



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“IMPLEMENTACIÓN DE UN SISTEMA DE
NAVEGACIÓN Y MAPEO EN VEHÍCULOS AÉREOS NO
TRIPULADOS UTILIZANDO DISPOSITIVOS
EMBEBIDOS Y SOFTWARE LIBRE”**

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

CARLOS MAURICIO GUALÁN LLANOS

CHRISTIAN DAVID CAMPODÓNICO DURANGO

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTOS

Quiero agradecer a mis padres por haberme forjado como la persona en la cual me he convertido, todo lo que he alcanzado se los debo a ustedes. Me formaron con reglas y libertades, pero siempre me motivaron a seguir mis sueños e ideales.

Carlos Gualán

Agradezco a mis padres por su amor incondicional, sus enseñanzas y los valores inculcados a lo largo de mi vida. Gracias a ustedes aprendí a afrontar la vida con responsabilidad sin importar las circunstancias y a seguir siempre el camino de la excelencia en todo lo que haga.

A mi hermano por su apoyo en todo momento, por ser mi ejemplo para seguir y por haberme enseñado a través de sus acciones que lo primero en la vida es la honra a nuestros padres.

A mis familiares y amigos por estar en los momentos más difíciles, por haberme otorgado su confianza, gracias por creer en mí, sin su ayuda nada de esto hubiera sido posible.

Christian Campodónico

DEDICATORIA

Dedico este proyecto a mis padres y hermanas quienes me brindaron su confianza y apoyo absoluto a lo largo de mi vida y de mis estudios universitarios, y también a quienes me brindaron su ayuda y sus consejos.

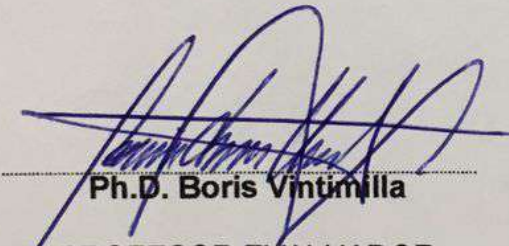
Carlos Gualán

Dedico este proyecto a mi familia, a mis amigos y a todas aquellas personas que día a día se esfuerzan por alcanzar sus metas.

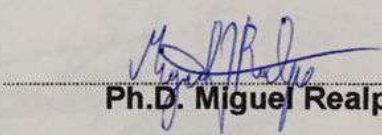
“Siempre intentaste. Siempre fallaste. No importa. Intenta otra vez. Falla de nuevo. Falla mejor”. Samuel Beckett.

Christian Campodónico

TRIBUNAL DE EVALUACIÓN



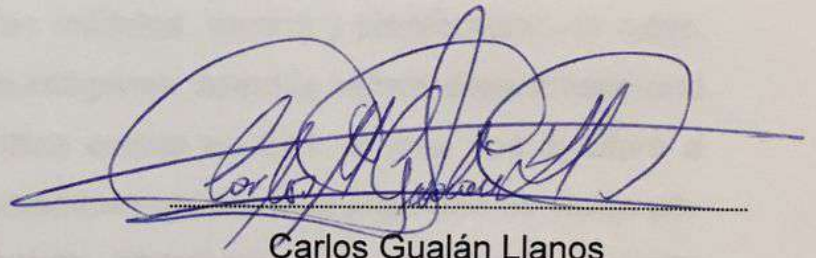
Ph.D. Boris Vintimilla
PROFESOR EVALUADOR



Ph.D. Miguel Realpe
PROFESOR EVALUADOR

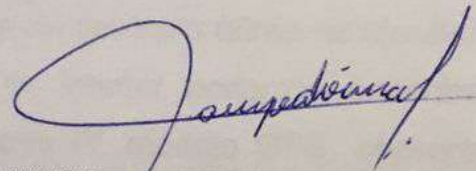
DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Handwritten signature in blue ink, appearing to read "Carlos Gualán Llanos".

Carlos Gualán Llanos



Handwritten signature in blue ink, appearing to read "Christian Campodónico Durango".

Christian Campodónico Durango

RESUMEN

El presente documento trata acerca de la implementación de un sistema de navegación y mapeo en vehículos aéreos no tripulados integrando tecnologías de software y hardware libres, y su aplicación en un entorno de trabajo real. En este caso empleamos dicha solución para capturar múltiples fotografías aéreas de una zona específica, posterior a eso realizamos un procesamiento imágenes mediante el uso de algoritmos de visión por computadora para obtener como resultado imágenes panorámicas de alta resolución.

El proyecto se encuentra dividido en tres módulos: control y planificación de rutas, adquisición de datos y procesamiento de imágenes, además hemos desarrollado una aplicación de escritorio con interfaz gráfica escrita en C++, la cual nos ayudará a realizar la generación de imágenes panorámicas utilizando el algoritmo de visión por computadora SIFT (*Scale-invariant feature transform*). Esta aplicación también integra las plataformas de software de código abierto *Mission Planner* y *OpenDroneMap*, las cuales nos permitirán realizar la planificación de rutas de vuelo y la fusión de imágenes a partir de su información geográfica, respectivamente.

Para la adquisición de imágenes, hemos ensamblado un vehículo aéreo no tripulado multirrotor conocido comúnmente como dron, en su interior podemos encontrar integrados dispositivos embebidos como el *Raspberry Pi*, módulo GPS, sensores IMU, cámara fotográfica digital y una tarjeta de piloto automático y control de vuelo llamada *Pixhawk*. Así mismo realizamos un análisis del funcionamiento de cada uno de estos componentes y la razón de su elección.

Para el análisis de resultados de las pruebas del sistema nos basamos en el desempeño y comportamiento del hardware durante las misiones de vuelo y en la calidad de las fotografías aéreas obtenidas para que el algoritmo de procesamiento de imágenes pueda cumplir con su objetivo. Por ejemplo, para cada misión se toman en cuenta las condiciones climáticas, el tiempo de duración, el tiempo de autonomía de vuelo del dron, la velocidad del vehículo, la cantidad superficie cubierta, la altura de ejecución, el tipo de terreno, la cantidad de imágenes capturadas y la calidad de

las mismas considerando variables tales como el porcentaje de superposición entre cada una de ellas, nitidez, resolución, rotación, contraste, exposición a la luz y distorsión geométrica de la lente de la cámara.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA	v
RESUMEN	vi
CAPÍTULO 1.....	1
1. PLANTEAMIENTO DEL PROBLEMA.	1
1.1 Antecedentes.....	1
1.2 Descripción del problema.	2
1.3 Objetivos.....	2
1.3.1 Objetivo general	2
1.3.2 Objetivos específicos.....	2
1.4 Revisión de literatura.	3
1.4.1 Soluciones similares.....	3
1.4.2 Creación de conceptos	4
1.4.2.1 Vehículos aéreos no tripulados	4
1.4.2.2 Planificación de rutas de cobertura	5
1.4.2.3 Stitching de imágenes	6
1.4.2.4 Dispositivos embebidos	7
1.5 Solución propuesta.....	7
1.5.1 Listado de ideas	7
1.5.2 Limitaciones.....	9
CAPÍTULO 2.....	10
2. METODOLOGÍA Y SOLUCIÓN.....	10
2.1 Metodología.....	10
2.2 Requerimientos de hardware.....	12
2.2.1 Frame Quadcopter	14

2.2.2	Motores, hélices y ESC	15
2.2.3	Batería LiPo y Power Module	18
2.2.4	Tarjeta de piloto automático y control de vuelo	20
2.2.5	Módulo GPS	22
2.2.6	Telemetría	23
2.2.7	Cámara de fotos	23
2.2.8	Raspberry Pi.....	24
2.3	Requerimientos de software.....	25
2.4	Diseño de la solución.	26
2.4.1	Módulo de control y planificación de rutas.....	28
2.4.2	Módulo de adquisición de datos	32
2.4.3	Aplicación de escritorio Hermes	34
2.4.4	Módulo de procesamiento de imágenes.....	38
CAPÍTULO 3.....		41
3.	PRUEBAS DEL SISTEMA Y RESULTADOS.....	41
3.1	Caso de prueba cancha ESPOL.....	43
3.2	Caso de prueba comuna Río Verde.	46
3.3	Caso de prueba dataset públicos.	47
CONCLUSIONES Y RECOMENDACIONES.....		50
BIBLIOGRAFÍA.....		53
ANEXOS.....		58
A.	Configuración y calibración del hardware con Mission Planner	58
B.	Planificación de rutas con Mission Planner	68
C.	Geotagging de imágenes con Mission Planner	75

CAPÍTULO 1

1. PLANTEAMIENTO DEL PROBLEMA.

1.1 Antecedentes.

En años recientes, se ha extendido el uso de drones con múltiples aplicaciones en áreas militares, civiles e industrias en general.

Los ejemplos de su uso en la vida cotidiana y en la fuerza laboral son muy amplios, podemos encontrar este tipo de tecnología siendo aplicada en el campo audiovisual para el desarrollo del turismo, el cine y la televisión, para situaciones de emergencia que involucren rescate de personas, controles de incendios forestales, inundaciones, en las industrias los vemos siendo utilizados como herramientas de seguridad y vigilancia, también en funciones de inspecciones de plantas eléctricas y en el sector de la construcción para el control de estructuras.

Solo por poner un ejemplo más, los drones tienen muchas aplicaciones en la agricultura y el sector ganadero, eso es un dato interesante ya que, en nuestro país la economía ha dependido de manera importante de este tipo de actividades.

En un estudio realizado por el Ministerio de Agricultura, Ganadería, Acuacultura y Pesca del Ecuador (MAGAP) durante los últimos 10 años, la contribución del sector agrícola y ganadero en la economía nacional ecuatoriana ha representado un 8% del PIB Agropecuario, con un crecimiento interanual del 4%. Este modelo económico se respalda fundamentalmente en la producción de banano, café, cacao y flores, las actividades asociadas; Acuacultura, Pesca, Silvicultura y Producción Pecuaria; donde más del 50% de su producción ha sido relacionado a actividades exportables [1].

Este crecimiento ha sido posible en mayor parte gracias a las condiciones establecidas por las nuevas políticas gubernamentales, lo que a su vez ha permitido una mayor facilidad para acceder a insumos agropecuarios, esto con el objeto de crear un mercado más favorable para el productor ecuatoriano [1].

Hoy en día es palpable que la tendencia mundial del sector agrícola apunta a la sistematización de la agricultura, a mejorar la eficiencia de sistemas de riego, al acceso a la información y las telecomunicaciones [2].

Dados estos sucesos, nuestra motivación es extender el uso de estas tecnologías, no solo en la agricultura, sino también en múltiples sectores que ayuden a contribuir en el desarrollo social, económico, tecnológico e investigativo del país, ya que Ecuador, por encontrarse catalogado como un país en vías de desarrollo, aún carece, en gran medida, de la infraestructura y del desarrollo científico que le permitan tener avances tecnológicos propietarios de este tipo.

1.2 Descripción del problema.

El problema considerado en este proyecto es la complejidad que existe, para realizar actividades tales como el reconocimiento de lugares, inspección, vigilancia, monitoreo, búsqueda, rescate, mapeo de superficies, entre otras, y que podrían ser simplificadas y tecnificadas utilizando vehículos aéreos no tripulados.

A pesar de que existen soluciones tecnológicas que satisfacen estas necesidades, las barreras económicas, la escasa difusión y el poco conocimiento que existe acerca de este tema, no ha permitido que se lleven a cabo implementaciones de este tipo con mayor frecuencia en nuestro país.

1.3 Objetivos.

1.3.1 Objetivo general

Integrar tecnologías de hardware y software libres que nos permitan implementar un sistema modular de navegación y mapeo en vehículos aéreos no tripulados.

1.3.2 Objetivos específicos

- Ensamblar un vehículo aéreo no tripulado que al ser operado por medio de radio control, sea capaz de realizar misiones de vuelo en modo de piloto automático, lo cual nos permitirá automatizar tareas como el mapeo de superficies.

- A partir de la integración de tecnologías de hardware y software libre existentes, implementar un sistema que funcione de manera modular, con una interfaz gráfica simple, que permita realizar las tareas de control, planificación de rutas y procesamiento de imágenes.
- Recolectar múltiples fotografías aéreas de un terreno específico y a partir ellas generar una sola imagen panorámica de alta resolución mediante un proceso conocido como *stitching* de imágenes.

1.4 Revisión de literatura.

En esta sección haremos una breve introducción sobre las soluciones existentes, así mismo repasaremos de manera general conceptos que tienen que ver con la planificación de rutas, vehículos aéreos no tripulados, procesamiento de imágenes usando algoritmos de visión por computadora y dispositivos embebidos, para ayudarnos a entender de una mejor manera el contexto del presente proyecto.

1.4.1 Soluciones similares

A nivel local, existen compañías que brindan servicios de consultoría acerca de agricultura de precisión. En la Figura 1.1 se muestra un ejemplo de solución para agricultura de precisión de la compañía AGROSCAN.



Figura 1.1: Solución para agricultura de precisión AGROSCAN [3].

Estos servicios están respaldados en una solución extranjera de la compañía de origen suizo, *SenseFly*, cuyo producto, el *eBee*, un dron utilizado para la agricultura de precisión es el encargado de realizar tareas como la captura de fotografías aéreas en alta resolución, geo-

posicionamiento, generación de índices mediante cálculos complejos, planificación de rutas, simulaciones de vuelo, monitoreo y control, todo esto gracias a un software llamado *eMotion*. [3].

Otra empresa ecuatoriana es LAS, *Latitude Aerospace Solutions*, una empresa que se especializa en el diseño, fabricación de componentes y ensamblaje de robots aéreos [4]. Entre sus ofertas se encuentran levantamientos topográficos, catastros aéreos, agricultura de precisión, fiscalización de obras, infraestructura aérea, grabación de videos profesionales, además del soporte y la asesoría técnica.

Cabe recalcar, que entre los programas que utilizan para realizar varias de las tareas antes mencionadas, se encuentra una solución propietaria extranjera, denominada PIX4D, la cual es una plataforma de software que permite realizar mapeo y modelamiento en 3D, y mapeo para agricultura de precisión [5].

1.4.2 Creación de conceptos

1.4.2.1 Vehículos aéreos no tripulados

Popularmente se los denomina drones, los vehículos aéreos no tripulados son aeronaves que no tienen un piloto a bordo, en su interior contienen tarjetas controladoras de vuelo y pueden ser manipulados a distancia desde un computador con un software de estación de control terrestre mediante las cuales se pueden preprogramar planes de vuelo. Los drones además poseen sensores inerciales como sistema de percepción para captar información del mundo exterior.



Figura 1.2: Drones utilizados en múltiples tareas: agricultura de precisión, topografía y mapeo de zonas devastadas [6].

Anteriormente su uso más frecuente se limitaba a prácticas y operaciones de carácter militar, pero hoy en día tiene múltiples aplicaciones, como podemos observar en la Figura 1.2, dentro de los distintos tipos de actividades se encuentran la agricultura de precisión, levantamientos topográficos, cobertura de eventos en donde existe una concurrencia masiva de personas, misiones de búsqueda y rescate, monitoreo de zonas rurales, control de fronteras, control de incendios forestales, investigaciones arqueológicas, investigaciones bilógicas, manipulación de materiales nocivos e inclusive pueden fungir como pequeños satélites.

1.4.2.2 Planificación de rutas de cobertura

La planificación de rutas de cobertura es la tarea de determinar un camino que pasa por todos los puntos de un área o volumen de interés mientras se evitan los obstáculos. Esta tarea es parte integral de muchas aplicaciones robóticas, tales como robots de limpieza por vacío, robots pintores, vehículos autónomos que crean mosaicos de imagen, robots de desactivación de minas, cortadoras de césped, cosechadoras automáticas, limpiadores de ventanas e inspección de estructuras complejas [7].

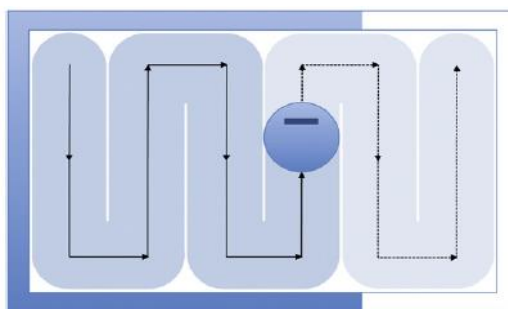


Figura 1.3: Ruta de cobertura en zigzag [7].

Existen varios tipos de algoritmos para realizar rutas de cobertura, en la Figura 1.3, podemos observar un ejemplo de un algoritmo de descomposición celular exacta [8], lo que hace es descomponer una región de interés en subregiones basado en los puntos críticos de la

función Morse, estas descomposiciones representan el área de desplazamiento del robot. Este tipo de descomposiciones tienen aplicaciones cartográficas, también se usan para completar recorridos en áreas sin dejar espacios.

1.4.2.3 Stitching de imágenes

El *stitching* de imágenes, muchas veces conocido también como *mosaicking*, es una técnica en la que, a partir de varias imágenes dentro del dominio superpuesto de la visión, se mezclan para dar lugar a una imagen panorámica de alta resolución. Muchos de los métodos de *stitching* de imágenes requieren intersecciones casi precisas entre imágenes e iluminación idéntica para obtener resultados perfectos.



Figura 1.4: Fusión de imágenes georreferenciadas [10].

En la Figura 1.4 se observa cómo a través del proceso de *stitching*, a partir de un conjunto de imágenes georreferenciadas que pertenecen a una misma escena, se obtiene una imagen panorámica de alta resolución. Las imágenes georreferenciadas son imágenes que están asociadas con la información geográfica desde donde fueron tomadas, esta información contiene la latitud y la longitud, también podría ser asociada la altitud o la orientación de la brújula. La información geográfica se anexa a la *metadata* de la imagen, por lo general el formato de las imágenes suele ser JPEG, pero esto se debe a que el procedimiento para georreferenciar imágenes en este formato se encuentra estandarizado. El estándar se llama EXIF (*Exchangeable*

Image File Format), y está encargado de especificar los formatos utilizados por las cámaras digitales para adjuntar *metadata* a las imágenes.

1.4.2.4 Dispositivos embebidos

Un sistema embebido es un sistema informático que se diseña para funciones de control específicas y está incorporado como parte del dispositivo completo que puede incluir elementos de hardware y partes mecánicas, el cerebro del sistema es un circuito integrado programable conocido como microcontrolador.

Por lo tanto, a diferencia de los computadores de propósito general, por ejemplo, las PC de escritorio, un sistema de embebido realiza unas cuantas tareas predeterminadas, con requerimientos específicos [11]. Ejemplos típicos de sistemas de embebidos incluyen equipos de audio, autos, el ascensor, hornos microondas, cámaras digitales y teléfonos móviles. Como parte de los componentes de un dron, pueden realizar tareas de control de sensores, control de vuelo, manipulación de cámaras, módulos FPV e inclusive tareas de procesamiento de información en tiempo real.

1.5 Solución propuesta.

En esta sección se explica de forma general la solución propuesta en respuesta al problema de la dificultad para realizar tareas específicas utilizando vehículos aéreos no tripulados, gracias a un sencillo esquema que dará a conocer cómo funciona la arquitectura de nuestra solución. Así mismo detallaremos aspectos importantes como lo son las limitaciones del presente proyecto, lo que nos permitirá tener una idea más precisa acerca del camino a seguir en este proyecto de ahora en adelante.

1.5.1 Listado de ideas

Actualmente en el Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales (CIDIS), que pertenece a la Escuela Superior Politécnica del Litoral, existe un proyecto de investigación afín, que se

realizó en semestres anteriores. Este proyecto abarca el sistema de navegación autónoma de un robot terrestre y el software desarrollado para el mismo solo nos permite definir la ruta que seguirá el robot, mediante el trazo de una línea sobre un mapa [12].

A continuación, en la Figura 1.5, podemos ver un esquema simplificado de la arquitectura Cliente/VANT (Vehículo aéreo no tripulado), de nuestro proyecto, el sistema estará compuesto por tres módulos.



Figura 1.5: Arquitectura Cliente/VANT.

El sistema está compuesto por tres módulos. En un computador tendremos un software que integra dos módulos, uno para procesamiento de imágenes y otro para el control y planificación de rutas de cobertura. A través de este software podremos realizar las configuraciones de planificación de vuelo y se enviarán estos datos a través de un medio inalámbrico al vehículo aéreo no tripulado o dron.

El tercer módulo del sistema se encuentra en el dron, es un módulo para la adquisición de datos, y consiste en un hardware embebido que permite manipular una cámara fotográfica para poder capturar imágenes del entorno, las cuales serán almacenadas en su memoria interna, para su posterior procesamiento en el computador.

La navegación aérea del dron será asistida por sensores inerciales, tales como acelerómetro, giroscopio, magnetómetro y barómetro, además de tener incorporado un módulo GPS (*Global Positioning System*) que permite rastrear y determinar su ubicación en el espacio.

En el capítulo dos ampliaremos más en detalle los requerimientos de hardware y software para la implementación completa del sistema.

1.5.2 Limitaciones

Antes de entrar en materia con la metodología, hardware y software que utilizaremos para llevar a cabo la consecución de los objetivos del proyecto, es necesario tener en cuenta algunos factores limitantes y que definen con mayor precisión el alcance de este proyecto:

- Parte del desarrollo del proyecto consiste en ensamblar y adecuar un dron multirrotor de tipo *quadcopter* para capturar imágenes aéreas, para ello se colocará a bordo una cámara fotográfica digital que estará sincronizada y será disparada con un *Raspberry Pi*.
- La planificación de misiones aéreas está enfocada para ser utilizada en vehículos aéreos no tripulados para exteriores, en zonas abiertas, pero no contempla la detección y evasión de obstáculos.
- El georreferenciamiento de las imágenes para aplicaciones en fotogrametría, generación de ortofotos, modelado 3D de terrenos o análisis multiespectral para agricultura de precisión son funcionalidades que pueden ser integradas en un futuro trabajo.

CAPÍTULO 2

2. METODOLOGÍA Y SOLUCIÓN.

En el apartado siguiente, primero se realizará la descripción de la metodología de desarrollo, luego una descripción de los requerimientos de hardware y software del proyecto, y finalmente explicaremos cómo trabajar con estas tecnologías de manera conjunta mediante el diseño de la solución.

2.1 Metodología.

Para el desarrollo de este proyecto utilizamos *Scrum* [13], una metodología de desarrollo ágil que está definida como un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible.

Entre los principios en los que está basado *Scrum* tenemos: la colaboración estrecha con el cliente, predisposición y respuesta al cambio, desarrollo incremental con entregas funcionales frecuentes, comunicación verbal directa entre los implicados en el proyecto, motivación y responsabilidad de los equipos por la autogestión, autoorganización y compromiso, simplicidad, y supresión de artefactos innecesarios en la gestión del proyecto [14].

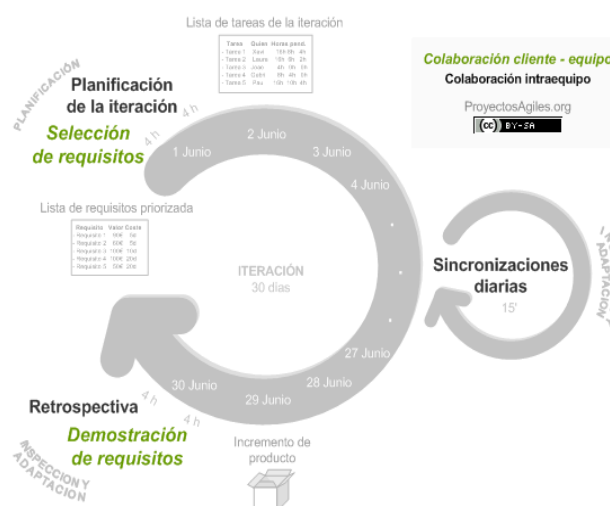


Figura 2.1: Proceso de ejecución de un proyecto utilizando Scrum [13].

En la Figura 2.1 podemos observar la representación del proceso de desarrollo de un proyecto utilizando *Scrum*, la ejecución se divide en bloques temporales cortos y fijos, sus iteraciones normalmente son de 2 semanas. Cada iteración debe generar un resultado completo, es decir un avance del producto final que debe ser presentado al cliente.

El inicio del proceso se realiza a partir de la lista de objetivos y requisitos del producto, su contenido debe estructurarse por orden de prioridades, la planificación del proyecto está basada en esta lista.

Las actividades que se realizan por cada iteración en *Scrum* son:

- **La planificación de la iteración**

Comprende la selección de requisitos, el cliente debe mostrar al equipo de trabajo la lista de requerimientos en orden de prioridad. A su vez el equipo de trabajo está en la obligación de hacer preguntas al cliente para disipar todo tipo de dudas y de esa manera van depurando y organizando la lista para determinar cuáles serán los entregables durante las primeras iteraciones.

Luego el equipo de trabajo debe crear una lista de tareas que corresponden a la iteración en curso y deben estimar el esfuerzo que se hace por cada tarea de manera conjunta. La repartición de tareas se realiza por auto asignación de los miembros.

- **Ejecución de la iteración**

Se empieza con una reunión de sincronización, la duración es de unos 15 minutos como mucho, esto es con el fin de informar a cada miembro del equipo el trabajo que el resto está realizando. Cada miembro del equipo debe responder a las siguientes preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Las iteraciones están supervisadas por una persona que funge como facilitador, el rol se denomina *Scrum Master*, y la idea es que vele por el cumplimiento de los objetivos de trabajo del equipo, protegiéndoles de cualquier interrupción externa que afecte su productividad y eliminando cualquier tipo de problemas que el equipo no pueda resolver por sí mismo.

Durante la iteración, se pueden volver a planificar los objetivos del proyecto junto con el cliente, en caso de ser necesario.

- **Inspección y adaptación**

En el último día de cada iteración se realiza una reunión de revisión en la cual se presentan los resultados basados en los objetivos planteados para la iteración en curso.

Al final el equipo de trabajo debe hacer un análisis sobre su desempeño en la ejecución de las tareas y sobre los posibles problemas que podrían afectar su productividad en las iteraciones posteriores.

2.2 Requerimientos de hardware.

Para el proyecto es necesario tener un hardware que permita realizar los recorridos de las rutas planificadas y el mapeo de zonas para obtener las imágenes que van a alimentar a nuestro módulo de procesamiento. Por lo tanto, hemos adquirido las partes y componentes del dron, para ensamblarlos, y no un dron comercial ensamblado. Primero por la disponibilidad, segundo por el alto costo que tiene un dron comercial adaptado para realizar el tipo de tareas que vamos a ejecutar y tercero porque mediante esta modalidad tenemos la libertad de personalizarlo, agregando y removiendo con facilidad cualquier tipo de periférico que nosotros deseemos.

Por lo general todo dron comercial utiliza una tarjeta controladora de vuelo, nosotros podríamos usar una tarjeta controladora más sofisticada e integrarnos con dispositivos embebidos como la *Raspberry Pi* a bordo del dron para realizar procesamiento de imágenes en tiempo real.



Figura 2.2: Componentes y piezas para ensamblar el dron [18].

Los componentes mínimos necesarios para poder ensamblar el dron se pueden observar en la Figura 2.2. La lista de piezas y partes para su armado es la siguiente:

- 1 *Frame* para dron *quadcopter*, tipo 'X'.
- 4 motores tipo *brushless*.
- 4 controladores de velocidad electrónicos (ESC).
- 4 hélices.
- 1 batería LiPo con *Power Module*.
- 1 tarjeta controladora de vuelo con autopiloto.
- 1 módulo de telemetría (emisor y receptor).
- 1 control remoto con su receptor.
- 1 módulo GPS.
- 1 cámara de fotos con conexión por Wifi.
- 1 *Raspberry pi* con regulador de voltaje BEC 5V.



Figura 2.3: Dron ensamblado y adaptado para misiones de reconocimiento y mapeo de superficies.

En la Figura 2.3 observamos al dron ensamblado en su totalidad. A continuación, explicaremos de manera breve los detalles técnicos de cada uno de los componentes, esto es importante, ya que nos ayudará a visualizar de mejor manera y entender, cuál es su funcionamiento y por qué son necesarios.

2.2.1 Frame Quadcopter

El *Frame* es el armazón o la estructura del dron. *Quadcopter* es una palabra compuesta, *quad* significa cuatro, y *copter* hace referencia a helicóptero. Que la aerodinámica del *frame* sea cómo la de un helicóptero quiere decir que el dron tendrá la capacidad de poder mantenerse en el aire si necesidad de desplazarse, a diferencia de los drones que tienen estructuras parecidas a las de un avión. Esta característica es importante porque para las tareas de toma de fotografía a veces es necesario que el dron se quede en un punto fijo en el aire.



Figura 2.4: Frame de dron multirrotor de 4 motores, tipo 'X' [19].

Existen muchos tipos de configuraciones, hay *frames* de 3, 4, 6 y hasta 8 motores, en nuestro caso hemos elegido un armazón para dron multirrotor

de 4 motores, con forma de 'X', como se observa en la Figura 2.4. Su tamaño depende de la cantidad y el tipo de componentes que adaptemos a su estructura. El material del que están hechos puede ser de plástico o fibra de carbono. El tamaño de este *frame* es de 450 mm, se mide en diagonal y su peso aproximado es de unos 280 g.

2.2.2 Motores, hélices y ESC

El motor es una máquina que convierte la energía eléctrica en energía mecánica, actualmente los motores para drones son de tipo *brushless*, estos motores ya no usan escobillas en su interior como sus antecesores, lo cual los hace más eficientes y más duraderos [20].

Los motores en conjunto con las hélices nos permitirán hacer que el dron se despegue del suelo. Para que esto suceda, al girar generan una fuerza conocida como Empuje, es importante saber cuánto mide esta fuerza ya que, a partir de ello, sabremos cuanta masa será capaz de mover el dron, de modo que no solo se le permita despegar del suelo, sino también flotar y desplazarse en el aire con facilidad. Podemos obtener este valor a partir de la hoja de datos del fabricante, y también debemos conocer cuánto pesa el dron en total, nuestro dron pesa alrededor de 1330 g, esto lo sabemos porque pesamos previamente cada uno de los componentes que van a bordo del dron.



Figura 2.5: Datos técnicos del motor para vehículos aéreos no tripulados tipo brushless [21].

En la Figura 2.5, podemos ver cuál es el motor que usamos y los datos técnicos dados por el fabricante, todos los datos son importantes, pero nos centraremos en la cantidad de Kv, que es una forma de especificar la

cantidad de revoluciones por minuto que da el motor por cada Voltio que recibe desde la fuente de energía. En este caso el motor dará una cantidad de 1000 vueltas en un lapso de un minuto al recibir 1 Voltio de entrada. A mayor número de Kv el motor dará más vueltas, pero su contra es que tiene menos Torque, esto hará que la fuerza de Empuje sea menor, por lo tanto, los motores levantarán menos peso. Tener 1000 Kv es un número relativamente bajo, y encaja muy bien para la cantidad de peso del dron que necesitamos levantar [21].

Propeller	Gear Ratio	Volts	Amps	Watts	RPM	Speed (mph)	Thrust (g)	Thrust (oz)
GWS HD 8x4	1	7	3.35	23	6630	25.1	226	7.97
GWS HD 8x4	1	7.9	4.1	32	7410	28.1	287	10.12
GWS HD 8x4	1	8.9	4.85	43	8220	31.1	347	12.24
GWS HD 8x4	1	9.9	5.65	55	8940	33.9	420	14.82
GWS HD 8x4	1	10.9	6.5	70	9660	36.6	495	17.46
GWS HD 9x5	1	6.9	5.5	37	6000	28.4	348	12.28
GWS HD 9x5	1	7.9	6.7	52	6660	31.5	436	15.38
GWS HD 9x5	1	8.9	7.85	69	7290	34.5	526	18.55
GWS HD 9x5	1	9.9	9.25	91	7920	37.5	627	22.12
APC E 10x5	1	6.9	7	48	5610	26.6	406	14.32
APC E 10x5	1	7.9	8.45	66	6120	29.0	505	17.81
APC E 10x5	1	8.9	9.9	88	6690	31.7	604	21.31
APC E 10x5	1	9.9	11.45	113	7170	34.0	702	24.76
APC E 10x5	1	10.9	13	141	7650	36.2	802	28.29
GWS HD 10x6	1	6.9	7.2	49	5610	31.9	424	14.96
GWS HD 10x6	1	7.9	8.7	68	6180	35.1	526	18.55
GWS HD 10x6	1	8.9	10.1	89	6690	38.0	617	21.76

Figura 2.6: Datos de prueba con diferentes tipos de hélices [21].

Así mismo para seleccionar las hélices, nos basamos en la hoja de datos de prueba del fabricante. En la Figura 2.6 hemos encerrado en un recuadro rojo la hélice que necesitamos.



Figura 2.7: Juego de 4 hélices 10X45 para un motor de 1000 Kv [22].

En la Figura 2.7 podemos observar las cuatro hélices del dron, sus medidas corresponden al diámetro de las hélices en pulgadas. Según la

hoja de datos, con esto tendremos un valor Empuje (*Thrust*) de aproximadamente 802 g por cada motor. Para llegar a ese valor se usa la siguiente fórmula:

$$T = \frac{W*2*1.2}{N} \quad (2.1)$$

En donde:

T = es el Empuje en Newtons.

W = es el peso total del dron en Kilogramos.

N = número total de motores del dron.

Entonces en nuestro caso, con un peso total del dron de 1330 g y 4 motores, tenemos un valor de Empuje de 798 g generados por cada motor para poder hacer volar al dron, lo cual está dentro de lo esperado. Y para ello también necesitamos tener una fuente de energía de unos 10.9 V, lo cual nos da unas luces para saber las características de la batería que necesitaremos para hacer funcionar los motores.

Por último, tenemos que el consumo de corriente que va a generar cada motor es 13 A, estos motores son trifásicos por lo tanto necesitamos algo que nos ayude a convertir la corriente de la batería que es directa a alterna, que es lo que necesitan los motores, y viceversa. Los circuitos electrónicos conocidos como ESC, que quiere decir *Electronic Speed Controller* (ver Figura 2.8), pueden realizar esa función, aunque el propósito principal para el que fueron diseñados es variar la velocidad con la que giran los motores.



Figura 2.8: Controlador electrónico de velocidad, ESC, con capacidad para administrar hasta 30 A. [23].

Las especificaciones vienen dadas por la cantidad de corriente que pueden administrar, por lo general se recomienda que su capacidad exceda a la cantidad de corriente máxima que puede consumir nuestro motor para trabajar sin sobrecalentarse.

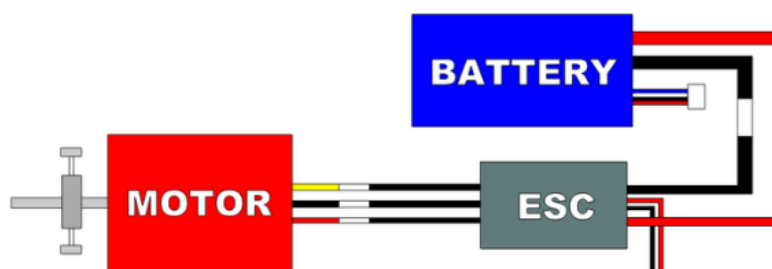


Figura 2.9: Diagrama de conexión de un ESC. Del lado izquierdo tenemos la conexión con el motor brushless trifásico, y del lado derecho tenemos la conexión con la batería [24].

En la Figura 2.9 podemos observar un ejemplo de cómo se deben realizar las conexiones de un ESC con el motor y la batería. En nuestro caso, un ESC de 20 amperios sería suficiente para conectarlo a nuestro motor ya que su máximo rango de operación en condiciones normales es de 13 A. Esto no quiere decir que no se pueda exceder ese valor, simplemente es un indicador para no hacer trabajar al motor más de la cuenta sin provocar un daño, es por esta razón que la cantidad de corriente de los ESC debe ser superior.

2.2.3 Batería LiPo y Power Module

La batería es un dispositivo electroquímico en el cual la energía eléctrica es convertida y almacenada de una forma química [25]. Las baterías son la fuente de energía del dron, por lo tanto, están encargadas de proporcionar la autonomía necesaria para realizar las misiones de vuelo.

Los parámetros importantes por considerar en las baterías son el peso, su capacidad de almacenamiento de carga y el voltaje, estos valores se pueden estimar de a partir de los parámetros de consumo de los motores, el resto de carga agregada de por los demás componentes electrónicos

por lo general no se consideran, ya que por naturaleza son de bajo consumo.



Figura 2.10: Batería de Litio-Polímero de 11.1V y 2450mAH.

Actualmente se usan baterías de Litio-Polímero para los drones, comúnmente se las conoce como LiPo (ver Figura 2.10), su ventaja sobre las demás, es que tienen la capacidad de almacenar más carga, son más livianas y pueden ofrecer tasas de descarga mucho más elevadas en comparación con las baterías de Níquel-Cadmio o Níquel-Metal hidruro.

La desventaja es que son bastante delicadas, el tope máximo de voltaje que puede tener una sola célula de batería LiPo es de 4.2 V, más allá de ese valor la batería podría explotar. Así mismo el valor mínimo no debe estar por debajo de los 3V, caso contrario la batería perdería sus propiedades para almacenar carga [26]. Por esta razón siempre es necesario adaptarle un sensor a la batería para controlar los valores de voltaje y corriente, a este tipo de sensores se les denomina *Power Module* [27] (ver Figura 2.11).

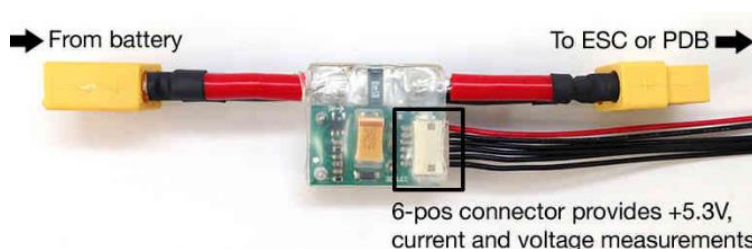


Figura 2.11: Power Module, sensor para medir el voltaje de la batería LiPo y el consumo de corriente [27].

Si recordamos los motores necesitan una Tensión de 10.9 V, y la cantidad de corriente máxima que soportan es de 13 A, entonces la batería debe ser de 3 celdas, lo cual nos daría un total de 11.1 V, con la posibilidad de tener una capacidad máxima de 12.6V.

La cantidad de carga que puede almacenar está dada en mili Amperios-Hora [mAh], hemos elegido una batería promedio de 2450 mAh, con una tasa de descarga de 35C, la C corresponde al valor de los 2450mAh. Esto quiere decir cuánto sería el valor máximo de carga sostenida que puede soportar la batería sin degradarse. Entonces nuestra batería puede soportar una corriente de carga sostenida de 85.75 A, lo cual es un límite superior bastante amplio para el consumo de nuestros cuatro motores que apenas consumen 13 A cada uno.

Una batería con estas características pesa alrededor de 200 g. El peso de la batería ya ha sido considerado dentro del peso total del dron al momento de elegir los motores, hay que tenerlo siempre en cuenta ya que podríamos caer en el error de querer utilizar una batería de mayor capacidad en amperios con el fin de obtener mayores tiempos de vuelo, sin embargo, se aplican los mismos principios como en cualquier batería, a mayor carga tendremos mayor peso y por ende mayor consumo de energía por parte de los motores para mantener el vehículo en el aire.

2.2.4 Tarjeta de piloto automático y control de vuelo

El piloto automático es un sistema que permite controlar la trayectoria de una aeronave sin necesidad de que sea controlada por una persona. Su intención no es reemplazar al operador, sino más bien asistirlo, de tal manera que pueda enfocarse en otros aspectos durante las operaciones de vuelo, como por ejemplo monitorear la trayectoria del vehículo, monitorear aspectos del entorno como el clima, la presión atmosférica o revisar el estado de los sistemas de sensores internos.

Existen varias tarjetas controladoras, por ejemplo, tenemos la *Pixracer*, *Erle-Brain*, *Beagle Bone Blue*, entre otras. Hemos escogido la tarjeta *Pixhawk* (Figura 2.12) porque se ajusta a nuestras necesidades y por

tener un costo en promedio más bajo respecto a las otras. Este controlador tiene la capacidad de funcionar como autopiloto, es decir, que nosotros podemos realizar misiones de vuelo con el dron sin necesidad de intervenir manualmente en su navegación mientras este realiza un recorrido en el aire.

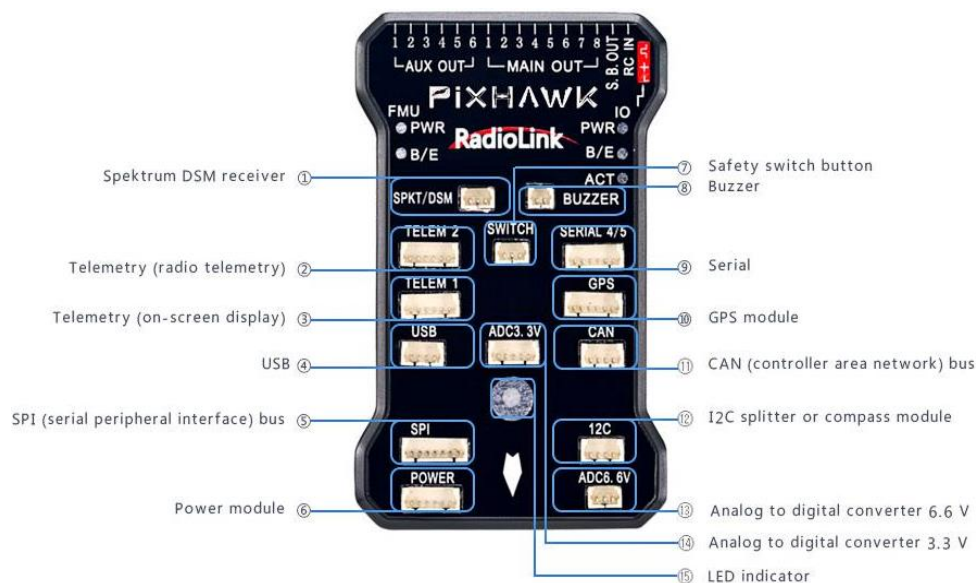


Figura 2.12: Interfaces de conexión de la tarjeta controladora de vuelo con soporte para piloto automático, Pixhawk de [28].

La tarjeta tiene su propia memoria interna para el almacenamiento de datos, allí es donde residen su sistema operativo, las configuraciones para interactuar con los diferentes sensores y dispositivos periféricos, y las rutas de vuelo programadas.

En su parte externa podemos encontrar algunas interfaces de conexión de tipo puerto serial que nos permitirán conectar los diferentes módulos físicos, tales como, GPS, baterías, sensores de voltaje, sensores de temperatura, telemetría, entre otros, los cuales le otorgan al dron la facultad de interactuar con su entorno. Además, posee los pines de salida para modulación de ancho de pulso (PWM) que nos permiten tener comunicación por medio de radio control, conectar cámara de fotos, paracaídas y motores.

2.2.5 Módulo GPS

El módulo GPS (*Global Positioning System*) nos ayuda a saber la ubicación de los drones midiendo cuánto tarda una señal en viajar desde un satélite. EL GPS también es capaz de dar una estimación de la altura de los drones, sin embargo, estos módulos son bastante inexactos y por si solo nos dará una posición dentro de 5 metros a la redonda. Aun así, sabemos que con la combinación de mediciones de otros sensores el controlador de vuelo puede obtener una mejor lectura de lo que está haciendo el dron.

La principal característica utilizada por el módulo de GPS es que permite volar al dron por sí mismo desde el momento en que despegue hasta su aterrizaje. En la Figura 2.13 podemos observar una representación de cómo funciona un sistema de GPS.

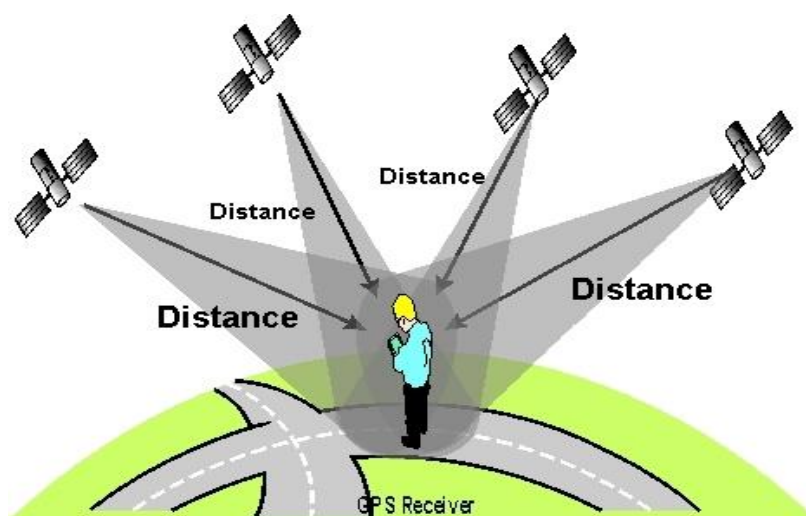


Figura 2.13: Esquema de cómo funciona un módulo GPS [29].

Hoy en día existen módulos GPS que son capaces de comunicarse a más de una red GPS, como por ejemplo la red rusa GLONASS. Pero esto sólo quiere decir que el receptor GPS puede recoger más señales de satélite, lo que mejora la fiabilidad y el rendimiento. Algo bueno es que no son muy costosos. Los módulos comunes que soportan redes GPS y GLONASS son los módulos *Ublox Neo 7N* y *Ublox Neo M8N227*.

2.2.6 Telemetría



Figura 2.14: Sistema de telemetría para envío y recepción de datos e información en tiempo real [30].

Un sistema de telemetría, como podemos ver en la Figura 2.14, es un medio de comunicación inalámbrico que trabaja en la frecuencia de los 915 MHz, nos permite enviar y recibir información en tiempo real, directamente desde un computador donde tengamos instalado un software de estación de control terrestre hacia el dron, y viceversa.

2.2.7 Cámara de fotos

La cámara necesaria para este proyecto tiene que cumplir con varios requisitos importantes, como por ejemplo soportar conexión a través de Wifi y recibir comandos a través del protocolo HTTP, también tiene que ser capaz de captar fotografías en movimiento y por lo tanto debe tener un mecanismo de estabilización incorporado para evitar ruidos, deformaciones y distorsiones que no nos permitan tener datos de calidad, debe tener excelente resolución en imágenes, debe tener un ángulo de visión amplio y además su peso debe ser muy ligero para que no sobrecargue al dron y de ese modo este consuma menos energía ya que necesitaría menos potencia para despegar y mantenerse en el aire.

La cámara SJ4000 cumple con estos requisitos y cuenta con un plus por su bajo costo, con respecto a marcas de renombre como GoPro, Sony, Canon o Nikon, la relación calidad-precio es excelente, en la Figura 2.15 podemos observar su diagrama.

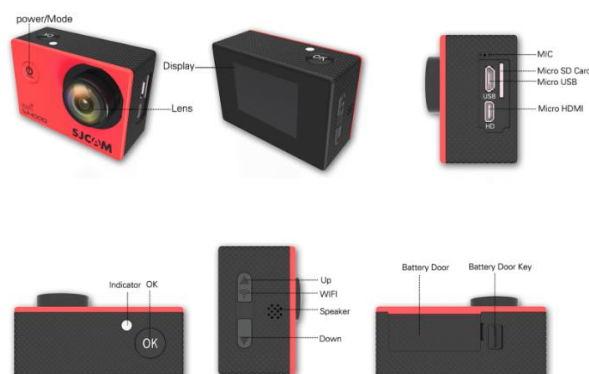


Figura 2.15: Características físicas la cámara fotográfica SJ4000 Wifi [31].

Algunas de las características de esta cámara son:

- Resolución de 12 Megapíxeles.
- Conexión por Wifi.
- Lente con gran angular de 170 grados.
- Zoom de 4X.
- Función WDR para capturar imágenes con mayor rango de luminosidad.
- Peso de 58g.
- Almacenamiento con tarjeta micro SD.
- Graba video a 1080p a 30fps.

2.2.8 Raspberry Pi

La *Raspberry Pi* es una computadora pequeña y de bajo costo, su tamaño es comparable al de una tarjeta de crédito, además posee interfaces de conexión para dispositivos periféricos como un TV, un monitor, un teclado o un mouse.

Sus usos más frecuentes son en proyectos de electrónica y robótica, aunque también se puede utilizar para muchas de las cosas que hace una PC de escritorio, como hojas de cálculo, procesamiento de textos, navegación por Internet, juegos o reproducción video de alta definición [32].



Figura 2.16: Raspberry Pi 3 Modelo B [33].

En el presente proyecto haremos uso de un *Raspberry Pi* de tercera generación modelo B (ver Figura 2.16), para controlar la cámara de fotos de manera remota mediante Wifi. Dentro de sus características tenemos: un CPU de 1.2GHz de 64-bits con 4 núcleos ARMv8, LAN inalámbrica 802.11n, Bluetooth 4.1, Bluetooth de bajo consumo (BLE), 1GB RAM, 4 puertos USB, 40 pines GPIO de propósito general, un puerto HDMI, un puerto Ethernet, un conector de audio y video de 3.5mm combinado, una interfaz para cámara (CSI), una interfaz para pantalla (DSI), 1 ranura para tarjeta Micro SD y un núcleo de gráficos VideoCore IV 3D [33].

2.3 Requerimientos de software.

Un requisito importante para la selección de software es que estas tecnologías sean libres y de código abierto, ya que esto nos permite usarlas sin restricciones y promueve el desarrollo colaborativo y la investigación con fines académicos [15]. Como primera consideración, el sistema operativo sobre el cual vamos a desarrollar la implementación del proyecto es una distribución de *Linux*, llamada *Arch Linux* [16].

En segundo lugar, para el presente proyecto vamos a necesitar desarrollar una interfaz gráfica, que será una aplicación de escritorio, y para ello utilizaremos un entorno de desarrollo integrado (IDE) llamado *Qt Creator*. Una de sus ventajas es que permite el desarrollo multiplataforma, además nos permite desarrollar en el lenguaje de programación C++ [17].

El propósito de la aplicación de escritorio es poder integrar varias funcionalidades relacionadas con la recolección de información realizada por vehículos no tripulados, agrupándolas dentro de funcionalidades en su interfaz.

En este proyecto implementaremos las dos primeras funcionalidades de la aplicación de escritorio, una tiene que ver con permitir la integración y ejecución de un software libre que sirve para el control y planificación de rutas en vehículos no tripulados. La segunda, tiene como objetivo generar imágenes panorámicas a partir de un conjunto de imágenes superpuestas por medio del proceso de *stitching*.

2.4 Diseño de la solución.

Para efectos de simplificar las explicaciones venideras, primero vamos asociar los nombres de las tecnologías de software y hardware libres que utilizaremos con los componentes que ya habíamos mencionado de manera general en las secciones anteriores. Así que tenemos lo siguiente:

- La aplicación para escritorio que se va a desarrollar en el IDE *Qt Creator*, la hemos denominado *Hermes* y será escrita en lenguaje C++.
- El software libre para el control y planificación de rutas de cobertura se llama *Mission Planner* [34].
- La tarjeta electrónica controladora de vuelo que soporta control por piloto automático y que se debe colocar abordo del dron, se llama *Pixhawk* [28].
- El software *Mission Planner* y la tarjeta controladora *Pixhawk*, fueron creados por la comunidad de desarrolladores que forman parte de la organización *Ardupilot*. Estas dos tecnologías interactúan entre sí por medio de una plataforma de software para control de piloto automático, que casualmente tiene el mismo nombre de la organización [35].
- El dispositivo embebido que se integrará con el *Pixhawk* y que se utilizará para manipular la cámara fotográfica mediante Wifi, es la *Raspberry Pi 3* modelo B [33].
- La cámara fotográfica para utilizar es la SJ4000 Wifi, de la marca SJCAM, tiene soporte para conexión a través de Wifi [31].

Dicho esto, podemos continuar con la explicación general del diseño de la solución del proyecto.

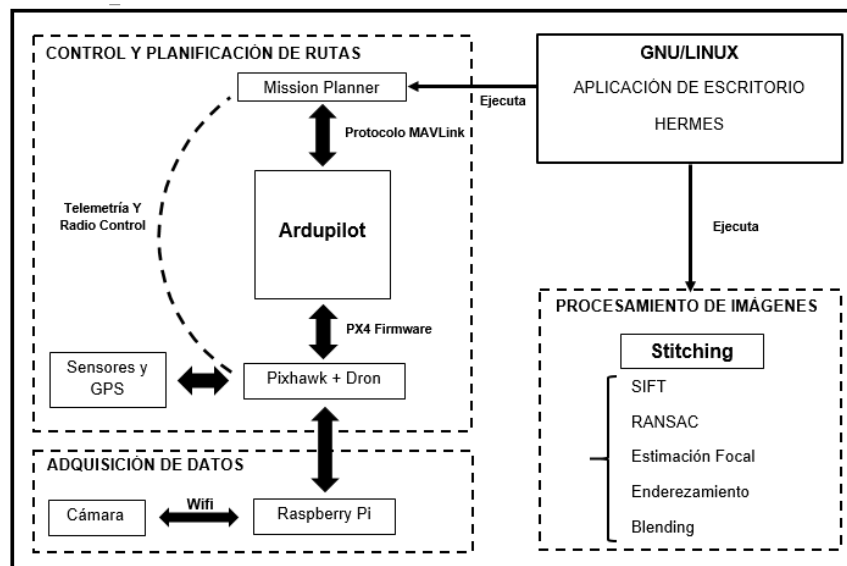


Figura 2.17: Esquema de la solución del proyecto. Tenemos los módulos de control y planificación de rutas, el módulo de procesamiento de imágenes y el módulo de adquisición de datos.

En la Figura 2.17 tenemos el esquema de la solución del proyecto, primero debemos empezar por entender la interacción de los componentes del sistema. Podemos observar que *Hermes* se está ejecutando sobre un sistema operativo GNU/Linux, a su vez podemos ver que de modo independiente se pueden ejecutar sus dos funcionalidades.

La primera funcionalidad nos permite ejecutar el módulo de control y planificación de rutas, en su interior tenemos al software *Mission Planner*, por medio de él podemos visualizar la interfaz de configuración y planificación de misiones de vuelo que serán programadas en el dron. Este módulo interactúa directamente con el módulo de adquisición de datos, el cual consiste en realizar el montaje de una cámara de fotos a bordo del dron, utilizando un dispositivo embebido entre ambos ya que no es posible integrarlos de manera directa por limitaciones de hardware. Esto nos permitirá tener comunicación programática entre el software *Mission Planner* y el dron.

Por último, tenemos la segunda funcionalidad de *Hermes*, esta nos permite ejecutar el módulo de procesamiento de imágenes, su función básicamente es tomar un dataset que está conformado por un conjunto de imágenes superpuestas correspondientes a una misma superficie, y aplicar el proceso de *stitching* para crear una sola imagen panorámica.

Para entender cómo funcionan internamente los módulos del sistema, dedicaremos una sección a cada uno de ellos, por lo tanto, hemos dividido la explicación de su implementación en cuatro partes:

- Módulo de control y planificación de rutas.
- Módulo de adquisición de datos.
- Aplicación de escritorio *Hermes*.
- Módulo de procesamiento de imágenes.

2.4.1 Módulo de control y planificación de rutas.

El propósito de este módulo es permitirnos crear rutas de vuelo para cualquier tipo de vehículo aéreo no tripulado utilizando un software libre llamado *Mission Planner*. Primero empezaremos hablando de su arquitectura interna y luego de la ejecución del software para planificar rutas de vuelo.

- **Arquitectura interna del módulo de control y planificación de rutas.**

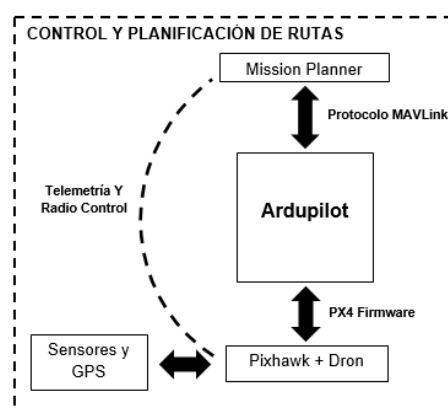


Figura 2.18: Esquema del módulo de control y planificación de rutas.

Dentro de este módulo existe una integración a través de la plataforma *Ardupilot*, entre *Mission Planner* y el hardware que, para nuestro caso, es un dron en cuyo interior se encuentra el controlador de vuelo *Pixhawk*. En la Figura 2.18 se observan los componentes que forman parte del este módulo y cómo están relacionados. *Ardupilot* ya viene empaquetado dentro de *Mission Planner* cuando realizamos su instalación en nuestro computador, así que no hay que preocuparse de hacer instalaciones independientes.

Para entender cómo se hace posible esta integración, podemos visualizar un esquema general de la arquitectura de *Ardupilot*, en la Figura 2.19 se muestra la interacción con los distintos componentes de hardware y software externos. Aquí es importante que nos enfoquemos en la capa central del gráfico, denominada *flight code*, ya que es ahí es donde se encuentra *Ardupilot* como tal, este se divide a su vez en tres subcapas.

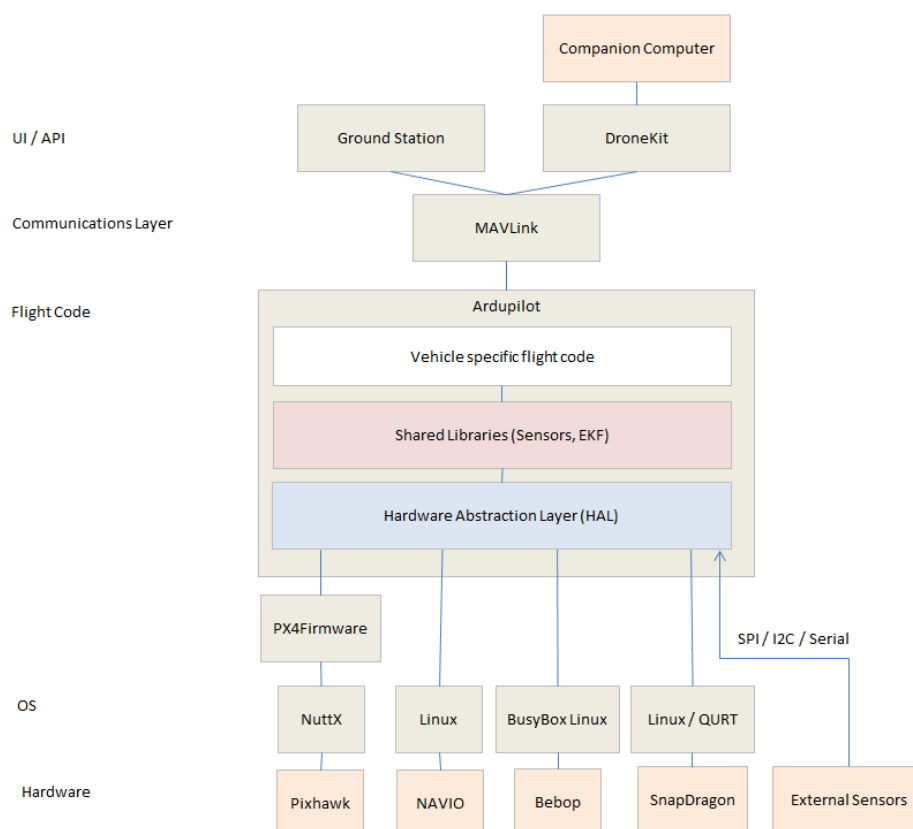


Figura 2.19: Esquema básico de la arquitectura de Ardupilot [35]

Describiéndolas de arriba hacia abajo, en la primera subcapa, *vehicle specific flight code*, están los *firmwares* para manejar a los distintos tipos de vehículos no tripulados (terrestres, aéreos, acuáticos, submarinos) y las librerías disponibles para crear programas clientes de control de vuelo, como es el caso del *Mission Planner*. En la segunda subcapa, *shared libraries*, tenemos a las librerías compartidas que nos permitirán interactuar con los controladores de sensores inerciales y módulos de GPS del hardware.

Por último, en la tercera subcapa, *hardware abstraction layer (HAL)*, es donde se encuentra la abstracción de software que permite ejecutar *ArduPilot* en diferentes plataformas (Windows, Linux, Mac OS, iOS, Android) y la abstracción de hardware para integrarse con diferentes tarjetas controladoras de vuelo basadas en AVR, PX4 o Linux, en nuestro caso la tarjeta *Pixhawk*.

- **Mission Planner para el control y planificación de rutas.**

Para la planificación de rutas primero debemos tener en cuenta que la tarjeta *Pixhawk* del dron debe estar conectada al computador vía USB o por telemetría. En la Figura 2.20, observamos un ejemplo del uso de *Mission Planner* como planificador de rutas.

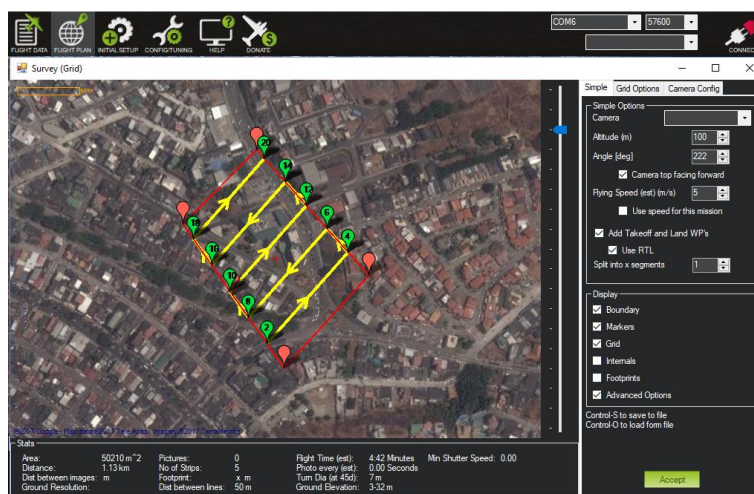


Figura 2.20: Ejemplo de cómo realizar una planificación de rutas de vuelo utilizando Mission Planner, como asistente de configuración.

La planificación de rutas se encuentra dentro de la opción *Flight Plan*. En este caso se ha creado una ruta de vuelo marcando varios puntos en el mapa, aquellos puntos de color rojo son los que delimitan el área dentro de la cual el dron se va a desplazar. Los puntos de color verde son aquellos por los cuales el dron tendrá que pasar de modo que cubra el área delimitada en su totalidad. Las líneas amarillas determinan la ruta que sigue el dron para cubrir esos puntos.

En la parte derecha nos permite configurar parámetros generales para modificar, por ejemplo, el ángulo de inclinación de las líneas de recorrido, el porcentaje de solapamiento de las escenas para la captura de fotos, el intervalo de tiempo entre la toma de cada foto, entre otros.

En la parte inferior del mapa, una vez que hemos graficado el curso de la misión de vuelo y revisado las configuraciones de parámetros, el software también nos muestra una serie de datos informativos, tales como, el área y la distancia a recorrer, la distancia entre líneas dentro del área o el tiempo estimado de duración total de la misión, un dato que es de suma importancia, ya que en parte, dependiendo de la cantidad de carga que pueda almacenar la batería del dron, este será capaz de realizar recorridos de mayor o menor extensión, según sea el caso.

Finalmente, una vez que hayamos aceptado los cambios, lo siguiente que veremos será una ventana con la ruta que hemos definido, como se observa en la Figura 2.21.



Figura 2.21: Cómo crear rutas de vuelo con Mission Planner.

En la parte inferior nos permite visualizar de manera detallada la información de cada punto en la ruta y nos permite modificarla si así lo deseamos. Si todo nos parece correcto, lo último que debemos hacer es dar clic en *Write WPs*, esta acción cargará nuestra misión de vuelo directo en la tarjeta *Pixhawk*.

Para entrar en más detalles del uso del programa para planificación de rutas de vuelo, configuración de hardware, calibración de sensores y controles puede consultar el Anexo B del documento.

2.4.2 Módulo de adquisición de datos

El presente módulo es el de adquisición de datos, tiene como objetivo permitirnos capturar fotografías aéreas de una superficie para su posterior procesamiento. Para conseguirlo, tenemos que realizar el montaje de una cámara de fotos a bordo del dron y lograr tener comunicación con la estación de control donde se encuentra *Mission Planner*. Por limitaciones de hardware, como no es posible conectar la cámara directamente a la tarjeta controladora de vuelo del dron, la solución consiste en utilizar un intermediario entre ambos, para ello usamos un dispositivo embebido conocido como *Raspberry Pi*.

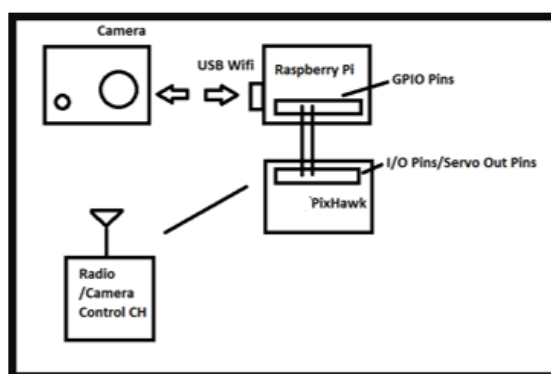


Figura 2.22: Esquema de conexión de la cámara de fotos con el Pixhawk mediante un Raspberry Pi.

En la Figura 2.22 podemos observar el esquema de conexión de hardware. Primero, estamos utilizando Wifi para establecer comunicación

entre la cámara de fotos y el *Raspberry Pi*, esto es posible debido a que la cámara SJ4000 soporta conexiones vía Wifi y está en la capacidad de recibir instrucciones usando el protocolo HTTP. Lo siguiente es programar un script en lenguaje *Python*, e instalarlo en el sistema operativo del *Raspberry Pi* para que se ejecute cada vez que el dispositivo se encienda. Dentro del script lo que se está haciendo es verificar primero que se haya establecido conexión por Wifi entre ambos dispositivos, luego por medio de instrucciones enviadas por HTTP, se establece el modo de la cámara en modo de captura de fotos, con su máxima resolución que es de 12 Megapíxeles, a partir de este momento el script queda corriendo como un proceso en el *Raspberry Pi*, a la espera de una señal de voltaje que recibirá a través de un par de sus pines de propósito general GPIO, que ya hemos configurado previamente como entrada de voltaje. En nuestro caso estamos utilizando los pines 29 y 30. La cámara tiene su propia batería así que no necesita conexiones adicionales, para energizar al *Raspberry Pi* se requieren 5V, por ende, se usan los pines de entrada 4 y 6, estos pines se cablean directamente a la batería del dron (ver Figura 2.23).

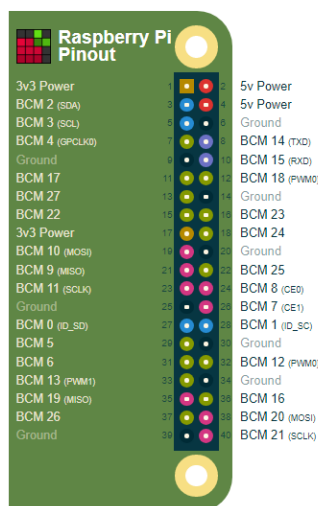


Figura 2.23: Numeración de pines del Raspberry Pi [37].

Por último, los pines 29 y 30 del *Raspberry pi* deben ser cableados de manera directa con un par de pines auxiliares del *Pixhawk* que son

configurando usando *Mission Planner*, en el cual indicamos sea por esos pines por donde envíe la señal de salida cada vez que el programa requiera tomar una foto durante la ejecución de una misión de vuelo. Un pin corresponde a una señal de 5V y el otro pin es la conexión a tierra.

2.4.3 Aplicación de escritorio Hermes

Hermes es una aplicación de escritorio compatible con las tecnologías de software libre seleccionadas. Para su desarrollo utilizamos el IDE *Qt Creator* y el lenguaje de programación C++.

En la Figura 2.24 podemos observar la pantalla principal de la aplicación, en la parte superior tenemos el menú donde constan las opciones que permitirán ejecutar sus funcionalidades.

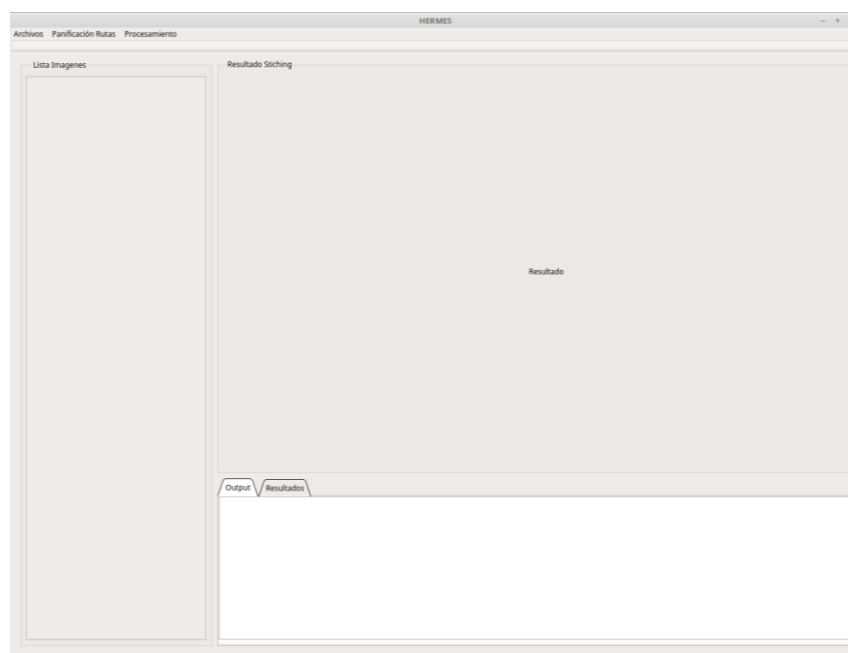


Figura 2.24: Pantalla inicial de la aplicación de escritorio Hermes.

Para empezar, en el menú de *Hermes* tenemos la opción *Archivos*, esta opción nos permitirá cargar los archivos de imágenes capturadas por el dron desde un directorio ubicado en nuestro computador local como se observa en la Figura 2.25. Se pueden seleccionar varias imágenes a la vez.

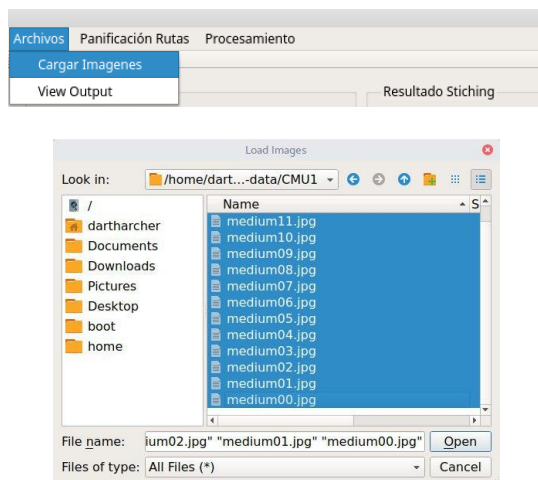


Figura 2.25: Proceso de selección de dataset de imágenes superpuestas, para su posterior procesamiento.

Luego tenemos la opción *Planificación Rutas*, con ella podemos ejecutar *Mission Planner*, el cual está siendo llamado por medio de una instrucción de línea de comandos. Con este software modo podremos realizar el proceso de calibración del dron o crear una ruta de vuelo para que sea ejecutada de manera automática (ver Figura 2.26).

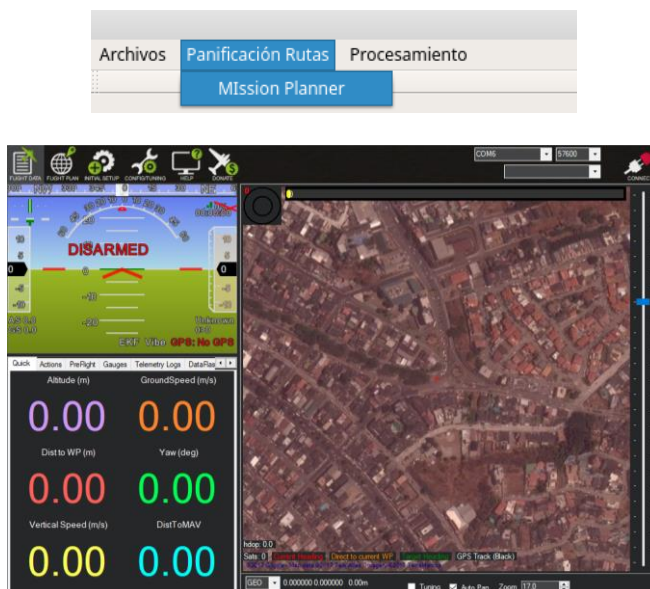


Figura 2.26: Pantalla inicial de Mission Planner, luego de haber sido ejecutado a través de Hermes.

Por último, tenemos la opción de *Procesamiento*. Como se observa en la Figura 2.27 podemos realizar el proceso de construcción de imágenes panorámicas ingresando a **Procesamiento | Panorámica | By Images Match**. En el lado izquierdo de la pantalla se muestra la lista de imágenes del dataset cada una con su vista en miniatura, al hacer clic sobre cada una de ellas se puede observar una vista preliminar. Con este método el *stitching* se realiza mediante un algoritmo que detecta características distintivas en cada imagen y permiten relacionarlas de ese modo se van ordenando para componer la imagen panorámica resultante. El resultado es mostrado en la parte central de la interfaz.

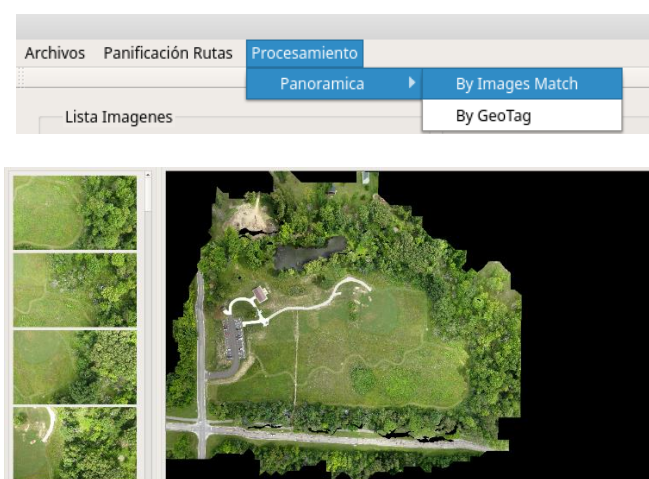


Figura 2.27: Proceso de stitching de imágenes a partir de una dataset seleccionado. El resultado final es una imagen panorámica de alta resolución.

Adicionalmente en este módulo se implementa la fusión de imágenes georreferenciadas, para poder realizar esto se ha integrado a Hermes un software de código abierto llamado *OpenDroneMap* [42], podemos realizar a esta función en el menú **Procesamiento | Panorámica | By Geotag**.

OpenDroneMap es una herramienta de post-procesamiento de datos obtenidos por vehículos aéreos que permite generación de *point clouds*, modelos de superficies con texturizado digital, ortorrectificación de

imágenes, ortomosaicos, modelos de elevación digital y reconstrucción de superficies en 3D. En nuestro caso lo utilizamos para hacer el stitching de imágenes, pero basado en la información geográfica de las de mismas, de este modo se realiza un procesamiento de imágenes más preciso ya que el ordenamiento de imágenes esta dado por las coordenadas de latitud y longitud lo cual permite a los algoritmos de creación de imágenes panorámicas reducir el costo de cálculo computacional y agilizar el proceso.

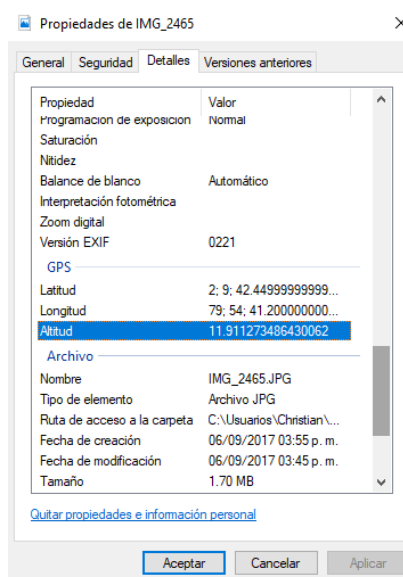


Figura 2.28: Ejemplo del contenido metadata de una imagen con información georreferenciada.

Como se observa en la figura 2.28 tenemos las propiedades de una imagen que incluye *metadata* con información geográfica de latitud, longitud y altura. Existen varias formas de agregar esta información, una de ellas es que la cámara fotográfica que se utilice para capturar la *data* tenga integrado un módulo GPS o se le pueda adaptar uno externo, de ese modo podrán generarse fotos con información geográfica de manera automática.

La otra opción de georreferenciamiento de imágenes también se la puede hacer de manera manual aprovechando los registros de vuelo generados

por *Mission Planner* en los cuales por cada misión se tiene el registro geográfico de latitud, longitud y altura al momento de tomar cada una de las fotos gracias a que el dron tiene conectado un módulo GPS y sensores que permiten determinar a qué altura se encuentra. En el Anexo C del documento se explica el proceso para georreferenciar imágenes con *Mission Planner*.

2.4.4 Módulo de procesamiento de imágenes

En este módulo vamos a realizar el procesamiento de imágenes, la entrada de información se produce a partir de las imágenes que se capturan por medio de la cámara fotográfica a bordo del dron durante las misiones de vuelo programadas.

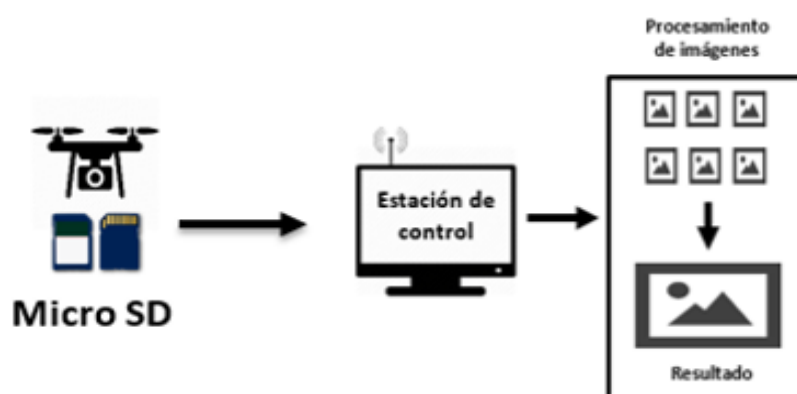


Figura 2.29: Interacción de los componentes módulo de procesamiento de imágenes.

Estas imágenes son almacenadas en la memoria micro SD de la cámara, como se muestra en la Figura 2.29. Una vez que el dron ha terminado su recorrido, debe regresar y aterrizar cerca de la estación de control de vuelo, luego debemos copiar las imágenes que se encuentran en la tarjeta micro SD a al computador.

Internamente en el módulo, para realizar el proceso de *stitching* de imágenes, se está utilizando una librería escrita en C++ llamada *OpenPano* [38]. Esta librería nos permite realizar el procesamiento de imágenes para generar panorámicas usando un conjunto de algoritmos

especializados, como podemos ver en la Figura 2.30, los cuales vamos a detallar a continuación [39].



Figura 2.30: Esquema del módulo de procesamiento de imágenes.

Primero necesitamos detectar *keypoints* o características invariantes dentro de una secuencia de imágenes aplicando el algoritmo de visión por computadora, llamado SIFT (*Scale-invariant feature transform*) [40]. El algoritmo enfoca este tema como un problema de relación de correspondencia entre imágenes, al final lo que nos permitirá hacer es, relacionar y ordenar un conjunto de imágenes que pertenecen a una misma escena, a partir de un conjunto de imágenes desordenadas, sin importar la escala o la perspectiva que estas tengan. Para ser más precisos, las características detectadas en las imágenes son invariantes respecto a los siguientes factores: la escala, traslación, rotación, iluminación y afinidades.

Como ejemplo, en la Figura 2.31, podemos observar dos imágenes que pertenecen a la misma escena, estas han sido captadas de manera superpuesta y los puntos graficados sobre ellas corresponden a las características invariantes de cada imagen obtenidas con SIFT.



Figura 2.31: Identificación de características en imágenes usando el algoritmo SIFT [40].

El siguiente paso consiste en aplicar el algoritmo RANSAC (*Random Sample Consensus*), un proceso iterativo que nos ayuda a estimar una matriz de homografía o transformación afín. Por homografía se entiende a toda transformación proyectiva que determina una correspondencia entre dos figuras geométricas de superficie plana [41].

Entonces a partir de los puntos características invariantes obtenidos con SIFT, como podemos ver en la Figura 2.32, que lo que nos permite hacer RANSAC es encontrar puntos que coincidan con una alta probabilidad en ambas imágenes y que obedezcan a un modelo matemático, a estos puntos los denomina *inliers*, de modo que aquellos puntos que no encajen en el modelo los denomina *outliers*, y son descartados del resultado final.

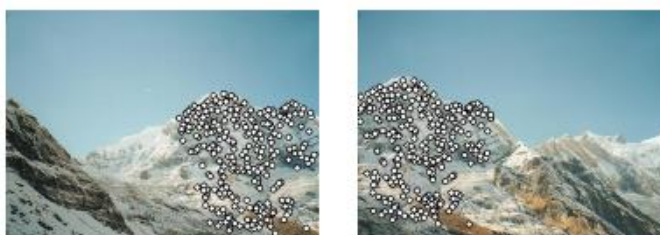


Figura 2.32: Imágenes relacionadas basado en el número de inliers obtenidos por RANSAC [40].

Para tener un correcto funcionamiento de los algoritmos, las imágenes deben tener entre un 75% hasta 85% de solapamiento, en algunos casos hasta 90%, esto se debe a que por ejemplo en imágenes donde las escenas o paisajes son muy homogéneas existe mayor dificultad para detectar puntos característicos o *keypoints*. Si los algoritmos no encuentran la cantidad de puntos necesarios, probablemente no realizarán el *stitching*. También suelen haber falencias si se tienen imágenes muy desenfocadas o con cambios de iluminación muy bruscos.

Finalmente, en el proceso se aplican métodos de la estimación focal y enderezamiento, combinando las imágenes, mitigando las diferencias de escala o de exposición lumínica generando una única imagen panorámica de alta resolución lo más homogénea posible.

CAPÍTULO 3

3. PRUEBAS DEL SISTEMA Y RESULTADOS.

En el presente capítulo analizamos las pruebas experimentales realizadas con el sistema funcionando en un entorno real con el fin de cumplir con los objetivos propuestos del proyecto. Antes de realizar cualquier tipo de maniobras con vehículos aéreos no tripulados es importante consultar las regulaciones internas establecidas por la Dirección General de Aviación Civil del Ecuador, con el fin de precautelar la seguridad operacional en las actividades aéreas, usuarios del transporte aéreo y público en general [45].

COMPONENTE	DESCRIPCIÓN	PESO
Hélices (4)	10 x 4.5	44 g
Motores brushless (4)	1000Kv	224 g
Batería de Litio y Polímero	2450 mAh	200 g
Controlador de Vuelo	Pixhawk 2.4.8	38 g
Cámara fotográfica	SJ4000 Wifi	160 g
Telemetría	3DR de 915 MHz	17 g
Módulo GPS	Ublox-M8N	45 g
Hardware auxiliar	Raspberry Pi modelo 3B	92 g
Receptor de radio control	FS-IA6B salida PPM	12 g
Switch de seguridad y buzzer	-	5 g
Armazón (frame)	Tipo quadcopter PVC	282 g

Tabla 1: Componentes utilizados a bordo del dron.

En la Tabla 1 podemos ver la lista del equipo usado a bordo del dron en todas las misiones de prueba para cumplir con estas especificaciones. La configuración del dron fue diseñada para que pueda levantar durante el vuelo un peso de poco más de

1300 g y tener una autonomía de vuelo de aproximadamente 8 minutos por cada misión utilizando una sola batería.

Se realizaron siete misiones de vuelo distribuidas en un período de tres semanas, de las cuales solamente escogimos la información de dos misiones para el análisis, esto fue a causa de la distorsión geométrica de la lente de la cámara SJ400 que tiene un efecto de ojo de pez (Fisheye), este efecto hace que los bordes de las imágenes tiendan a curvarse y a contraer la información en esas zonas (ver Figura 3.1). Esto a su vez reduce el porcentaje de superposición entre cada imagen, por lo tanto, genera una mala correspondencia y no permite generar la imagen panorámica.



Figura 3.1: Imagen captada con distorsión de ojo de pez (Fisheye) a una altura de 20 metros.

La primera misión se la realizó dentro de las instalaciones de la Escuela Superior Politécnica del Litoral en el área de tecnologías en las canchas deportivas, la segunda misión se llevó a cabo en la provincia de Santa Elena, en la comuna Río Verde en una pequeña finca del sector. Al final del capítulo se presentan resultados con dataset públicos de 77 y 211 imágenes, esto es para comprobar la validez de los algoritmos

implementados y corroborar que los resultados erróneos obtenidos se deben en parte a la influencia del efecto ojo de pescado de la lente de la cámara.

Para la ejecución de las pruebas del sistema primero realizamos de manera obligatoria el procedimiento de configuración y calibración de hardware explicado en el anexo A del documento. Luego para cada una de las misiones se siguieron los pasos de planificación rutas de mapeo con *Mission Planner* detallados en el anexo B del documento.

3.1 Caso de prueba cancha ESPOL.

En la Tabla 2 observamos los datos generados para dicha misión, los parámetros estimados son obtenidos con *Mission Planner* en el momento en que se está planificando la ruta de vuelo, como podemos observar existe una diferencia notable de más de 2 minutos en el tiempo estimado de vuelo, esto se debe a la oposición que ejerce el viento durante la ejecución de la misión, ya que como no se cuenta con un sensor de velocidad de viento a bordo del dron, el software estima el tiempo de vuelo asumiendo condiciones ideales basado en la superficie que debe de cubrir y en la velocidad promedio programada para el dron que es de 5m/s.

PARÁMETROS	ESTIMACIÓN	PRUEBA
Superficie	1322 m ²	1322 m ²
Resolución de imágenes	0.42 cm/pixel	0.42 cm/pixel
Velocidad	5 m/s	5 m/s
Altura	12 m	12 m
Tiempo de vuelo	2m 17s	4m 30s
Cantidad de Imágenes	105	65
Tiempo de captura entre cada foto	1 s	1.61 s

Tabla 2: Comparación de los datos de vuelo estimados para la misión Cancha ESPOL.

La cantidad de imágenes obtenidas durante esta misión fueron de 65 en comparación a las 105 imágenes que se estimaron, esto se debe a la limitación física que tiene la cámara fotográfica.

La velocidad de captura es mucho más lenta y el software hace una estimación de captura de fotos cada segundo basado en que el porcentaje de superposición configurado entre cada imagen es del 80%. La resolución de la cámara es de 12 Megapíxeles, por lo tanto, cada imagen tiene un tamaño de 4032 x 3024 píxeles (ver Figura 3.2).



Figura 3.2: Imagen de un sector de la cancha tomada a 12 m de altura.

Con este dataset de 65 imágenes primero realizamos el *stitching* por correspondencia de imágenes usando SIFT y sin considerar georreferenciamiento. Las imágenes han sido cargadas en el orden cronológico en que fueron capturadas, esto es un requisito obligatorio para que el algoritmo pueda realizar una mejor estimación en sus resultados. El algoritmo no fue capaz de resolver la relación entre todas las imágenes debido a la distorsión en los bordes causada por el efecto *Fisheye* de la lente de la cámara. Podemos ver en el cuadro de salida que nos indica que luego de haber realizado 7 iteraciones no pudo encontrar relación alguna entre 58 de las imágenes y por lo tanto nos muestra una imagen de color negro en respuesta a que no se pudo generar la panorámica (ver Figura 3.3).

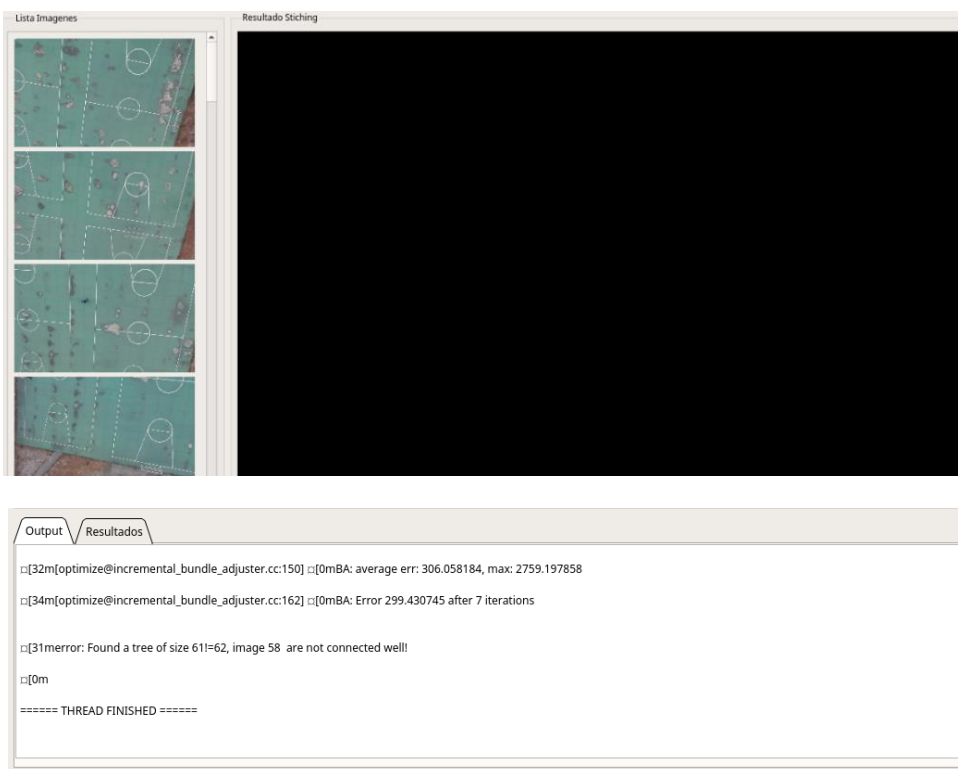


Figura 3.3: Resultado de fusión de imágenes por el método de correspondencia de imágenes usando SIFT, sin georreferenciamiento.

Finalmente ejecutamos el proceso de *stitching* utilizando el georreferenciamiento de las imágenes con la opción *By Geotag* de *Hermes* como se puede observar en la Figura 3.4, este algoritmo usa por detrás la integración con la plataforma *OpenDroneMap*, nuevamente el software no fue capaz de generar la imagen panorámica a partir de las imágenes provistas generando un mensaje de excepción en la pantalla de salida.



Figura 3.4: Resultado de fusión de imágenes por georreferenciamiento usando algoritmo de OpenDroneMap.

3.2 Caso de prueba comuna Río Verde.

La misión duró casi 10 minutos cubriendo un área de poco más de 80 metros cuadrados, la velocidad promedio programada para el dron fue de 5m/s, la altura fue de 20 metros, para evitar chocar con los árboles presentes en el sector. Esta misión se dividió en dos partes ya que cada batería nos dio una autonomía de vuelo de 8 minutos. La cantidad de imágenes obtenidas fue de 86 frente a las 225 estimadas, la velocidad de captura de la cámara fue mucho más lenta en comparación a la misión de la cancha ESPOL.

PARÁMETROS	ESTIMACIÓN	PRUEBA
Superficie	6450 m ²	6450 m ²
Resolución de imágenes	0.70 cm/pixel	0.70 cm/pixel
Velocidad	5 m/s	5 m/s
Altura	20 m	20 m
Tiempo de vuelo	5m 07s	9m 50s
Cantidad de Imágenes	225	86
Tiempo de captura entre cada foto	1.13 s	2.61 s

Tabla 3: Comparación de los datos de vuelo estimados para la misión en la comuna Río Verde.

En la Tabla 3 observamos los datos estimados generados para esta misión en comparación con los resultados de la prueba. Con este dataset de 86 imágenes primero realizamos el *stitching* por correspondencia de imágenes usando SIFT y luego consideramos el georreferenciamiento. El algoritmo en esta ocasión tampoco fue capaz de encontrar una relación entre todas las imágenes debido a la distorsión en los bordes causada por el efecto *Fisheye* de la lente de la cámara.

En la Figura 3.5 podemos ver que en el cuadro de salida se nos indica que luego de haber realizado 8 iteraciones no pudo encontrar relación alguna entre 85 de las imágenes y por lo tanto nos muestra una imagen de color negro en respuesta a que no se pudo generar la panorámica.

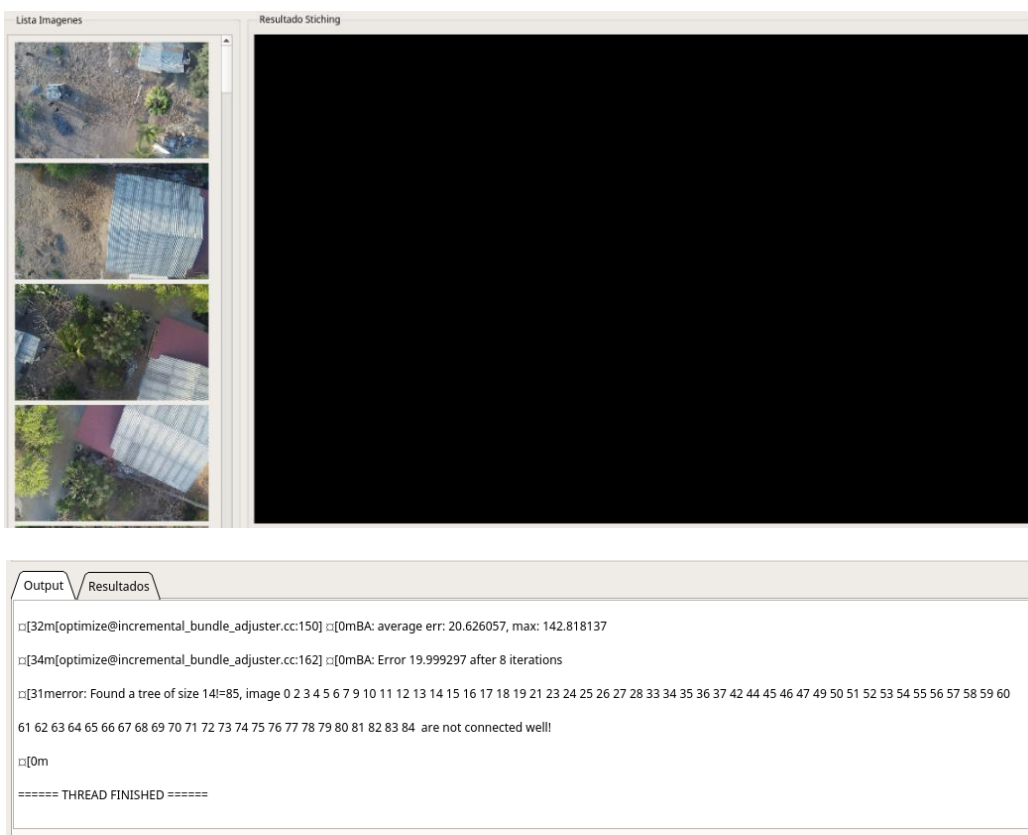


Figura 3.5: Resultado de fusión de imágenes por el método de correspondencia de imágenes de la misión en la comuna Río Verde.

3.3 Caso de prueba dataset públicos.

Para este caso hemos tomado imágenes de dos dataset públicos disponibles en la Web, el primero es el de 77 imágenes fue tomado del repositorio de *Github* de *OpenDroneMap* que ha sido generado por la comunidad [48]. El segundo es un dataset de ejemplo de la página Web de *Sensefly* [49], está compuesto por 211 imágenes,

En la Tabla 4 observamos los datos comparativos de estos dataset, para el de 77 imágenes tenemos imágenes de mayor tamaño tomadas con una cámara SONY DSC-WX220 con una resolución de 18 MP y para el de 211 se utilizó una cámara Canon IXUS 125 HS con una resolución de 15.9 MP, ambas no tienen distorsión del efecto ojo de pescado lo cual permite obtener imágenes de mejor calidad.

PARÁMETROS	Dataset de 77 imágenes	Dataset de 211 imágenes	Dataset comuna Río Verde	Dataset Cancha ESPOL
Modelo de Cámara	SONY DSC-WX220	Canon IXUS 125 HS	SJ4000 Wifi	SJ4000 Wifi
Distancia Focal	4.5 mm	4.3 mm	4.4 mm	4.4 mm
Megapíxeles	18	15.9	12	12
Tamaño de cada imagen	4896 x 3672	4608 x 3456	4032 x 3024	4032 x 3024
Altura GPS	346.315 m	162 m	20 m	12 m
cm/pixel	-	5cm/px	0.70cm/px	0.70cm/px

Tabla 4: Información de las imágenes de los dataset obtenidos del repositorio de OpenDroneMap [48].

En las Figura 3.6A y 3.6B podemos observar el resultado de las imágenes panorámicas obtenida al ejecutar el proceso de *stitching* en los dataset de 77 y 211 imágenes respectivamente, utilizando la opción por georreferenciamiento.

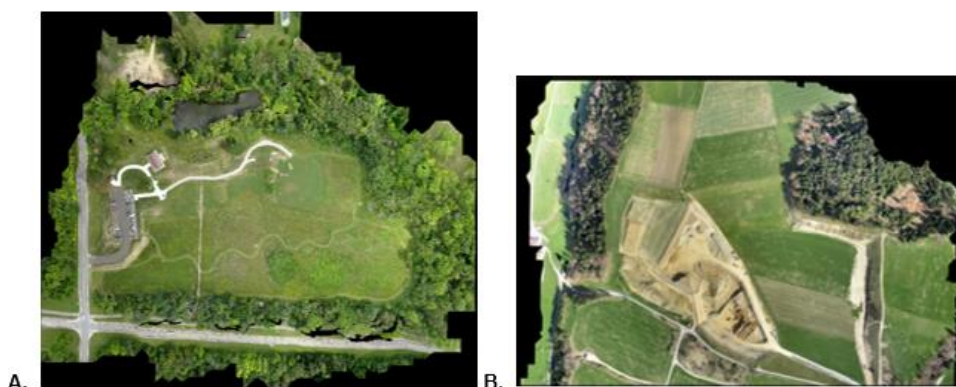


Figura 3.6: A. Resultado de fusión de imágenes usando georreferenciamiento, para el dataset de 77 imágenes. B. Resultado de fusión de imágenes usando georreferenciamiento, para el dataset de 211 imágenes.

En la Tabla 5 tenemos los resultados de la ejecución de las dos pruebas, la máquina en la cual se realizó el procesamiento fue una portátil con procesador Core i7 de 3.5 GHz y 8 GB de memoria RAM. El dataset de 211 imágenes demoró 8 horas en terminar de procesar y generó un archivo final de 473 MB. El dataset de 77 imágenes generó un archivo de 87 MB y el tiempo que le tomó el procesamiento fue de 2 horas.

PARÁMETROS	77 imágenes	211 imágenes
Tiempo de ejecución	2h 06m 11s	8h 06m 10s
Tamaño archivo	87 MB	473 MB
Tipo de archivo	PNG	PNG
Megapíxeles	51.5	264.4
Tamaño de imagen panorámica	7675 x 6709	16551 x 15976
Tipo de color	RGB con Alfa	RGB con Alfa

Tabla 5: Resultados del procesamiento de imágenes usando georreferenciamiento en Hermes.

CONCLUSIONES Y RECOMENDACIONES

Como conclusiones podemos decir que:

- Basado en los resultados obtenidos de las pruebas de vuelo realizadas, se pudo evidenciar que el tiempo de duración de las rutas de mapeo depende directamente de la cantidad de carga que puedan almacenar las baterías del dron, de la aerodinámica del vehículo y de la eficiencia en el consumo de energía que tengan los motores, basado en el peso que tenga que soportar el vehículo para ejecutar las misiones de vuelo.
- Usar SIFT para la generación de imágenes panorámicas no siempre resulta ser efectivo, sobre todo en casos donde las imágenes obtenidas corresponden a superficies homogéneas, en tales casos la detección de características invariantes se ve afectada. Además, SIFT calcula los gradientes por cada píxel, esto genera un costo computacional muy elevado, lo cual lo hace inadecuado para ser utilizado en aplicaciones que requieran procesamiento de datos en tiempo real.
- Gracias a la herramienta de software *Mission Planner*, fuimos capaces de planificar rutas para el mapeo de superficies de manera automática mediante la creación de polígonos cerrados sobre un mapa virtual. Dentro del proceso de configuración y creación de rutas se pueden personalizar parámetros fundamentales como el tamaño de la superficie de interés, las características técnicas de la cámara fotográfica, el porcentaje de superposición de las imágenes y la altitud relativa a la que el dron debe realizar la misión de vuelo.
- Se logró integrar tecnologías de hardware y software libres para poder crear un ecosistema que nos permita monitorear tareas de navegación y mapeo de manera autónoma, sin la intervención de una persona en el control de un vehículo aéreo no tripulado.

Recomendaciones de hardware:

- Para obtener imágenes georreferenciadas de manera automática se puede optar por una cámara fotográfica con GPS incorporado.
- La lente de la cámara no debe tener efecto de ojo de pez (*Fisheye*), ya que esto distorsiona los bordes de las fotografías captadas a gran altura y reduce el porcentaje de superposición en las imágenes lo cual no permitirá realizar el proceso de *stitching* de manera satisfactoria.
- Para casos en donde se requiera realizar actividades de mapeo de superficies para la elaboración de ortofoto mapas, generación de modelos 3D de terrenos, cálculo de volúmenes, trabajos topográficos o análisis multiespectral de vegetación en campos agrícolas, es mucho mejor utilizar un dron de ala fija que tenga forma de aeroplano. Su ventaja frente a los drones multirrotores es que consumen menos energía, en consecuencia, sus tiempos de vuelo son mucho más extendidos, hablamos de unos 45 a 60 minutos de vuelo por cada misión, por lo tanto, esto nos permitirá tener una mayor cobertura de terreno.
- Para trabajos en agricultura donde se necesitan por ejemplo conocer el estado de salud de los cultivos, estado de fertilidad del suelo, detección de plagas o enfermedades, se debe utilizar cámaras multiespectrales. Estas cámaras utilizan bandas de onda verde, roja, borde rojo (*red-edge*) e infrarrojo cercano (NIR) para capturar imágenes visibles e invisibles de los cultivos y la vegetación en general. Soluciones comerciales de alta calidad existentes para drones son las cámaras Canon, Mapir, Sequoia y Rededge de Micasense.
- Se puede conectar y configurar dispositivos embebidos como la *Raspberry Pi* para que pueda comunicarse con el controlador de vuelo *Pixhawk* usando el protocolo *MAVLink* [46] por medio de conexión serial. Esto se puede utilizar para tareas adicionales a bordo como el reconocimiento de imágenes. También puede adecuarse como un módulo de vista en primera persona (FPV) para recibir transmisión de video en directo.
- El *Pixhawk* permite la conexión de un dispositivo de hardware llamado *Gimbal*, básicamente es un estabilizador de cámaras, su uso es muy recomendable para casos en los que un dron sea utilizado para tareas como grabación de video o transmisión de video en tiempo real. Soporta *Gimbals* con 3 ejes de

movimiento, y para manipular los servomotores que controlan el movimiento del Gimbal necesitamos de un control remoto con un sexto canal.

Recomendaciones de software:

- Para casos donde tenemos un conjunto de imágenes de contenido homogéneo en las cuales se dificulta la detección de puntos característicos invariantes, se podría mejorar el proceso de *stitching*, realizando una fase de *pre-stitching* para fusionar las imágenes de manera ordenada etiquetándolas por estampa de tiempo y utilizando parámetros que ya conocemos como la altura a la que se tomaron las imágenes.
- Una de las alternativas para la mejora de rendimiento en la detección de características invariantes en imágenes, es el algoritmo SURF (*Speed-up Robust Features*) [47], es tres veces más rápido en comparación a SIFT ya que utiliza una menor cantidad de puntos invariantes para relacionar las imágenes. Aun así, nos es muy bueno cuando se trata de imágenes que tienen cambios de iluminación y perspectiva.
- Se pueden agregar a la aplicación *Hermes* funciones adicionales de *OpenDroneMap* que permitan realizar, aparte del *stitching* de imágenes panorámicas, la generación de nubes de puntos, modelos de superficies con texturizado digital, ortorrectificación de imágenes, ortomosaicos, modelos de elevación digital y reconstrucción de superficies en 3D.

BIBLIOGRAFÍA

[1] S. Salvador Sarauz y A. Monteros Guerrero, "Panorama agroeconómico del Ecuador. Una visión del 2015", 2015. [En línea]. Disponible en:

http://sinagap.agricultura.gob.ec/pdf/estudios_agroeconomicos/panorama_agroeconomico_ecuador2015.pdf.

[2] "Las 5 tendencias agrícolas para el 2016", Agriculturers.com | Red de Especialistas en Agricultura, 2016. [En línea]. Disponible en: <http://agriculturers.com/las-5-tendencias-agricolas-2016/>.

[3] "eBee: senseFly SA", Sensefly.com, 2017. [En línea]. Disponible en: <https://www.sensefly.com/drones/ebee.html>.

[4] "Servicios Profesionales con Robots Aéreos (Drones) en Ecuador", Servicios Profesionales con Robots Aéreos (Drones) en Ecuador, 2017. [En línea]. Disponible en: <https://www.latitideas.com/>.

[5] "Generate 2D and 3D information, purely from images with Pix4D", Pix4D, 2017. [En línea]. Disponible en: <https://pix4d.com/>.

[6] "Agricultura de Precisión, VANT Comercial y Drones Agricultores para la Venta", Precisionhawk.com, 2017. [En línea]. Disponible en:

<http://www.precisionhawk.com/es/>.

[7] "A survey on coverage path planning for robotics", Sciencedirect.com, 2013. [En línea]. Disponible en:

<http://www.sciencedirect.com/science/article/pii/S092188901300167X>.

[8] E. Acar, H. Choset and A. Rizzi, "Morse Decompositions for Coverage Tasks", The International Journal of Robotics Research - Ercan U. Acar, Howie Choset, Alfred A. Rizzi, Prasad N. Atkar, Douglas Hull, 2002", Journals.sagepub.com, 2002. [En línea]. Disponible en: <http://journals.sagepub.com/doi/10.1177/027836402320556359>.

[9] "A Methodical Review on Image Stitching and Video Stitching Techniques", ripublication.com, 2017. [En línea]. Disponible en:

https://www.ripublication.com/ijaer16/ijaerv11n5_80.pdf.

[10] "Image Stitching a Simplistic Tutorial - BitsMakeMeCrazy Kushal Vyas's Blog", Kushalvyas.github.io, 2017. [En línea]. Disponible en:

<http://kushalvyas.github.io/stitching.html>

[11] S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems", Academia.edu, 2014. [En línea]. Disponible en:

https://www.academia.edu/4186102/A_survey_of_techniques_for_improving_energy_efficiency_in_embedded_computing_systems.

[12] R. Maya and M. Loayza, "Desarrollo de una aplicación para el trazo de rutas planificadas de un vehículo a escala reducida usando un dispositivo embebido y GPS", Dspace.espol.edu.ec, 2017. [En línea]. Disponible en:

<http://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/37297/D-103445.pdf?sequence=-1&isAllowed=y>.

[13] "Qué es SCRUM", Proyectos Ágiles, 2017. [En línea]. Disponible en:

<https://proyectosagiles.org/que-es-scrum/>.

[14] "¿Qué significa Scrum?", Mintakastar, 2017. [En línea]. Disponible en:

<https://mintakastar.wordpress.com/2012/03/09/que-significa-scrum/>.

[15] "License (GPLv3) — Dev documentation", Ardupilot.org, 2017. [En línea]. Disponible en: <http://ardupilot.org/dev/docs/license-gplv3.html>.

[16] "Arch Linux", Archlinux.org, 2017. [En línea]. Disponible en:

<https://www.archlinux.org/>.

[17] "Qt Creator Manual", Doc.qt.io, 2017. [En línea]. Disponible en:

<http://doc.qt.io/qtcreator/>.

[18] Aliexpress.com, 2017. [En línea]. Disponible en: <https://www.aliexpress.com/item-img/4-axle-RC-Multi-QuadCopter-Drone-RTF-ARF-KK-V2-3-Circuit-board-1000KV-Motor-30A/32751238739.html?spm=a219c.10010108.1000017.1.eMBO7c#>.

[19] F. PDB and U. Tech, "F450 Quadcopter Frame with integrated PDB", Unmanned Tech, 2017. [En línea]. Disponible en: <https://www.unmannedtechshop.co.uk/f450-quadcopter-frame-with-integrated-pdb/>.

[20] D. Mahoney, What's So Great About Brushless Motors?, Popular Mechanics, 2017. [En línea]. Disponible en:

<http://www.popularmechanics.com/home/reviews/a8109/whats-so-great-about-brushless-motor-power-tools/>.

[21] "A2212/13T TECHNICAL DATA", rhydolabz.com, 2017. [En línea]. Disponible en: http://www.rhydolabz.com/documents/26/BLDC_A2212_13T.pdf.

[22] "HobbyKing Slowfly Propeller 10x4.5 Black (CW/CCW) (4pcs)", Hobbyking, 2017. [En línea]. Disponible en: https://hobbyking.com/en_us/hobbyking-slowfly-propeller-10x4-5-black-cw-ccw-4pcs.html?___store=en_us.

[23] "DONG YANG MODEL TECHNOLOGY CO., LIMITED|Servo|Brushless motor|Speed controller|Propeller|Multi-Rotor Brushless Motor|Gimbal Brushless Motor|Multicopter|FPV VideoTX RX|RC accessories", Dys.hk, 2017. [En línea]. Disponible en: <http://www.dys.hk/ProductShow.asp?ID=50>.

[24] "Ten R/C Mistakes to Avoid", Crash Test Hobby "Fly More - Fix Less", 2017. [En línea]. Disponible en: <http://www.crashtesthobby.com/ten-rc-mistakes-to-avoid.html>.

[25] "Li-Po Battery (Theory) : Energy Storage Labs : Mechanical Engineering : Amrita Vishwa Vidyapeetham Virtual Lab", Vlab.amrita.edu, 2017. [En línea]. Disponible en: <http://vlab.amrita.edu/?sub=77&brch=270&sim=1631&cnt=1>.

[26] "A Guide to Understanding LiPo Batteries", Roger's Hobby Center, 2017. [En línea]. Disponible en: <https://rogershobbycenter.com/lipoguide/>.

[27] "Common Power Module — Copter documentation", Ardupilot.org, 2017. [En línea]. Disponible en:

<http://ardupilot.org/copter/docs/common-3dr-power-module.html>.

[28] "Radiolink PIXHAWK - Radiolink Electronics Co., Ltd.", Radiolink.com.cn, 2017. [En línea]. Disponible en: <http://radiolink.com.cn/doce/product-detail-116.html>.

- [29] "Beginners guide to drone autopilots (flight controllers) and how they work", DroneTrest, 2015. [En línea]. Disponible en: <http://www.dronetrest.com/t/beginners-guide-to-drone-autopilots-flight-controllers-and-how-they-work/1380>.
- [30] "3DR-Radio-V2", 3dr.com, 2013. [En línea]. Disponible en: <https://3dr.com/wp-content/uploads/2013/10/3DR-Radio-V2-doc1.pdf>.
- [31] "SJCAM SJ4000 Wifi • SJCAM Official Website", SJCAM Official Website, 2017. [En línea]. Disponible en: <http://sjcamhd.com/sjcam-sj4000-wifi/>.
- [32] "Raspberry Pi FAQs - Frequently Asked Questions", Raspberry Pi, 2017. [En línea]. Disponible en: <https://www.raspberrypi.org/help/faqs/>.
- [33] "Raspberry Pi 3 Model B - Raspberry Pi", Raspberry Pi, 2017. [En línea]. Disponible en: <https://www.raspberrypi.org/products/raspberrypi-3-model-b/>.
- [34] "Mission Planner Overview — Mission Planner documentation", Ardupilot.org, 2017. [En línea]. Disponible en: <http://ardupilot.org/planner/docs/mission-planner-overview.html>.
- [35] "Learning ArduPilot — Introduction — Dev documentation", Ardupilot.org, 2017. [En línea]. Disponible en: <http://ardupilot.org/dev/docs/learning-ardupilot-introduction.html>.
- [36] "How to control a GoPro Camera on APM/Pixhawk with a Raspberry PI using Wifi – DIY Drones", Diydrone.com, 2017. [En línea]. Disponible en: <http://diydrone.com/m/blogpost?id=705844%3ABlogPost%3A1991119>.
- [37] "Raspberry Pi GPIO Pinout", Pinout.xyz, 2017. [En línea]. Disponible en: <https://pinout.xyz/>.
- [38] "ppwwyyxx/OpenPano", GitHub, 2017. [En línea]. Disponible en: <https://github.com/ppwwyyxx/OpenPano>.
- [39] Y. Wu, "OpenPano: How to write a Panorama Stitcher", Yuxin's Blog, 2016. [En línea]. Disponible en: <http://ppwwyyxx.com/2016/How-to-Write-a-Panorama-Stitcher/>.

[40] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching using Invariant Features", <http://matthewalunbrown.com>, 2007. [En línea]. Disponible en: <http://matthewalunbrown.com/papers/ijcv2007.pdf>.

[41] P. Flores and J. Braum, "Algoritmo RANSAC: fundamento teórico", <http://iie.fing.edu.uy>, 2011. [En línea]. Disponible en:

<http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/FundamentoRANSAC.pdf>.

[42] "OpenDroneMap", [Opendronemap.org](http://opendronemap.org), 2017. [En línea]. Disponible en:

<http://opendronemap.org/>.

[43] "Connect ESCs and Motors — Copter documentation", [Ardupilot.org](http://ardupilot.org), 2017. [En línea]. Disponible en: <http://ardupilot.org/copter/docs/connect-escs-and-motors.html>.

[44] "Accelerometer Calibration in Mission Planner — Copter documentation", [Ardupilot.org](http://ardupilot.org), 2017. [En línea]. Disponible en:

<http://ardupilot.org/copter/docs/common-accelerometer-calibration.html>.

[45] "Resolución No. 251/2015", [Aviacioncivil.gob.ec](http://www.aviacioncivil.gob.ec), 2015. [En línea]. Disponible en: <http://www.aviacioncivil.gob.ec/wp-content/uploads/downloads/2015/09/Resol.-251-2015-Normas-Operacion-Drones.pdf>.

[46] "MAVLink Micro Air Vehicle Communication Protocol - QGroundControl GCS", [Qgroundcontrol.org](http://qgroundcontrol.org), 2017. [En línea]. Disponible en:

<http://qgroundcontrol.org/mavlink/start>.

[47] "Introduction to SURF (Speeded-Up Robust Features) — OpenCV 3.0.0-dev documentation", [Docs.opencv.org](http://docs.opencv.org), 2017. [En línea]. Disponible en:

http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html.

[48] "OpenDroneMap/odm_data_aukerman", [GitHub](https://github.com), 2016. [En línea]. Disponible en: https://github.com/OpenDroneMap/odm_data_aukerman.

[49] "Example Datasets: senseFly SA", [Sensefly.com](https://www.sensefly.com), 2017. [En línea]. Disponible en: <https://www.sensefly.com/drones/example-datasets.html>.

ANEXOS

A. Configuración y calibración del hardware con Mission Planner

Previo a la configuración del hardware se asume que el dron ya ha sido ensamblado y se ha instalado el programa Mission Planner en el computador.

Para realizar configuraciones adicionales que no se especifican en este apartado se puede acceder a la documentación oficial de *Ardupilot* en la siguiente dirección: <http://ardupilot.org/copter/index.html>.

- **Instalación del firmware en la tarjeta controladora de vuelo Pixhawk**

Como primer paso debemos instalar el *firmware* en la tarjeta controladora de vuelo *Pixhawk*. Para establecer conexión se lo puede hacer mediante cable USB.



Figura A.1.

Lo siguiente es abrir *Mission Planner*, seleccionar el puerto COM en la esquina superior derecha de la pantalla y la tasa de Baudios con el valor de **115200**. No presionar **Connect**.

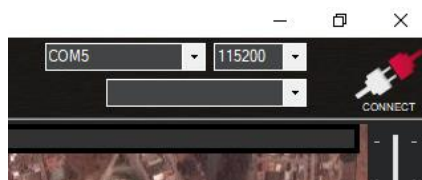


Figura A.2.

Ir a la opción **Initial Setup | Install Firmware** y seleccionar el ícono en la pantalla que se ajuste al tipo de dron que tenemos. En nuestro caso es un dron *quadcopter*, por lo tanto, seleccionamos **APM:Copter V3.5.2 Quad**. Cuando pregunte si estamos seguros, seleccionamos **Yes**.

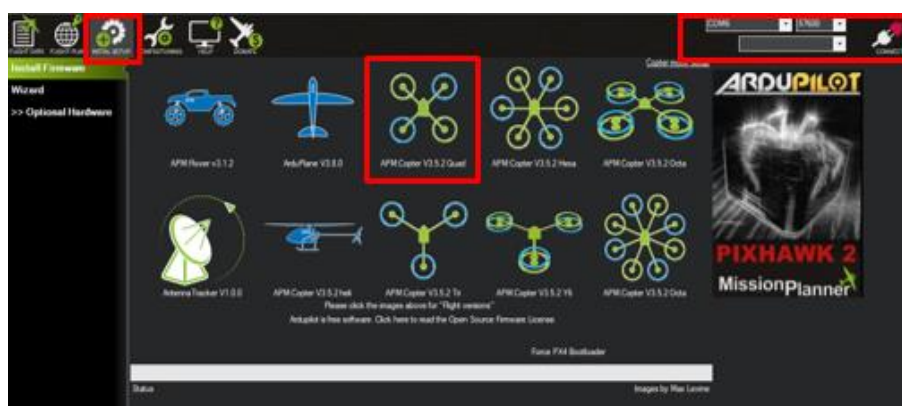


Figura A.3.

Mission Planner detectará la tarjeta, y nos pedirá desconectar la tarjeta, presionar **OK** y volver a conectarla. El programa empezará a descargar el firmware y procederá con la instalación. Para probar que se ha instalado correctamente hay que ir a la pantalla **Flight Data** y presionar el botón **Connect**.

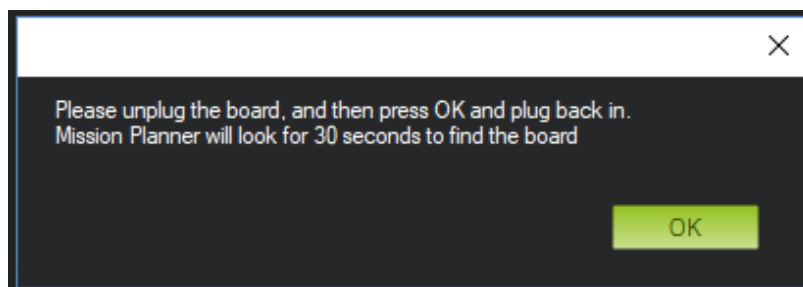


Figura A.4.

El programa iniciará la lectura de parámetros, y la interfaz deberá empezar a responder a los movimientos de inclinación de la tarjeta.

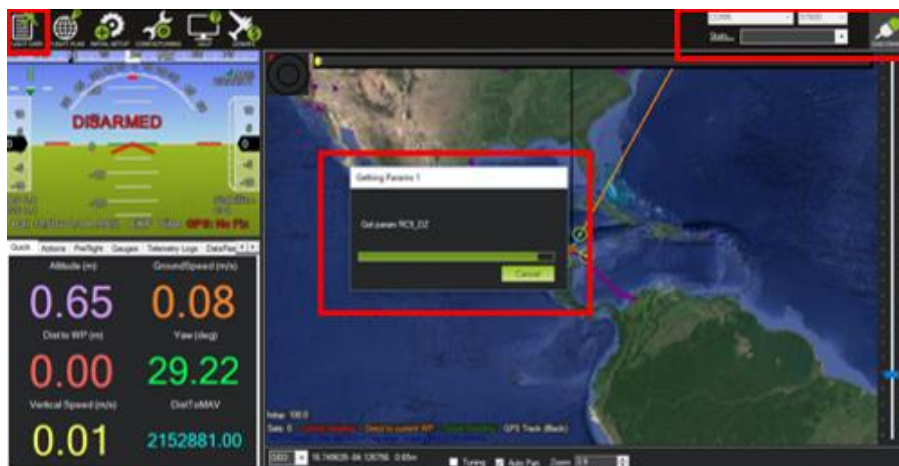


Figura A.5.

- **Selección del Frame**

Ir a la opción **Initial Setup | Mandatory Hardware | Frame Type** y seleccionar el frame **QUAD** tipo **X**.



Figura A.6.

- **Calibración de acelerómetro**

1. Ir a la opción **Initial Setup | Mandatory Hardware | Accel Calibration**.

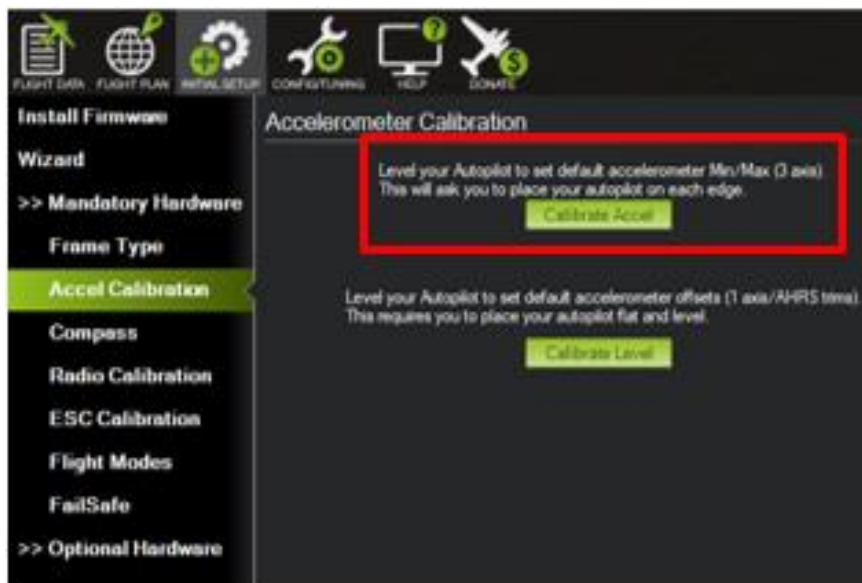


Figura A.7.

2. Presionar el botón **Calibrate Accel**, el asistente empezará a indicarnos automáticamente que coloquemos el dron en distintas posiciones sin desplazarlo.



Figura A.8.

Level: posición nivelada respecto al horizonte.

Left side: apoyar sobre el lado izquierdo.

Right side: apoyar sobre el lado derecho.

Nose down: parte delantera del dron apuntando hacia abajo.

Nose up: parte delantera del dron apuntando hacia arriba.

Back side: voltear el dron de modo que sus patas apunten hacia arriba.

Al finalizar la calibración el programa nos mostrará el mensaje “**Calibration Successful**”.

- **Calibración de compás**

Ir a la opción **Initial Setup | Mandatory Hardware | Compass**.

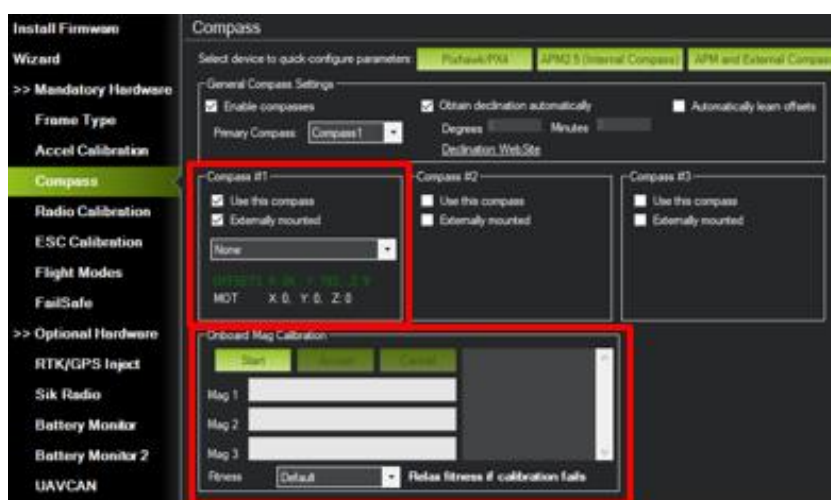


Figura A.9.

Primero debemos notar es que siempre debe estar marcado un solo un compás **Compass #1**, el cual hace referencia al compás interno del *Pixhawk*. Si se tiene más de un compás, por ejemplo, el GPS externo suele tener un compás que se conecta en el puerto I2C del *Pixhawk*, entonces se puede activar la casilla **Compass #2**.

En la sección **Onboard Mag Calibration** presionar el botón **Start** para iniciar el proceso de calibración. La calibración del compás o brújula es bastante similar a la del acelerómetro, debemos colocar al dron en distintas posiciones, la diferencia es que mientras lo hacemos, debemos mover el dron con nuestras manos como si este estuviera volando. A medida que el vehículo gira y se mueve las barras verdes

deben extenderse más y más hacia la derecha hasta que la calibración finalice.

Al finalizar con éxito se emitirán tres tonos y aparecerá una ventana con el mensaje "**Please reboot the autopilot**". Se debe desconectar y conectar el *Pixhawk* para reiniciarlo.

- **Calibración de radio control**

Encender el transmisor RC y comprobar que el transmisor esté conectado al receptor (el receptor muestra una luz verde sólida) y que está configurado para utilizar el modelo correcto para su vehículo. Por razones de seguridad debe desconectar la batería y retirar las hélices antes de realizar la calibración de la radio.

Ir a la opción **Initial Setup | Install Firmware | Radio Calibration**. Si su receptor RC (Rx) y el transmisor (Tx) están enlazados, debería ver las barras verdes moverse cuando mueva los sticks.



Figura A.10.

Para empezar, debemos dar clic en el botón **Calibrate Radio** que está en la parte inferior derecha de la ventana. El proceso consiste en mover los *sticks* del control hasta alcanzar sus valores máximos y mínimos, estos límites están marcados en pantalla con una línea de color rojo. La interfaz nos muestra hasta 14 canales para calibrar, pero nuestro

control solo soporta 6 canales: **throttle, roll, pitch, yaw, radio 5 y radio 6**.

Al finalizar, damos clic en el botón **Click When Done** para guardar los cambios y apagamos el transmisor. *Mission Planner* mostrará un resumen de los datos de calibración. Los valores normales son alrededor de 1100 para los mínimos y 1900 para los máximos.

- **Modos de vuelo**

Encender el transmisor RC e ir a la opción **Initial Setup | Mandatory Hardware | Flight Modes**.



Figura A.11.

Existen 14 modos de vuelo distintos, pero elegimos tres que van acorde a lo que necesitamos. *Mission Planner* los distribuye automáticamente de la siguiente manera: modo 1 **Stabilize**, modo 4 **Loiter**, modo 6 **Auto**.

- ✓ **Stabilize:** nos permite control del dron en modo totalmente manual, el cual solo se recomienda usar si el piloto tiene las destrezas necesarias.
- ✓ **Loiter:** este es un modo seguro de vuelo semi asistido por el piloto automático, que nos permitirá mantener la altura del dron en el aire

sin tocar los *sticks* del control, es muy útil en caso de haber problemas con otros modos de vuelo, nos permite un mejor control.

- ✓ **Auto:** es un modo útil para realizar las misiones de vuelo de mapeo de superficies en piloto automático.

- **Calibración del sensor de la batería**

Ir a la opción **Initial Setup | Optional Hardware | Battery Monitor**.

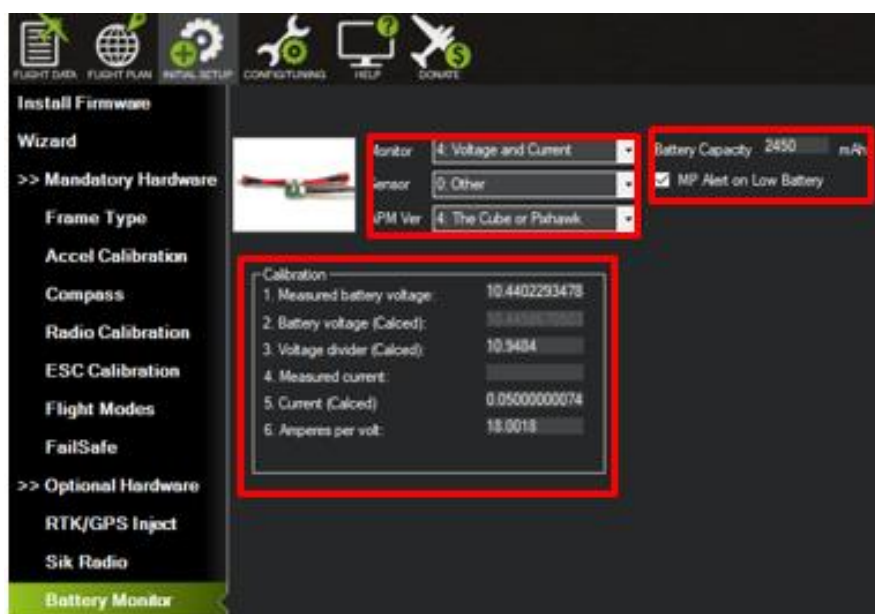


Figura A.12.

Se debe seleccionar las propiedades del módulo:

- ✓ **Monitor:** voltaje y corriente.
- ✓ **Sensor:** "Otro".
- ✓ **APM ver:** *Pixhawk*.
- ✓ **Capacidad de la batería:** Capacidad de la batería en **mAh**.

En la sección **Calibration**, en el campo **Measured battery voltage** se debe escribir el valor del voltaje medido de la batería. Luego presionar la tecla Tab o hacer clic fuera para que el valor en **Battery voltage (Calced)** y en **Voltage divider (Calced)** se actualice de manera automática.

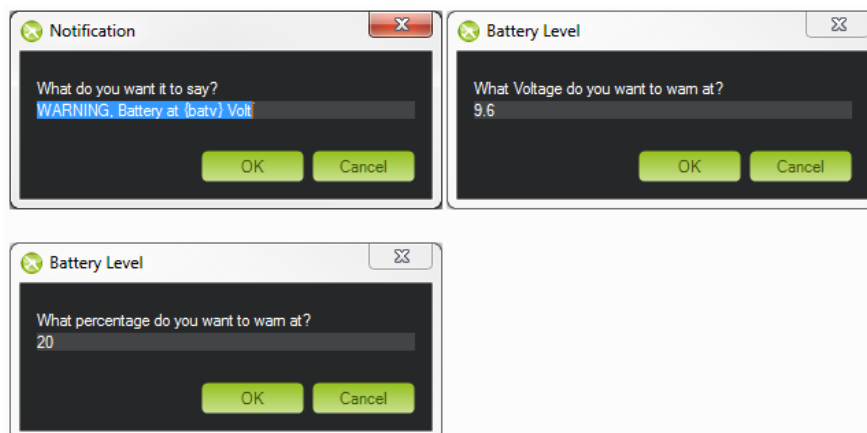


Figura A.13.

Finalmente, para configurar Mission Planner para que le avise verbalmente cuando la batería está baja (usando una voz computarizada) simplemente hay que marcar la casilla de verificación **MP Alert on Low Battery** e introducir la advertencia que desea escuchar, el nivel de voltaje y finalmente el porcentaje de corriente restante.

- **Configuración de variables para cámara fotográfica**

Para poder usar y activar la cámara debemos configurar variables en el sistema de modo que pueda ser activada por medio de los pines de salida auxiliares del *Pixhawk*.

Ir a la opción **Config Tuning | Standard Params**, con el botón **Find** buscamos el término “**camera**”.

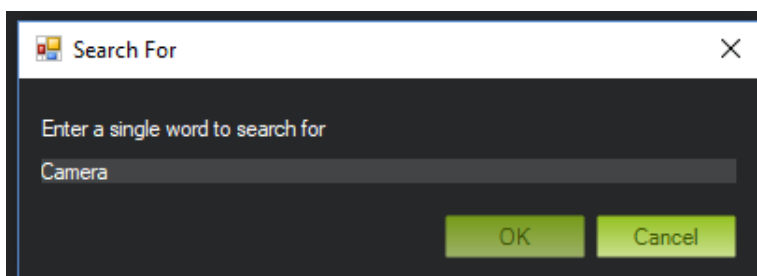
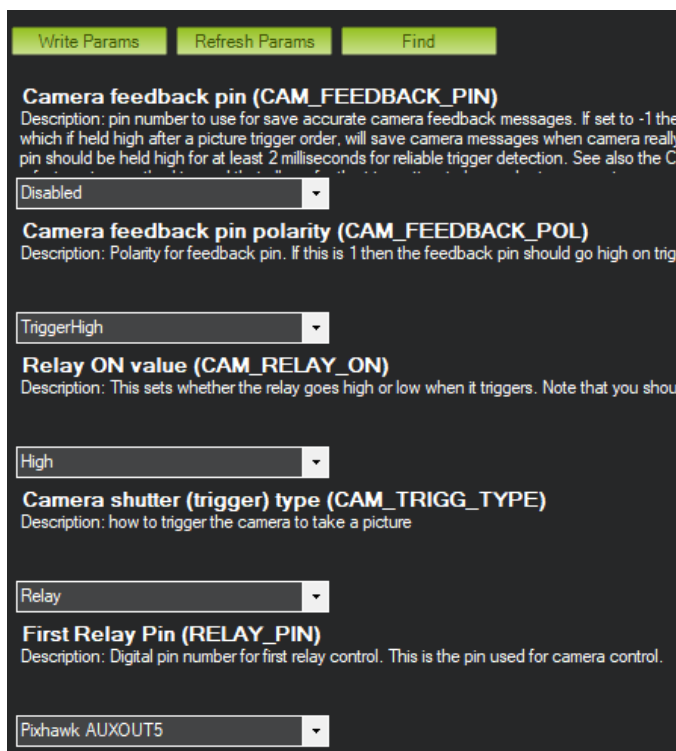


Figura A.14.

Luego debemos configurar los siguientes parámetros:

- ✓ **Camera feedback pin polarity:** TriggerHigh.
- ✓ **Relay ON Value (CAM_RELAY_ON):** High.
- ✓ **Camera shutter (trigger) type (CAM_TRIGG_TYPE):** Relay.
- ✓ **First Relay Pin (RELAY_PIN):** *Pixhawk* AUXOUT 5.



The screenshot shows a configuration interface with three buttons at the top: "Write Params", "Refresh Params", and "Find". Below the buttons are five parameter settings, each with a title, a description, and a dropdown menu:

- Camera feedback pin (CAM_FEEDBACK_PIN)**
Description: pin number to use for save accurate camera feedback messages. If set to -1 the which if held high after a picture trigger order, will save camera messages when camera really pin should be held high for at least 2 milliseconds for reliable trigger detection. See also the C
Dropdown: Disabled
- Camera feedback pin polarity (CAM_FEEDBACK_POL)**
Description: Polarity for feedback pin. If this is 1 then the feedback pin should go high on trig
Dropdown: TriggerHigh
- Relay ON value (CAM_RELAY_ON)**
Description: This sets whether the relay goes high or low when it triggers. Note that you shou
Dropdown: High
- Camera shutter (trigger) type (CAM_TRIGG_TYPE)**
Description: how to trigger the camera to take a picture
Dropdown: Relay
- First Relay Pin (RELAY_PIN)**
Description: Digital pin number for first relay control. This is the pin used for camera control.
Dropdown: Pixhawk AUXOUT5

Figura A.15.

Finalmente, en la parte superior presionamos el botón **Write Params** para guardar los cambios en el programa.

B. Planificación de rutas con Mission Planner

1. Se debe conectar el dron al computador por cable USB o de manera inalámbrica por telemetría. Esto es para que el programa detecte la posición del dron y la altura relativa del terreno sobre el cual se encuentra. Si no se realiza esto es posible que se tengan problemas para determinar la altitud relativa a la que debe volar el dron y podría estrellarse.
2. Abrir *Mission Planner*, elegir el puerto COM, la tasa de baudios (115200 USB; 57600 Telemetría) y presionar conectar en la esquina superior derecha de la pantalla.

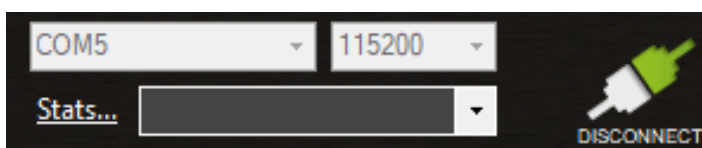


Figura B.1.

3. Ir a **Flight Plan**.
4. En esta pantalla tendremos un mapa en donde veremos la posición del dron que es dada manera automática por el GPS y un pin de color verde que se llama **Home position**, es decir la posición inicial donde el dron está siendo configurado, este pin también puede ser ubicado manualmente.

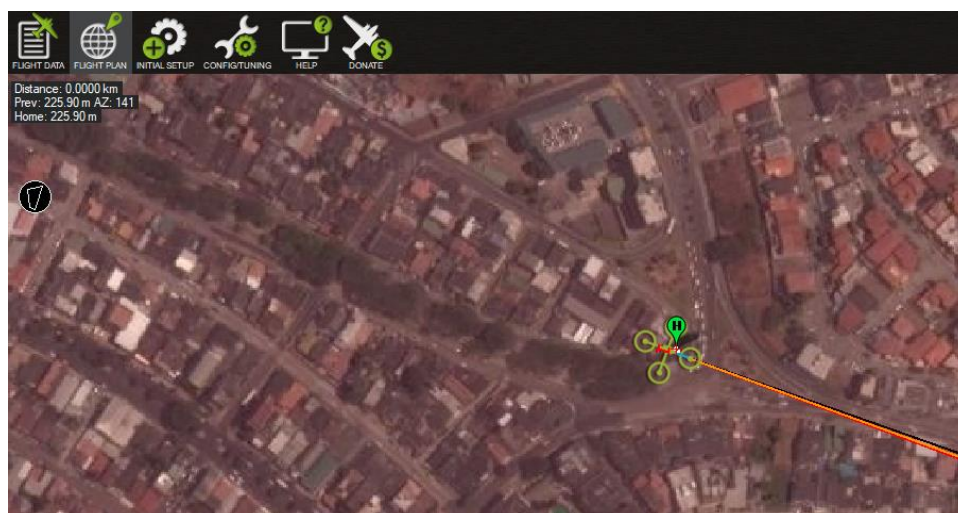


Figura B.2.

5. Lo siguiente es dar clic derecho sobre el mapa y escoger **Draw Polygon | Add Polygon Point**, y damos clic en **Ok** para aceptar.

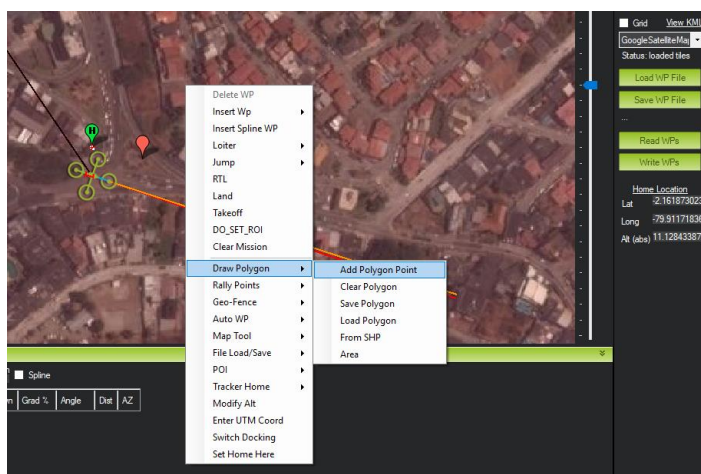


Figura B.3.

6. Entraremos en modo de dibujo de polígonos. Con esto podemos definir un área de interés sobre la cual queremos hacer el mapeo. Dando clic sobre el mapa se irán agregando pines de color rojo. Se puede dibujar cualquier forma poligonal.

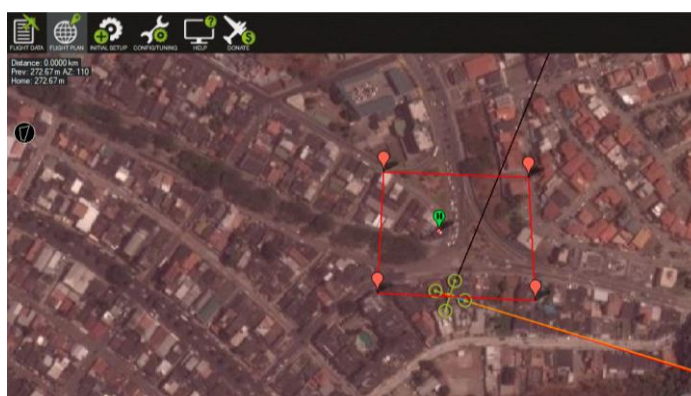


Figura B.4.

7. Una vez que hemos definido el área, lo siguiente es hacer clic derecho y escoger la opción **Auto WP | Survey (Grid)**. Esto muestra una segunda ventana **Survey (Grid)** con los datos de mapeo para misión de vuelo.

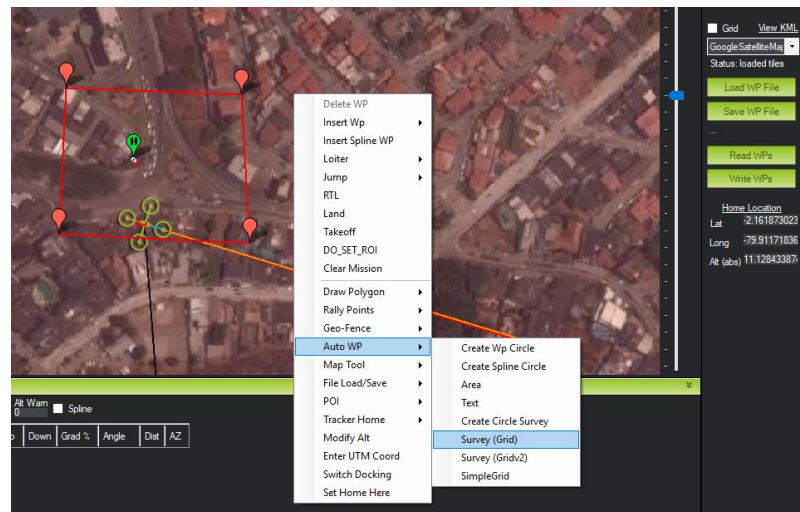


Figura B.5.

8. Lo primero es configurar la cámara, en la parte derecha escogemos la pestaña **Camera Config**. Debemos presionar el botón **Load Sample Photo**, esto es para cargar una imagen de ejemplo tomada con nuestra cámara, de ese modo el programa detectará los parámetros como distancia focal o resolución en pixeles de manera correcta. Con el botón **Save** podemos guardar nuestra configuración personalizada en el disco para que esté disponible en otras misiones. En nuestro caso lo hemos grabado como SJ4000.

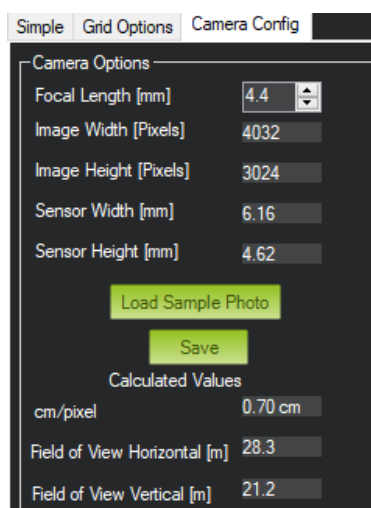


Figura B.6.

9. Dentro de la misma pestaña, en la sección **Trigger Method** debemos establecer lo siguiente:

- ✓ **CAM_TRIGG_DIST**: constante que debe activarse para que la cámara tome fotografías durante la misión de vuelo.
- ✓ **Breakup Starts**: desmarcar esta casilla.

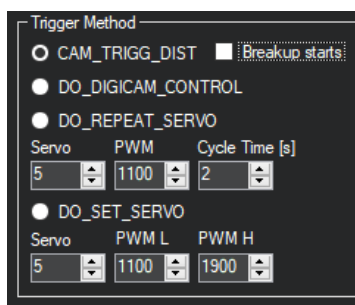


Figura B.7.

El resto de parámetros los dejamos como están por defecto.

10. Lo siguiente es ir a la pestaña Simple y verificar los siguientes parámetros:

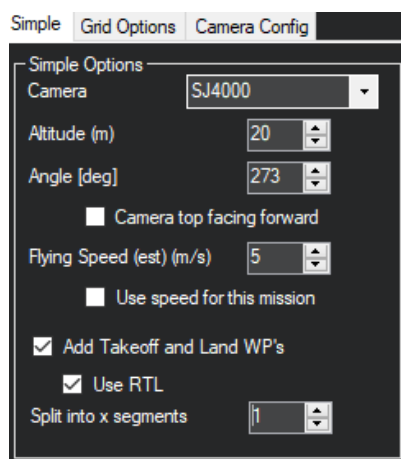


Figura B.8.

- ✓ **Altitude**: altura en metros a la que deseemos que vuele el dron.
- ✓ **Angle**: el ángulo de inclinación entre 0 a 360 grados para las líneas de color amarillo dibujadas sobre el mapa que representan la trayectoria que seguirá el dron para realizar el mapeo de la zona.

- ✓ **Flying speed:** velocidad en m/s a la que queremos que vuele el dron, para usar esta velocidad debe marcar la casilla **Use this speed for this mission**, caso contrario usara la velocidad por defecto que es 5 m/s.
- ✓ **Add Takeoff and Land Waypoints:** debe marcarse para que agregue puntos de inicio y fin de la ruta.
- ✓ **Use RTL:** Esta opción significa **Return To Land**, debe estar marcada y sirve para que en caso de emergencia por falta de batería el dron obedezca al comando RTL y realice un aterrizaje forzoso.

11. En la pestaña Grid debemos configurar lo siguiente:

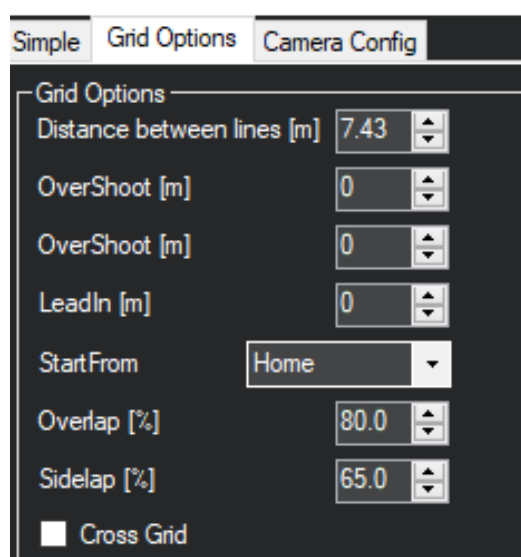


Figura B.9.

- ✓ **Start From:** el punto de partida del dron.
 - ✓ **Overlap:** determinamos el porcentaje de superposición que debe existir entre cada foto se recomienda que sea de 80% o más.
 - ✓ **Sidelap:** se determina el porcentaje de superposición lateral entre cada imagen, se recomienda que sea entre 65% a 75%.
12. Luego de configurar estos parámetros, en la parte inferior podemos ver los datos estimados para la misión de vuelo, tenemos el área de la superficie, la cantidad de fotos que serán tomadas, el tiempo de vuelo estimado, la distancia entre cada imagen, entre otras. De esta manera si vemos inconsistencias en los datos podemos reajustarlos.

Stats							
Area:	18805 m ²	Pictures:	524	Flight Time (est):	11:22 Minutes	Min Shutter Speed:	1/1428
Distance:	2.73 km	No of Strips:	17	Photo every (est):	1.13 Seconds		
Dist between images:	6 m	Footprint:	28.3 x 21.2 m	Turn Dia (at 45d):	7 m		
Ground Resolution:	0.70 cm	Dist between lines:	7.43 m	Ground Elevation:	3-17 m		

Figura B.10.

13. Para aceptar los cambios regresamos a la pestaña **Simple** y damos clic en el botón **Accept**.
14. Una vez que aceptamos los cambios se muestran en pantalla. En la parte inferior se muestra la lista de **Waypoints**, que son los puntos que forman parte de la ruta que seguirá el dron para realizar el mapeo de superficies. Estos parámetros se pueden modificar, a continuación, indicaremos para que sirve cada uno de estos valores:

Waypoints															
WP Radius	Loiter Radius	Default Alt	Relative			<input type="checkbox"/> Verify Height	Add Below	Alt Warn	<input type="checkbox"/> Spline						
2		100						0							
	Command	Delay				Lat	Long	Alt	Delete	Up	Down	Grad %	Angle	Dist	AZ
1	TAKEOFF	20	0	0	0	0	0	30	X	⬆️	⬆️	0	0	0	0
2	WAYPOINT	0	0	0	0	-2.1613418	-79.9122229	20	X	⬆️	⬆️	24.4	13.7	83.8	316
3	DO_SET_...	5.66436767578125	0	0	0	0	0	0	X	⬆️	⬆️	0	0	0	0
4	WAYPOINT	0	0	0	0	-2.1614003	-79.9109024	20	X	⬆️	⬆️	0.0	0.0	146.9	93

Figura B.12.

- ✓ **WP Radius:** especifica el tamaño en metros del radio de cada waypoint.
- ✓ **Default Alt:** es el valor de la altura en metros de cada waypoint por defecto en caso de no especificarse.
- ✓ **Verify Height:** esta casilla debe estar marcada para que el dron realice un pre-chequeo de la altura relativa del sitio en el que se encuentra. El combo box que se encuentra a un costado debe tener seleccionada la opción **Relative**.
- ✓ **Add Below:** nos permite agregar un nuevo waypoint a la ruta de manera manual.
- ✓ **Alt Warn:** valor en metros que especifica una altura a la que deseemos tener un mensaje de advertencia.

- ✓ **Command:** especifica el tipo de waypoint. Por ejemplo, si especificamos el valor **TAKE_OFF** quiere decir que es un punto de partida, **DO_SET_CAM_TRIGG_DIST** especifica que a partir de ese punto se activa la cámara para empezar a tomar fotos.
 - ✓ **Delay:** es un valor en metros.
 - ✓ **Lat:** el valor de la latitud del waypoint.
 - ✓ **Long:** el valor de la longitud del waypoint.
 - ✓ **Alt:** altura en metros del waypoint.
15. Para guardar un archivo de la misión lo podemos hacer en el lado derecho de la pantalla presionando el botón **Save WP File**. Del mismo modo para cargar alguna misión que hayamos creado con antelación lo podemos hacer con el botón **Load WP File**.

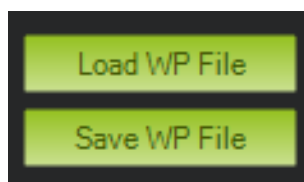


Figura B.12.

16. Para escribir la misión en la tarjeta controladora del dron lo hacemos con el botón **Write WPs**.

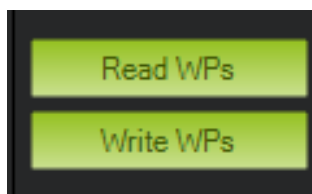


Figura B.13.

17. Lo siguiente es colocar el dron en el punto de partida que determinamos en el mapa, armar el sistema con el radio control y proceder a ejecutar la misión.

C. Geotagging de imágenes con Mission Planner

Antes de iniciar las misiones de vuelo con Misión Planner, es necesario realizar la calibración obligatoria de los componentes del dron, a continuación, mostraremos de manera breve la calibración realizada de los elementos más importantes, para tener una correcta operación del dron.

1. Abra *Mission Planner* y presione **Ctrl + F**. Esto abrirá una pantalla oculta como la que se muestra a continuación.



Figura C.1.

2. Haga clic en el botón **Geo ref images**. Le dará acceso a la herramienta Geo Tagging.

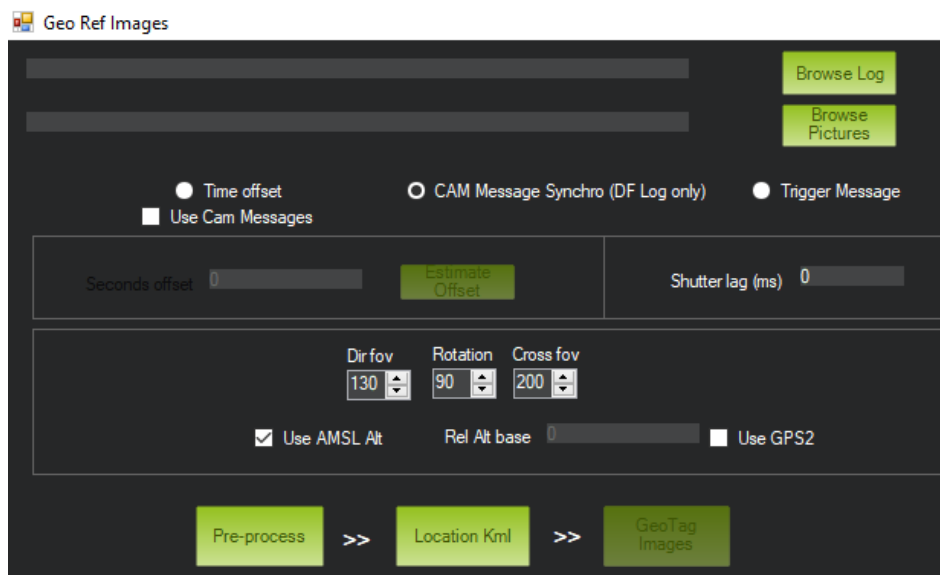


Figura C.2.

3. Haga clic en el botón **Browse log** y seleccione el registro (**.tlog** o **.log**) del vuelo relacionado con la sesión de tomas de imágenes.
4. Haga clic en el botón **Browse directory** y seleccione la carpeta donde se descargaron las imágenes aéreas de su cámara.

Tener en cuenta que la carpeta de imágenes de la cámara puede contener otros archivos aparte de las imágenes. Sin embargo, en términos de imágenes, debe contener sólo los que pertenecen al vuelo.

5. Los siguientes pasos dependen de si está utilizando una de las siguientes opciones **CAM Messages** o **Time offset**.

Para procesar con **CAM Messages**, el archivo de la misión tenía que haber sido definido para usar el disparador de cámara remota por medio de cualquiera de los comandos de misión disponibles (**CAM_TRIGG_DISTANCE**, **DO_DIGICAM_CONTROL**, etc.):

- Seleccione el método **CAM Messages** en la sección **Geotag Mode**.
- Introduzca el valor del campo **Shutter Lag** (en milisegundos) si lo conoce. Si no, dejar con 0.

- Seleccione la casilla **AMSL Alt** si desea una altitud absoluta. De lo contrario, dejar desmarcado.
- Hacer clic en **Pre-process**.

Si el preprocesamiento funciona correctamente, verá un mensaje de **Done** al final del área de mensajes de proceso.

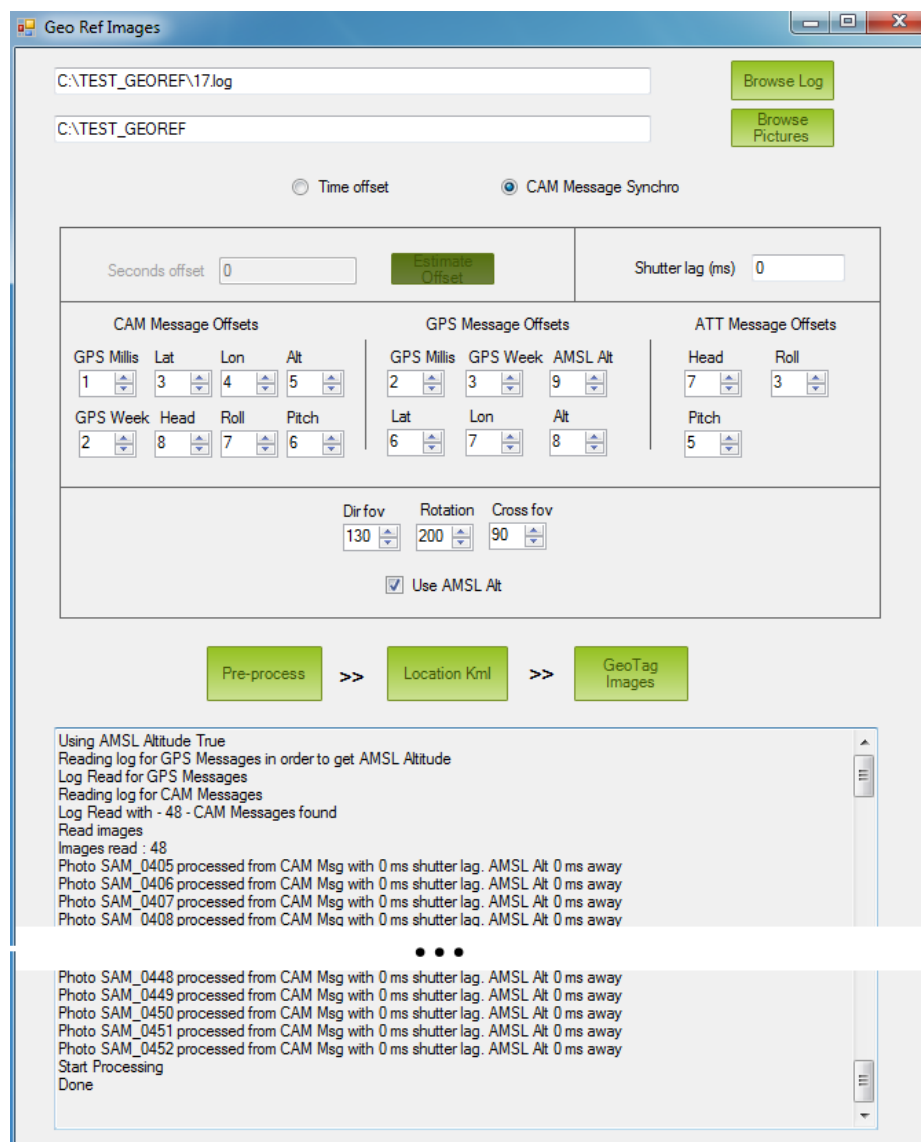


Figura C.3.

Para procesar con el método **Time Offset** hacer los siguiente:

- Haga clic en el botón de **Estimate offset**.

La herramienta calculará la diferencia de tiempo entre la primera imagen y la primera hora GPS en el log.

- Copiar el offset calculado impreso en el área de mensajes de proceso y pegarlo en el campo **Seconds offset**.
- Hacer clic en **Pre-process**.

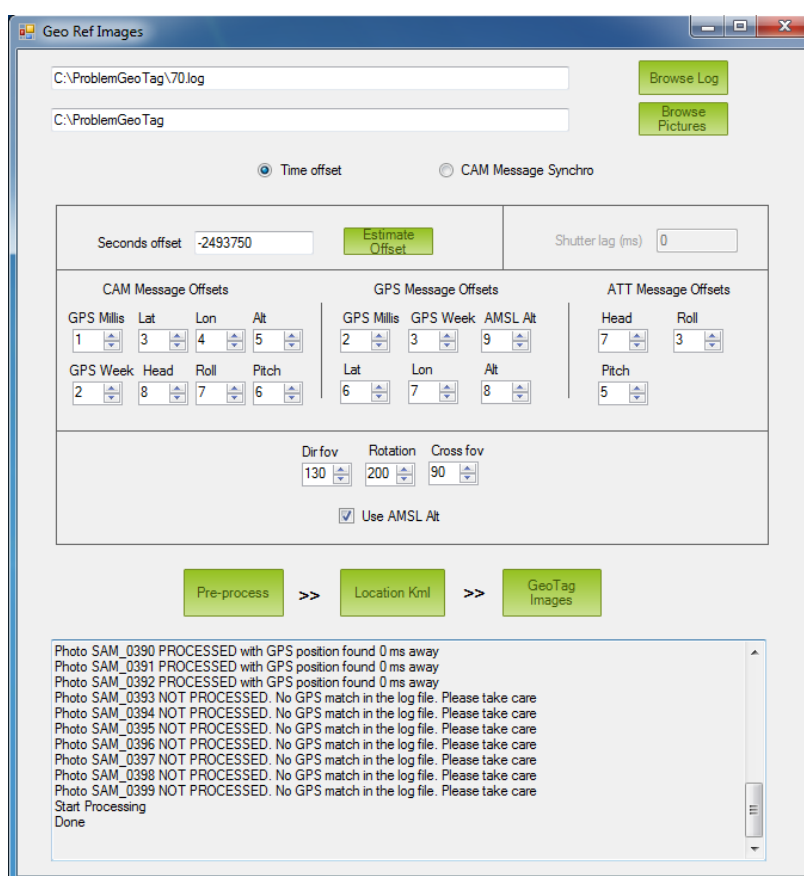


Figura C.4.

6. Los pasos restantes consisten en comprobar que las ubicaciones de imágenes propuestas son correctas y escribir la información de ubicación en los datos EXIF de la imagen:
 - Haga clic en el botón **Locate Kml**. Google Earth se abrirá y mostrará las ubicaciones de imágenes propuestas (como se muestra a continuación):

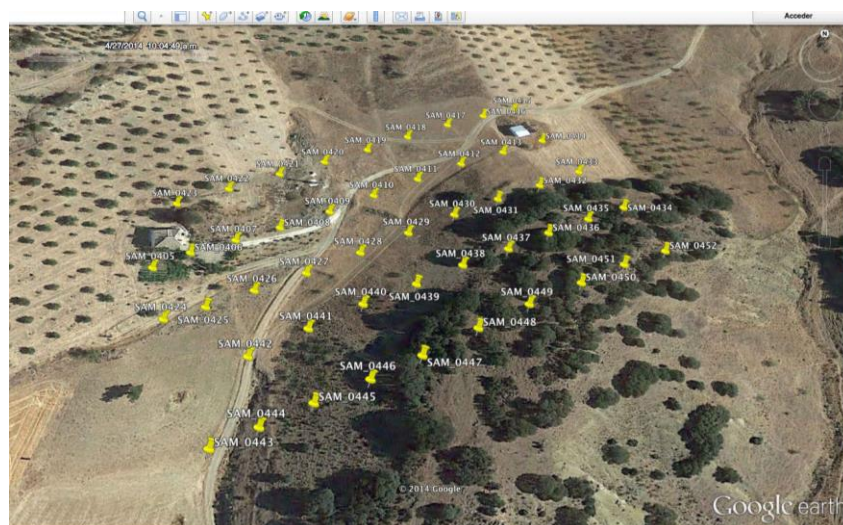


Figura C.5.

- Si se ha utilizado el modo de **CAM message** no debe haber ningún problema (si lo hay, sólo debe ordenar las imágenes en orden alfabético y comprobar que la primera imagen de la lista es también la más antigua). Si utilizó **Time Offset**, simplemente intente sumar o restar segundos a la compensación de tiempo.
- Haga clic en el botón **GeoTag Images** y espere a ver el mensaje de procesamiento **Geo Tagging FINISHED**

Los archivos georreferenciados se pueden encontrar en la carpeta **geotagged** dentro de la carpeta original de sus imágenes.