución en la mecánica y electrónica que involucra poder realizar el vuelo, en donde se destacan la miniaturización de los componentes y el aumento en la capacidad en las fuentes de energía que se utilizan .

Al no existir la necesidad de una cabina que proteja al piloto, el peso y tamaño del dispositivo son menores brindándole un acceso a sitios inalcanzables para vehículos tripulados. Dentro de los vehículos aéreos no tripulados (UAV), los cuadricópteros o aeronaves de cuatro motores son los más populares, debido a su gran capacidad de carga y alta maniobrabilidad en comparación a otro tipo de UAVs.

El cuadricóptero cambia de dirección manipulando la velocidad de las hélices que producen el empuje de cada uno de los cuatro motores que están sujetos a la estructura, por lo que un cuadricóptero puede ser visto como un sistema en el que existen cuatro fuerzas de entrada provenientes de cada uno de los motores y 6 estados de salida ( $x, y, z, \theta, \psi, \omega$ ). Bajo este esquema las coordenadas XYZ representan la posición del centro de masa de la aeronave en relación con su estructura, mientras que  $\theta, \phi$  y  $\psi$  representan los ángulos de Euler: Roll o  $\theta$ , determina la rotación de cuadricóptero alrededor del eje Y.

Para comandar los UAV, típicamente se hace uso de un radio control, lo que significa que el vehículo es completamente dependiente de un constante control y monitoreo por parte de un operario. Para lograr mantener el vuelo se requiere de un piloto con gran experiencia y habilidad en esta tarea.

En algunos vehículos, los controles ingresados por el operador se han convertido en ajustes encargados de comandar el rumbo de las aeronaves. Este principio es conocido como fly-by-wire. Haciendo uso de este principio el vehículo obtiene un alto grado de autonomía en comparación a los vehículos comandados por control remoto, pero siguen dependiendo de una supervisión por parte de un operador. Un ejemplo de sistemas fly-by-wire puede observarse en los jets de combate como el F-18 Hornet, el cual es capaz

de auto estabilizarse en caso de que el piloto deje de manipular los controles; sin embargo, necesita de la asistencia del piloto para tareas como dirigirse a un punto en específico.

Para considerar que un UAV es autónomo, este debe tener incorporado un sistema de inteligencia de navegación y control que le dé la capacidad de ser operado por si solo es decir poder realizar tareas como despegue, aterrizaje, dirigirse hacia un lugar deseado y además el vehículo puede ser operado mediante simples comandos e instrucciones.

Tomando como base la capacidad de mantener el vuelo estacionario, un cuadricóptero se vuelve una excelente plataforma sobre la cual se pretende implementar un sistema de vigilancia basado en video transmitido en tiempo real, mismo que será capaz de realizar su misión de manera autónoma, es decir deberá despegar, trazar su trayectoria, dirigirse a un punto geográfico definido y retornar al punto de despegue automáticamente.

Al estar este vehículo en fase de prototipo es necesario tener actualizada la información durante el vuelo para así poder tomar decisiones en caso que se presente alguna falla en los sistemas empleados en el vuelo para lograr este flujo de información se hará uso del protocolo de comunicaciones MAVlink el cual es comprendido nativamente por la tarjeta controladora de vuelo, además el propósito de crear este canal de comunicación será el poder cambiar de objetivos durante el vuelo sin necesidad de que el vehículo retorne a la base para ser reprogramado.

### **1.1. ESTADO DEL ARTE**

Las soluciones basadas en UAV han experimentado un aumento sustancial en la última década, especialmente en los últimos cinco años. Ya en 2004, los expertos de la NASA definieron un amplio conjunto de aplicaciones civiles para vehículos aéreos no tripulados (T. H. Cox, 2004), destacando su potencial en un futuro cercano en áreas como comercial, ciencias de la tierra, seguridad nacional y gestión de la tierra. Este informe preliminar fue ratificado años más tarde por autores como Hugenholtz . (C. H. Hugenholtz, 2012), quien explicó cómo el uso de UAV podría revolucionar los métodos de investigación en los campos de las Ciencias de la Tierra y la teledetección. En (Pajares, 2015), los autores muestran los resultados de un estudio detallado sobre diferentes aspectos de los vehículos aéreos no tripulados, mostrando su aplicabilidad en agricultura y silvicultura, monitoreo de desastres, localización y rescate, vigilancia, monitoreo ambiental, monitoreo de la vegetación, fotogrametría, y más. Enfocándose específicamente en la investigación utilizando cuadricópteros o multicópteros, autores como Gupte (S. Gupte, 2012) y

Colomina & Molina consideran que, dada su alta maniobrabilidad, compacidad y versatilidad, se encuentran diferentes aplicaciones para estos dispositivos en áreas que incluyen ingeniería civil, búsqueda y rescate, respuesta de emergencia, seguridad nacional y patrullaje fronterizo, así como en otras áreas como Ciencias de la Tierra, donde se pueden utilizar los UAV para estudiar el cambio climático, la dinámica de los glaciares y la actividad volcánica o para el muestreo atmosférico, entre otros.

Hay trabajos centrados en el uso de vehículos aéreos no tripulados para tareas específicas que involucran movimientos autónomos. Un ejemplo es (J. Sanchez-Garcia, 2015), donde los autores presentan un modelo de movilidad para el autodespliegue de una red aérea ad hoc en un escenario de desastre con el fin de crear una infraestructura de comunicaciones flexible que las víctimas puedan usar. El modelo de movilidad propuesto se basa principalmente en la métrica de similitud de Jaccard para controlar el despliegue de los vehículos aéreos no tripulados que componen la red. Un trabajo similar se presenta en (O. Briante, 2015), donde en cambio se usa un análisis de densidad dentro de la red para seleccionar las áreas físicas que necesitan ser visitadas por un robot volador.

Centrándose exclusivamente en las propuestas existentes que abordan los modelos de movilidad, se encontró trabajos como (O. Bouachir, 2014) donde los autores proponen el Modelo de Movilidad de Paparazzi (PPRZM) definiendo cinco tipos de movimientos: Permanecer, Waypoint, Ocho, Escanear y Ovalado, siguiendo una máquina de estado definida con diferentes probabilidades para cambiar entre estados. Incluso hay estudios siguiendo patrones de navegación basados en animales. Un ejemplo de este tipo de trabajo es (P. Basu, 2004), donde los autores investigan la colocación de UAV y las estrategias de navegación con el objetivo final de mejorar la conectividad de la red, utilizando reglas locales de flocking que los seres vivos aéreos como pájaros e insectos suelen seguir.

Sin embargo, el uso de cuadricópteros para realizar tareas de video vigilancia de manera autónoma en un área específica no está presente en la literatura científica, y este trabajo puede verse como uno de los primeros enfoques en esta dirección.

## **1.2. PROBLEMA**

La vigilancia puede definirse como una actividad en donde se observa a las personas, grupos de personas, comportamientos, actividades, edificios, etc., con el propósito de protegerlos o dirigirlos según sea el caso. Existen varios métodos diferentes de vigilancia,

incluyendo rastreo satelital, cámaras de observación, sistemas de vigilancia biométrica entre otros.

Los métodos de vigilancia convencionales se encuentran limitados típicamente por la naturaleza estática de las cámaras que los componen, ya que usualmente se encuentran fijados a un trípode u otro tipo de estructura. La vigilancia aérea puede ser realizada por aeronaves como aviones o helicópteros, este método de vigilancia cumple con el resultado deseado, pero a costa de un elevado presupuesto de mantenimiento y despliegue.

El problema se evidencia en la Ciudadela Sixto Duran Ballén, donde no existe un sistema video vigilancia desplegado capaz de cubrir toda el área que comprende el sector en cuestión. Actualmente la principal referencia en cuanto a video vigilancia que se encuentra disponible es una cámara del Sistema Integrado de Seguridad ECU 911, misma que se encuentra ubicada en la intersección de la avenida 9 de octubre y Madrid, tomando en cuenta la delimitación de la ciudadela, esta cámara se ubica en el extremo noreste de la ciudadela, lo que deja prácticamente toda el área fuera del alcance de la cámara. Es por estas razones que un sistema basado en un vehículo aéreo no tripulado provee una solución ideal a los problemas y limitaciones de los otros métodos de vigilancia.

Por tal motivo, se justifica el presente estudio para la implementación y utilización de un robot cuadricóptero que integre un sistema de videovigilancia, mismo que deberá realizar patrullajes autónomos y acudir a un lugar específico en caso de que se requiera.

## **CAPITULO II**

### **2. OBJETIVOS.**

#### **2.1. OBJETIVO GENERAL.**

Diseñar, construir e implementar un cuadricóptero, para realizar tareas de video vigilancia de manera autónoma en la Ciudadela Sixto Duran.

#### **2.2. OBJETIVOS ESPECIFICOS.**

- Desarrollar un modelo matemático que permita representar los parámetros necesarios para que el cuadricóptero pueda lograr un vuelo estable.
- Simular el comportamiento del modelo variando los parámetros para obtener las características físicas mínimas con las que debe contar el cuadricóptero para mantener un vuelo estable.
- Analizar los componentes mínimos con los que debe contar el vehículo para poder realizar su misión.
- Diseñar una interfaz que sea capaz de comunicar el cuadricóptero con la estación base con la finalidad de visualizar los datos de telemetría y comandar la aeronave.

## CAPITULO III

### 3. FUNDAMENTACIÓN TEÓRICA.

#### 3.1. Tipos de drones según los métodos de generación de sustentación

Los drones tienen un gran número de configuraciones y campos de aplicación en los que pueden ser empleados, debido a estos factores, no es posible aplicar una clasificación general, sino se suele utilizar categorizaciones en función de características en común entre las aeronaves.

Para ilustrar de mejor manera las características de los drones, se realizará la clasificación tomando en cuenta como la aeronave genera la sustentación necesaria para mantener un vuelo constante, utilizando este método de clasificación, los drones se clasifican en 2 grandes grupos que son los aerodinos o drones más pesados que el aire y los aerostatos que logran levantar el vuelo haciendo uso de gases más livianos que el aire. En la figura 1 se presenta el esquema de dicha clasificación.

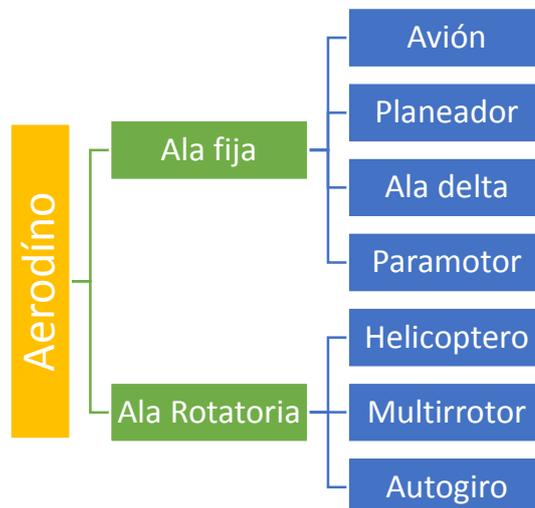


Figura 1. Clasificación de Drones según la sustentación  
Fuente: Autor

#### 3.2 Tipos de Drones según el tipo de ala.

Centrándose en los drones aerodinos, se puede clasificar las aeronaves dependiendo como se genera la sustentación necesaria para mantenerse en el aire teniendo la siguiente clasificación:

- Drones de Ala Fija
- Drones de Ala Rotatoria
- Drones Híbridos

### 3.2.1 Drones de Ala Fija

Este tipo de aeronaves generan la sustentación utilizando el principio de planos, en los que se crea un perfil aerodinámico que genera una diferencia de presión entre la parte superior e inferior del ala; en este tipo de configuración se utiliza estructuras de geometría fija que se encuentran pegadas al cuerpo del dron.

Los drones de ala fija se catalogan dependiendo en que parte del fuselaje se encuentra empotrada el ala; en donde se encuentra la siguiente clasificación:

#### Drones de ala alta

Con sus alas ubicadas en la parte superior del fuselaje, este dron tiene una gran estabilidad en el aire, aunque sacrificando maniobrabilidad como se muestra en la figura 2.



Figura 2. *Dron Ala alta*  
Fuente: (Paul Ghering Falstrom, 2012)

#### Drones de ala media

Este tipo de dron cuenta con las alas empotradas en la mitad de la aeronave, esta configuración le otorga prestaciones medias tanto en estabilidad como en maniobrabilidad, este tipo de estructura se la puede apreciar en la figura 3.



Figura 3. *Dron Ala media*  
Fuente: (Paul Ghering Falstrom, 2012)

#### Drones de ala baja

La figura 4 muestra un dron con el ala ubicada en la parte inferior del fuselaje, con lo que el dron presenta grandes prestaciones en cuanto a maniobrabilidad.



Figura 4. *Drone Ala baja*  
Fuente: (Paul Ghering Falstrom, 2012)

### **Drones ala volante**

En este tipo de dron e el ala se ha fusionado totalmente con el fuselaje de la aeronave como se muestra en la figura 5, gracias a esta singular forma este tipo de aeronaves tienen una muy baja resistencia aerodinámica y un elevado nivel de maniobrabilidad.



Figura 5. *Drone Ala volante*  
Fuente: (Paul Ghering Falstrom, 2012)

La característica principal de los drones que se sustentan por medio de alas fijas, es la elevada eficiencia aerodinámica lo que les proporciona un excelente rendimiento energético es decir pueden lograr prolongados tiempos de vuelo, este método de sustentación también les permite tener fuselajes sencillos y de fácil construcción. En contraparte este tipo de estructura requiere de elementos adicionales para realizar tareas como despegar o aterrizar, además este tipo de aeronaves no pueden lograr tareas complejas como vuelos estacionarios o en reversa, lo que les impide desplegarse en lugares que exijan complejidad en maniobras o espacios reducidos.

### **3.2.3 Drones de Ala Rotatoria.**

Este tipo de aeronaves sustituyen las alas por hélices, mismas que al girar a altas velocidades alrededor de un eje produce la sustentación necesaria para elevar el dron e;

este método de sustentación presenta varias configuraciones dependiendo el número de rotores utilizados se dividen en los siguientes tipos:

### **Drones con un motor principal y un motor de cola**

En la figura 5 se muestra un drone donde la sustentación se genera por un motor unido a una hélice de gran tamaño que se encuentra en la parte superior del fuselaje mientras que el rotor de cola sirve para evitar que el drone gire descontroladamente sobre su propio eje, con esta configuración se obtiene buenas prestaciones de control y estabilidad a costa de un diseño mecánico más complejo.



Figura 6. CAMCOPTER® S-100  
Fuente: (Corporation, 2017)

### **Drone con un único rotor**

Dotado de un único motor ubicado en su parte central como se observa en la figura 7 este tipo de drone tiene una alta eficiencia energética; para compensar la fuerza de torque hace uso de alerones evitando que gire sobre su propio eje, con este tipo de configuración se obtiene una estructura relativamente más simple que otros drones de ala fija a costa de una baja capacidad de control.

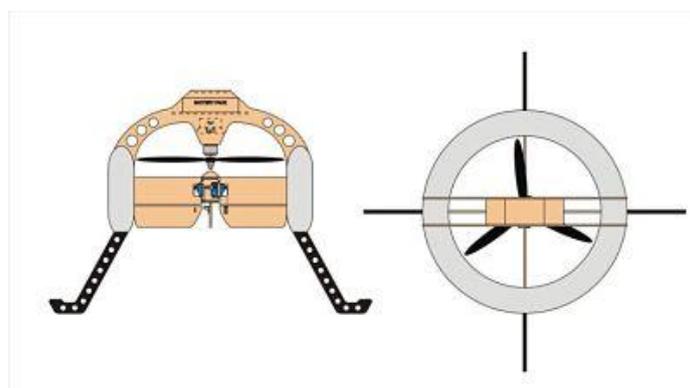


Figura 7. Drone con un único rotor  
Fuente: (Team, 2016)

## Drone coaxial

En la figura 8 se muestra un drone que tiene colocados sus rotores uno sobre el otro y sus ejes se encuentran en ubicados de forma coaxial es decir uno dentro del otro , estos rotores giran en direcciones contrarias para que las sumatoria de los torques sea igual a 0, utilizando esta forma de sustentación se consigue estructuras compactas pero a costa de un elevado desperdicio de energía ya que el empuje de un rotor se ve interferido por el otro.

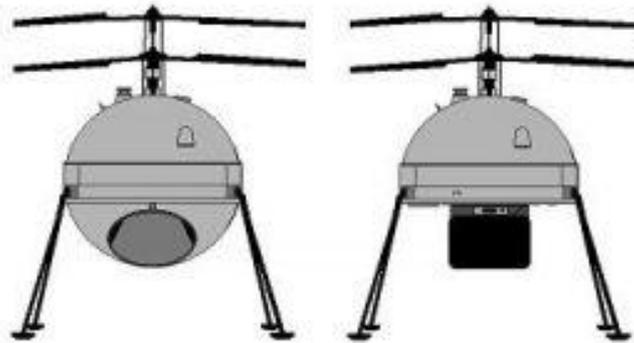


Figura 8. *Drone coaxial*  
Fuente: (Team, 2016)

## Drones con dos rotores en configuración tándem

La configuración tándem utiliza dos rotores dispuestos a lo largo del eje longitudinal del fuselaje como se aprecia en la figura 9, al igual que el drone coaxial los rotores giran en sentidos opuestos para eliminar el torque producido, este tipo de aeronaves tienen una alta capacidad de carga, pero requieren de mecanismos más complejos para su funcionamiento.



Figura 9. *DP-12 Rhino UAS*  
Fuente: (AviationWeek, 2017)

## Drones Multirrotores

Un dron se categoriza como multirrotor cuando en su fuselaje presenta mas de 2 rotores, a este tipo de drones se los llama por lo general por el numero de rotores que utilizan para lograr el vuelo que van desde los equipados con tres rotores o tricópteros, pasando por los de cuatro rotores o cuadricópteros; hasta los que requieren de 8 o más rotores para volar; la figura 10 muestra la mayoría de configuraciones conocidas para multirrotores.

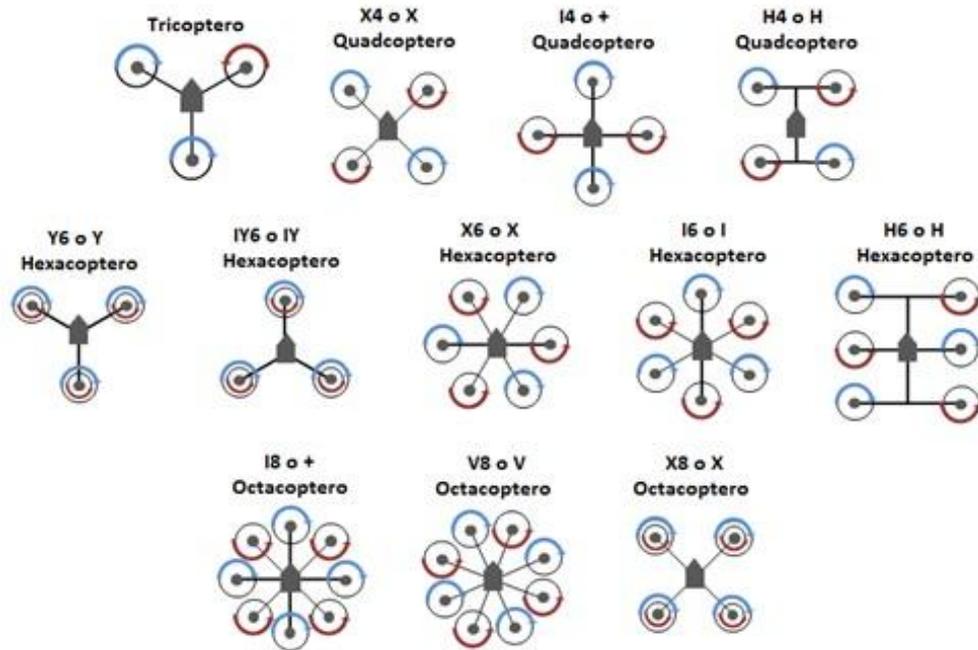


Figura 10. Configuración multirrotores  
Fuente: (Team, 2016)

La principal particularidad de los multirrotores es que utilizan hélices de paso fijo para generar la sustentación necesaria para volar, además sus rotores son de revolución variable, el giro de los rotores es opuesto para contrarrestar los torques producidos por cada motor; este tipo de configuración les confiere altas prestaciones tanto en maniobrabilidad como en control este tipo de drones tienen la capacidad de realizar movimientos instantáneos en cualquiera de los ejes cartesianos además de tener la capacidad de permanecer en vuelo estacionario, gracias a su versatilidad son del tipo de drones más utilizados.

Los multirrotores también son llamados vehículos VTOL o vehículos de despegue y aterrizaje vertical, esta habilidad les permite desplegarse en casi cualquier superficie sin necesidad de aditamentos o infraestructuras adicionales; otra característica especial de estos drones es la capacidad de volar lentamente y a muy baja altura por lo que son utilizados en aplicaciones como reconocimiento de terrenos y topología; en contraparte a

todas estas ventajas este tipo de aeronaves tiene un elevado consumo energético lo que restringe su tiempo de vuelo y área de cobertura.

### 3.4. Cuadricópteros

La principal característica de un cuadricóptero es la disposición simétrica de los rotores con respecto al centro de gravedad, para su funcionamiento se usan cuatro motores con hélices, dos giran en el sentido de las agujas del reloj y las otras dos en sentido contrario como se muestra en la figura 11. Las cuatro hélices trabajan al mismo tiempo para crear la fuerza de empuje necesaria para llevar al vehículo hacia arriba. Variando el empuje que se ejerce con cada hélice se puede conseguir una estabilidad completa del aparato. El peso total del cuadricóptero se divide entre cada uno de los motores y modificando el par de cada uno de ellos, es posible controlar el vuelo.

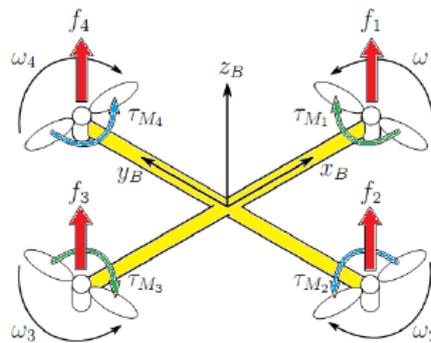


Figura 11. Fuerzas que actúan sobre un cuadricóptero  
Fuente: Autor

Un cuadricóptero tiene cuatro tipos de movimiento: guiñada (hacia la derecha o izquierda del eje vertical), alabeo (hacia la derecha o izquierda del eje longitudinal), cabeceo (rotación hacia delante o atrás con respecto al eje transversal) y altitud (elevación en vertical).

Estos movimientos están controlados por la variación o el ajuste de la propulsión en cada hélice. Por ejemplo, si se desea que el cuadricóptero se incline o cabecee, se debe incrementar la propulsión de una de las hélices y al mismo tiempo reducir la propulsión de la hélice opuesta.

Esto debe hacerse de forma simultánea y al mismo nivel lo que producirá que la fuerza del vector se divida en dos componentes, la horizontal y la vertical, que tendrá dos consecuencias: La primera es que el cuadricóptero empezará a volar en sentido contrario

al componente horizontal, y la segunda es que el componente vertical será más pequeño causado por la disminución de la fuerza del vector. Esto provocará el descenso vertical en el funcionamiento del cuadricóptero. Para que esto no ocurra, es necesario incrementar la propulsión de las cuatro hélices.

### 3.5 Componentes de un Cuadricóptero.

#### 3.5.1 Pixhawk

Pixhawk es un proyecto independiente, en hardware abierto cuyo objetivo es proporcionar un dispositivo de hardware para piloto automático de gama alta a bajo costo y alta disponibilidad a la academia, para hobby y comunidades industriales. PIXHAWK es uno de los proyectos que constituyen la Fundación DroneCode de Linux. Es apoyado por el Proyecto PIXHAWK de Computer Vision y Geometry Lab de ETH Zurich (Instituto Federal Suizo de Tecnología), por el Laboratorio de Sistemas Autónomos y el Laboratorio de Control Automático, y también de 3D Robotics

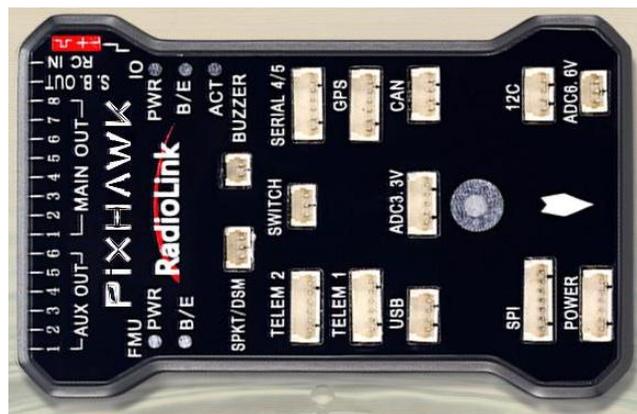


Figura 12. PIXHAWK  
Fuente: (Team, 2016)

#### 3.5.2 Características Pixhawk

Pixhawk es un piloto automático de código abierto para modelos de radio control, bien sean multirrotores, aviones o vehículos terrestres. Dentro de sus características fundamentales cuenta con:

- Tiene un procesador de 32 bits ARM Cortex con un S.O. NuttX RTOS.
- 14 salidas PWM/servo.
- Puertos de conexión UART, I2C, CAN.
- Sistema de copia de seguridad integrado para la recuperación en vuelo y control manual
- Entradas de alimentación redundantes y conmutación automática ante error.

- Botón externo de seguridad para la activación del motor.
- Indicador LED multicolor.

### 3.5.3. Motores brushless

La característica principal de los motores brushless es que realiza la misma función de un motor DC normal, pero reemplazando el conmutador y las escobillas por switches de estado sólido que funcionan con una lógica para la conmutación de los embobinados.

Se puede concluir que la gran ventaja de los motores brushless, frente a los demás motores DC es baja periodicidad de mantenimiento.



Figura 13. DJI MOTOR 920KV  
Fuente (DJI, 2017)

### 3.5.4. Controlador de velocidad.

Estos controladores son drivers trifásicos usados como interface entre la salida del controlador de vuelo y los motores, permitiendo controlar la velocidad de estos. Esto se consigue modificando el ancho del pulso PWM de entrada (con un periodo estándar de 50 Hz) es decir entre 1 y 2 milisegundos.



Figura 14. Esc EMAX 4 in 1  
Fuente (EMAXmodel, 2017)

### 3.6 Protocolo MAVlink.

MAVlink es un protocolo de comunicación especialmente desarrollado para el intercambio de información entre las estaciones de base en tierra y el UAV. Dicha información se encuentra contenida en ficheros con extensión .xml, lo que permite que esta pueda ser estructurada y empaquetada empleando diferentes lenguajes de programación, consiguiendo así a facilitar el desarrollo de soluciones software para drones.

MAVlink es muy fácil de integrar en cualquier sistema tanto de en el vehículo como en la estación de control ya que la definición de los mensajes se realiza mediante cabeceras en lenguaje C. A pesar de que la librería contiene una útil variedad de mensajes, es posible crear nuevos mensajes personalizados. MAVlink posee licencia LGPL lo que posibilita su uso tanto en proyectos close-source como open-source.

#### 3.6.1 Paquete MAVlink

El paquete MAVlink consiste en una secuencia codificada de bytes. Los protocolos Controller Area Network bus (CAN bus) y SAE AS-4 sirvieron de inspiración para la creación de la estructura de los paquetes de este protocolo para drones. En la siguiente figura se presenta la estructura de un paquete MAVlink.



Figura 15. Componentes de un mensaje Mavlink  
Fuente: (QGroundControl, 2017)

Como se puede observar en la figura anterior, la cabecera está conformada por un conjunto de 6 bytes que indican diferentes metadatos asociados al mensaje. En la tabla 1 se explica la función de cada byte de la cabecera:

Tabla 1. Parámetros MAVLink  
Fuente Autor

Byte	Explicación
Inicio	Este byte indica el comienzo del paquete MAVLink.
Longitud	Indica el número de bytes que se van a enviar como payload del mensaje. Puede tomar un valor entre 0 y 255.
Número de secuencia	Muestra el número de paquete enviado según su pertenencia a la secuencia de los mismos con el objetivo de reordenar éstos en el destino o detectar paquetes perdidos. Puede tomar un valor entre 0 y 255.
Identificación del sistema	Identifica el sistema, mediante un número natural, que ha enviado el mensaje permitiendo diferenciar entre varios sistemas de la misma red. Puede tomar un valor entre 0 y 255.

Identificación del componente del sistema	Señala, mediante un número natural, el componente de dicho sistema que ha enviado el mensaje permitiendo de este modo diferenciar varios componentes dentro del mismo sistema. Puede tomar un valor entre 0 y 255.
Identificación del mensaje	Indica el tipo de mensaje enviado, señalando cómo se tiene que leer el payload que viene a continuación. Utilizando un símil, es una leyenda que indica cómo se debe leer y traducir la carga útil del mensaje. Puede tomar un valor entre 0 y 255.

En el Payload o carga útil del paquete es donde se encuentra la información que se desea transmitir. Su tamaño se especifica en el segundo byte de la cabecera correspondiente a la longitud del mensaje. Puede transportar desde 0 hasta 255 bytes de datos. El control de errores se realiza gracias a la información del Checksum.

### 3.6.2 Mensajes MAVLINK

Por otra parte, todos los mensajes creados usando este protocolo poseen el siguiente formato dentro de cada archivo xml:

```

-<message id="0" name="HEARTBEAT">
-<description>
  The heartbeat message shows that a system is present and responding. The type of the MAV and Autopilot hardware allow the
  interface based on the autopilot).
</description>
-<field type="uint8_t" name="type">
  Type of the MAV (quadrotor, helicopter, etc., up to 15 types, defined in MAV_TYPE ENUM)
</field>
-<field type="uint8_t" name="autopilot">
  Autopilot type / class, defined in MAV_AUTOPILOT ENUM
</field>
-<field type="uint8_t" name="base_mode">
  System mode bitfield, see MAV_MODE_FLAG ENUM in mavlink/include/mavlink_types.h
</field>
<field type="uint32_t" name="custom_mode">A bitfield for use for autopilot-specific flags.</field>
<field type="uint8_t" name="system_status">System status flag, see MAV_STATE ENUM</field>
-<field type="uint8_t" mavlink version" name="mavlink version">
  MAVLink version, not writable by user, gets added by protocol because of magic data type: uint8_t_mavlink_version
</field>
</message>

```

Figura 16. Estructura XML de un mensaje Mavlink

Fuente: (QGroundControl, 2017)

Como se puede ver en la figura 16, todos los mensajes son encerrados dentro de las etiquetas `<message></message>`. El byte de identificación del mensaje viene representado por el parámetro *id*. El nombre que ha sido dado al mensaje es indicado en el parámetro *name*. Para describir la funcionalidad del mensaje se emplea la etiqueta `<description></description>` parámetro name indica el nombre que toma el mensaje dentro del protocolo MAVLINK.

La etiqueta indica la funcionalidad del mensaje y cualquier información que fuera necesaria para que éste pueda ser entendido por el desarrollador. En cada etiqueta `<type></type>` se indica cada parámetro que necesita el mensaje para su correcta implementación. El campo *type* indica el tipo de dato aceptado por un determinado parámetro del mensaje. Estos parámetros pueden ser de tipo integer de 8, 16, 32 y 64 bit de longitud y de tipo float. El campo *name* indica el nombre del parámetro utilizado dentro del mensaje.

## CAPÍTULO IV

### 4. METODOLOGÍA

#### 4.1 TIPO DE ESTUDIO.

**4.1.1. Descriptivo:** Consiste en realizar una exposición narrativa, numérica y/o gráfica, lo más detallada y exhaustiva posible de la realidad que se investiga.

**4.1.2. Explicativo:** Estudia las causas y efectos que se generan del problema de estudio.

**4.1.3 Cuantitativo:** Proceso formal, sistemático en el que se usan datos numéricos para obtener información sobre el objeto de estudio.

**4.1.4 Experimental:** Evalúa los efectos de la intervención de un sistema, ya sea preventiva, curativa o rehabilitadora.

**4.1.5 Experimental Puro:** Permite la distribución de usuarios en un grupo de control y otro de experimentación.

#### 4.2 Técnicas.

**4.2.1 Observación** Esta técnica consiste en la recolección de información que sea de apoyo para el desarrollo del proyecto, dando las pautas necesarias para el diseño y construcción e implementación de un cuadricóptero autónomo de video vigilancia en la Cdla Sixto Durán de la ciudad de Riobamba.

#### 4.3 Hipótesis

¿El diseño, construcción e implementación de un cuadricóptero autónomo de videovigilancia permitirá realizar misiones de video vigilancia en la Ciudadela Sixto Duran de la ciudad de Riobamba?

#### 4.3.1 Identificación de variables

##### 4.3.1.1 Variable independiente

El diseño, construcción e implementación de un cuadricóptero autónomo de videovigilancia.

##### 4.3.1.2 Variable dependiente

La Transmisión de video y datos de telemetría en Tiempo real.

#### 4.4 Operacionalización De Las Variables

Tabla 2. Operacionalización De Las Variables  
Fuente Autor

VARIABLES	CONCEPTO	INDICADORES	INSTRUMENTOS
Diseño, construcción e implementación de un cuadricóptero autónomo de videovigilancia	Vehículo aéreo no tripulado diseñado para operaciones de videovigilancia	- Diseño del cuadricóptero	- Simulación del comportamiento del vehículo  - Característica de los elementos a bordo
		- Diseño del sistema de videovigilancia	
Transmisión de video y datos de telemetría	Conjunto de datos que muestran la situación actual del cuadricóptero	Recopilación de datos generados por los sensores y cámara durante el vuelo de vuelo	- Pruebas de vuelo - Pruebas de autonomía - Pruebas de transmisión de video

#### 4.5 Población

Debido al tipo de proyecto no es posible aplicar una población de estudio.

#### 4.6 Arquitectura del sistema

Para lograr el objetivo planteado es necesario esquematizar el proyecto para definir los módulos que componen el sistema, la figura 16 muestra en forma general cada uno de los componentes necesarios para que el sistema cumpla con los objetivos planteados.

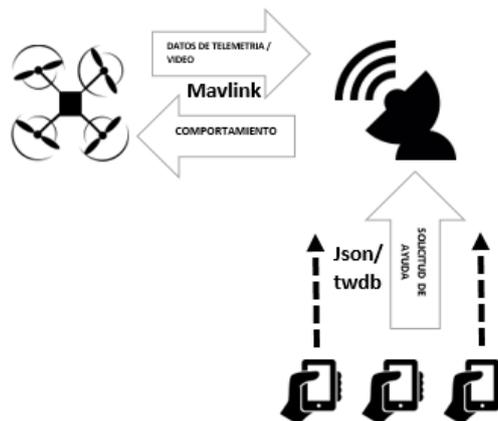


Figura 17. Arquitectura del sistema  
Fuente Autor

La arquitectura del sistema propuesto en la figura 17., muestra la necesidad de mantener el intercambio de información de manera constante y fluida hacia la estación base, motivo por el cual es necesario desarrollar una interface que minimice los tiempos de bloqueo durante la etapa de recolección de datos ya sea desde el cuadricóptero o el Smartphone. Para poder satisfacer esta condición, la interfaz está desarrollada sobre Python, que posee la característica de ser un lenguaje asíncrono, esto significa que las llamadas de las funciones se realizan de manera indistinta teniendo como resultado una interfaz inmune a los bloqueos debido a que los procesos de solicitud de información desde la estación base hacia el resto de componentes del sistema se realiza en segundo plano.

Al tener una capacidad limitada de procesamiento a bordo del cuadricóptero, se requiere tener un canal de comunicación el cual mantenga la información actualizada en tiempo real entre el vehículo y la estación base. La comunicación se encuentra basada sobre el protocolo MAVlink, el cual es comprendido de manera nativa por la tarjeta controladora de vuelo.

#### **4.6 Modelamiento y simulación del chasis.**

Para desarrollar sistemas de control precisos para la plataforma del cuadricóptero, es necesario analizar las ecuaciones del movimiento que definen el sistema, el cuadricóptero se define por un conjunto de ecuaciones no lineales que dificultan la simulación precisa y el control.

Una vez que se desarrollan todos los elementos de las ecuaciones, es posible simplificar las ecuaciones del movimiento, después de hacer varias suposiciones sobre el método en el que el vehículo será operado. El objetivo es hacer un modelo que sea simple y al mismo tiempo lo más realista posible.

Un cuadricóptero funciona con el concepto de torques y empujes variables. Cada rotor se compone de un motor generalmente brushless y una hélice con paso fijo. Los motores están dispuestos en pares a lo largo de los ejes horizontal y vertical, el par vertical se encuentra girando en el sentido de las agujas del reloj mientras que el par horizontal girando en sentido antihorario. Este diseño da como resultado que los torques de reacción de los pares de motores estén exactamente opuestos entre sí, si todos los rotores están girando a la misma velocidad. La eliminación del momento de rotación permite que el vehículo mantenga un rumbo constante durante el vuelo.

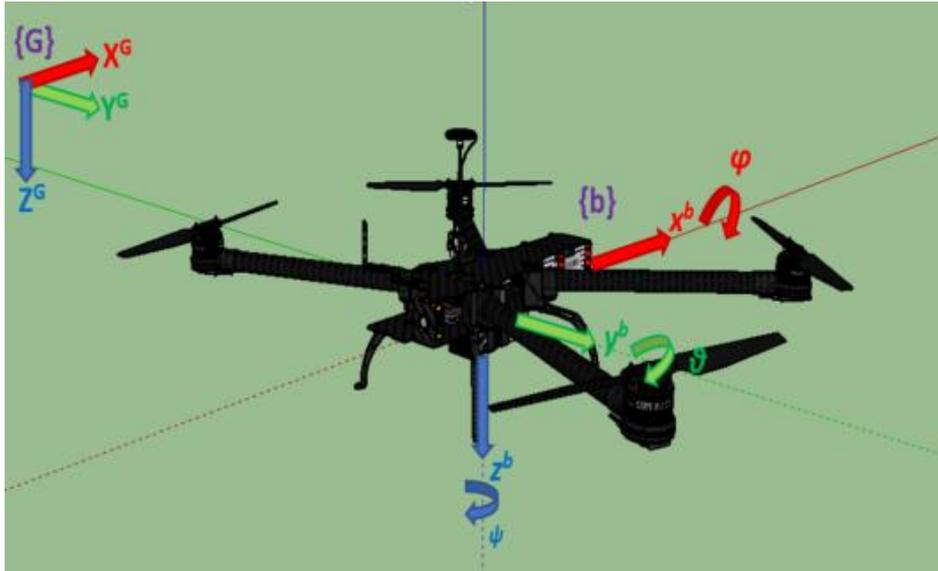


Figura 18. Sistemas de referencia considerados en el cuadricóptero.

Fuente Autor

La estructura básica del cuadricóptero utilizada para el desarrollo del modelo se muestra en la figura 18, en la que se representan los ángulos roll, pitch y yaw, el sistema de coordenadas del chasis representado por  $\{b\}$  y el sistema de coordenadas globales representado por  $\{G\}$ . Para simplificar el análisis de la cinemática del cuadricóptero, el sistema estará basado en varios supuestos:

- La estructura del cuadricóptero es un cuerpo rígido y simétrico con su centro de masa alineado con el centro del chasis del vehículo.
- El empuje de los motores es proporcional al cuadrado de la velocidad de los motores.
- Las hélices se consideran que son rígidas, es decir se consideran despreciables las deformaciones producidas al girar a elevadas revoluciones.
- Las corrientes de viento circundantes son despreciables.
- El área de despegue es plana y estática.

La tabla 3 muestra la lista de las variables de estado que se tomarán en cuenta para el modelamiento del sistema, esta notación está basada en el sistema de coordenadas NED. La posición del vehículo está dada en el sistema de coordenadas globales mientras que la velocidad y la velocidad angular están definidas dentro del sistema de coordenadas del chasis. Es importante distinguir en cual sistema de referencia actúan cada uno de los sensores, por ejemplo, el sensor GPS, mide la posición del cuadricóptero en el sistema de referencia global, mientras que los acelerómetros y giroscopios realizan las mediciones en el sistema de referencia del chasis.

Tabla 3. Variables del chasis de cuadricóptero  
Fuente Autor

Variable	Descripción
$X^G$	Posición ( <b>Norte</b> ) del cuadricóptero en las coordenadas globales (m)
$Y^G$	Posición ( <b>Este</b> ) del cuadricóptero en las coordenadas globales (m)
$Z^G$	Posición ( <b>Abajo</b> ) del cuadricóptero en las coordenadas globales (m)
$x^b$	Posición ( <b>Norte</b> ) del cuadricóptero en las coordenadas del chasis (m)
$y^b$	Posición ( <b>Este</b> ) del cuadricóptero en las coordenadas del chasis (m)
$z^b$	Posición ( <b>Abajo</b> ) del cuadricóptero en las coordenadas del chasis (m)
$\Phi$	El ángulo roll sobre $x^b$ (radianes)
$\Theta$	El ángulo pitch sobre $y^b$ (radianes)
$\Psi$	El ángulo yaw sobre $z^b$ (radianes)
$P$	Velocidad angular de roll (radianes/segundo)
$Q$	Velocidad angular de pitch (radianes/segundo)
$R$	Velocidad angular de yaw (radianes/segundo)

#### 4.6.1 Cinemática de traslación.

Las variables de estado de la velocidad se encuentran en las coordenadas del chasis y las variables de la posición se encuentran en las coordenadas globales, es por esta razón que es necesario definir una matriz de rotación para transformar estas variables entre los distintos sistemas de coordenadas.

Para hallar las coordenadas NED del chasis a partir de las coordenadas globales se utiliza la ecuación 1.

$$x^b = R_G^b X^G = R(\Phi)R(\Theta)R(\varphi)X^G \quad (1)$$

En donde

$$x^b = R_G^b X^G = R(\Phi)R(\Theta)R(\varphi)X^G \quad (2)$$

$$R(\Theta) = \begin{bmatrix} \cos(\Theta) & 0 & -\text{sen}(\Theta) \\ 0 & 1 & 0 \\ \text{sen}(\Theta) & 0 & \cos(\Theta) \end{bmatrix} \quad (3)$$

$$R(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \text{sen}(\varphi) \\ 0 & -\text{sen}(\varphi) & \cos(\varphi) \end{bmatrix} \quad (4)$$

#### 4.6.2 Cinemática rotacional.

Dado que las velocidades angulares están definidas en el sistema de coordenadas del chasis y los ángulos de Euler se encuentran representados a través del sistema de coordenadas globales, es posible usar la derivada de la matriz de rotación con el objetivo de determinar la relación entre la velocidad angular y la derivada de los ángulos de Euler con respecto al tiempo. Las velocidades angulares son vectores que apuntan a lo largo de cada uno de los ejes de rotación y no son iguales a la derivada de los ángulos de Euler.

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = R(\phi) R(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (5)$$

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = S \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (6)$$

$$S^{-1} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (8)$$

#### 4.7. Diseño del cuadricóptero

Para maximizar el tiempo de vuelo, el cuadricóptero está concebido con la idea de ser lo más liviano posible, motivo por el cual requiere llevar a bordo únicamente los componentes mínimos necesarios para poder mantener el vuelo y transmitir los datos a la estación base que es en donde se realiza el procesamiento de la información y la toma de decisiones, estos componentes se encuentran descritos en la tabla 4.

Tabla 4. Componentes a bordo del cuadricóptero  
Fuente Autor

Parte central del chasis	3 placas de fibra de vidrio
Brazos	4 brazos de aluminio
Controlador de vuelo	RadioLink Pixhawk PX4 32 Bit ARM
Sistema de comunicación	915 MHz Telemetry kit
Motorización	4 brushless <i>DJI</i> a2212 @ 920 kv
Hélices	10x4,5"
ESC	Emax 4 in 1 25A ESC
GPS	NEO-M8N
Batería	Floureon LiPo 3s 5500 mAh
Transmisor de video	BossCam 600 mw
Peso total	1.225 kg.

#### 4.7.1 Diagrama eléctrico

La figura muestra esquemáticamente las conexiones eléctricas de los componentes a bordo del cuadricóptero.

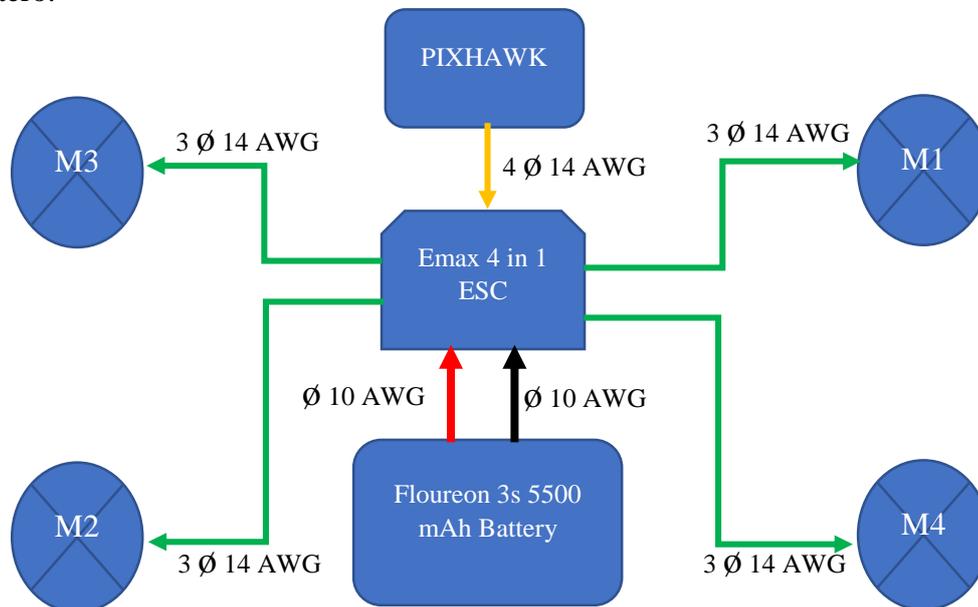


Figura 19. Esquema de conexiones eléctricas  
Fuente Autor

#### 4.7.2. Componentes del chasis

Al ser la estructura del chasis simétrica, el cuadricóptero se compone de 4 de 21,5 cm de largo y 3,5 cm de ancho como se muestra en la figura 20.



Figura 20. Brazo Cuadricóptero  
Fuente Autor

Los brazos se ensamblan al centro de la estructura que está conformada por 3 placas de fibra de vidrio y se encuentran alineadas al centro de gravedad del cuadricóptero, como se muestra en la figura 21

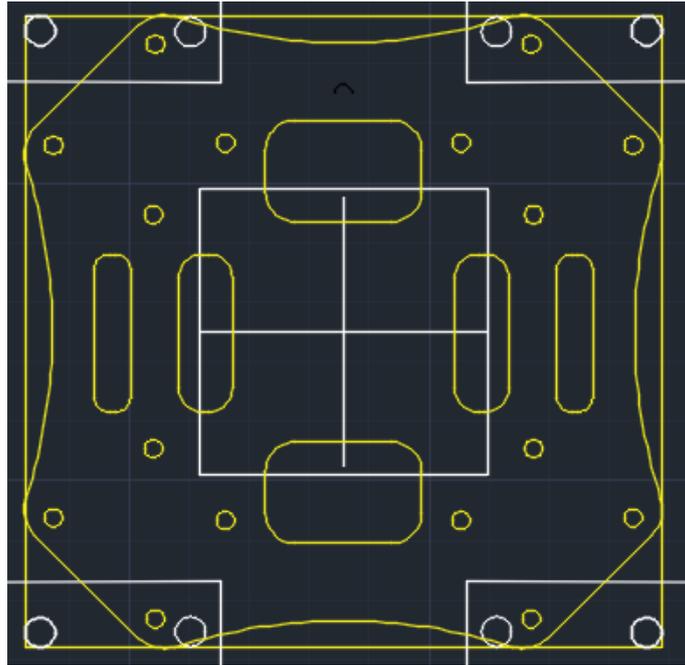


Figura 21. Diseño placa central  
Fuente Autor



Figura 22. Prototipo terminado  
Fuente: Autor

#### 4.8 Estación Base

Al tener una limitada capacidad de procesamiento a bordo del cuadricóptero, el sistema requiere de una estación que será la encargada de procesar los datos de telemetría, monitorear el estado del botón de pánico instalado en el smartphone y de enviar las instrucciones necesarias para que el cuadricóptero realice las tareas encomendadas. La comunicación entre el cuadricóptero y la estación base se encuentra desarrollada sobre el protocolo MAVlink, el cual es interpretado nativamente por la tarjeta controladora de vuelo. Este enlace es de tipo full

dúplex, desde el cuadricóptero se envían los datos de telemetría que incluyen estados del IMU, posición global, voltaje de la batería etc.; por parte de la estación base se envían puntos de destino y el modo de operación del cuadricóptero, la tabla 5 muestra las características técnicas del enlace de telemetría.

Tabla 5. Características enlace de telemetría  
Fuente Autor

Frecuencia de operación	915 MHz
Potencia de transmisión	20 dBm
Modo de transmisión	Transparente
Data rate	56 kbps
Protocolo	MAVlink

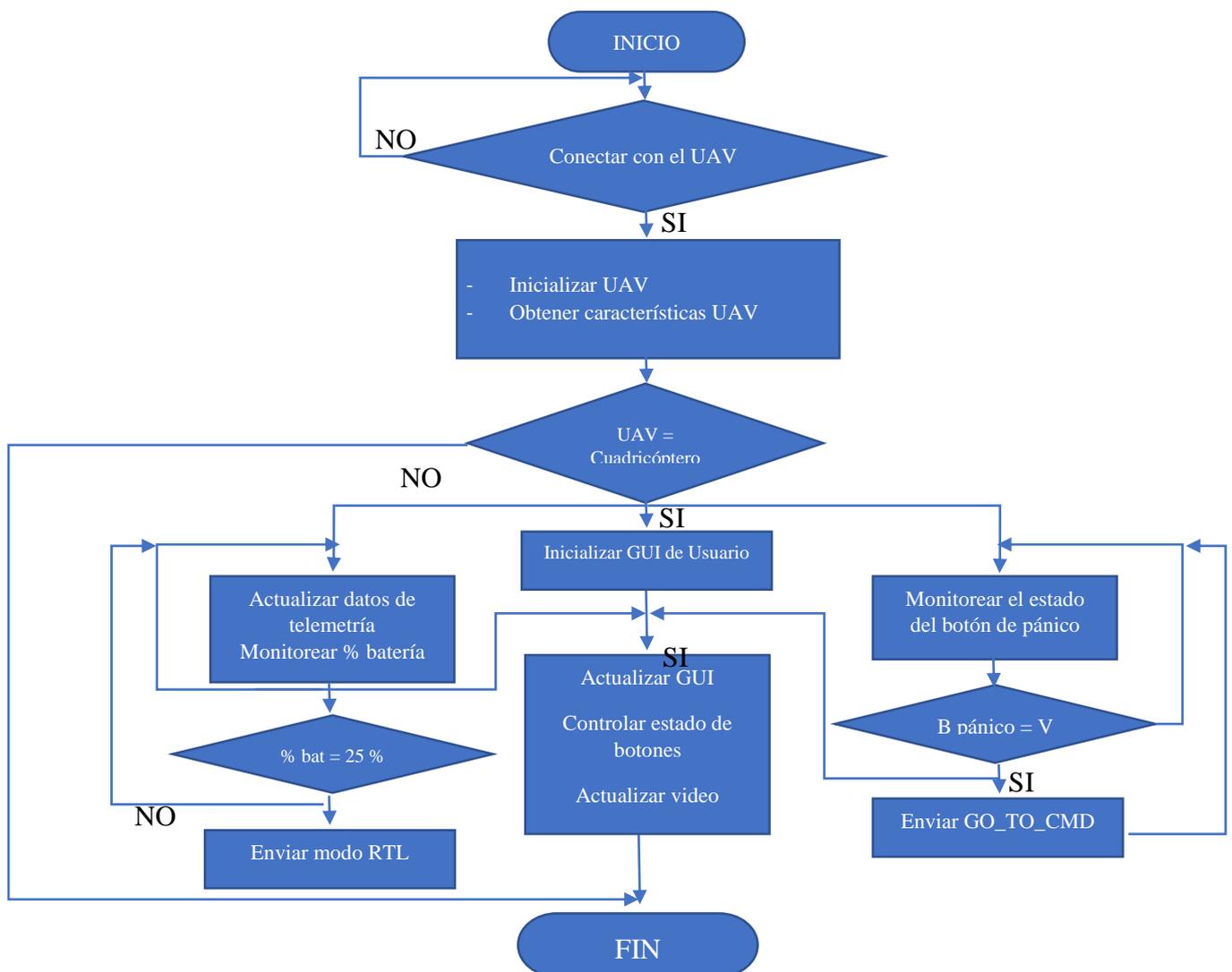


Figura 23. Diagrama de flujo de la estación base  
Fuente Autor

Para evitar tiempos de retardo durante la ejecución del programa especialmente en las tareas de recolección fue necesario dividir la ejecución del programa en 3 hilos o threads, la figura 23 muestra el funcionamiento de cada uno de los threads y como se interconectan entre ellos.

**Main thread:** es el encargado de gestionar la interfaz gráfica de usuario, es decir mantiene actualizada la información proveniente del cuadricóptero, actualiza la posición actual del cuadricóptero en el mapa y actualiza la ventana de video.

**Thread 1:** se encarga de solicitar la información del estado del cuadricóptero, enviar los datos recibidos al main thread, monitorear el estado de la batería y regresar al cuadricóptero a tierra cuando este tenga un porcentaje igual o menor al 25 %.

**Thread 2:** Este subproceso monitoriza el estado del botón de pánico conectándose a la base de datos hospedada en <http://appinvtinywebdb.appspot.com/getvalue> con una tag drone\_value; producto de esta operación se obtienen los datos de latitud, longitud y el estado del botón de pánico; cuando este botón es presionado, se cambia el estado del carácter ubicado en la décimo séptima posición de la trama, al ser detectado el cambio de estado; la estación base envía el comando MAVlink con las coordenadas del smartphone a donde debe dirigirse. La tabla 6 muestra los parámetros de la trama que envía el smartphone hacia la base de datos web, estos valores se envían como caracteres alfanuméricos en formato json, para luego ser decodificados por la estación base.

Tabla 6. Trama enviada desde el Smarthphone  
Fuente Autor

Latitud	Longitud	Bandera botón presionado	Bandera autenticación
8 char	8 char	1 char [0,1]	1 char [0-9]

Ejemplo

Latitud	Longitud	Bandera botón presionado	Bandera autenticación
-1688072	-7863717	1	0



Figura 24. Botón de pánico en Android  
Fuente Autor

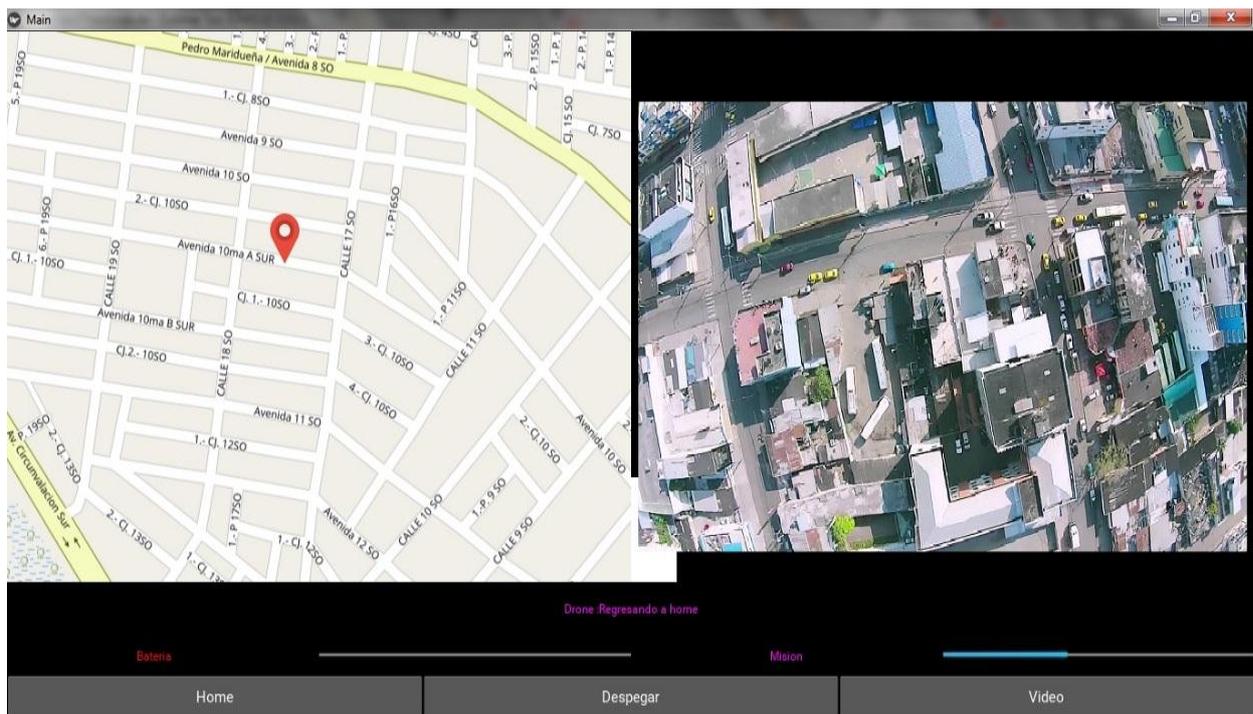


Figura 25. Interfaz gráfica de usuario de la estación base  
Fuente Autor

El enlace entre el usuario y el cuadricóptero se realiza por medio de la interfaz gráfica de usuario que se muestra en la figura 25; esta consta de dos paneles principales, a la izquierda se encuentra el mapa donde se visualiza la posición en tiempo real del cuadricóptero y la posición de la estación base, en el panel del lado izquierdo se muestra el video envía desde el

cuadricóptero. Debajo de estos paneles se encuentra la barra de información que muestra el estatus del vehículo.

La interfaz también cuenta con tres botones los cuales permiten al usuario forzar el comportamiento del cuadricóptero en caso de que el usuario lo requiera.

**Botón Home:** presionando este botón, se envía el comando para cambiar el modo de vuelo a RTL (Return to launch), en este modo de vuelo fuerza al regreso de la aeronave cancelando la misión actual, la altura de retorno es igual a la altura de la última acción ejecutada.

**Botón Despegar:** con el uso de este botón se fuerza a iniciar una misión de patrullaje previamente guardada en la memoria del cuadricóptero; debido a que el cuadricóptero no consta de sensores de detección frontal, es necesario que este alcance una altura de vuelo segura antes de empezar a dirigirse a los puntos de patrullaje, esta secuencia se muestra en la figura 26.

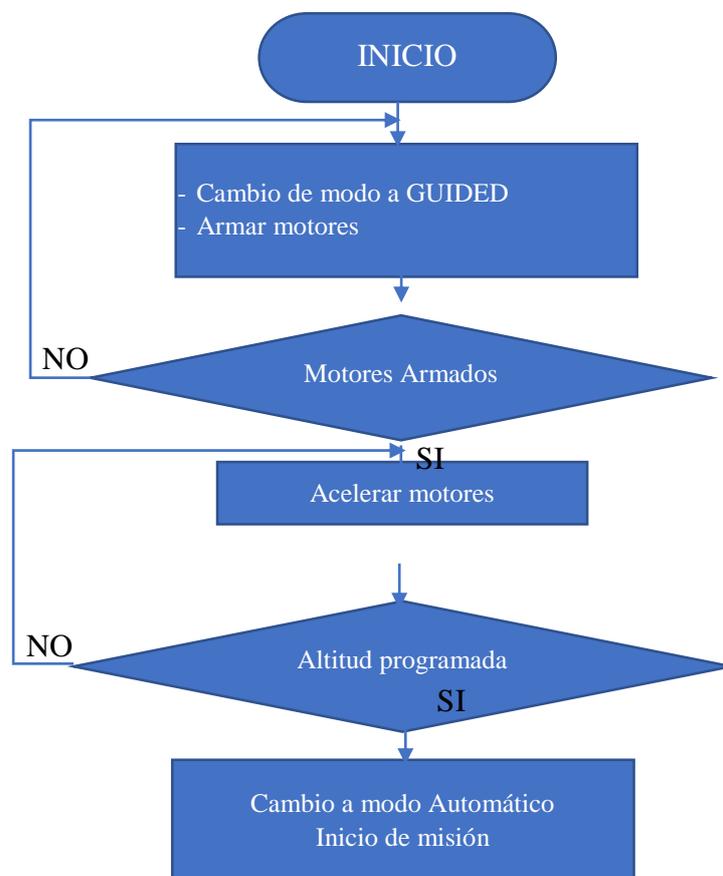


Figura 26. Secuencia de despegue  
Fuente Autor

## Capítulo V

### 5. Resultados

En la metodología fueron desarrolladas las ecuaciones necesarias para simular el comportamiento en vuelo del cuadricóptero, se realizó el modelamiento matemático utilizando Matlab en donde se implementó una interfaz gráfica que visualiza el comportamiento de la aeronave bajo los supuestos considerados anteriormente. Las figuras 27 y 28 muestran el comportamiento del cuadricóptero desde el despegue hasta que llega a mantener un estado de vuelo estacionario en referencia a los sistemas de coordenadas globales tanto en el eje X como en el eje Y, en donde logra una estabilización en un tiempo de 3 segundos, esto se debe a la simetría que tiene el chasis en relación con estos 2 ejes.

La figura 29 muestra el comportamiento correspondiente a la altura que desea ser mantenida, en relación a los ejes X,Y; en el eje Z, la gráfica denota una menor variación en la respuesta que tiene que ejercer el cuadricóptero para alcanzar la altura deseada debido a que para mantener su altura, es necesario mantener la misma velocidad en todos los motores, cosa que no sucede a lo largo de los otros ejes, en donde para lograr un equilibrio se requiere un constante cambio de velocidad en cada uno de los motores hasta equilibrar las fuerzas.

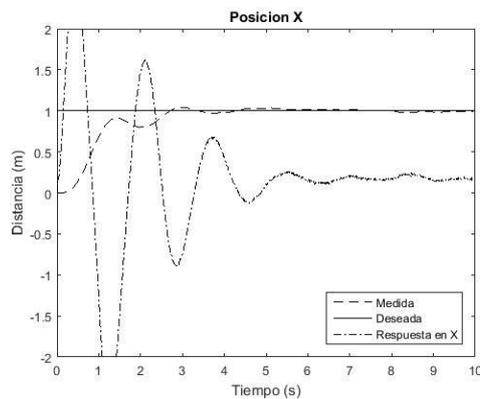


Figura 27. Estabilización eje X  
Fuente Autor

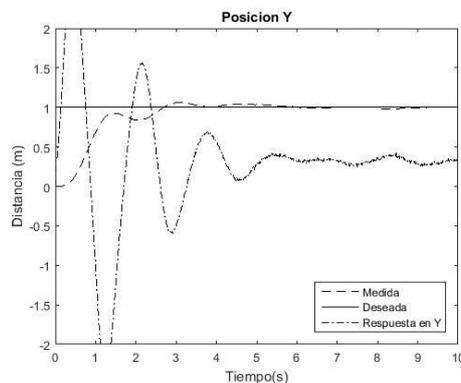


Figura 28. Estabilización eje Y  
Fuente Autor

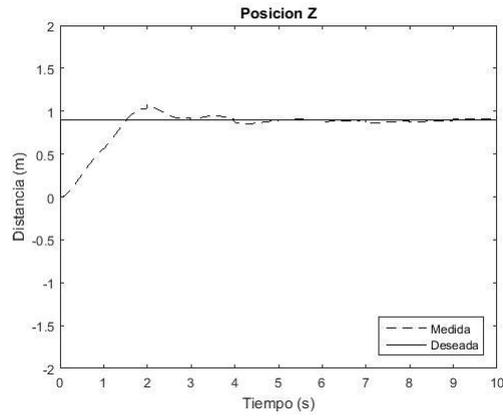


Figura 29. Estabilización eje Z  
Fuente Autor

En cuanto a los ángulos de Euler las gráficas demuestran un mayor esfuerzo para equilibrar los ángulos Roll y Pitch figuras 30 y 31, por su parte el Angulo Yaw muestra variaciones que no superan a los 2 grados de inclinación.

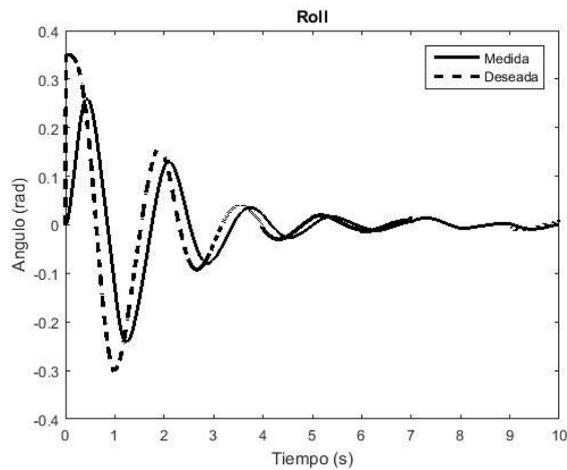


Figura 30. Respuesta Roll  
Fuente Autor

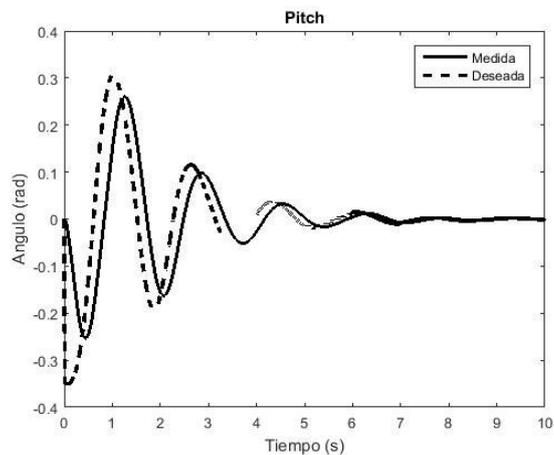


Figura 31. Respuesta Pitch  
Fuente Autor

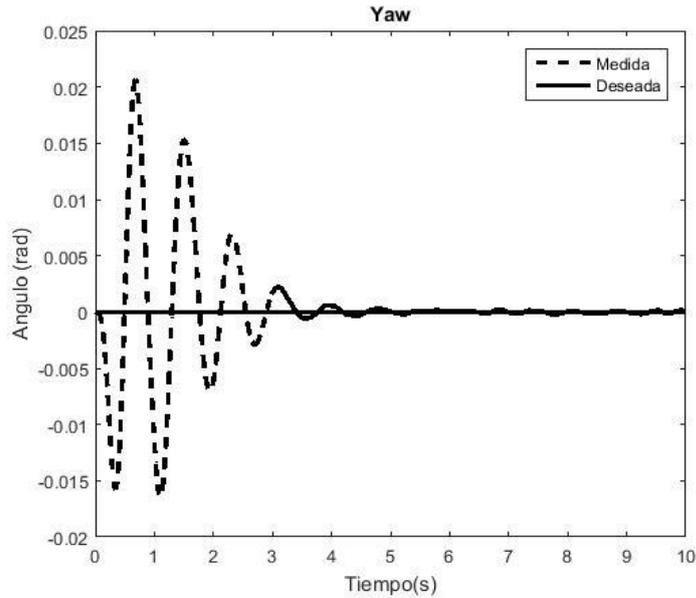


Figura 32. Respuesta Yaw  
Fuente Autor

Una vez realizado el análisis del comportamiento en el simulador se obtuvo los valores de la tabla 7, con los cuales debería contar la estructura del chasis del cuadricóptero para poder realizar un vuelo estable priorizando la eficiencia energética y maximizando el tiempo de vuelo.

Tabla 7. Parámetros del chasis  
Fuente Autor

Distancia del eje del rotor – centro de masa	0,25 m
Radio de la hélice	0,127 m
Peso máximo	2,5 kg
Kv	900 – 1200

De acuerdo con el plan propuesto, el resultado final de este documento conduce al desarrollo de una plataforma capaz de comandar autónomamente el cuadricóptero que gracias al análisis inicial logra tener un vuelo estable.

### 5.1 Pruebas de autonomía.

Para determinar el tiempo de vuelo se realizaron pruebas en dos escenarios con los parámetros iniciales de la tabla 8.

Tabla 8. Parámetros Pruebas de autonomía  
Fuente Autor

Parámetros	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Altura m.s.n.m	6	6	2745	2745
Ambiente	Interior	Exterior	Interior	Exterior
Velocidad del viento (km/h)	0	4	0	9
Dirección del viento	0	E	0	SO
Peso (g)	1500	1500	1500	1500
Altura de vuelo estacionario (m)	6	6	6	6
Porcentaje de batería inicial %	100	100	100	100

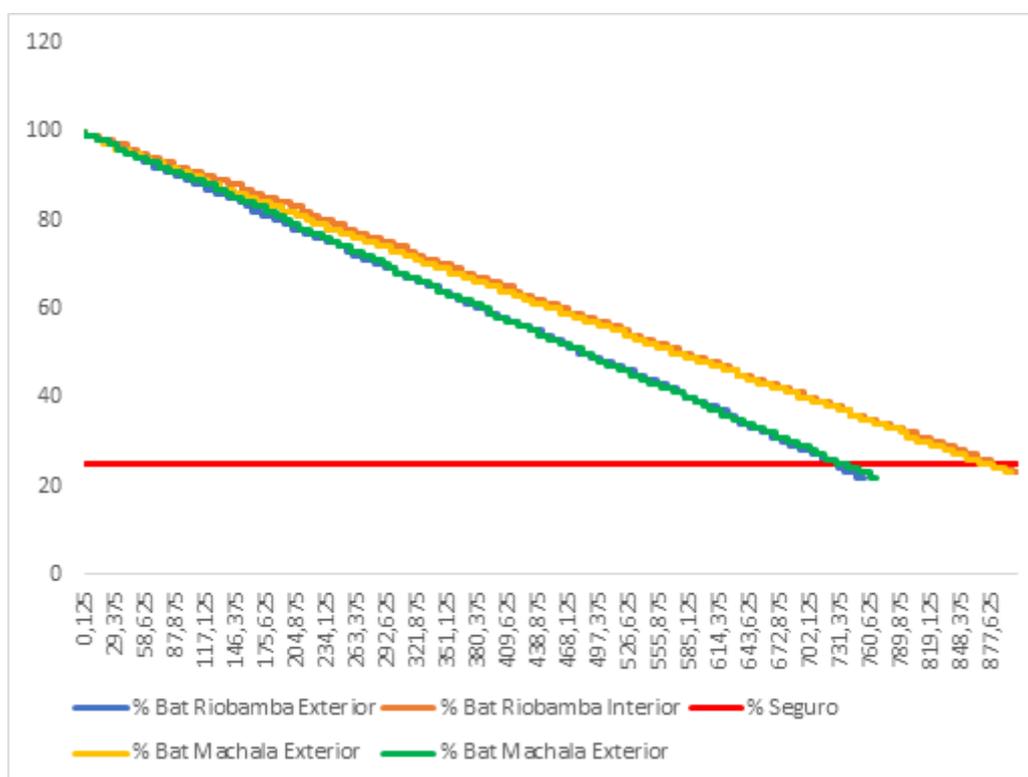


Figura 33. Pruebas de autonomía de la batería  
Fuente Autor

## 5.2 Pruebas de Mantenimiento de Posición

La figura 33 muestra el consumo de batería en cada una de las pruebas realizadas, en donde se evidencia un tiempo de vuelo mayor cuando se realiza el vuelo en ambientes cerrados donde no existe corrientes de aire, bajo estas condiciones se logró obtener tiempos de vuelo de 15 minutos antes de llegar a un nivel mínimo de seguridad del 25 % de la carga de batería es decir

un voltaje de 3.7 v por cada celda; para vuelos en exteriores, se obtuvo una autonomía cercana a los 13 minutos.



Figura 34. Variación de la posición respecto a home  
Fuente Autor

En concordancia con el análisis cinemático, el cuadricóptero es capaz de mantener la posición fija incluso en condiciones de viento, los puntos azules de la figura 34 representan la posición durante un vuelo estacionario donde se observa un pequeño desplazamiento debido a la fuerza ejercida por el viento.

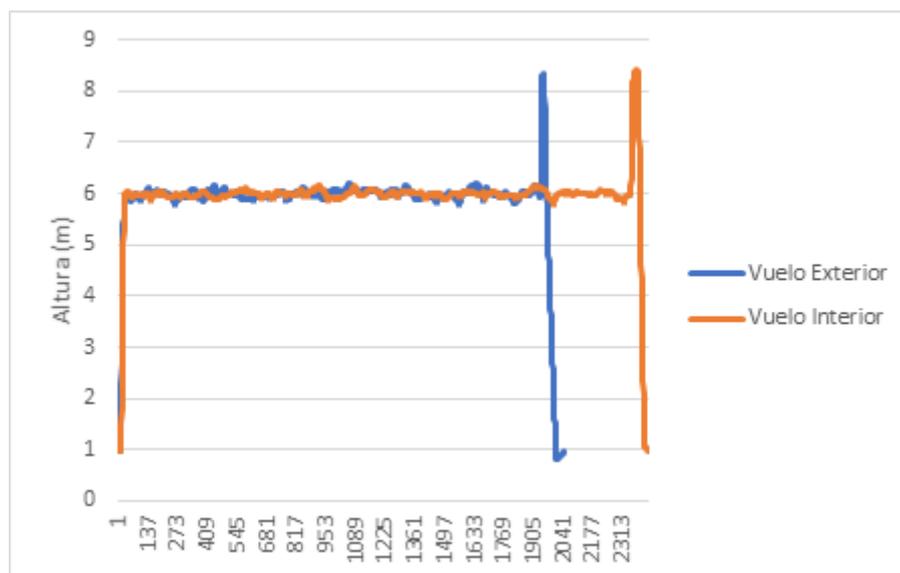


Figura 35. Comparación mantenimiento de altura  
Fuente Autor

En cuanto al mantenimiento de la altura, la figura 35 muestra un comportamiento similar tanto en vuelos en exteriores como en interiores, la altura del vuelo se ve poco afectada por el viento debido al peso de la aeronave.

### 5.3 Pruebas de Transmisión de Telemetría y Video

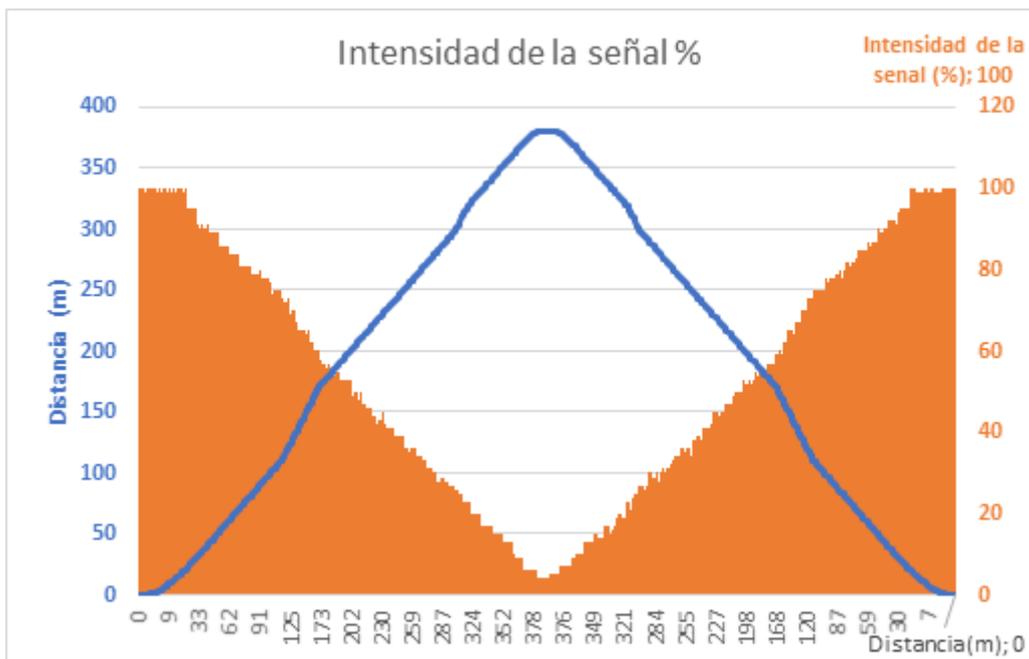


Figura 36. Distancia máxima enlace de telemetría  
Fuente autor

La figura 36 muestra el alcance del enlace entre la estación base y el cuadricóptero, la línea azul muestra la distancia en metros mientras que el área anaranjada muestra en porcentaje la intensidad de la señal, siendo el máximo 100 % a una distancia de 0 m y una intensidad de señal menor al 5% en distancias de 380m.



Figura 37. distancia máxima enlace de video  
Fuente Autor

Para la transmisión de video, el cuadricóptero está provisto de un transmisor que trabaja a una frecuencia de 5.8 Ghz. con 200 mw de potencia, la figura 37 se observa como disminuye la calidad de video hasta perder por completo la señal a una distancia mayor a 720 m. Al no contar con el instrumental necesario para poder analizar los parámetros técnicos de la señal de video como son: ancho de banda, intensidad de la señal, potencia de transmisión, la tabla 9 muestra los datos del fabricante del transmisor de video; se realizará un análisis comparando la calidad de la imagen obtenida versus la distancia, la tabla 10 muestra los resultados obtenidos.

Tabla 9. Características del transmisor  
Fuente Autor

<b>Modulación</b>	WBFM
<b>Formato de video</b>	NTSC/PAL
<b>Potencia de salida</b>	21-23 dBm
<b>Frecuencia de operación</b>	5705 – 5945 MHz
<b>Voltaje de operación</b>	12 V
<b>Corriente</b>	150 mA

Tabla 10. Calidad de imagen vs distancia  
Fuente Autor

<b>Distancia (m)</b>	<b>Calidad de video</b>	<b>Observaciones</b>
0	Optima	No se presentan distorsiones de ningún tipo.
100	Optima	No se presentan distorsiones de ningún tipo.
200	Clara	Presencia intermitente de líneas
450	Visible	Presencia de lluvia y líneas horizontales.
600	Borrosa	Se logra distinguir imagen distorsionada con gran cantidad de lluvia.
720 +	No visible	No se logra distinguir imagen Perdida completa del enlace.

#### 5.4 Misiones de patrullaje

Para que el cuadricóptero pueda realizar su misión de patrullaje de manera autónoma es necesario cargar el plan de vuelo en su memoria el cual consta de los waypoints hacia los cuales se dirigirá y las acciones que hará en cada uno de ellos, la tabla 11 muestra los parámetros de

la trayectoria de patrullaje. La figura 38 muestra el proceso que se realiza para cargar una misión de patrullaje a la memoria del cuadricóptero.

#### 5.4.1 Protocolo Waypoint.

El protocolo está compuesto de diferentes transacciones, cada transacción se completa con éxito o falla de manera que los estados anteriores de los waypoints en el cuadricóptero no cambian. Una transacción puede ser inicializada únicamente cuando no existen otras transacciones activas entre la estación base y el cuadricóptero.

Después de que cada mensaje es enviado, la estación base inicia un contador, si este no recibe un mensaje de confirmación de recepción después de un tiempo especificado, el mensaje es enviado nuevamente; este proceso se reintenta varias veces y si este no recibe respuesta, se cambia de estado a la bandera de falla de transacción.

#### 5.4.2 Envío de waypoints al cuadricóptero

Para enviar una lista de waypoints, se envía un mensaje que contiene el número de puntos que conforman la misión, el cuadricóptero se prepara para la recepción e inicia la transacción de los waypoints; enviando solicitudes a la estación base, estas solicitudes son secuenciales iniciando desde el punto 0.

Cuando el ultimo waypoint es recibido con éxito, el cuadricóptero envía un mensaje ACK a la estación base indicando el fin de la transacción. Si un waypoint es enviado fuera de este proceso, el cuadricóptero contestara con el mensaje ACK sin añadir este punto a la misión.



Figura 38. Proceso de carga de misión  
Fuente Autor

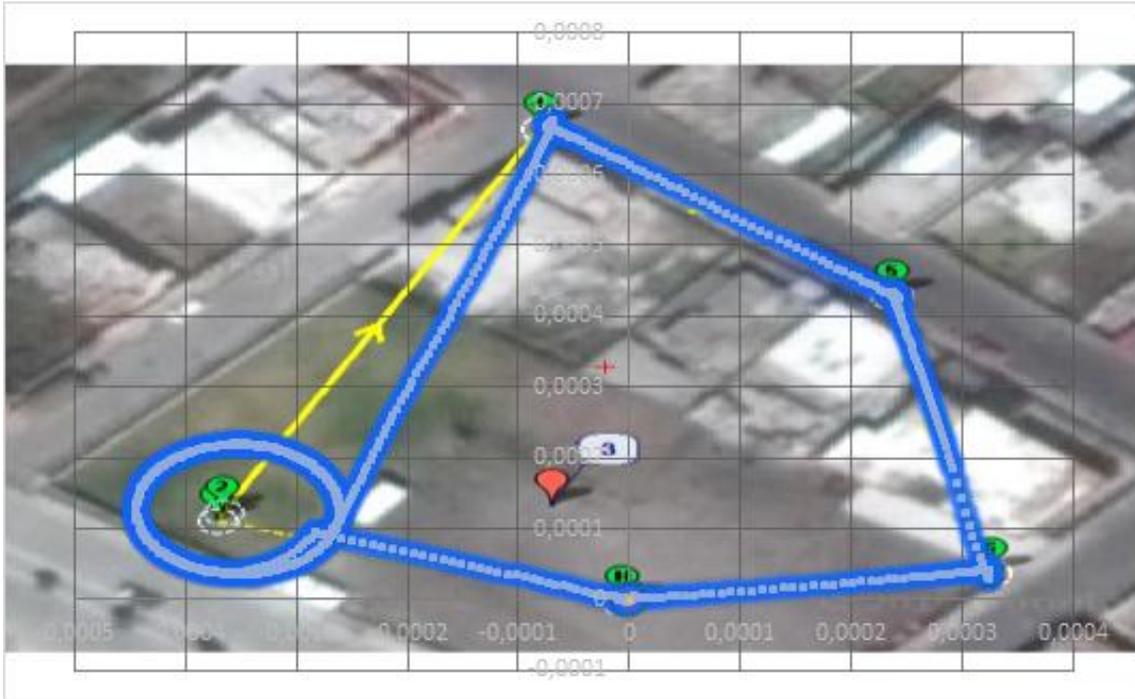


Figura 39. Trayectoria de patrullaje  
Fuente Autor

Tabla 11. Parámetros misión de patrullaje  
Fuente Autor

No	P1	P2	P3	P4	P5	P6	P7	Latitud	Longitud	Alt	UAV	Tipo
0	1	0	16	0	0	0	0	-1.688751	-78.637971	2735	1	HOME
1	0	3	16	0	0	0	0	-1.688635	-78.638329	40	1	Wp 1
2	0	3	18	3	0	0	0	-1.688625	-78.638324	40	1	Región de interés
3	0	3	201	0	0	0	0	-1.688614	-78.638034	6	1	Wp2
4	0	3	16	30	0	0	0	-1.688083	-78.638045	40	1	Wp3
5	0	3	16	30	0	0	0	-1.688319	-78.637734	40	1	Wp4
6	0	3	16	30	0	0	0	-1.68871	-78.637648	40	1	Wp5
7	0	3	20	0	0	0	0	0	0	0	1	Regreso a Home

### 5.4.3 Resultados misiones de patrullaje

Tabla 12. Resumen de misiones realizadas  
Fuente Autor

	N Waypoints	Waypoints alcanzados	Vuelos completos batería 100%	% final de batería
Misión 1	4	4	1	52%
Misión 2	5	5	2	36 %
Misión 3	1	1	4	22%
Misión 4	3	3	3	29%
Misión 5	2	2	4	43%
Misión 6	6	5	0	22%

La tabla 12 muestra el resumen de las misiones realizadas utilizando diferentes números de waypoints, las pruebas consistieron en observar cuantas veces el cuadricóptero logra completar su misión iniciando con un porcentaje de batería igual al 100%, en todas las pruebas la aeronave fue capaz de cumplir la tarea al menos una vez, las misiones fueron suspendidas automáticamente por la estación base; cuando el cuadricóptero alcanzó un nivel de batería de 25 %.

### 5.5 Comprobación de la hipótesis.

El método de comprobación escogido para demostrar la hipótesis es la prueba Chi para Bondad de Ajuste, donde su estadístico se representa con la formula 9.

$$\chi^2_{(r-k-1)} = \sum_{i=1}^r \frac{(o_i - e_i)^2}{e_i} \quad (9)$$

Tabla 13. Tabla de frecuencias observadas  
Fuente Autor

# Waypoints	Vuelos Completados
1	4
2	4
3	3
4	1
5	2
6	0
Total	<b>14</b>

La tabla 13 los resultados obtenidos en las diferentes trayectorias con los cuales se hará la prueba de la comprobación de la hipótesis; continuando con la prueba, el siguiente paso a seguir es calcular los parámetros de estimación, es decir la media y la varianza, la tabla 14 muestra los resultados de los cálculos.

Tabla 14. Parámetros de estimación.  
Fuente Autor

<b>Media Muestral</b>	2.00
<b>Varianza Muestral</b>	1.96
<b>Desvío Muestral:</b>	1.40

La tabla 15 muestra el cálculo del valor empírico en función de los valores esperados y los valores observados en la experimentación, el valor empírico será comparado con el valor crítico

con un grado de libertad de 1, aplicando la función inversa de la prueba chi cuadrado da como resultado un valor crítico de 6.635.

Tabla 15. Calculo del valor empírico  
Fuente Autor

Prob. Acumulada		Probabilidad en Intervalo	Valores esperados	Valores Observados	(V. Obs-Val Esp)^2/Val Esp.
0.0766	0.2376	0.1610	2.25	4	1.35
0.2376	0.5000	0.2624	3.67	4	0.03
0.5000	0.7624	0.2624	3.67	3	0.12
0.7624	0.9234	0.1610	2.25	1	0.70
0.9234	0.9839	0.0605	0.85	2	1.57
			<b>1.30</b>	0	1.30
			14	14	<b>5.07</b>
					<b>Valor empírico</b>

## H0

El cuadricóptero es capaz de realizar misiones de video vigilancia de manera autónoma en la Ciudadela Sixto Duran de la ciudad de Riobamba.

## H1

El cuadricóptero no es capaz de realizar misiones de video vigilancia de manera autónoma en la Ciudadela Sixto Duran de la ciudad de Riobamba.

Tabla 16. Resultados de la prueba Chi para Bondad de Ajuste  
Fuente Autor

### Decisiones Posibles según distintos niveles de significancia:

Nivel se significancia	Valores Críticos según nivel de significancia	Valor empírico	Decisiones
0.01	<b>6.635</b>	<b>5.068</b>	<b>NO se rechaza H0</b>

La tabla 16 muestra que el valor empírico es menor que el valor crítico para 1 grado de libertad por lo que se acepta la hipótesis H0.

## Capítulo VI

### 6.1. Conclusiones

- El desarrollo del modelo matemático permitió entender de mejor manera como es el comportamiento general de un cuadricóptero e identificar cuáles son los parámetros físicos que afectan el comportamiento de la aeronave.
- El realizar una simulación previa a la construcción del chasis permitió estimar las características físicas con las que debe contar el cuadricóptero para que logre un vuelo estable que permita transmitir el video de forma clara.
- Los resultados de la simulación mostraron similitudes en la señal de respuesta en los ejes X e Y, debido a la simetría del chasis con respecto a estos.
- El uso del lenguaje de programación Python ayudo al rápido desarrollo de la interfaz de la estación base, ya que cuenta con librerías especializadas que permitieron establecer la comunicación con el cuadricóptero y el smartphone.
- El uso de procesos en segundo plano permitió minimizar los tiempos de bloqueo durante la ejecución del programa al punto que para el usuario son prácticamente imperceptibles.
- Gracias a la implementación de un carácter de identificación dentro de la trama que envía el botón de pánico hacia la estación base; el sistema es capaz de identificar hasta 10 smartphones conectados al sistema.
- El uso de una secuencia de despegue previo al inicio de cualquier misión es indispensable debido a que, gracias a esta el cuadricóptero alcanza una altura de vuelo segura antes de dirigirse hacia su objetivo, evitando así posibles colisiones durante el despegue.
- En general los tiempos de vuelo alcanzados tanto a nivel del mar como en la ciudad de Riobamba tuvieron prácticamente el mismo valor rondando los 15 minutos en ambientes cerrados y 13 minutos en ambientes exteriores con viento.
- El alcance del enlace de comunicación del sistema es de 380 m mientras que el área en cuestión en su punto más alejado tiene una distancia de 190 m con relación a la estación base, lo que significa que el sistema logra cubrir toda el área propuesta en el planteamiento del problema.

## 6.2. Recomendaciones.

- Tener botones de emergencia redundantes en caso de que se presente una emergencia y se deba regresar el cuadricóptero a tierra.
- Depurar el programa de la estación base antes de realizar la conexión con el cuadricóptero para evitar perder link de comunicaciones cuando este se encuentre en el aire.
- Corroborar el voltaje de la batería con un multímetro y compararlos con el sensor a bordo.
- Contar con más de una batería para evitar retrasos entre vuelos.
- Tener el control RC conectado al momento de realizar las pruebas de vuelo.
- Realizar una revisión previa al primer uso de los motores busheles para evitar problemas al momento de estar en el aire.
- Hacer el estudio previo de los materiales y componentes disponibles en el mercado, para evitar retrasos al momento de construir el chasis.
- Continuar la línea de investigación partiendo del trabajo aquí realizado y añadiendo una red de sensores que permitan al cuadricóptero evitar posibles obstáculos que se encuentren en su trayectoria.

## Capítulo VI

### Bibliografía

- AviationWeek. (2017). *Large Tandem-Rotor UAS Aimed At Heavy-Lift Markets*. Obtenido de <http://aviationweek.com/technology/large-tandem-rotor-uas-aimed-heavy-lift-markets>
- Blyenburgh, P. v. (2006). *UAV systems: global review*. Amsterdam.
- C. H. Hugenholtz, B. J. (2012). Small unmanned aircraft systems for remote sensing and Earth science research. *Transactions American Geophysical Union*, 236.
- Centre, T. J. (2010). *Strategic concept of employment for unmanned aircraft systems in NATO*. Kalkar.
- Corporation, S. (2017). *Schiebel's CAMCOPTER® Unmanned Air System*. Obtenido de <https://schiebel.net/products/camcopter-s-100/>
- DJI. (2017). *DJI store*. Obtenido de [https://store.dji.com/product/snail-racing-propulsion-system?from=menu\\_products](https://store.dji.com/product/snail-racing-propulsion-system?from=menu_products)
- EMAXmodel. (2017). *EMAX Simon Series 25A For Muti-Copter*. Obtenido de <https://www.emaxmodel.com/emax-simon-series-25a-for-muti-copter.html>
- Foundation, .. P. (2017). *Python*. Obtenido de <https://www.python.org/>
- Gordon Ononiwu, A. O. (2016). Design and Implementation of a Real Time Wireless Quadcopter for Rescue. *American Journal of Engineering Research*.
- J. Sanchez-Garcia, J. M.-C. (2015). A Self Organising Aerial Ad Hoc Network Mobility Model for Disaster Scenarios. *Proceedings of the 8th International Conference on Developments in eSystems Engineering*, 35–40.
- jDRONES. (2016). *MAVLink Commands*. Obtenido de <http://ardupilot.org/dev/docs/mavlink-commands.html>
- Kivy.org. (2017). *Kivy*. Obtenido de <https://kivy.org/#home>
- mbH, I.-B. (2001). *CARE innovative action – preliminary study on integration of unmanned aerial vehicles into future air traffic management*. Berlin.
- Molina, I. C. (2014). Unmanned aerial systems for photogrammetry and remote sensing: a review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 79–97.
- NASA. (12 de 03 de 2017). *NASA - Digital Fly By Wire*. Obtenido de [https://www.nasa.gov/vision/earth/improvingflight/fly\\_by\\_wire.html](https://www.nasa.gov/vision/earth/improvingflight/fly_by_wire.html)
- O. Bouachir, A. A. (2014). A mobility model for UAV ad hoc network. *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems*, 383–388.
- O. Briante, V. L. (2015). COMVIVOR: An Evolutionary Communication Framework Based on Survivors' Devices Reuse,. *Wireless Personal Communications*, 2021–2040.

- P. Basu, J. R. (2004). Coordinated flocking of UAVS for improved connectivity of mobile ground nodes. *MILCOM 2004 - 2004 IEEE Military Communications Conference*, 1628–1634.
- Pajares, G. (2015). Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). *Photogrammetric Engineering and Remote Sensing*, 281–329.
- Paul Ghering Falstrom, T. J. (2012). *Introduction to UAV Systems*. New Delhi: Wiley.
- QGroundControl. (2017). *MAVLink Micro Air Vehicle Communication Protocol*. Obtenido de <http://qgroundcontrol.org/mavlink/start>
- S. Gupte, P. I. (2012). A survey of quadrotor unmanned aerial vehicles. *Proceedings of the Proceeding of the IEEE Southeastcon*, 1-6.
- T. H. Cox, C. J. (2004). *Civil uav capability assessment*.
- Team, A. D. (2016). *ArduPilot* . Obtenido de Autopilot Suite: <http://ardupilot.org/ardupilot/index.html>
- Technology, M. I. (2017). *Anyone Can Build Apps That Impact the World*. Obtenido de <http://appinventor.mit.edu/explore/>

## CAPITULO VII

### Publicaciones

El presente proyecto fue aceptado para ser expuesto en el IV Congreso Internacional de Ciencia Tecnología e Innovación para la Sociedad CITIS 2017 organizado por la Universidad Politécnica Salesiana sede Guayaquil.



## ANEXOS

### ANEXO 1

#### CODIGO

```
def timeconsuming1(self):
    while True:
        data = {"tag": "drone_value"}
        r = requests.post('http://appinvtinywebdb.appspot.com/getvalue', data)
        data=r.json()
        print data[2]
        lat=data[2][1:11]
        lon=data[2][12:24]
        print lat
        print lon

def change_Label_text1(self):
    #self.button1.disabled = True
    threading.Thread(target=self.timeconsuming).start()

def timeconsuming(self):
    self.dronemensaje = "Verificando parametros..."

    while not vehicle.is_armable:
        self.dronemensaje = " Inicializando cuadricoptero..."
        time.sleep(1)

    self.dronemensaje = "Armando motores"

    vehicle.mode = VehicleMode("GUIDED")
    vehicle.armed = True
```

```

while not vehicle.armed:
    self.dronemensaje = " Esperando confirmacion de armado..."
    time.sleep(1)

self.dronemensaje = "Despegando..."
vehicle.simple_takeoff(6)

while True:
    self.dronemensaje = " Altitud actual: "+ str(vehicle.location.global_relative_frame.alt)
    if vehicle.location.global_relative_frame.alt>=6*0.95: #Trigger just below target alt.
        self.dronemensaje = "Altitud segura alcanzada..."
        break
    time.sleep(1)
# Set mode to AUTO to start mission
self.dronemensaje = "Iniciando mission..."
vehicle.mode = VehicleMode("AUTO")

def update(self,dt):
    global vehicle
    self.map.center_on(self.posdrone[6], self.posdrone[7])
    self.posdrone[0] = vehicle.location.global_frame.lat
    self.posdrone[1] = vehicle.location.global_frame.lon
    self.posdrone[2] = vehicle.location.global_relative_frame.alt
    self.posdrone[3] = vehicle.heading
    self.posdrone[4] = vehicle.battery.level
    self.posdrone[5] = vehicle.commands.next
    self.statusdrone[0]= vehicle.mode.name
    self.statusdrone[1]= vehicle.armed

```

```

if vehicle.mode.name == "AUTO":
    self.dronemensaje = "Mision "
if vehicle.mode.name == "RTL":
    self.dronemensaje = "Regresando a home "
if vehicle.battery.level == 25:
    self.dronemensaje = "Bateria Baja, regresando a Home "
    vehicle.mode = VehicleMode("RTL")

```

```

def contar_1(self,dt):
self.contador_1 = str(int(self.contador_1)+1)
if self.contador_1 == "5":
    self.obj1.cancel()
    sleep(5)
#self.consulta()

```

```

def contar_2(self,dt):
    self.contador_2 = str(int(self.contador_2)+1)
    Clock.schedule_once(self.contar_2,2)

```

```

def conectar(self):
    global vehicle
    vehicle = connect("127.0.0.1:14550", wait_ready=True)
    vehicle.wait_ready('autopilot_version')
    self.posdrone[0] = -3.259018
    self.posdrone[1] = -79.974671
    self.posdrone[6] = vehicle.home_location

```

```

def arm_and_takeoff(self,aTargetAltitude):
    """
    Arms vehicle and fly to aTargetAltitude.
    """

    self.dronemensaje = "Configurando Cuadricoptero"

    while not vehicle.is_armable:
        self.dronemensaje = " Esperando inicializacion"
        time.sleep(1)

    self.dronemensaje = "Armando motores"
    # Copter should arm in GUIDED mode
    vehicle.mode = VehicleMode("GUIDED")
    vehicle.armed = True

    while not vehicle.armed:
        self.dronemensaje = " Esperando ..."
        time.sleep(1)

    self.dronemensaje = "Despegando"
    vehicle.simple_takeoff(aTargetAltitude)
    self.dronemensaje = "Ejecutando mission"
# Reset mission set to first (0) waypoint
    vehicle.commands.next=0
    time.sleep(1)
# Set mode to AUTO to start mission
    vehicle.mode = VehicleMode("AUTO")

```

```

def armar(self):
    self.arm_and_takeoff(6)

def aterrizar(self):
    vehicle.mode = VehicleMode("RTL")
    self.dronemensaje = "Regresando a home"

def set_marker_position(self, mapview, marker):
    x, y = mapview.get_window_xy_from(marker.lat, marker.lon, mapview.zoom)
    marker.x = int(x - marker.width * marker.anchor_x)
    marker.y = int(y - marker.height * marker.anchor_y)
    if isinstance(marker, MapMarkerPopup):
        marker.placeholder.x = marker.x - marker.width / 2
        marker.placeholder.y = marker.y + marker.height

def get_location_metres(original_location, dNorth, dEast):

    earth_radius=6378137.0
    dLat = dNorth/earth_radius
    dLon = dEast/(earth_radius*math.cos(math.pi*original_location.lat/180))

    newlat = original_location.lat + (dLat * 180/math.pi)
    newlon = original_location.lon + (dLon * 180/math.pi)
    return LocationGlobal(newlat, newlon,original_location.alt)

def get_distance_metres(aLocation1, aLocation2):
    """
    Returns the ground distance in metres between two LocationGlobal objects.

```

This method is an approximation, and will not be accurate over large distances and close to the

earth's poles. It comes from the ArduPilot test code:

<https://github.com/diydrones/ardupilot/blob/master/Tools/autotest/common.py>

```
"""
```

```
dlat = aLocation2.lat - aLocation1.lat
```

```
dlong = aLocation2.lon - aLocation1.lon
```

```
return math.sqrt((dlat*dlat) + (dlong*dlong)) * 1.113195e5
```

```
def distance_to_current_waypoint():
```

```
"""
```

```
Gets distance in metres to the current waypoint.
```

```
It returns None for the first waypoint (Home location).
```

```
"""
```

```
nextwaypoint = vehicle.commands.next
```

```
if nextwaypoint==0:
```

```
    return None
```

```
missionitem=vehicle.commands[nextwaypoint-1] #commands are zero indexed
```

```
lat = missionitem.x
```

```
lon = missionitem.y
```

```
alt = missionitem.z
```

```
targetWaypointLocation = LocationGlobalRelative(lat,lon,alt)
```

```
    distancetopoint = get_distance_metres(vehicle.location.global_frame,  
targetWaypointLocation)
```

```
    return distancetopoint
```

```
class MainApp(App):
```

```
def build(self):
    return WidAlfa()

if __name__ == "__main__":
    MainApp().run()
```

## **ANEXO 2**

### **CODIGO GUI**

```
# -*- coding: utf-8 -*-
```

```
<WidAlfa>:
```

```
orientation: "vertical"
```

```
map: map
```

```
BoxLayout:
```

```
    #height: "40dp"
```

```
    #size_hint_y: None
```

```
MapView:
```

```
    id: map
```

```
    zoom: 17
```

```
MapMarkerPopup:
```

```
    lat: root.posdrone[0]
```

```
    lon: root.posdrone[1]
```

```
    popup_size: dp(130), dp(130)
```

```
MapMarkerPopup:
```

```
    lat: root.posdrone[6]
```

```
    lon: root.posdrone[7]
```

```
    popup_size: 1000,1000
```

```
Bubble:
```

```
    Image:
```

source: "ctx.jpg"

mipmap: True

#Camera:

#id: camera

#resolution: (640, 480)

#play: False

BoxLayout:

height: "50dp"

size\_hint\_y: None

Label:

height: "40dp"

id: label1

font\_size: 12

color: 1,0,1,1

text:"Drone :" + root.dronemensaje

BoxLayout:

orientation: "horizontal"

height: "40dp"

size\_hint\_y: None

Label:

text\_size: cm(1.5), cm(0.4)

id: label1

font\_size: 12

color: 1,0,0,1

text: "Bateria"

ProgressBar:

max: 100

value: root.posdrone[4]

Label:

height: (30,30)

id: label1

font\_size: 12

color: 1,0,1,1

text: "Mision"

ProgressBar:

max: 5

value: root.posdrone[5]

BoxLayout:

height: "40dp"

size\_hint\_y: None

Button:

text: "Home"

size\_hint\_x: 25

on\_press: root.aterrizar()

Button:

text: "Despegar"

size\_hint\_x: 25

on\_press: root.change\_Label\_text1()

ToggleButton:

text: 'Video'

size\_hint\_x: 25

on\_press: camera.play = not camera.play

BoxLayout:

height: "50dp"

size\_hint\_y: None

Label:

height: "40dp"

id: label1

font\_size: 12

color: 1,0,1,1

text:"Posición Home : " + "\n" + "Lat : " + str(root.posdrone[6]) + "\n" + "Lon : " + str(root.posdrone[7])

Label:

height: "40dp"

id: label1

font\_size: 12

color: 0,1,0,1

text:"Posición Actual : " + "\n" + "Lat : " + " " + str(root.posdrone[0]) + "\n" + "Lon : " + " " + str(root.posdrone[1])

Label:

height: "40dp"

id: label1

font\_size: 12

color: 0,1,0,1

text: "Estatus" + "\n" + root.statusdrone[0] + "\n" + str(root.statusdrone[1])

Label:

height: "40dp"

id: label1

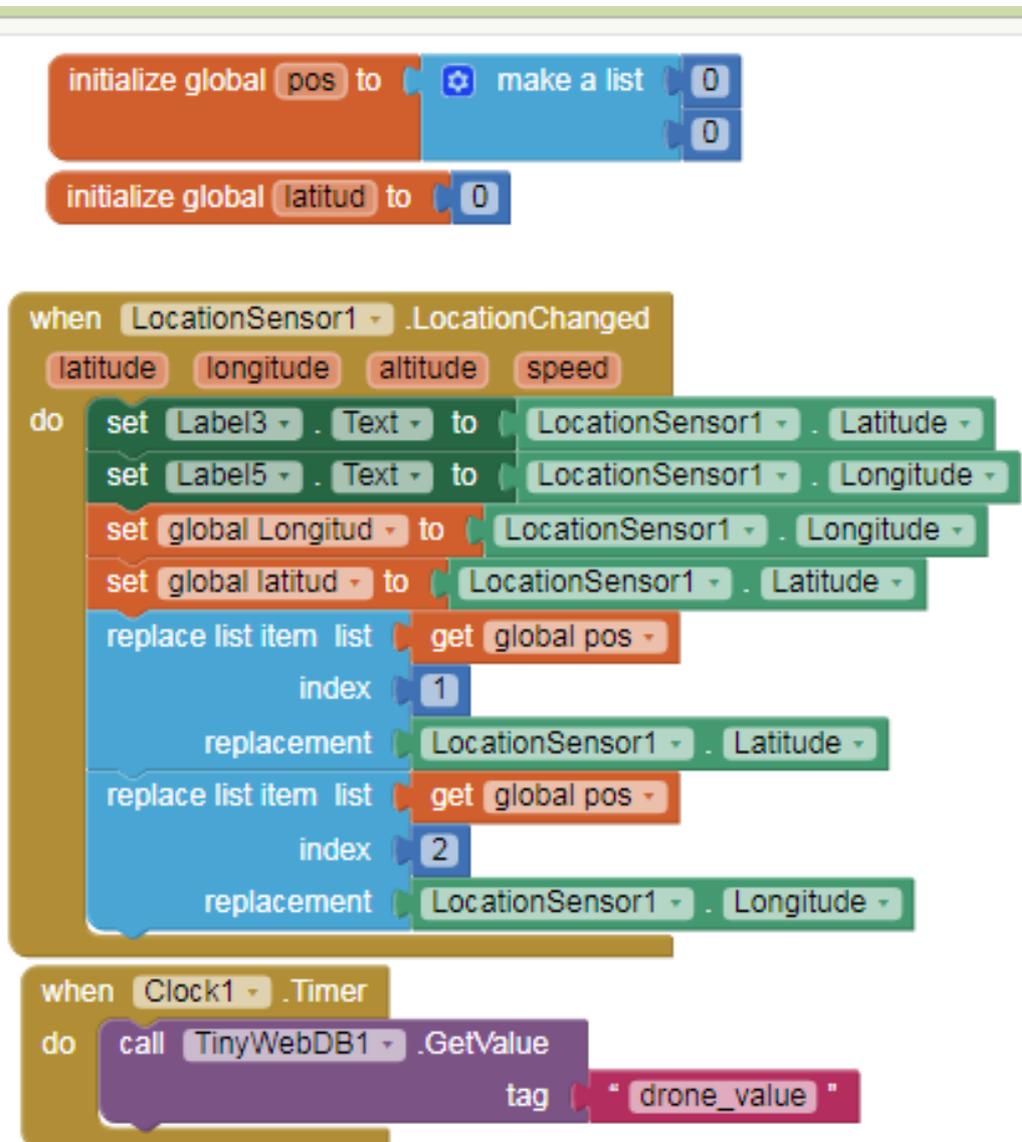
font\_size: 12

color: 0,1,0,1

text: "Altitud : " + " " + str(root.posdrone[2]) + "\n" + "Heading : " + str(root.posdrone[3]) + "\n" + "battery : " + str(root.posdrone[4])

### ANEXO 3

#### DIAGRAMA BLOQUES APP INVENTOR





```
initialize global Longitud to 0
```

```
when Button1 .Click  
do call TinyWebDB1 .StoreValue  
    tag "drone_value"  
    valueToStore get global pos
```

```
when TinyWebDB1 .GotValue  
    tagFromWebDB valueFromWebDB  
do set Label1 .Text to get valueFromWebDB
```

