

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO AUXILIAR PARA LA CAÍDA Y LOCALIZACIÓN DE UN UAV

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES

Roberth David Vega Escobar

roberth.vega@epn.edu.ec

DIRECTOR: ING. ALEX FABRICIO OÑA ÑACATA, MSC.

alex.oña@epn.edu.ec

CODIRECTOR: ING. CARLOS ORLANDO ROMO HERRERA, MSC.

carlos.romo@epn.edu.ec

Quito, julio 2021


CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr. Vega Escobar Roberth David como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo mi supervisión:



Ing. Alex Fabricio Oña
Ñacata Msc.

DIRECTOR DEL PROYECTO



Ing. Carlos Orlando Romo Herrera
Msc.


CODIRECTOR DEL PROYECTO

DECLARACIÓN

Yo Vega Escobar Roberth David con CI: 1725193666 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



Roberth David Vega Escobar

CI: 1725193666

Teléfono: 0991948749 / 02-30401

Correo: roberth.vega@epn.edu.ec

DEDICATORIA

Mi proyecto de titulación va dedicado principalmente a nuestro Padre Celestial, por darme las fuerzas y el conocimiento necesario para culminar con una de mis metas.

A mis padres Luis y Lucia, quienes son una parte importante en mi vida, porque junto a ellos aprendí que la vida es difícil, pero en ocasiones la vida te puede gratificar cuando se trabaja con amor, fe, pasión, dedicación, persistencia, disciplina, coraje, creatividad, etc.

A mis hermanos Stalin y Esteban, ya que ellos han estado presentes y me apoyan moralmente en cada logro o fracaso de mi vida.

A todas las personas que me apoyaron y han aportado con un granito de arena para que esta meta sea un éxito.

AGRADECIMIENTO

Agradezco a Dios por brindarme sabiduría y la fortaleza necesaria para trabajar cada día en mis metas.

A mi director y tutor de tesis Ing. Alex Oña, quien, con su guía, consejos, experiencia, conocimientos, correcciones pude culminar con mi proyecto de titulación, él se ha convertido como en mi mentor y le agradezco por toda su ayuda.

A mis padres, ya que ellos me apoyaron incondicionalmente en el transcurso de toda mi vida, en mi carrera universitaria y en el desarrollo de mi tesis.

A todas las personas (familia y amigos) que me alentaron en el transcurso de mis estudios universitarios y me apoyaron de una u otra forma.

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN.....	12
1.1	Objetivo general	13
1.2	Objetivos específicos.....	13
1.3	Fundamentos.....	13
	Vehículos Aéreos No Tripulados.....	13
	Unidad de Medición Inercial.....	14
	GPS.....	15
	GSM	16
2	METODOLOGÍA.....	17
3	RESULTADOS Y DISCUSIÓN	18
3.1	Requerimientos para el diseño del prototipo	18
	Sistemas del paracaídas.....	18
	Sistemas de localización.....	20
	Requerimientos del prototipo auxiliar	22
	Prototipo auxiliar	24
3.2	Selección de los dispositivos a usar en el sistema auxiliar.....	24
	Selección del módulo con tecnología GSM y GPS.....	24
	Unidad de Medición Inercial.....	27
	Actuador	28
	Módulo de radio frecuencia.....	29
	Módulo de <i>bluetooth</i>	29
	<i>Buzzer</i>	30
	Microcontrolador	30
	Fuente de alimentación.....	31
3.3	Implementación del prototipo en base a los dispositivos seleccionados	34
	Diseño de los diagramas del prototipo	34
	Implentación de la placa electrónica	35

Diseño del paracaídas del prototipo auxiliar.....	38
Diseño de la caja protectora del prototipo	42
3.4 Diseño e implementación del algoritmo de control del sistema	45
Diseño de la interfaz de usuario.....	48
3.5 Pruebas y Análisis de Resultados.....	51
Pruebas del módulo SIM808 para la localización GPS	51
Pruebas de la alimentación eléctrica del sistema (PCB)	53
Pruebas de los enlaces de comunicación	54
Pruebas con el sistema del paracaídas	58
Pruebas del prototipo con el UAV	61
3.6 Manual de Uso y Mantenimiento.....	67
4 CONCLUSIONES Y RECOMENDACIONES	68
4.1 Conclusiones.....	68
4.2 Recomendaciones.....	69
5 REFERENCIAS BIBLIOGRÁFICAS.....	71
ANEXOS.....	77
Anexo 1: Certificado de Funcionamiento.....	77
Anexo 2: Costo del Proyecto.....	78
Anexo 3: Planos de la Caja del Bloque 1	80
Anexo 4: Planos de la Caja del Bloque 2	83
Anexo 5: Código de Programación del Sistema	86
Anexo 6: Código de Programación de la Aplicación Móvil	98
Anexo 7: Código QR para Acceder a la Aplicación Móvil	120

ÍNDICE DE FIGURAS

Figura 1.1 (A) UAV de ala rotatoria (B) UAV de ala fija	13
Figura 1.2 (A) Colisión del <i>DJI Phantom</i> (B) Colisión del UAV <i>Ghost</i>	14
Figura 1.3 Ángulos de <i>Euler</i> de un UAV	15
Figura 1.4 Segmentos del GPS.....	16
Figura 3.1 Sistema ubicado en el UAV modelo M600	18
Figura 3.2 Sistema <i>SafeAir</i> para un UAV modelo DJI M200.....	19
Figura 3.3 Dron <i>DJI Mavic 2</i> con el sistema de emergencia denominado <i>SafeAir</i>	19
Figura 3.4 Modelos de sistemas de paracaídas para los drones.....	20
Figura 3.5 Módulo RTK de la empresa DJI	20
Figura 3.6. Equipo D-RTK GNSS de la empresa DJI	21
Figura 3.7 Sistema D-RTK2	21
Figura 3.8 Diagrama de bloques del sistema propuesto.....	23
Figura 3.9 Diagrama de bloques del prototipo auxiliar.....	24
Figura 3.10 Módulo SIM808 v2 2.9.1 con tecnología GSM y GPS	27
Figura 3.11 Módulo IMU modelo GY-521.....	27
Figura 3.12 Micro servomotor modelo MG 90 S.....	28
Figura 3.13 Módulo relé	28
Figura 3.14 Módulos RF modelo mRo 915 (MHz)	29
Figura 3.15 Módulo de <i>Bluetooth</i> modelo HC-05.....	29
Figura 3.16 <i>Buzzer</i>	30
Figura 3.17 Microprocesador Atmega 328P	31
Figura 3.18 Batería Li-Po	33
Figura 3.19 Conexión en serie de las baterías Li-Po de 3.7 (V)	33
Figura 3.20 Batería alcalina de 9 (V).....	34
Figura 3.21 Diagrama de bloques de la alimentación eléctrica del bloque 1 y 2	35
Figura 3.22 Diagrama de bloques del suministro de energía del boque 3	35
Figura 3.23 Componentes <i>SMD</i>	36
Figura 3.24 Diagrama circuital del bloque 1	36
Figura 3.25 Diseño circuital del bloque 3.....	37
Figura 3.26. (A)Vista superior (B)Vista inferior de la placa electrónica del bloque1	37
Figura 3.27 (A) Vista superior (B) Vista inferior de la placa electrónica del bloque 3 ..	38
Figura 3.28 Descenso de un modelo aéreo utilizando un paracaídas	38
Figura 3.29 Dimensiones del cuadricóptero	40

Figura 3.30	Paracaídas del proyecto	41
Figura 3.31	Paracaídas.....	42
Figura 3.32	Diseño de la caja del bloque 2	42
Figura 3.33	Diseño de la caja protectora del bloque 1.....	44
Figura 3.34	Caja prefabricada para el bloque 3 del proyecto	45
Figura 3.35	Diagrama de flujo del sistema principal	47
Figura 3.36	Interfaz principal de la aplicación del proyecto)	48
Figura 3.37	Opciones de la interfaz secundaria de la aplicación	49
Figura 3.38	Diagrama de flujo de la funcionalidad de la aplicación móvil	50
Figura 3.39	Pruebas del GPS del módulo SIM808	51
Figura 3.40	Recepción de la trama de la navegación por satélite	52
Figura 3.41	(A) Coordenadas GPS en <i>Google Maps</i> (B) Distancia entre los puntos ..	52
Figura 3.42	Implementación del proyecto en una placa de pruebas.....	53
Figura 3.43	Medición del voltaje del módulo SIM808	54
Figura 3.44	Equipos del proyecto.....	55
Figura 3.45	Pruebas de comunicación RF del proyecto	55
Figura 3.46	Pruebas de comunicación del enlace GSM del proyecto.....	56
Figura 3.47	Mensajes de texto recibidos en el dispositivo móvil.....	57
Figura 3.48	(A) Localización en <i>Google Maps</i> (B) Localización en el mapa <i>OsmAnd</i> ..	57
Figura 3.49	Sistema ubicado en el cuadricoptero.....	58
Figura 3.50	Expulsión de forma manual el paracaídas.....	59
Figura 3.51	Expulsión de forma automática del paracaídas	59
Figura 3.52	Lectura del ángulo de orientación de la aeronave	60
Figura 3.53	Código QR de las pruebas con el sistema del paracaídas	60
Figura 3.54	(A) Dispositivos utilizados en las pruebas (B) Área de la prueba.....	61
Figura 3.55	Distancia recorrida para la prueba de comunicación con línea de vista ..	62
Figura 3.56	(A) Datos solicitados por el enlace RF (B) Led indicador del enlace RF ..	63
Figura 3.57	(A) Ubicación del sistema en <i>Google Maps</i> (B) Ubicación en <i>OsmAnd</i>) ..	64
Figura 3.58	(A) Datos solicitado por el enlace GSM (B) Led indicador del enlace GSM	65
Figura 3.59	Prueba en campo del paracaídas (Fuente propia).....	65
Figura 3.60	(A) Elevación del UAV (B) UAV previo a la expulsión del paracaídas	66
Figura 3.61	UAV descendiendo con el paracaídas.....	66
Figura 3.62	Código QR de las pruebas en campo del sistema montado en el UAV ...	67
Figura 3.63	Código QR para el manual de uso del proyecto	67
Figura 3.64	Código QR del manual de mantenimiento del proyecto.....	68

ÍNDICE DE TABLAS

Tabla 3.1	Especificaciones técnicas de los módulos con tecnología GSM y GPS	25
Tabla 3.2	Criterios y ponderación para la selección del módulo GSM y GPS	26
Tabla 3.3	Selección del módulo con tecnología GSM y GPS.....	26
Tabla 3.4	Requerimientos al microprocesador de los módulos.....	30
Tabla 3.5	Consumo de energía de cada módulo	32
Tabla 3.6	Consumo de energía de los módulos del bloque 3	33
Tabla 3.7	El coeficiente de arrastre de acuerdo con la forma del objeto	39
Tabla 3.8	Descripción de cada pieza de la caja de bloque 2	43
Tabla 3.9	Descripción de las piezas que conforman la caja protectora.....	44
Tabla 3.10	Voltajes experimentales y teóricos de los módulos del proyecto	54

RESUMEN

El presente proyecto tuvo como propósito la implementación de un prototipo auxiliar que ayude a reducir los daños por la caída de un Vehículo Aéreo No Tripulado (VANT) del inglés *Unmanned Aerial Vehicle (UAV)*, así mismo, permite la geolocalización del mismo en caso de que aterrice o caiga en un lugar de difícil acceso, el proyecto está previsto para los UAV's de ala rotatoria como por ejemplo un cuadricóptero.

En la primera sección se dispone de la introducción, del objetivo general, los objetivos específicos y del apartado fundamentos, en el cual se describe de forma general los temas fundamentales que fueron necesarios para el desarrollo y ejecución del proyecto.

En la segunda sección se explica el tipo de metodología que se utilizó en el proceso de investigación, diseño e implementación del prototipo auxiliar.

En la tercera sección se encuentra el desarrollo y la implementación de cada etapa que conforma el sistema, las diferentes pruebas y los resultados obtenidos.

En la cuarta y quinta sección se redactan las conclusiones y recomendaciones que se obtuvieron con el desarrollo y ejecución del prototipo auxiliar, además, se encuentra el apartado de las referencias bibliográficas las cuales fueron necesarias para el proceso de investigación

Finalmente, en la última sección se encuentra el apartado de los anexos donde se exponen los diferentes diagramas y códigos de programación.

PALABRAS CLAVE: UAV, geolocalización, cuadricóptero, paracaídas.

ABSTRACT

The document purpose is the implementation of an auxiliary prototype that helps to reduce the damage caused by the fall of an Unmanned Aerial Vehicle (UAV), likewise, it's the geolocation of the UAV when it descends in an unknown or difficult to access place, the project is intended for a UAV of rotary wing such as a quadcopter.

The first section contains the introduction, the general objective, the specific objectives and the fundamentals, this section describes the fundamental themes for the development and execution of the project.

The second section briefly explains the type of methodology used in research process design and implementation of auxiliary prototype.

The third section contains the development and implementation of each stage of system the different tests and results obtained.

The fourth and fifth sections contains the conclusions and recommendations obtained from the implementation of auxiliary prototype and the section of the references bibliographic which were necessary for the process of research of the project.

Finally, in the last section are the annexes, where I exposed the different diagrams, codes of programming and the plans of the boxes of the proyect.

KEYWORDS: UAV, geolocation, quadcopter, parachute.

1 INTRODUCCIÓN

En el Ecuador en la actualidad se están utilizando con más frecuencia los UAV`s para diferentes usos, como: la seguridad y vigilancia, la fotografía, los reportajes, la cartografía de zonas, la fumigación de cultivos, la ubicación y rescate de personas, entre otras aplicaciones.

La mayoría de los drones están equipados con cargas de pago (cámaras, sensores, etc.) que tienen un precio elevado, los cuales al recibir un golpe contundente producido por una caída pueden dañarse por completo, para solucionar el problema las empresas han fabricado diferentes equipos auxiliares que tienen el propósito de expulsar un paracaídas cuando el dron se encuentra con dificultades técnicas y al piloto se le imposibilita realizar un descenso controlado, esto permite que la caída no sea brusca.

Las empresas que diseñan y fabrican los UAV`s disponen de equipos de emergencia que contienen varios sensores e incluso un paracaídas, estos tienen un costo alrededor de los cientos o miles de dólares debido a que utilizan microcomponentes electrónicos y los materiales de la caja protectora son resistentes e impermeables para evitar filtraciones de alguna sustancia líquida. Por tal razón, se ha considerado proponer un prototipo de paracaídas de bajo costo que ayude en casos de emergencia como el expuesto.

Otro problema que dispone un dron comercial básico es cuando desciende en lugares que no tiene una amplia visibilidad de la zona, es posible que se extravíe la aeronave, por lo tanto, se consideró la implementación de un prototipo auxiliar que permita la geolocalización del vehículo aéreo.

El prototipo auxiliar que se ha implementado permite la geolocalización del sistema con base en el Sistema de Posicionamiento Global del inglés *Global Positioning System (GPS)*, otra función que se agregó para localizar el prototipo en lugares con poca visibilidad fue el encendido de una alarma auditiva tipo *beep*; también se incorporó un mecanismo que tiene la capacidad de desplegar un paracaídas para amortiguar la caída de la aeronave, para la activación automática del mismo se utilizó un sensor de orientación (IMU). Además, se creó una aplicación en el programa *Android Studio* para el control manual de las funciones del prototipo auxiliar y la presentación de los datos, el medio de comunicación se consideró redundante para que el usuario realice el envío de datos a través de módulos de radio frecuencia y por la red celular, por último, la caja

protectora del proyecto fue construida con filamento PLA procurando que sea ligera y con un cierto grado de hermeticidad.

El prototipo que se implementó fue sometido a varias pruebas de funcionamiento, cuyo certificado de funcionamiento se muestra en el Anexo 1.

1.1 Objetivo general

Diseñar un prototipo auxiliar para el accionamiento y la localización de un UAV utilizando una comunicación inalámbrica redundante.

1.2 Objetivos específicos

- Revisar la información para determinar los requerimientos para el diseño del prototipo.
- Seleccionar los dispositivos a usar en el sistema auxiliar.
- Implementar el prototipo en base a los dispositivos seleccionados.
- Diseñar e implementar el algoritmo de control del sistema.
- Probar el prototipo construido en un UAV básico de ala rotatoria.

1.3 Fundamentos

Vehículos Aéreos No Tripulados

Los vehículos aéreos no tripulados son aeronaves sin tripulación controlados por un piloto a través de un mando a distancia. Existen dos tipos básicos de UAV de ala fija o de ala rotatoria los cuales se pueden observar en la **Figura 1.1**. Los UAV de ala fija se caracterizan por realizar vuelos a velocidades, alturas superiores para recolección de datos cartográficos y el despliegue de la mayoría es de forma horizontal, en cambio, los UAV de ala rotatoria pueden ser tricópteros, cuadricópteros, hexacóptero y octocópteros de acuerdo con el número de motores, son requeridos en vuelos de baja velocidad, para realizar maniobras en forma estacionaria y el despliegue es de forma vertical [1].

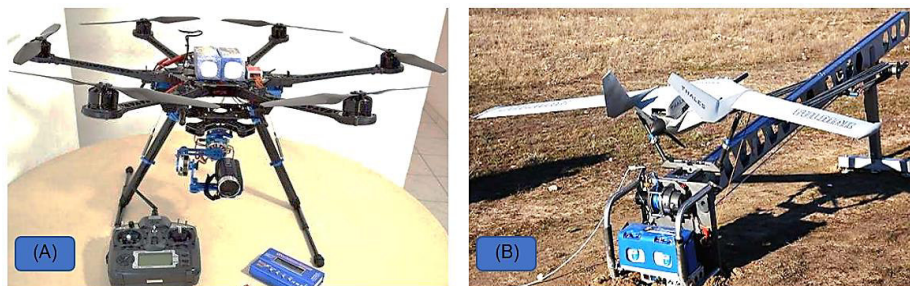


Figura 1.1 (A) UAV de ala rotatoria (B) UAV de ala fija [2]

Al principio los drones fueron diseñados exclusivamente de uso militar, para brindar apoyo a tropas, ejecutar misiones de ataque o de reconocimiento; sin embargo, en la actualidad los UAV se encuentran en el ámbito civil desempeñando diferentes aplicaciones, las más frecuentes son: imagen y video, monitoreo y vigilancia, inspección de infraestructuras, búsqueda y rescate, mapeo de terrenos, riego de cultivos, entre otros. En el Ecuador está la empresa *Latitude Aerospace Solutions* (LAS) la cual se dedica a la construcción, adaptación y venta de drones, en un artículo se menciona que el precio de un UAV varía entre cientos y miles de dólares de acuerdo con la aplicación, el tipo de UAV, el tiempo y la distancia de vuelo, las cargas de pago (cámaras, sensores, actuadores, etc.), entre otras cosas que requiera el usuario [1] [3] [4].

Un UAV utiliza diferentes dispositivos para realizar vuelos de forma manual o autónoma, estos equipos son propensos a daños, fallas o errores debido a factores humanos por parte del usuario, así como factores internos y/o externos de la aeronave, por consiguiente, puede ocasionar la pérdida de forma parcial o total el control del UAV durante el vuelo, esto ocasiona una colisión donde puede ser afectado el equipo en conjunto así como la integridad de terceras personas, ver **Figura 1.2**.



Figura 1.2 (A) Colisión del *DJI Phantom* (B) Colisión del UAV *Ghost* [5] [6]

Además, otra consecuencia que puede producirse cuando algún dispositivo del UAV falla durante el vuelo, es que descienda en un lugar con poca visibilidad por ejemplo un bosque, es posible que se pierda de vista y comunicación con la aeronave, siendo necesario sistemas auxiliares que permitan lograr una localización aún si el equipo está desconectado por la caída.

Unidad de Medición Inercial

Una de las funciones del sistema del proyecto es la activación de forma automática del mecanismo de apertura del paracaídas, por lo que se ha considerado la unidad de medición inercial del inglés *Inertial Measurement Unit* (IMU) para medir la orientación

del UAV y en el caso de sobrepasar un determinado ángulo de inclinación (umbral) se expulsa el paracaídas.

La IMU es un dispositivo electrónico conformado por acelerómetros y giroscopios los cuales pueden medir la velocidad angular, así como, la fuerza de la aceleración lineal en relación con la gravedad, con base en varios algoritmos matemáticos y los microsensores es posible determinar la orientación y los ángulos alrededor de los ejes (x, y, z), estos también son expresados como los ángulos de *Euler* denominados *roll*, *pitch* y *yaw* (ver **Figura 1.3**) [7].

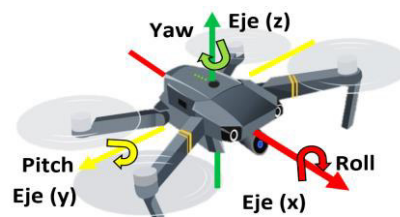


Figura 1.3 Ángulos de *Euler* de un UAV (Fuente propia)

El sensor de movimiento IMU es muy usado en sistemas de navegaci3n en veh3culos a3reos y espaciales, rob3tica, accesorios automotrices, microcirug3a, entre otros [8]. Un sensor IMU tiene diferentes caracter3sticas para su selecci3n, en este proyecto se ha considerado: factibilidad de determinaci3n de los 3ngulos de *Euler*, bajo consumo de energ3a, tama1o reducido, peso ligero y bajo costo.

GPS

Otra de las funciones del prototipo auxiliar, es proporcionar al usuario la ubicaci3n geogr3fica del sistema, por lo tanto, se ha considerado un dispositivo GPS de bajo consumo el3ctrico.

El GPS es un sistema que permite la navegaci3n por sat3lite desarrollado por el Departamento de Defensa de los Estados Unidos, el sistema inici3 como un proyecto exclusivamente de uso militar, sin embargo, actualmente est3 a la disposici3n de uso civil. El GPS funciona a trav3s de tres segmentos los cuales son: control, espacial y usuario como se puede visualizar en la **Figura 1.4** [9].

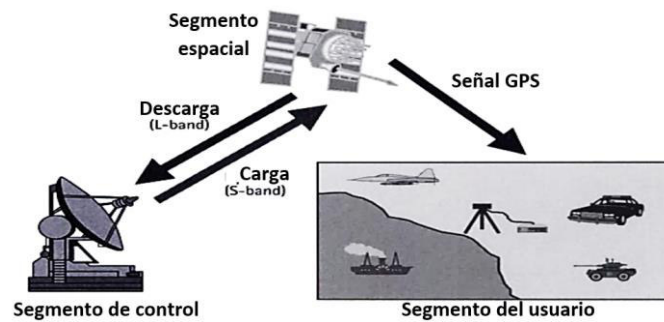


Figura 1.4 Segmentos del GPS [9]

El segmento de control está constituido por una red de instalaciones terrestres que permite monitorear a cada satélite que se encuentra alrededor del mundo, el segmento espacial conformado por al menos 24 satélites que están en la órbita MEO y el segmento usuario constituye todos los usuarios civiles y militares que disponen de una antena GPS receptora para determinar la posición geográfica. El sistema de GPS tiene múltiples aplicaciones como en *robots* móviles, navegación de vehículos, ubicación de equipos o personas, entre otros [9] [10] [11].

GSM

En el proyecto se ha considerado el uso de la red celular como método de comunicación inalámbrica entre el usuario y el prototipo auxiliar, esto gracias a su amplia cobertura geográfica en comparación a otros medios de comunicación.

El GSM es una tecnología digital creada para la telefonía celular que permite establecer una comunicación entre los dispositivos móviles, esta tecnología está conformada por una serie de antenas que cubren una zona determinada como "celda", los mismos que se encuentran en una estación base que se conectan a un centro de conmutación móvil que administra los servicios (mensajes de texto, llamadas, conexión a internet) del usuario [12] [13] [14].

2 METODOLOGÍA

La investigación aplicada se orienta a la resolución de problemas de la vida cotidiana, buscando la consolidación del conocimiento para su aplicación y otorga un aporte al desarrollo cultural y científico con una innovación tecnológica [15].

En el presente proyecto se ha utilizado la investigación aplicada porque se realizó el diseño e implementación de un prototipo aplicando conceptos revisados en la carrera, procurando solucionar un problema en la operación de los UAV de ala rotatoria.

El prototipo auxiliar implementado dispone de varias funciones, por ejemplo: cuando ocurre una avería o daño en la aeronave durante el vuelo y el mismo empieza a caer, una de las funciones es la activación (manual/automática) de la apertura de un paracaídas con la finalidad de amortiguar la caída del UAV, esta función tiene el propósito de disminuir la probabilidad de la destrucción de la aeronave por la caída y reducir el riesgo de un posible accidente a terceras personas o dañar propiedad privada.

En caso de que la aeronave caiga en un sitio con poca visualización se consideró en el prototipo la geolocalización del sistema utilizando el módulo SIM808 con tecnología GSM y GPS; también se ha utilizado una alarma de tipo *beep* como guía auditiva. Para el enlace entre el proyecto y el usuario se han considerado dos dispositivos de comunicación con diferentes métodos de conexión, por una parte se dispone del servicio de mensajes de texto de la telefonía celular que dispone el módulo SIM808 y el otro medio de transmisión de datos a través del dispositivo de radio frecuencia modelo "mRo 915 (MHz)", así pues, en el momento de que el usuario ha perdido la conexión con uno de los dispositivos de comunicación tiene la oportunidad de enlazarse con el otro equipo y disminuir la probabilidad de pérdida del control del prototipo.

Uno de los propósitos del proyecto que se ha procurado lograr, es que el prototipo implementado sea modular y adaptable a cualquier UAV de ala rotatoria, por consiguiente, los dispositivos utilizados son de dimensiones pequeñas con el fin que el sistema no sea demasiado grande para un UAV.

3 RESULTADOS Y DISCUSIÓN

3.1 Requerimientos para el diseño del prototipo

A continuación, se presenta información de diferentes sistemas comerciales de paracaídas y equipos con GPS disponibles para un UAV, estos son diseñados y fabricados por diferentes empresas a nivel mundial.

Sistemas del paracaídas

La empresa DJI que se dedica al diseño y fabricación de un UAV ofrece un sistema de paracaídas auxiliar para drones de ala rotatoria, los mismos que funcionan de forma independiente de la aeronave, el sistema utiliza sensores de movimiento que permite detectar variaciones de inclinación irregulares del vehículo aéreo las cuales no fueron planificadas por el usuario. El sistema inicia una serie de procesos previos al despliegue del paracaídas previniendo la destrucción del UAV, el sistema tiene un peso aproximado de 620 (g), el paracaídas se encuentra en un contenedor de forma tubular de materiales resistentes como se puede visualizar en la **Figura 3.1** [16].

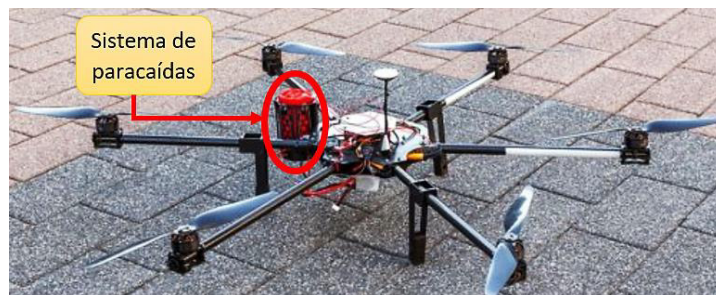


Figura 3.1 Sistema ubicado en el UAV modelo M600 [16]

La empresa *ParaZero* dedicada a la fabricación, personalización y distribución del UAV disponen del sistema *SafeAir*, el cual utiliza varios sensores independientes para monitorear los patrones de vuelo; además, identifica de inmediato las fallas de la aeronave, lo que desencadena una serie de medidas de seguridad antes de desplegar el paracaídas, esto con el fin de precautelar el bienestar de la aeronave y de terceras personas, el sistema es diseñado para drones de ala rotatoria como se puede visualizar en la **Figura 3.2** [17].



Figura 3.2 Sistema *SafeAir* para un UAV modelo DJI M200 [17]

Para los drones que desempeñan la función de vuelo de exhibición y fotografía como el modelo *DJI Mavic*, la empresa *ParaZero* dispone de un reducido sistema de paracaídas como se puede visualizar en la **Figura 3.3**, esta nueva versión puede identificar de manera autónoma los riesgos durante el vuelo y desplegar el paracaídas, pero el costo de este sistema se encuentra alrededor de cientos de dólares [18] [19].



Figura 3.3 Dron *DJI Mavic 2* con el sistema de emergencia denominado *SafeAir* [18]

La empresa *Skycat* dedicada al diseño y distribución de sistemas de paracaídas para drones, disponen de una variedad de sistemas de acuerdo con las especificaciones y la función que cumple el UAV, las características que tiene en consideración es el peso, las dimensiones, entre otras características de la aeronave como se visualiza en la **Figura 3.4** [20].

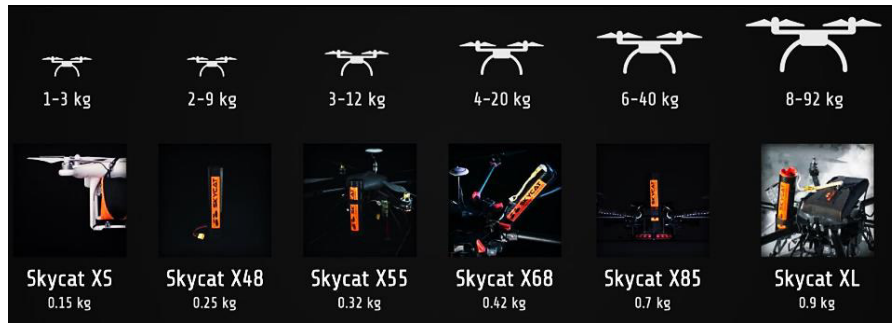


Figura 3.4 Modelos de sistemas de paracaídas para los drones [20]

De acuerdo con la información revisada anteriormente, se puede destacar varios puntos importantes, por ejemplo estos sistemas disponen de uno o varios sensores que permiten monitorear los movimientos del UAV y determinar que la aeronave tiene problemas durante el vuelo al punto de una inevitable caída, para lo cual estos sistemas inician un proceso programado para el despliegue del paracaídas. Otro punto referente son las dimensiones del paracaídas debido a que existen diferentes UAV's con un peso específico y en ocasiones al mismo se le añade cargas de pago que incrementan el peso total de la aeronave, por consiguiente, estos aspectos han sido considerados para el diseño del prototipo.

Sistemas de localización

En cuanto a los sistemas auxiliares para localización de un UAV disponibles en el mercado, se tienen varias opciones como se indican a continuación.

La empresa *DJI* dispone de varios equipos y accesorios para la localización y navegación de los UAV's, por ejemplo se dispone del módulo denominado RTK (ver **Figura 3.5**) para el posicionamiento del UAV, el módulo proporciona al usuario la ubicación de la aeronave con una alta precisión a nivel de los centímetros y el equipo es de tamaño reducido para evitar problemas de sobrepeso en la aeronave [21].



Figura 3.5 Módulo RTK de la empresa DJI [21]

El equipo D-RTK GNSS (ver **Figura 3.6**) es un sistema de navegación y posicionamiento de alta precisión, el sistema proporciona al UAV un posicionamiento 3D preciso a nivel

de los centímetros la cual es esencial para aplicaciones que requieran precisión durante los vuelos programados, el equipo dispone de varias características entre las más relevantes son: el peso aproximado de 139.5 (g), los interfaces de comunicación (UART,CAN y USB), las medidas son de 112.3 x 63 x 18.6 (mm), etc [22].



Figura 3.6. Equipo D-RTK GNSS de la empresa DJI [22]

También se tiene el sistema D-RTK 2 (ver **Figura 3.7**), el cual es un receptor móvil con sistema global de navegación por satélite (GNSS) de alta precisión, el equipo proporciona los datos en tiempo real para que los drones logren una alta precisión de posicionamiento en vuelos programados, el tipo de comunicación es a través de la tecnología GSM, Wi-Fi y LAN la cual garantiza una transmisión ininterrumpida en cualquier escenario, el sistema es compatible con cuadricopteros modelos DJI, *Phantom 4*, *Matrice 300*, entre otros [23].



Figura 3.7 Sistema D-RTK2 [23]

En base a los equipos comerciales mencionados, se puede destacar que estos sistemas son fabricados de manera compacta, resistentes y ligeros para evitar sobrecargar de peso al UAV, algunos de los sistemas revisados disponen de varios medios de comunicación con el usuario para asegurar la transmisión de datos, así pues, evitar pérdida de conexión, por consiguiente, para la implementación del sistema de ubicación del proyecto se ha considerado el uso de un dispositivo GPS y comunicación redundante para la localización del UAV.

Requerimientos del prototipo auxiliar

En base a la información revisada anteriormente, se procede a establecer los requerimientos para el diseño del prototipo, los cuales se detalla a continuación:

- Sistema de comunicación redundante.
- Sistema electromecánico del paracaídas.
- Sensor de orientación.
- Sistema GPS.
- Sistema auditivo.
- Sistema de control.
- Suministro de energía.
- Aplicación móvil.

Se requiere un sistema de comunicación redundante con el fin de garantizar la activación manual del sistema electromecánico del paracaídas y la ubicación GPS del sistema, por lo tanto, se considera el uso de los módulos GSM y RF. La tecnología GSM permite una amplia cobertura de la señal, sin embargo, el usuario debe pagar para acceder a los servicios de la red GSM, por otra parte, al utilizar los módulos RF no conlleva ningún costo para el usuario porque utiliza un rango de frecuencia libre, pero de acuerdo con la frecuencia de trabajo y en ciertas situaciones para establecer un enlace de comunicación se requiere de línea de vista entre los dispositivos. El módulo RF tiene la capacidad de una conexión a través de un cable Micro USB y transferir los datos del sistema al dispositivo móvil, pero existe la posibilidad que ocurra una desconexión del medio de transmisión por diversos factores, por lo tanto, con el fin de garantizar esta comunicación, se considera utilizar un módulo *Bluetooth* (BT) para generar un enlace inalámbrico entre estos dispositivos.

Se considera un sistema electromecánico con el propósito de expulsar el paracaídas acoplado al UAV, para lo cual es vital el uso de un actuador que permita activar el mecanismo (resortes) cuando la situación lo amerite.

Se requiere de un sensor IMU con el propósito de conocer la orientación del UAV y cuando la aeronave sobrepase un determinado umbral se active de forma automática el sistema electromecánico del paracaídas.

Se considera un sistema auditivo y el sistema GPS con el fin de ubicar la aeronave de forma acústica y utilizando el mapa de la aplicación móvil con las coordenadas captadas de la antena GPS.

En el sistema de control se considera una placa electrónica con un microcontrolador que permita procesar los datos de cada sistema mencionado anteriormente, por otra parte, con el propósito de detener el giro de los motores y evitar que los hilos del paracaídas se enreden con las hélices, se considera el uso de un actuador que permita suprimir el suministro de energía de la aeronave, para que a su vez los motores se apaguen.

Se ha optado por un suministro de energía independiente al UAV, esto con el fin de que el sistema auxiliar pueda operar en caso de falla de la alimentación del vehículo aéreo.

Finalmente, para la interacción del usuario con las funciones del sistema se debe considerar una aplicación móvil compatible con dispositivos con el sistema operativo *Android*.

Es necesario mencionar que todo el sistema en conjunto se conforma de tres bloques, los mismos que se detalla a continuación:

- **Bloque 1:** está conformada por el sistema de control, el sistema GPS, el sistema auditivo, el sistema de comunicación RF y GSM, el sensor de orientación y un suministro de energía.
- **Bloque 2:** se encuentra el sistema electromecánico con el paracaídas.
- **Bloque 3:** destinado al usuario está conformado por un suministro de energía y el sistema de comunicaciones (módulo RF y *Bluetooth*).

En la **Figura 3.8** se puede observar un diagrama de bloques representativo de cada subsistema del proyecto, cabe mencionar que cada subsistema se encontrará en su correspondiente caja protectora.

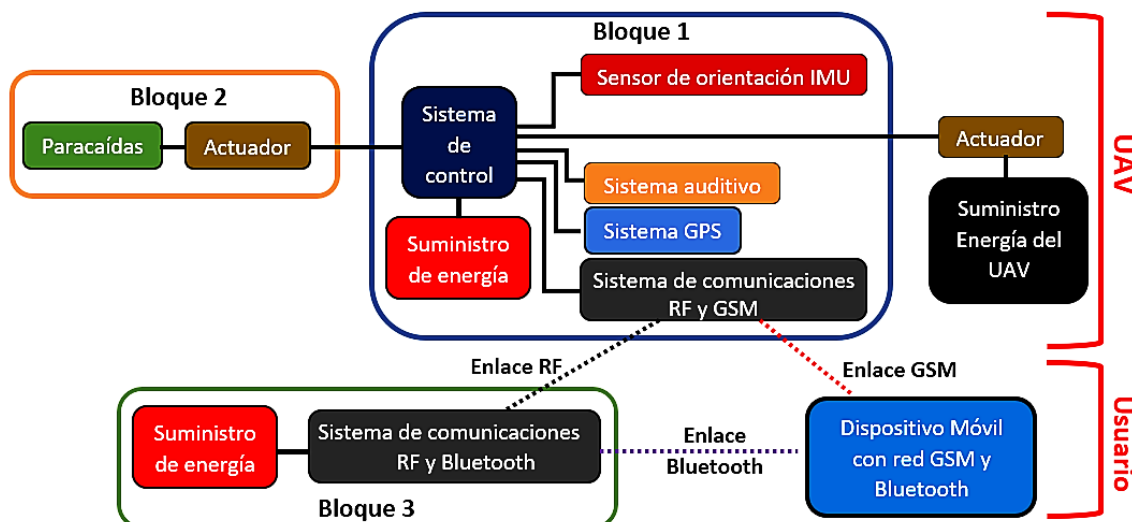


Figura 3.8 Diagrama de bloques del sistema propuesto (Fuente propia)

Prototipo auxiliar

El prototipo auxiliar propuesto en este proyecto permite al usuario la geolocalización del equipo utilizando un GPS y tiene la capacidad de desplegar un paracaídas de forma manual o automática, en este caso la manera manual es a través de un dispositivo móvil con la aplicación *Android*, en cambio, la forma automática es utilizando un sensor de orientación (IMU). El prototipo dispone de una comunicación redundante el cual emplea dos dispositivos con diferentes maneras de conexión, uno de ellos es utilizando el servicio de mensajes de texto del Sistema Global Móvil del inglés *Global System Mobile* (GSM), así mismo, el otro dispositivo usado para enviar o recibir información son los módulos de radio frecuencia (RF) de corto alcance. En la **Figura 3.9** se puede observar un diagrama de bloques representativo de la funcionalidad del prototipo auxiliar.

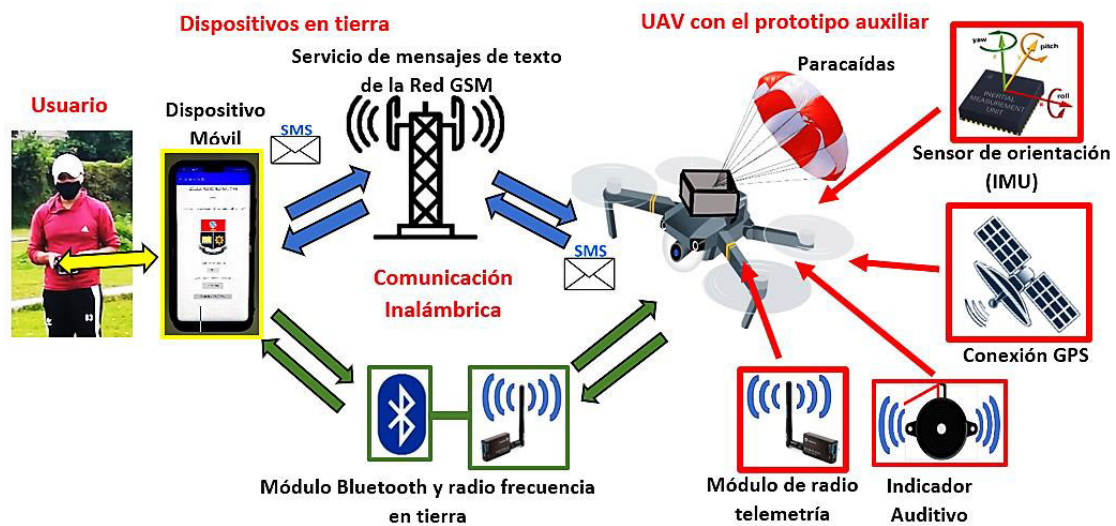


Figura 3.9 Diagrama de bloques del prototipo auxiliar (Fuente propia)

3.2 Selección de los dispositivos a usar en el sistema auxiliar

A continuación, se procede a realizar la selección y la descripción de cada dispositivo considerado para el sistema.

Selección del módulo con tecnología GSM y GPS

En el mercado comercial existen varios módulos que incorporan la tecnología GSM y GPS, por lo tanto, se procede a considerar tres módulos que dispongan de la conectividad GSM y GPS como se puede observar en la **Tabla 3.1**. Con el fin de tener un sistema más compacto, se procede a realizar la selección de un módulo que incorpore la tecnología GSM y GPS utilizando la herramienta matriz de priorización, la

cual con base a la aplicación de criterios y ponderaciones permite seleccionar la mejor alternativa entre varias opciones [24].

Tabla 3.1 Especificaciones técnicas de los módulos con tecnología GSM y GPS [25], [26], [27], [28], [29], [30]

Especificaciones	Módulo A7	Módulo SIM808 EVB v3.2	Mini módulo SIM808 v2 2.9.1
Banda de frecuencia de la comunicación GSM	850/900/1800/1900 (MHz)	850/900/1800/1900 (MHz)	850/900/1800/1900 (MHz)
Capacidad de conexión GPS	Sí	Sí	Sí
Ganancia de la antena GSM	2 (dBi)	3 (dBi)	3 (dBi)
Ganancia de la antena GPS	3 (dB)	28 (dB)	28 (dB)
Consumo de la corriente	2 (A)	0.355 - 2 (A)	1.6 (A)
Tensión de entrada	5 - 9 (V)	3.4 - 26 (V)	3.4 - 18 (V)
Plataformas compatibles para el procesamiento de información	Plataforma Arduino y Raspberry Pi	Plataforma Arduino	Plataforma Arduino
Dimensiones de la placa del módulo	6.58 x 4.73 (cm)	8.5 x 5.5 (cm)	5.5 x 4.5 (cm)
Peso del módulo incluyendo las antenas GPS y GSM	49 (g)	79 (g)	40 (g)
Costo aproximado	\$ 44	\$ 45	\$ 40
Imagen referencial			

En la **Tabla 3.2** se presentan los criterios y ponderaciones que se ha considerado para la selección del módulo.

Tabla 3.2 Criterios y ponderación para la selección del módulo GSM y GPS

Criterio	Descripción	Ponderación
1	Ganancia de las antenas (GPS y GSM)	0.2
2	Plataforma compatible para el procesamiento de la información.	0.1
3	Consumo de corriente y voltaje.	0.2
4	Dimensiones	0.2
5	Peso	0.2
6	Costo	0.1
Total		1

Para la matriz de priorización se considera un puntaje de 0 a 5 para cada criterio del módulo GSM y GPS como se visualiza en la **Tabla 3.3**.

Tabla 3.3 Selección del módulo con tecnología GSM y GPS

Criterio	Ponderación	Alternativa 1: Módulo A7		Alternativa 2: Módulo SIM808 EVB v3.2		Alternativa 3: Módulo SIM808 v2 2.9.1	
		Puntaje	Subtotal	Puntaje	Subtotal	Puntaje	Subtotal
1	0.2	2	0.4	5	1	5	1
2	0.1	4	0.4	4	0.4	4	0.4
3	0.2	2	0.4	3	0.6	4	0.8
4	0.2	3	0.6	2	0.4	4	0.8
5	0.2	4	0.8	2	0.4	4	0.8
6	0.1	3	0.3	2	0.2	4	0.4
		Total 1:	2.9	Total 2:	3	Total 3:	4.2

De acuerdo con los resultados de la **Tabla 3.3**, se identifica que la alternativa número tres es la adecuada para el proyecto; por esta razón, se considera el uso del módulo SIM808 v.2 2.9.1 (ver **Figura 3.10**).

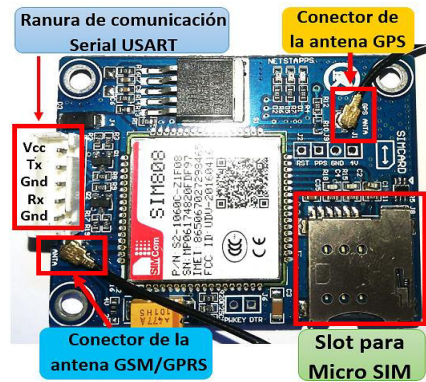


Figura 3.10 Módulo SIM808 v2 2.9.1 con tecnología GSM y GPS (Fuente propia)

Las características más importantes que dispone el módulo SIM808 GSM con GPS son los siguientes:

- El control del módulo es a través de comandos AT.
- Dispone de un puerto serial USART para la interacción con un microprocesador.
- Soporta una fuente de alimentación de 3.4 – 18 (V) con una corriente de consumo mínimo de 355 (mA) y máximo de 2 (A).
- La precisión de las coordenadas GPS en posición horizontal es de < 2.5 (m).
- Pesa aproximadamente 40 (g) [31].

Unidad de Medición Inercial

En el mercado existen diferentes módulos para determinar la orientación de un equipo, en este caso se ha considerado la IMU modelo GY-521 (ver **Figura 3.11.**) porque es un módulo pequeño, de bajo costo y suministra el dato (ángulo de giro de un objeto) que se requiere en el proyecto.

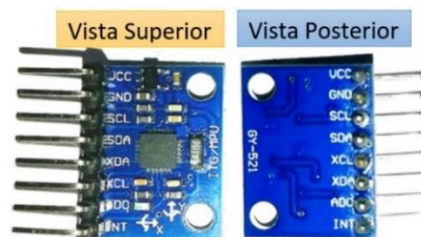


Figura 3.11 Módulo IMU modelo GY-521 (Fuente propia)

Las características relevantes que dispone el módulo IMU GY-521 son las siguientes:

- La tensión de operación del dispositivo varía entre los 3 - 5 (V) y una corriente de operación de 6 (mA).
- Con un peso de 4 (g).

- Para la comunicación se utiliza el interfaz I2C a 400 (KHz) [32].

Actuador

De acuerdo con los requerimientos del proyecto, el sistema del paracaídas requiere de un actuador, por lo tanto, se ha considerado el micro servomotor modelo MG 90S (ver **Figura 3.12**) debido a su tamaño reducido y consumo de energía.



Figura 3.12 Micro servomotor modelo MG 90 S [33]

Las características más relevantes del micro servomotor son:

- La tensión de operación de 4.8 – 6 (V) y una corriente de operación aproximado de 100 (mA).
- El peso es de 13.4 (g) debido a engranes metálicos.
- Torque de 1.8 (kgf/cm) con 4.8 (V) y 2.2 (kgf/cm) con 6 (V).
- Un ciclo de trabajo de 1 – 2 (ms) y con un período de PWM de 20 (ms) o 50 (Hz) [33].

Con el fin de controlar el suministro de energía del UAV, el segundo actuador que se ha considerado para el proyecto es un módulo relé (ver **Figura 3.13**).

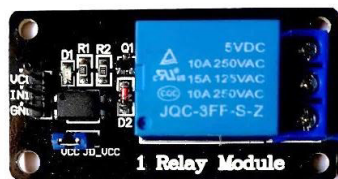


Figura 3.13 Módulo relé (Fuente propia)

Las características más relevantes del módulo son:

- La tensión de operación de 5 (V) y una corriente de operación aproximado de 15 a 20 (mA).
- La activación del contacto es con 0 lógico.
- El peso es de 15 (g) [34].

Módulo de radio frecuencia

Como se estableció que el proyecto disponga de una comunicación redundante y uno de los enlaces considerados es utilizando módulos RF, se ha seleccionado el modelo mRo 915 (MHz) (ver **Figura 3.14**) porque estos equipos tienen aplicaciones para drones, son de bajo consumo de energía y son compatibles con microcontroladores que tienen un interfaz de comunicación serial UART, por ejemplo, el Atmega 328P.



Figura 3.14 Módulos RF modelo mRo 915 (MHz) [35]

Las características más relevantes de los módulos son:

- La tensión de operación de 4.8 – 5.6 (V) y una corriente de operación aproximada de 90 (mA).
- Peso de los equipos es de 13.4 (g).
- Disponen de un interfaz de comunicación serial UART.
- El rango de operación es aproximadamente de 300 (m) [35].

Módulo de *bluetooth*

De acuerdo con los requerimientos del sistema para el enlace *Bluetooth* del bloque 3 con el dispositivo móvil, se seleccionó el modelo HC-05 (ver **Figura 3.15**) debido a su bajo costo de adquisición y bajo consumo de energía.

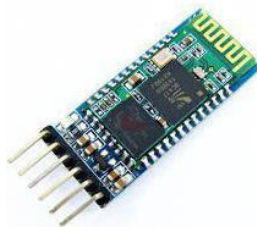


Figura 3.15 Módulo de *Bluetooth* modelo HC-05 [36]

Las características más destacadas del módulo son:

- La tensión de operación de 3 – 6 (V) y una corriente de operación aproximado de 50 (mA).
- Peso de 3 (g).
- Banda de frecuencia de 2.4 (GHz).
- Alcance aproximado de 10 (m) [36].

Buzzer

Una de las funciones del sistema es emitir una señal auditiva con el fin de ayudar a la localización del UAV, por consiguiente, se consideró el uso de un *buzzer* como se observa en la **Figura 3.16**.



Figura 3.16 Buzzer [37]

Las características relevantes son las siguientes:

- La tensión de operación varia entre los 4 - 8 (V) y una corriente de operación de 30 (mA).
- Las dimensiones son de 9.5 (mm) de alto y con un diámetro de 12 (mm).
- Con un peso de 2 (g) [37].

Microcontrolador

Como se ha mencionado anteriormente se requiere un microprocesador para procesar los datos de los módulos y activar los actuadores, por lo tanto, con base a los componentes electrónicos seleccionados se establecen los requerimientos en cuanto a pines y características de comunicación del sistema, los mismos se presenta en la **Tabla 3.4**.

Tabla 3.4 Requerimientos al microprocesador de los módulos

Módulo	Requerimientos al microprocesador
SIM808	Puerto serial UART
RF Sik mRo	Puerto serial UART
IMU GY-521	Puerto I2C
Micro servomotor MG 90S	Salida PWM

<i>Buzzer</i>	Salida digital
Relé	Salida digital
Led indicador 1	Salida digital
Led indicador 2	Salida digital
Led indicador 3	Salida digital

En base a la **Tabla 3.4** se ha seleccionado el microcontrolador Atmega 328-P (ver **Figura 3.17**) debido a su disponibilidad de pines digitales, puertos de comunicación y es de bajo consumo de energía.



Figura 3.17 Microprocesador Atmega 328P (Fuente propia)

Las características más relevantes son:

- La tensión de operación varía entre los 1.8 – 5.5 (V) y una corriente de operación aproximada de 2.7 (mA).
- Dispone de 23 pines de I/O digitales.
- Capacidad de comunicación utilizando el protocolo I2C, USART, SPI.
- La memoria Flash es de 32 (KB) y una memoria RAM de 2 (KB) [38] [39].

Fuente de alimentación

El sistema montado en el UAV (bloque 1 y 2) requiere de un suministro de energía externo a la aeronave, por lo tanto, se ha considerado el uso de una batería Li-Po porque son livianas y tienen una mejor capacidad de corriente. El sistema en tierra (interfaz del usuario – bloque 3) requiere de una batería, ya que es un sistema portátil, en este caso se ha considerado una de tipo alcalina debido a su bajo costo y pequeñas dimensiones.

Con la **Ecuación 3.1** es posible conocer la capacidad de la batería Li-Po requerida para energizar al bloque 1 y 2, para lo cual se debe establecer la autonomía y la corriente total del sistema.

$$\text{Autonomía} = \frac{C_{\text{Batería}}}{I_{\text{Sistema}}}$$

Ecuación 3.1 Autonomía del sistema [40]

Donde:

Autonomía : (h) tiempo de descarga de la batería.

$C_{\text{Batería}}$: (mAh) capacidad de la batería.

I_{Sistema} : (mA) corriente de consumo.

El tiempo de vuelo de algunos UAV's de ala rotatoria se encuentra entre los 15 a 30 minutos y en casos especiales de acuerdo con la aplicación de la aeronave puede superar las 2 horas [41] [42], por lo tanto, se ha considerado de 2 (h) la autonomía del sistema.

Con el propósito de estimar la corriente total del sistema, se procede a listar los valores de la tensión y corriente de cada módulo como se observa en la **Tabla 3.5**.

Tabla 3.5 Consumo de energía de cada módulo [31] [32] [33] [35] [37] [38] [39]

Módulo	Tensión de alimentación (V)	Corriente (mA)
ATmega 328P	1.8 – 5.5	2.7
SIM808	3.4 – 18	355
Radio frecuencia Sik mRo	4.8 - 6	90
IMU GY-521	3 - 5	6
Micro servomotor MG 90S	4.8 - 6	100
<i>Buzzer</i>	4.8 - 8	30
Relé	5	15
Corriente Total:		598.7

Con la información descrita anteriormente y la **Ecuación 3.1** se procede a realizar los cálculos para obtener la capacidad de la batería.

$$C_{\text{Batería}} = I_{\text{Sistema}} * \text{Autonomía}$$

Donde:

Autonomía : 2 (h) tiempo de duración de la batería

I_{Sistema} : 598.7 (mA) corriente total del sistema

Lo cual se obtiene como resultado:

$$C_{\text{Batería}} = 1\,197.4 \text{ (mAh)}$$

De acuerdo con el resultado obtenido de 1197.4 (mAh) de la capacidad de la batería se consideró la adquisición de la batería Li-Po de 3.7 (V) con 1200 (mAh) como se observa en la **Figura 3.18**.



Figura 3.18 Batería Li-Po (Fuente propia)

Con el fin de energizar los módulos que requieren 5 (V) se procede a realizar la conexión en serie de dos baterías Li-Po (ver **Figura 3.19**) de esta manera se obtiene un voltaje aproximado de 7.4 (V) conservando la corriente de 1200 (mAh), por consiguiente, con este nuevo cambio en la fuente de energía, se considera el uso de reguladores de voltaje en la placa electrónica del sistema.

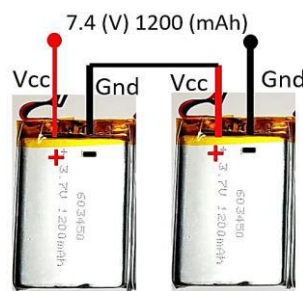


Figura 3.19 Conexión en serie de las baterías Li-Po de 3.7 (V) (Fuente propia)

A continuación, se procede a determinar la fuente de alimentación del bloque 3 con la **Ecuación 3.1** y tomando en cuenta la autonomía de 2 (h) del sistema (bloque 1 y 2) montado en el UAV.

En la **Tabla 3.6** se puede observar el consumo de energía de los módulos del bloque 3.

Tabla 3.6 Consumo de energía de los módulos del bloque 3 [35] [36]

Módulo	Tensión de alimentación (V)	Corriente (mA)
Radio frecuencia Sik mRo	4.8 - 6	90
Bluetooth HC-05	3 - 6	50
Corriente Total:		140

Con la información descrita en la **Tabla 3.6** y la **Ecuación 3.1**, se procede a realizar los cálculos para obtener la capacidad de la batería para el bloque 3.

$$C_{\text{Batería}} = I_{\text{Sistema (3)}} * \text{Autonomía}$$

Donde:

Autonomía : 2 (h) tiempo de duración de la batería

$I_{\text{Sistema (3)}}$: 140 (mA) corriente total del bloque 3

Lo cual se obtiene como resultado:

$$C_{\text{Batería}} = 280 \text{ (mAh)}$$

De acuerdo con el resultado obtenido de 280 (mAh) de la capacidad de la batería, se consideró la adquisición de la batería alcalina de 9 (V) con 500 (mAh) (ver **Figura 3.20**).



Figura 3.20 Batería alcalina de 9 (V) [43]

3.3 Implementación del prototipo en base a los dispositivos seleccionados

En el siguiente apartado se presenta los diagramas y los diseños para la implementación de la placa electrónica del prototipo.

Diseño de los diagramas del prototipo

En la **Figura 3.21** se puede observar el diagrama de la alimentación eléctrica del sistema (bloque 1 y 2) que va a ir en la aeronave, la conexión eléctrica del módulo SIM808 es directa con la batería de Li-Po porque el módulo soporta una alimentación superior a los 3.5 (V), sin embargo, para energizar los otros módulos del sistema se considera el uso de un regulador de tensión de 5 (V).

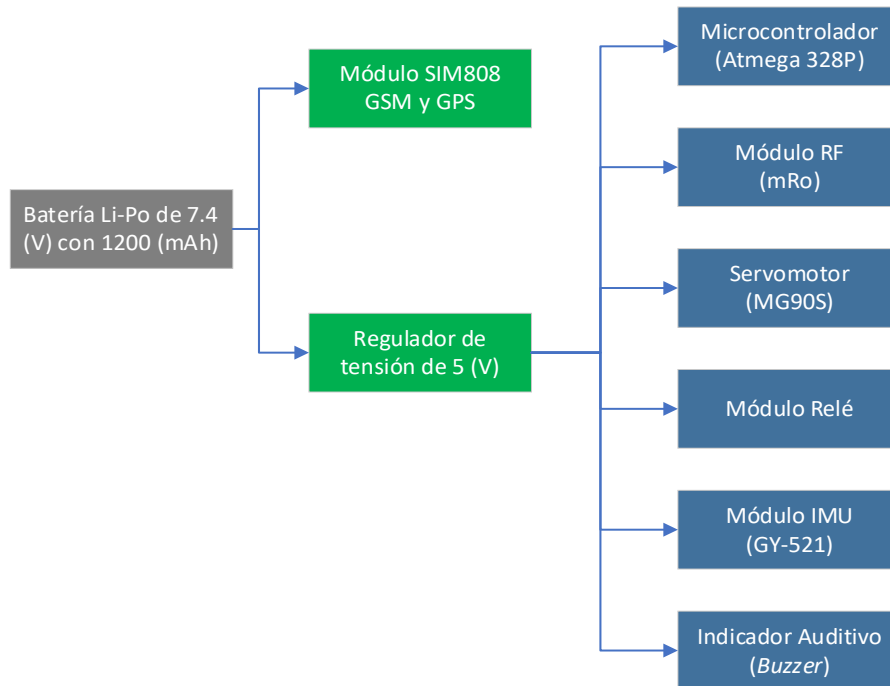


Figura 3.21 Diagrama de bloques de la alimentación eléctrica del bloque 1 y 2
(Fuente propia)

En la **Figura 3.22** se puede observar el diagrama de alimentación eléctrica del bloque 3, en el cual dispone de una batería 9 (V) a 500 (mAh) y un regulador de voltaje de 5 (V).

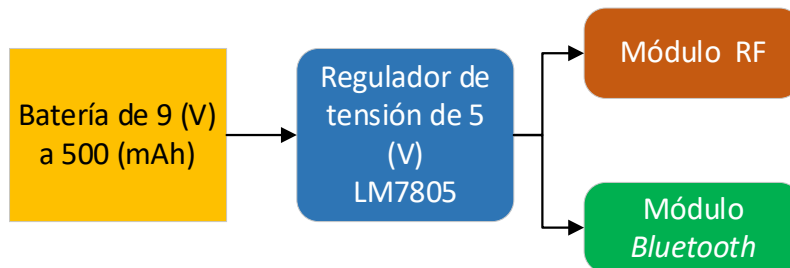


Figura 3.22 Diagrama de bloques del suministro de energía del bloque 3
(Fuente propia)

Implementación de la placa electrónica

Con el fin de implementar una placa liviana y de tamaño reducido, se ha considerado usar componentes de montaje superficial (SMD) por ejemplo: resistores, capacitores, diodo rectificador y un regulador de tensión (ver **Figura 3.23**); también se estima el uso de componentes de tecnología de agujeros pasantes (THT) debido a que en el mercado son los más comunes y fáciles de adquirir.

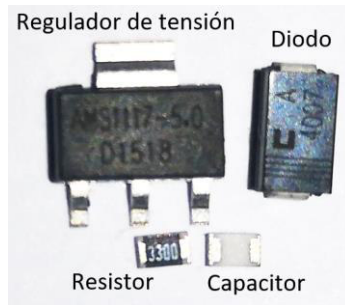


Figura 3.23 Componentes SMD (Fuente propia)

Previo a la fabricación de las placas electrónicas del bloque 1 y 3, se realizaron los diseños circuitales en el programa Proteus 8.5. En la Figura 3.24 se puede observar el diagrama del bloque 1.

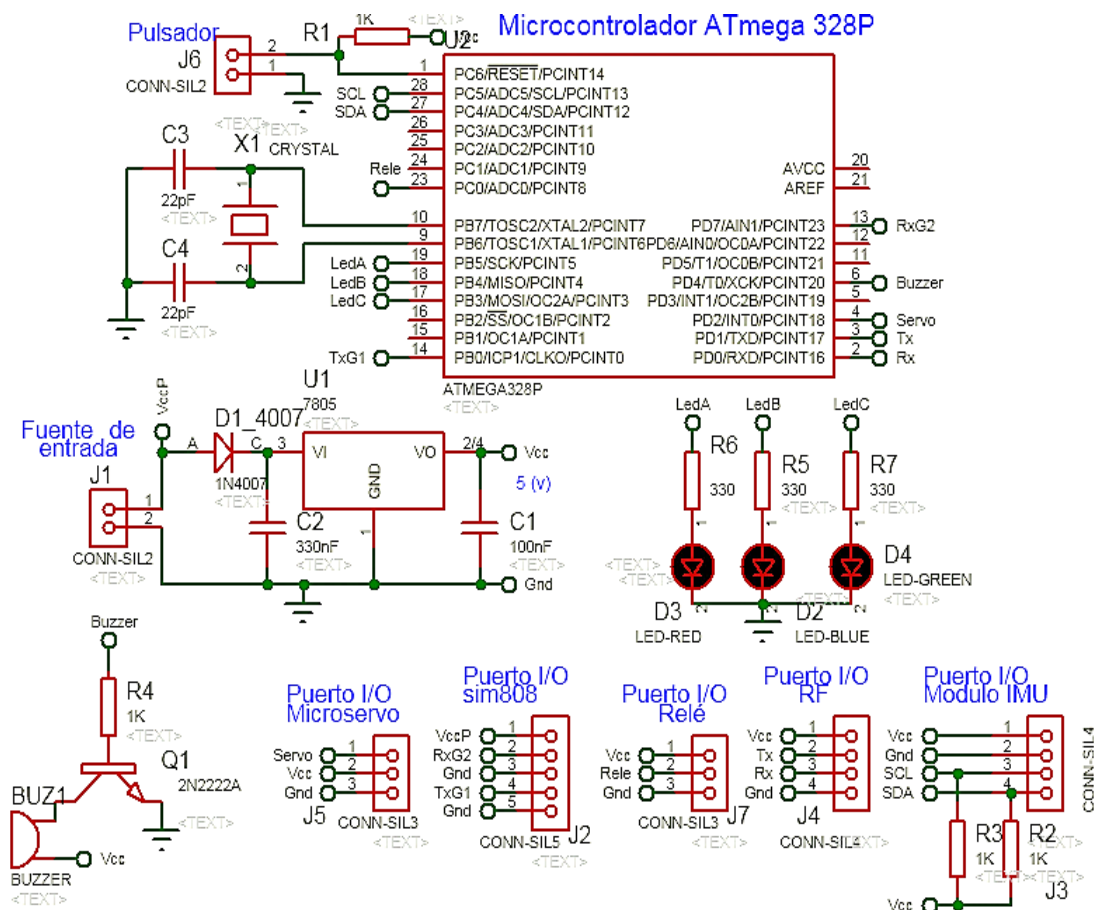


Figura 3.24 Diagrama circuitual del bloque 1 (Fuente propia)

En la Figura 3.25 se puede visualizar el diagrama circuitual del bloque 3.

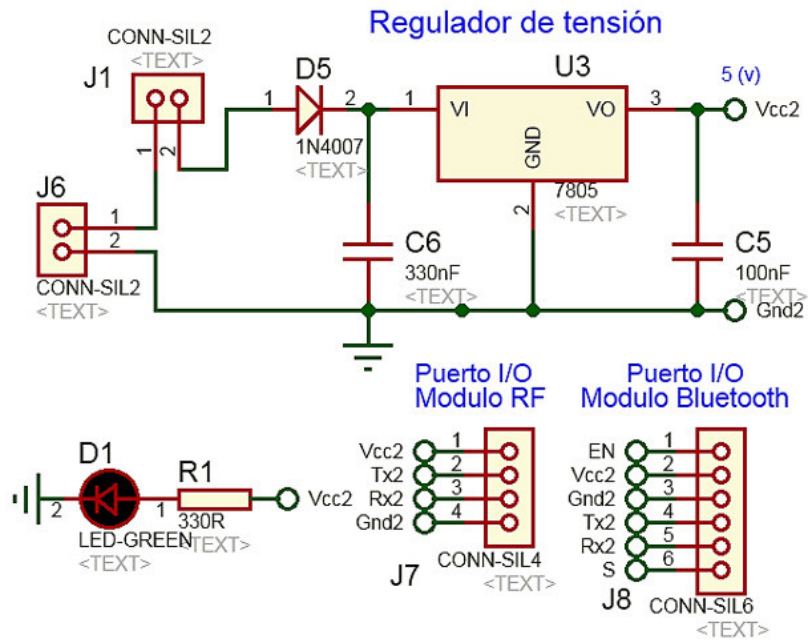


Figura 3.25 Diseño circuital del bloque 3 (Fuente propia)

En la **Figura 3.26** se puede observar la placa electrónica con dimensiones de 6 x 3.6 (cm) del bloque 1 del sistema, la misma está implementada con componentes electrónicos SMD, THT y los conectores para la conexión de los módulos.

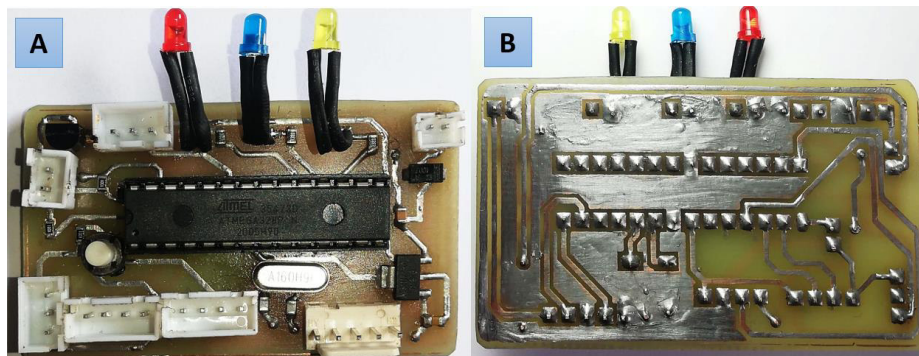


Figura 3.26. (A) Vista superior (B) Vista inferior de la placa electrónica del bloque 1 (Fuente propia)

En la **Figura 3.27** se puede visualizar la placa electrónica del bloque 3, tiene dimensiones de 4.8 x 2.8 (cm) y está conformada por componentes THT, conectores para el módulo RF y *Bluetooth*.

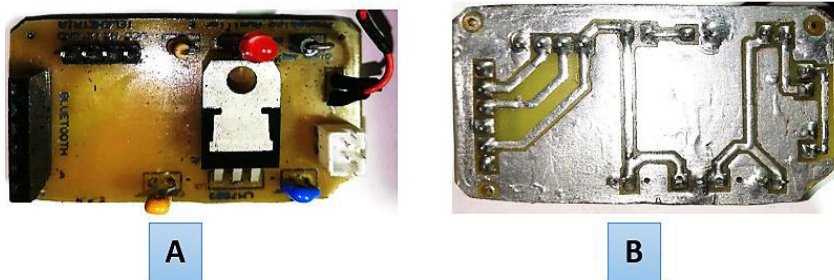


Figura 3.27 (A) Vista superior (B) Vista inferior de la placa electrónica del bloque 3
(Fuente propia)

Diseño del paracaídas del prototipo auxiliar

De acuerdo con los requerimientos del proyecto se requiere de un sistema que permita amortiguar la caída del UAV, por lo tanto, para el diseño del paracaídas se debe considerar las fuerzas físicas generadas cuando la aeronave inicia el descenso con este sistema como se puede observar en la **Figura 3.28** [44].



Figura 3.28 Descenso de un modelo aéreo utilizando un paracaídas [44]

Considerando que las fuerzas que se generan son vectores es posible igualar (ver **Ecuación 3.2**) para poder disponer del cálculo del área del paracaídas.

$$m * g = K * v^2$$

Ecuación 3.2 Igualación de las variables de la fuerza de rozamiento y gravedad [44]

Donde:

- m : (kg) masa del modelo incluyendo el paracaídas.
- g : (m/s^3) aceleración de la gravedad.
- K : Constante del paracaídas.
- v : (m/s) velocidad del descenso del objeto.

Para calcular la constante proporcional (K) de la fuerza de rozamiento se debe utilizar la **Ecuación 3.3**.

$$K = \frac{\rho \cdot A_p \cdot C_d}{2}$$

Ecuación 3.3 Constante del paracaídas [44]

Donde:

- K : Constante del paracaídas.
- ρ : (kg/m³) densidad del aire, al nivel del mar es de 1.223 (kg/m³)
- A_p : (m²) área del paracaídas.
- C_d : Coeficiente de arrastre de acuerdo con la forma del paracaídas.

Remplazando la **Ecuación 3.3** en la **Ecuación 3.2** y en base a cálculos matemáticos es posible disponer del área del paracaídas como se observa en la **Ecuación 3.4**.

$$A_p = \frac{2 \cdot g \cdot m}{\rho \cdot C_d \cdot v^2}$$

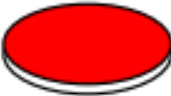

Ecuación 3.4 Área del paracaídas [44]


Donde:

- A_p : (m²) área del paracaídas.
- m : (kg) masa del modelo incluyendo el paracaídas.
- g : (m/s³) aceleración de la gravedad.
- C_d : Coeficiente de arrastre de acuerdo con la forma del paracaídas.
- v : (m/s) velocidad del descenso del objeto.
- ρ : (kg/m³) densidad del aire.

En la **Tabla 3.7** se puede visualizar los valores del coeficiente de arrastre (C_d) que permite definir la forma del diseño de un paracaídas.

Tabla 3.7 El coeficiente de arrastre de acuerdo con la forma del objeto [44]

Nombre	Forma del objeto	Valor aproximado de C _d
Disco circular rígido		1.2
Hemisferio		0.8

Semi-hemisferio plano		0.75
--------------------------	---	------

Con el fin de facilitar la construcción del paracaídas en función del valor del diámetro, se considera el área de un círculo como se observa en la **Ecuación 3.5**.

$$A_c = \pi \cdot \left(\frac{d}{2}\right)^2$$

Ecuación 3.5 Área de un círculo [45]

Donde:

A_c : (m²) área de la forma plano circular del paracaídas.

d : (m) diámetro del área de un círculo.

Posteriormente de realizar un proceso matemático de igualación y despeje con la **Ecuación 3.4** y **Ecuación 3.5** se obtiene la **Ecuación 3.6**, la misma que permite calcular el diámetro del paracaídas.

$$d = \sqrt{\frac{8 \cdot g \cdot m}{\rho \cdot C_d \cdot \pi \cdot v^2}}$$

Ecuación 3.6 Diámetro del paracaídas plano circular [44]

Antes de proceder a realizar el cálculo del diámetro del paracaídas, es primordial mencionar que para el proyecto se ha considerado un cuadricoptero (ver **Figura 3.29**) con dimensiones de 49.5 x 24.5 (cm) y con una masa aproximado de 1 (kg), también se considera que el prototipo auxiliar dispondrá de una masa de 500 (g).



Figura 3.29 Dimensiones del cuadricóptero (Fuente propia)

De acuerdo con el valor de la masa del UAV y del proyecto se procede a calcular el diámetro del paracaídas plano circular de forma Semi-hemisferio utilizando la **Ecuación 3.6**.

Donde:

- m_{total} : 1.5 (kg) masa aproximada del modelo del UAV incluyendo el proyecto.
- g : 9.81 (m/s^2) aceleración de la gravedad.
- C_d : 0.75 coeficiente de arrastre de acuerdo con la forma Semi-hemisferio del paracaídas.
- v : 4,46 (m/s) velocidad del descenso del objeto.
- ρ : 1.223 (kg/m^3) densidad del aire al nivel del mar.
- d : (m) diámetro para el diseño de un paracaídas plano circular.

Usando la **Ecuación 3.6** se obtiene:

$$d = 1.43 \text{ (m)}$$

En la **Figura 3.30** se puede observar el diseño del paracaídas con un diámetro de 1.43 (m) para amortiguar la caída de una masa de 1.5 (kg).

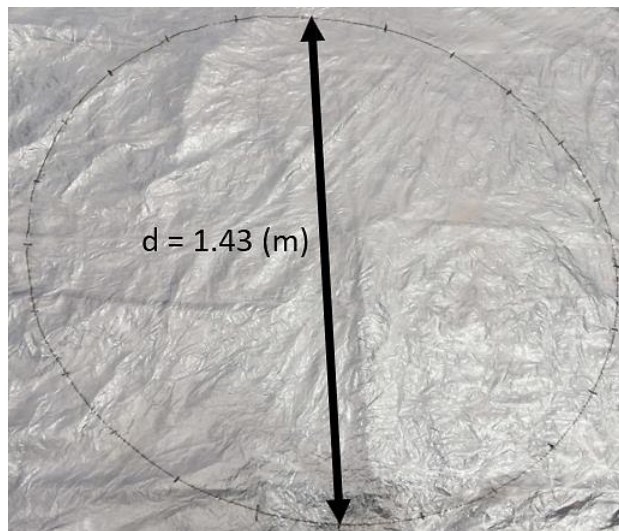


Figura 3.30 Paracaídas del proyecto (Fuente propia)

En la **Figura 3.31** se puede observar el paracaídas plegado con medidas de 6 x 8 (cm), las mismas que se han considerado para el diseño de la caja del bloque 2 del sistema.

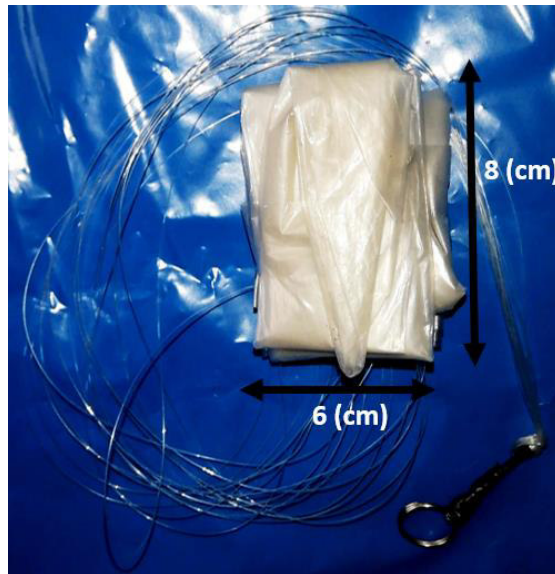


Figura 3.31 Paracaídas (Fuente propia)

Diseño de la caja protectora del prototipo

En el mercado existen diferentes cajas prefabricadas ideales para conservar el proyecto, sin embargo, el peso para esta aplicación puede ser alto debido al grosor del material y a las dimensiones estandarizadas, por esta razón se ha considerado el diseño en 3D de las cajas protectoras del bloque 1 y 2, las mismas que son diseñadas en el programa *Autodesk Fusion 360* y la impresión del modelo se lo realizó en filamento de ácido poliláctico denominado PLA.

Para el diseño de la caja protectora (ver **Figura 3.32**) que concentra el bloque 2, se consideró las dimensiones del plegado del paracaídas y el sistema electromecánico.

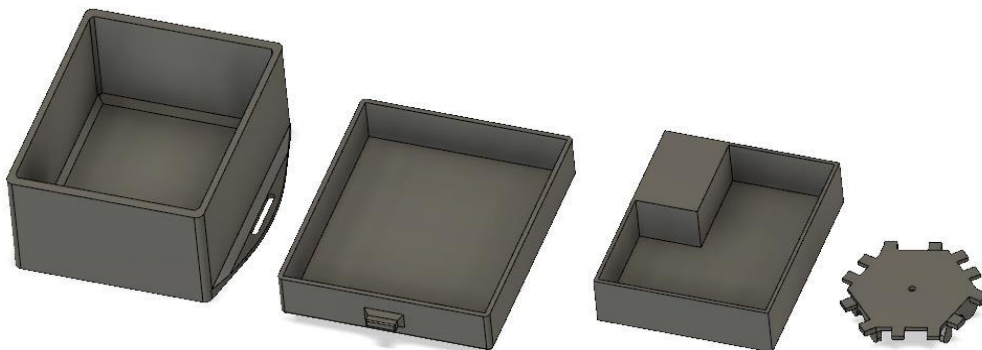


Figura 3.32 Diseño de la caja del bloque 2 (Fuente propia)

En la **Tabla 3.8** se puede observar la descripción de cada pieza de la caja del sistema del paracaídas.

Tabla 3.8 Descripción de cada pieza de la caja de bloque 2

Pieza	Descripción
	Hexágono giratorio para la sujeción de resortes.
	Base portadora del paracaídas.
	Tapa de la caja.
	Caja protectora.

Para el diseño de la caja (ver **Figura 3.33**) del bloque 1 del sistema, se ha considerado las dimensiones de la placa electrónica del sistema de control, los módulos y la fuente de alimentación.

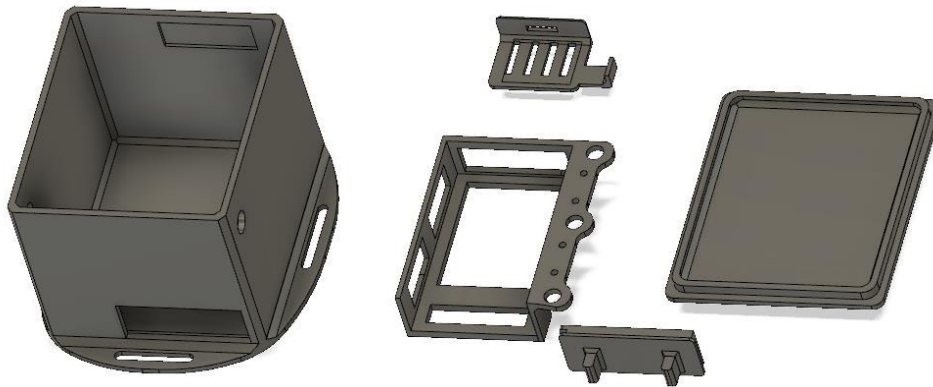





Figura 3.33 Diseño de la caja protectora del bloque 1 (Fuente propia)

En la **Tabla 3.9** se puede observar la descripción de cada pieza que conforma la caja protectora dl bloque 1.

Tabla 3.9 Descripción de las piezas que conforman la caja protectora

Pieza	Descripción
	Base de la batería y soporte de la placa electrónica.
	Soporte de la IMU.
	Tapa del compartimiento de la batería.
	Tapa de la caja protectora.

	<p>Caja protectora de los módulos y la placa electrónica.</p>

El bloque 3 del proyecto requiere de un armazón que conserve la placa electrónica, el suministro de energía y los módulos BT y RF, debido a que este sistema no se acopla en la aeronave, se ha utilizado una caja comercial como se indica en la **Figura 3.34**.

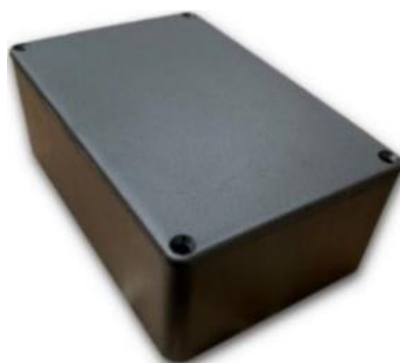


Figura 3.34 Caja prefabricada para el bloque 3 del proyecto (Fuente propia)

3.4 Diseño e implementación del algoritmo de control del sistema

Para la implementación del algoritmo de control, se han considerado las diferentes funciones que dispone el proyecto, las mismas se detalla a continuación:

- **Comunicación redundante:** con el fin de una comunicación redundante en el sistema se encuentra habilitado dos medios de comunicación utilizando los enlaces GSM y RF, para lo cual solo depende del usuario la selección del tipo de enlace para solicitar la información al sistema montado en el UAV.
- **Ubicación GPS del prototipo:** el usuario solicita al sistema las coordenadas GPS, los datos de localización del SIM808 son procesados por el microcontrolador, el mismo envía el dato de longitud y latitud al dispositivo móvil a través del canal de comunicación por donde se realizó la solicitud.

- **Activación manual del sistema del paracaídas:** esta función permite al usuario a través del dispositivo móvil solicitar al microcontrolador la activación del relé para suprimir el suministro de energía que está conectado al UAV y expulsar el paracaídas.
- **Activación automática del sistema del paracaídas:** periódicamente el módulo IMU GY-521 envía los datos de orientación de la aeronave al microcontrolador, el cual compara el dato con el umbral establecido, en el caso de sobrepasar el umbral el sistema activa el relé para apagar el UAV y después de un corto tiempo expulsar el paracaídas.
- **Información del estado del sistema:** el usuario solicita el nivel de intensidad señal de la red GSM del sistema, la cual es proporcionada por el módulo SIM808 y procesada por el microcontrolador, por último, este dato es enviado al dispositivo móvil a través del enlace de comunicación preseleccionado.
- **Sistema Auditivo:** el *buzzer* se activa por un corto periodo después de la expulsión del paracaídas y de forma manual utilizando la opción de la aplicación móvil.

En la **Figura 3.35** se puede observar de forma general el diagrama de flujo de la funcionalidad del sistema.

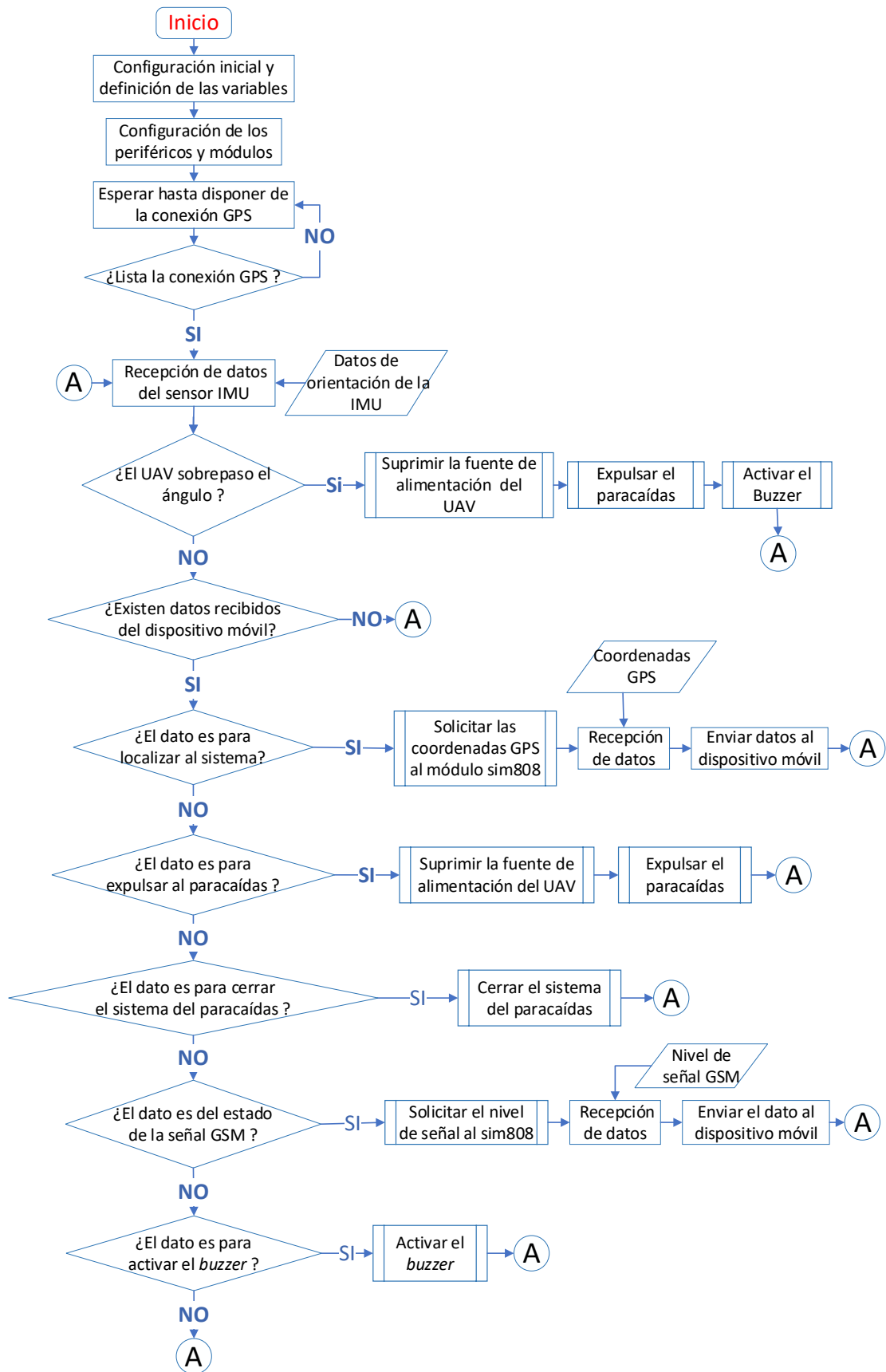


Figura 3.35 Diagrama de flujo del sistema principal (Fuente propia)

Diseño de la interfaz de usuario

Con el fin de proporcionar la interacción entre el dispositivo móvil del usuario con el sistema montado en el UAV, se considera el diseño de una aplicación con las funciones que dispone del proyecto, la misma fue desarrollada en el programa *Android Studio 3.6.3* y lleva como nombre de *Geoparachute*.

La aplicación *Geoparachute* está compuesta por un interfaz principal y un secundario, el interfaz principal dispone de las opciones para determinar el tipo de enlace de comunicación como se observa en la **Figura 3.36**.



Figura 3.36 Interfaz principal de la aplicación del proyecto (Fuente propia)

Las opciones del interfaz principal son las siguientes:

1. **GSM:** se presenta las opciones para la comunicación a través de la red GSM utilizando los mensajes de texto.
2. **RF-BT:** se despliega las opciones para la comunicación BT con el bloque 3.
3. **Salir de la aplicación:** permite al usuario salir de la aplicación *Android*.

En la **Figura 3.37**. se puede observar el interfaz secundario de la aplicación con las funciones del sistema.



Figura 3.37 Opciones de la interfaz secundaria de la aplicación (Fuente propia)

Las opciones de la interfaz secundaria son las siguientes:

1. **Señal GSM:** permite conocer el estado de conexión y la intensidad de señal recibida (RSSI) de la red GSM del sistema.
2. **Expulsar paracaídas:** activa el relé para suprimir la energía del UAV y expulsar el paracaídas.
3. **Asegurar el sistema del paracaídas:** desactiva el relé para suministrar la energía a la aeronave y el servomotor regresa al estado inicial para guardar el paracaídas.
4. **Coordenadas GPS:** solicita las coordenadas de localización del sistema.

5. **Mapa OsmAnd:** permite visualizar la ubicación del sistema en el mapa de la aplicación *OsmAnd*.
6. **Google Maps:** localiza el sistema en el mapa de *Google*.
7. **Sistema Auditivo:** permite al usuario por un corto periodo activar el *buzzer*.

En la aplicación *Geoparachute* se recomienda el uso de la aplicación *OsmAnd*, la cual permite al usuario localizar al sistema en el mapa cuando el dispositivo móvil no disponga de conexión a Internet, por lo tanto, en el dispositivo móvil se debe de instalar la aplicación *OsmAnd* y se recomienda descargar previamente los mapas.

En la **Figura 3.38** se puede observar el diagrama de flujo de la funcionalidad de la aplicación móvil.

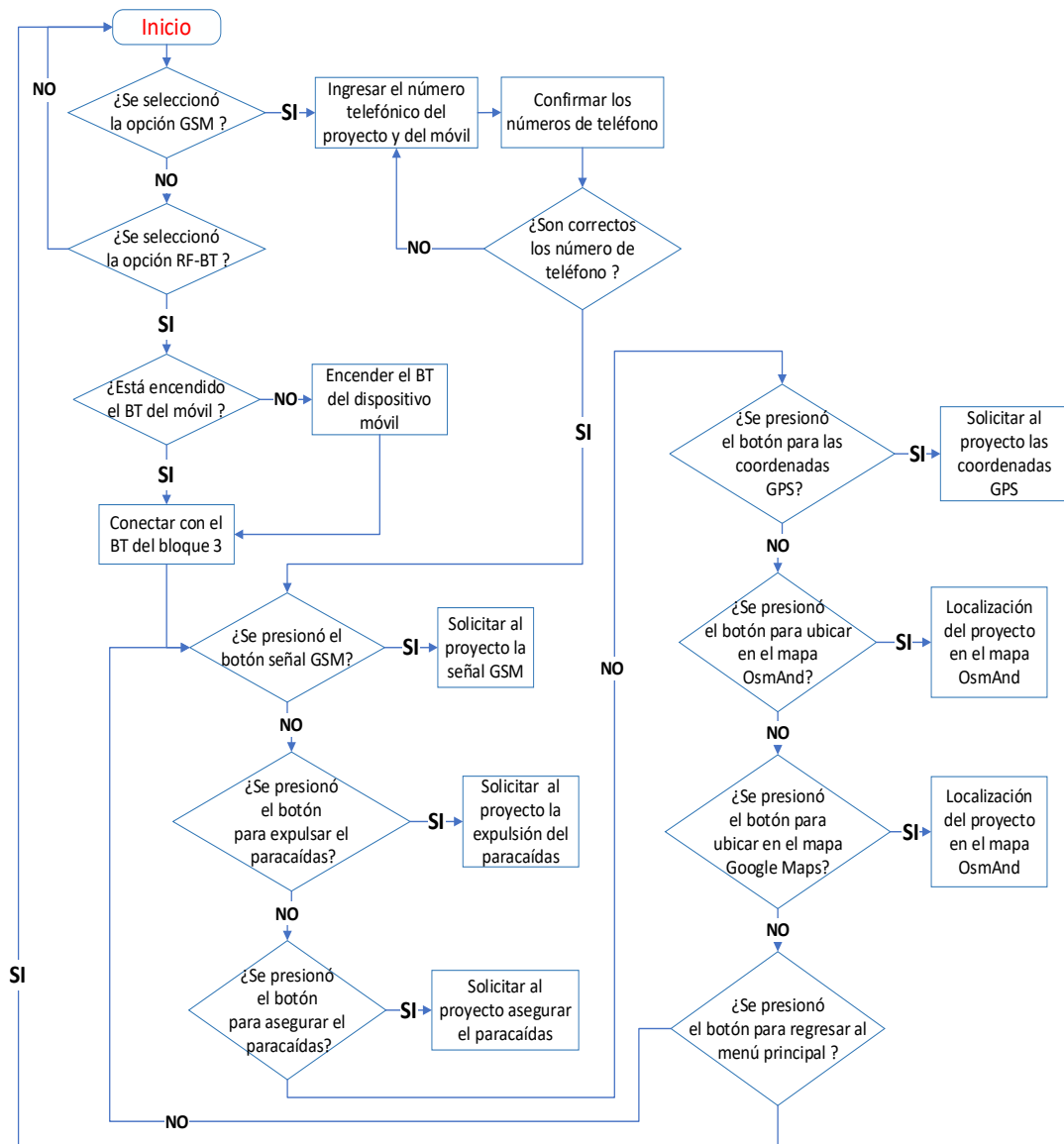


Figura 3.38 Diagrama de flujo de la funcionalidad de la aplicación móvil

3.5 Pruebas y Análisis de Resultados

En el siguiente apartado se presenta las pruebas y los resultados obtenidos del proyecto.

Pruebas del módulo SIM808 para la localización GPS

Con el propósito de verificar la recepción de los datos GPS del módulo SIM808, se efectuaron varias pruebas externas (aire libre) con el fin de que el módulo se conecte de forma eficiente con los satélites, para esta prueba se utilizó la placa Arduino Uno como se observa en la **Figura 3.39**.

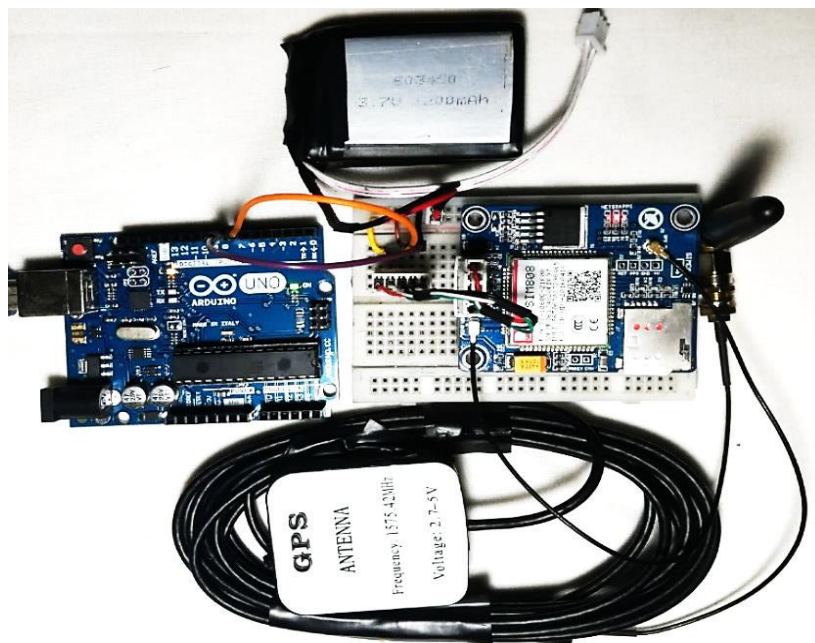


Figura 3.39 Pruebas del GPS del módulo SIM808 (Fuente propia)

En la placa Arduino Uno se ejecutó un código para la interacción con el módulo SIM808, los mismos fueron conectados a través de un enlace serial UART y se utilizaron varios comandos AT para obtener las coordenadas GPS (ver **Figura 3.40**).

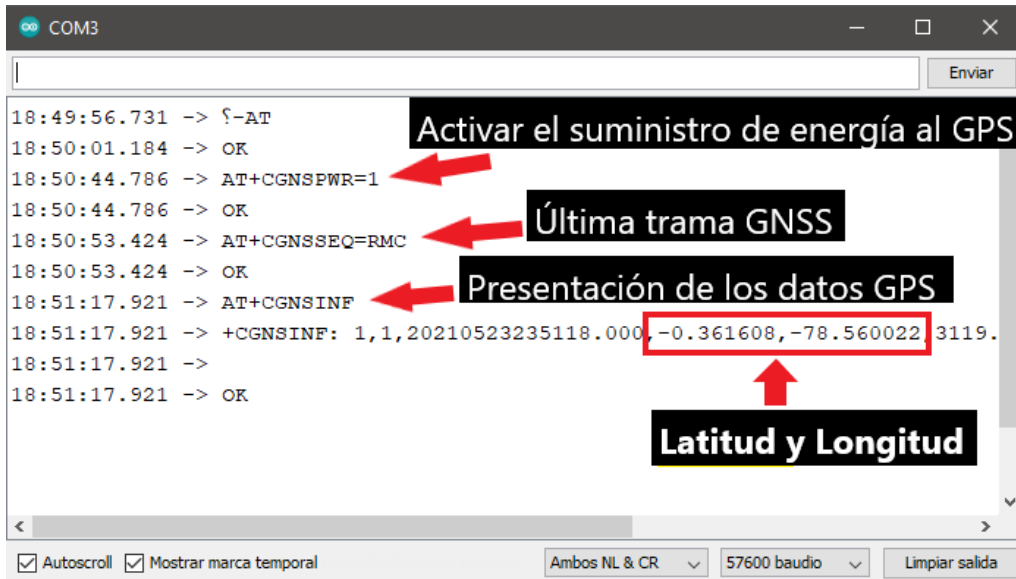


Figura 3.40 Recepción de la trama de la navegación por satélite (Fuente propia)

Por último, para verificar la distancia existente entre el módulo SIM808 y las coordenadas recibidas, los datos fueron ingresados en la aplicación *Google Maps* de un dispositivo móvil, En la **Figura 3.41** se puede observar que hay una distancia de separación de 1 (m).



Figura 3.41 (A) Coordenadas GPS en *Google Maps* (B) Distancia entre los puntos (Fuente propia)

De acuerdo con las pruebas efectuadas con el SIM808, se considera que, al energizar el módulo, se debe esperar un tiempo hasta que se conecte con los diferentes satélites antes de solicitar los datos GPS; también es importante que el módulo esté fuera de cualquier infraestructura para disminuir el margen de error.

Pruebas de la alimentación eléctrica del sistema (PCB)

Con el fin de verificar las pruebas de alimentación eléctrica, se conectó todo el sistema con la batería (ver **Figura 3.42**) y se procedió a medir el voltaje de los módulos usando un multímetro.



Figura 3.42 Implementación del proyecto en una placa de pruebas (Fuente propia)

Por ejemplo, en la **Figura 3.43** se puede visualizar la medición del módulo SIM808, el mismo que tiene un voltaje de 7.38 (V).

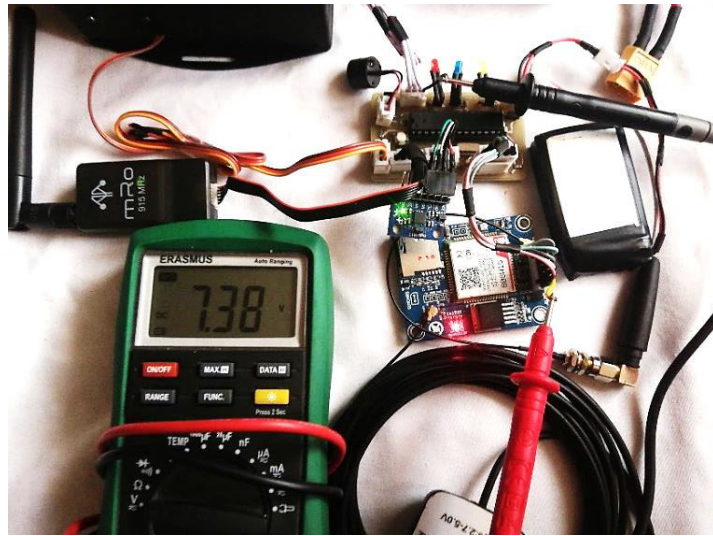


Figura 3.43 Medición del voltaje del módulo SIM808 (Fuente propia)

En la **Tabla 3.10** se puede visualizar los datos de voltaje teórico y experimental de los módulos.

Tabla 3.10 Voltajes experimentales y teóricos de los módulos del proyecto

Dispositivo	Voltaje Teórico (V)	Voltaje Experimental (V)
SIM808	7.4	7.38
Sensor IMU GY-521	5	5.05
Micro servomotor	5	5.17
Radio frecuencia mRo	5	5.06
<i>Buzzer</i>	5	5.16
Relé	5	5.11
Atmega 328P	5	5.10

Con las pruebas de alimentación eléctrica realizadas al sistema, se verificó que el voltaje de alimentación de los dispositivos es el requerido de acuerdo con las especificaciones técnicas.

Pruebas de los enlaces de comunicación

Con la finalidad de verificar la comunicación y transferencia de datos de los enlaces del sistema con el dispositivo móvil, se procedió a realizar varias pruebas, para lo cual se utilizó el sistema (bloque 1 y 3) y un dispositivo móvil como se observa en la **Figura 3.44**

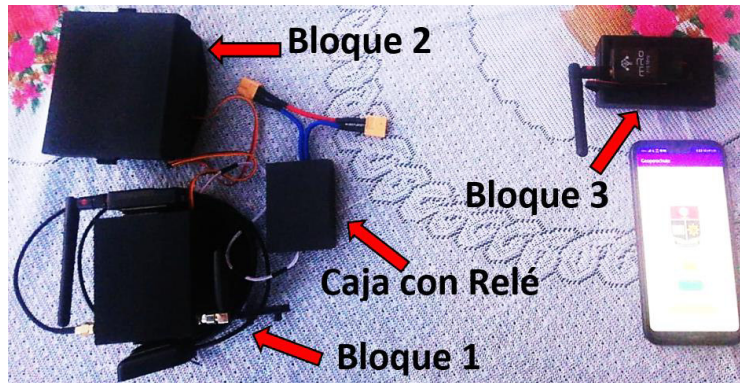


Figura 3.44 Equipos del proyecto (Fuente propia)

En el primer caso, se procede a probar la comunicación de los módulos RF, para lo cual se procedió a encender el bloque 3 y enlazar vía BT con el dispositivo móvil, en cambio, el bloque 1 y 2 se enlaza automáticamente siempre y cuando exista línea de vista entre los equipos RF.

En la **Figura 3.45** se pueden observar los resultados obtenidos en la aplicación móvil después de solicitar los datos al sistema (bloque 1) por el enlace de radiofrecuencia de los módulos RF y BT.



Figura 3.45 Pruebas de comunicación RF del proyecto (Fuente propia)

En el segundo caso, se procede a probar el enlace de comunicación GSM, en la aplicación móvil se ingresó el número del Micro SIM del sistema y del dispositivo, después se procedió a solicitar los datos al bloque 1, los resultados obtenidos se pueden observar en la **Figura 3.46**.



Figura 3.46 Pruebas de comunicación del enlace GSM del proyecto (Fuente propia)

Con el propósito de validar los datos que se presenta en la **Figura 3.46**, en la **Figura 3.47** se puede observar cada mensaje de texto recibido.



Figura 3.47 Mensajes de texto recibidos en el dispositivo móvil (Fuente propia)

En los dos casos anteriores después de solicitar las coordenadas GPS, se procede a localizar en el mapa de la aplicación OsmAnd y en el *Google Maps* como se observa en la **Figura 3.48**.

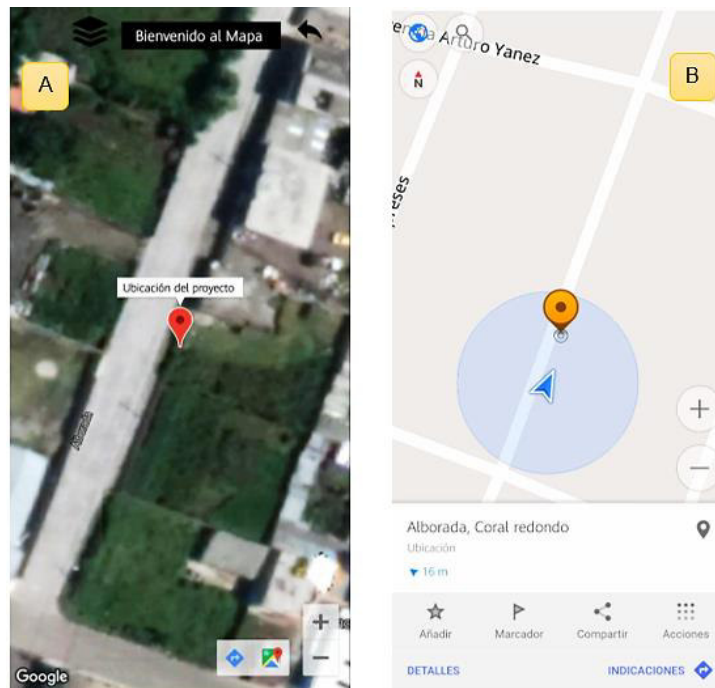


Figura 3.48 (A) Localización en *Google Maps* (B) Localización en el mapa OsmAnd (Fuente propia)

De acuerdo con las pruebas realizadas de comunicación, se puede destacar que el enlace con los módulos RF tiene una rápida respuesta en comparación con el enlace GSM, esto se debe a la distancia y el procesamiento de datos que realiza el servicio de mensaje de texto.

Pruebas con el sistema del paracaídas

Con el propósito de verificar el funcionamiento de la expulsión del paracaídas de forma manual y automática, se consideró realizar varias pruebas en tierra del sistema montado en el UAV, se procede a ubicar el sistema en el centro de la aeronave y es asegurado con cinta elástica como se observa en la **Figura 3.49**.

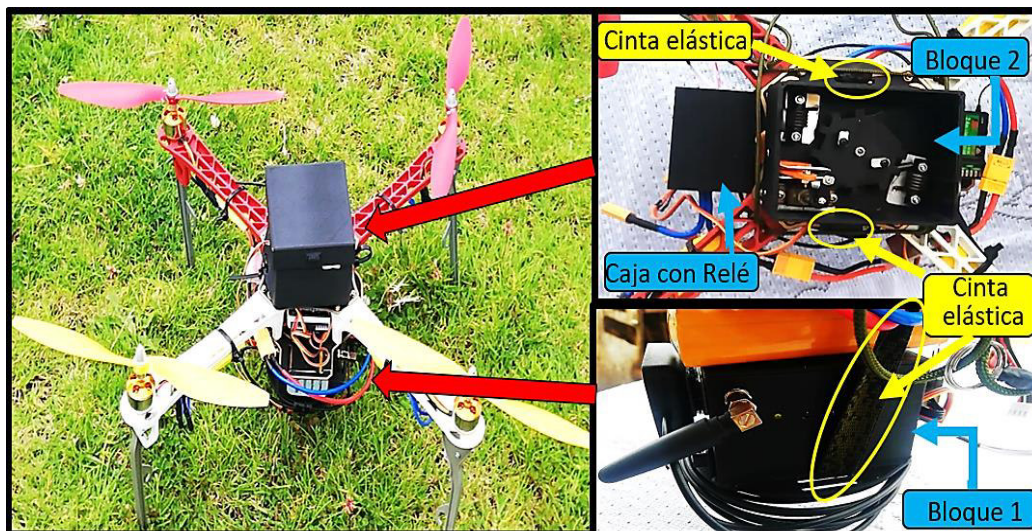


Figura 3.49 Sistema ubicado en el cuadricoptero (Fuente propia)

En el primer caso, se procede a expulsar el paracaídas de forma manual, simulando que la aeronave está realizando un vuelo se energiza el UAV y con el radiocontrol de la aeronave se incrementa la velocidad de giro de los motores, posteriormente utilizando la aplicación móvil se solicita expulsar el paracaídas, por lo tanto, el sistema procede a suprimir el suministro de energía del cuadricoptero y después de un corto periodo se procede a expulsar el paracaídas como se observa en la **Figura 3.50**.



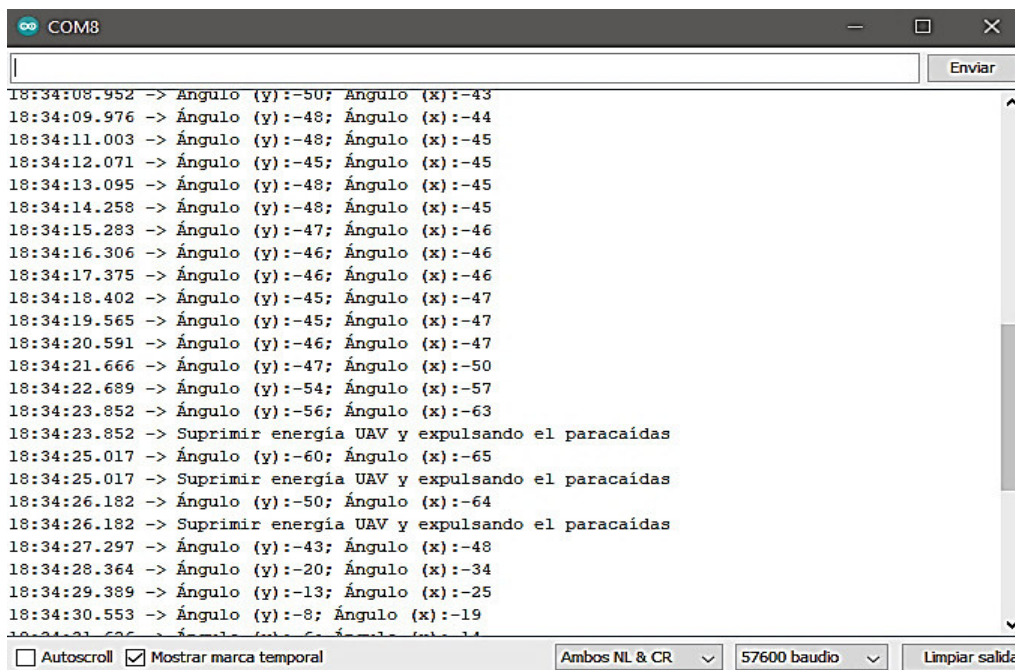
Figura 3.50 Expulsión de forma manual el paracaídas (Fuente propia)

El segundo caso para verificar la activación de forma automática se procede a girar la aeronave al punto de sobrepasar el umbral de 60° , el sistema activa el relé para suprimir el suministro de energía del UAV y después de un periodo corto se expulsa el paracaídas como se observa en **Figura 3.51**.



Figura 3.51 Expulsión de forma automática del paracaídas (Fuente propia)

En la **Figura 3.52** se puede observar la lectura del sensor IMU, el sistema al identificar el ángulo de giro de la aeronave y al sobrepasar el umbral se ejecuta la actividad de suprimir la energía del UAV y la expulsión del paracaídas.



```
COM8
18:34:08.952 -> Ángulo (y):-50; Ángulo (x):-43
18:34:09.976 -> Ángulo (y):-48; Ángulo (x):-44
18:34:11.003 -> Ángulo (y):-48; Ángulo (x):-45
18:34:12.071 -> Ángulo (y):-45; Ángulo (x):-45
18:34:13.095 -> Ángulo (y):-48; Ángulo (x):-45
18:34:14.258 -> Ángulo (y):-48; Ángulo (x):-45
18:34:15.283 -> Ángulo (y):-47; Ángulo (x):-46
18:34:16.306 -> Ángulo (y):-46; Ángulo (x):-46
18:34:17.375 -> Ángulo (y):-46; Ángulo (x):-46
18:34:18.402 -> Ángulo (y):-45; Ángulo (x):-47
18:34:19.565 -> Ángulo (y):-45; Ángulo (x):-47
18:34:20.591 -> Ángulo (y):-46; Ángulo (x):-47
18:34:21.666 -> Ángulo (y):-47; Ángulo (x):-50
18:34:22.689 -> Ángulo (y):-54; Ángulo (x):-57
18:34:23.852 -> Ángulo (y):-56; Ángulo (x):-63
18:34:23.852 -> Suprimir energía UAV y expulsando el paracaídas
18:34:25.017 -> Ángulo (y):-60; Ángulo (x):-65
18:34:25.017 -> Suprimir energía UAV y expulsando el paracaídas
18:34:26.182 -> Ángulo (y):-50; Ángulo (x):-64
18:34:26.182 -> Suprimir energía UAV y expulsando el paracaídas
18:34:27.297 -> Ángulo (y):-43; Ángulo (x):-48
18:34:28.364 -> Ángulo (y):-20; Ángulo (x):-34
18:34:29.389 -> Ángulo (y):-13; Ángulo (x):-25
18:34:30.553 -> Ángulo (y):-8; Ángulo (x):-19
18:34:31.626 -> Ángulo (y):-6; Ángulo (x):-14
18:34:32.700 -> Ángulo (y):-5; Ángulo (x):-13

 Autoscroll  Mostrar marca temporal
Ambos NL & CR 57600 baudio Limpiar salida
```

Figura 3.52 Lectura del ángulo de orientación de la aeronave (Fuente propia)

Con las pruebas ejecutadas de la expulsión del paracaídas, se puede mencionar que el umbral preseleccionado está en un rango adecuado para la activación automática del paracaídas, teniendo en cuenta que el sistema es destinado para las aeronaves que no realicen giros bruscos, en comparación a los drones que tienen aplicación para carreras de velocidad o FPV.

Con el fin de una mejor apreciación de las pruebas realizadas con el sistema del paracaídas, se consideró la elaboración de un video demostrativo, el mismo es accesible mediante el siguiente código QR de la **Figura 3.53**.



Figura 3.53 Código QR de las pruebas con el sistema del paracaídas

Pruebas del prototipo con el UAV

Con la finalidad de verificar el funcionamiento del sistema montado en el UAV, se procedió a realizar varias pruebas en el exterior y se utilizaron los equipos que se observa en la **Figura 3.54**.

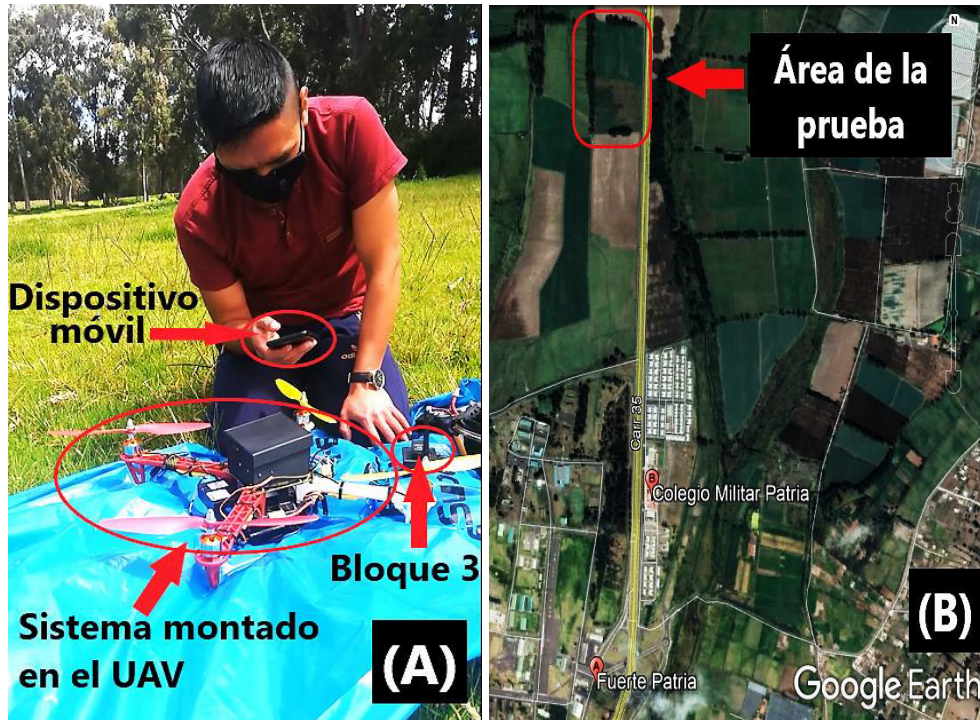


Figura 3.54 (A) Dispositivos utilizados en las pruebas (B) Área de la prueba
(Fuente propia)

En el primer caso se verifica la distancia de comunicación del enlace RF del sistema (bloque 1 y 3) con el dispositivo móvil, el usuario (Punto B) con línea de vista al sistema (Punto A), el usuario utilizando el dispositivo móvil procedió a solicitar los datos de información y activar el *buzzer*, en este caso los dos puntos se encuentran a una distancia de 340.34 (m) como se observa en la **Figura 3.55**



Figura 3.55 Distancia recorrida para la prueba de comunicación con línea de vista
(Fuente propia)

En la **Figura 3.57** se pueden observar los datos solicitados por el usuario al sistema y la activación del led indicador de color azul notificando que la solicitud de datos es a través del enlace RF.

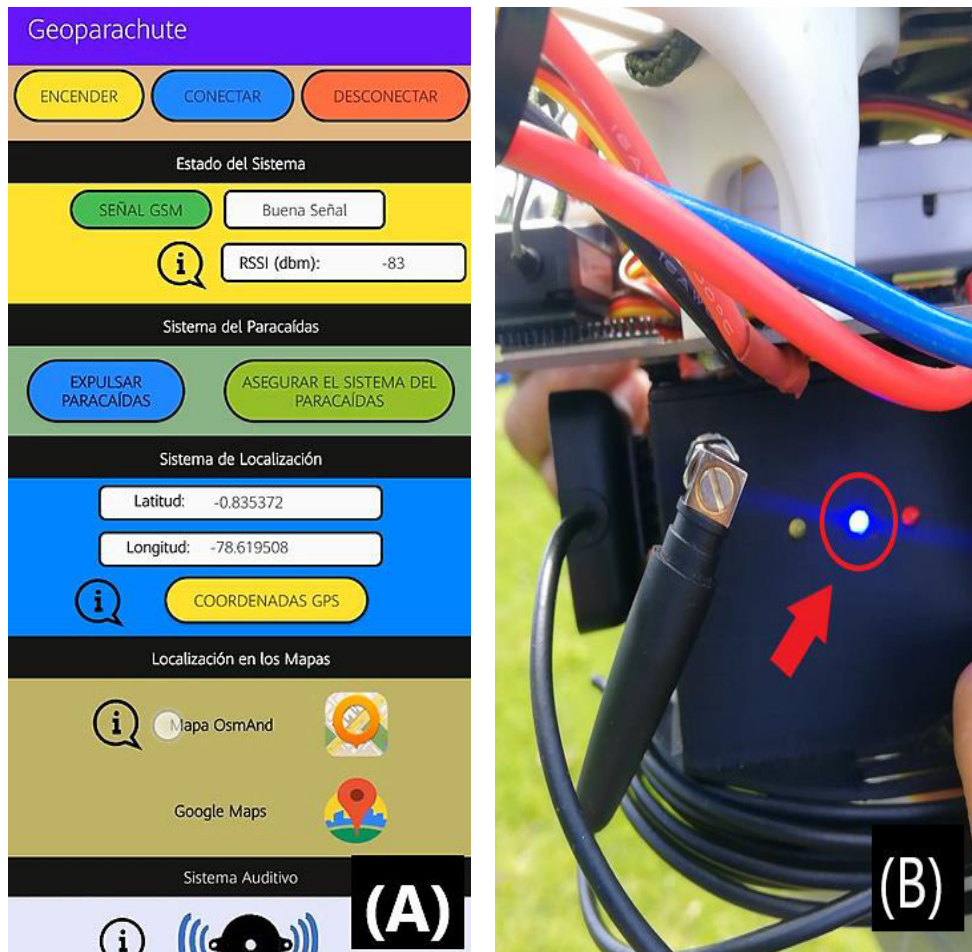


Figura 3.56 (A) Datos solicitados por el enlace RF (B) Led indicador del enlace RF
(Fuente propia)

En la **Figura 3.57** se puede evidenciar la localización del sistema en los mapas de disponibles de la aplicación móvil.



Figura 3.57 (A) Ubicación del sistema en *Google Maps* (B) Ubicación en *OsmAnd*
(Fuente propia)

En el segundo caso, sin línea de vista se comprueba el enlace de la red GSM, para esto el usuario se ubica detrás de los árboles que se encuentra en el área de la prueba, utilizando la aplicación se procede a solicitar la información y el sistema enciende el led indicador como se observa en la **Figura 3.58**.

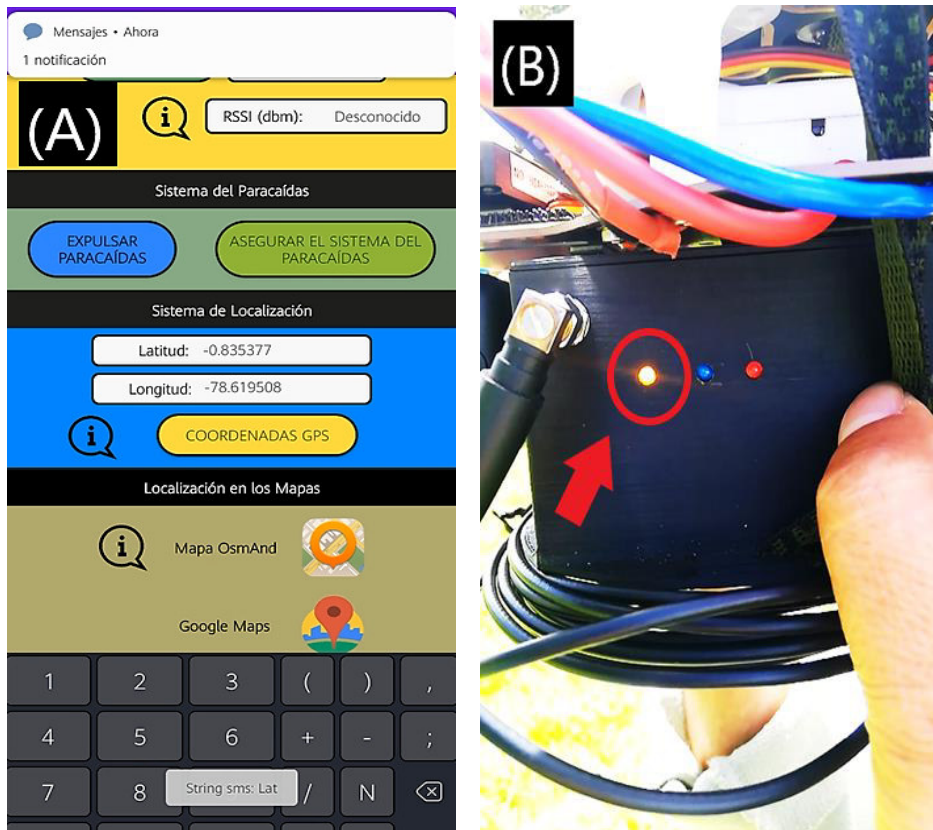


Figura 3.58 (A) Datos solicitado por el enlace GSM (B) Led indicador del enlace GSM (Fuente propia)

En el tercer caso se procede a comprobar la expulsión y apertura del paracaídas del sistema durante se realiza un vuelo con el UAV, para lo cual la aeronave es situada en el suelo como se observa en la **Figura 3.59**.



Figura 3.59 Prueba en campo del paracaídas (Fuente propia)

El piloto con el radio control eleva al UAV a una altura considerable (ver **Figura 3.60**) para permitir el despliegue del paracaídas del sistema, el usuario utilizando el dispositivo móvil procede a solicitar la expulsión del paracaídas y la aeronave empieza a descender como se observa en la **Figura 3.61**.

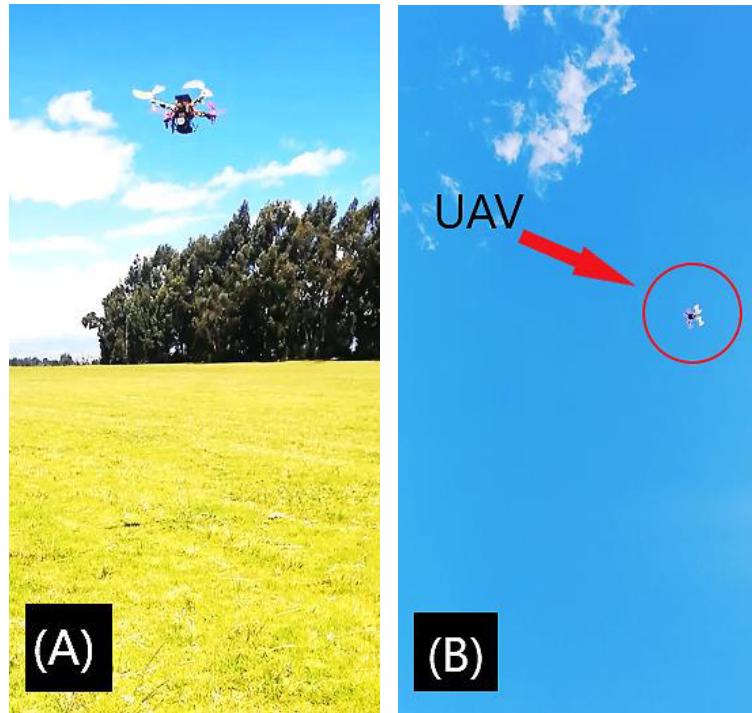


Figura 3.60 (A) Elevación del UAV (B) UAV previo a la expulsión del paracaídas
(Fuente propia)



Figura 3.61 UAV descendiendo con el paracaídas (Fuente propia)

Con el propósito de una mejor percepción de las pruebas en campo del sistema montado en el UAV, se consideró la elaboración de un video demostrativo, el mismo es accesible mediante el siguiente código QR de la **Figura 3.62**.



Figura 3.62 Código QR de las pruebas en campo del sistema montado en el UAV

Dentro del proyecto se han utilizado varios componentes, el cual conlleva a un costo total aproximado de 275.24 dólares y el desglose se lo presenta en el Anexo 2. Es necesario aclarar que no se considera los costos por el diseño, códigos de programación de la aplicación móvil y mano de obra.

3.6 Manual de Uso y Mantenimiento

Con el fin de revisar a detalle el uso del prototipo auxiliar, se recomienda visualizar el video explicativo, el cual es direccionado a través del código QR que se observa en la **Figura 3.63**.



Figura 3.63 Código QR para el manual de uso del proyecto

En la **Figura 3.64** se puede observar el código QR, el mismo permite redireccionar al video de mantenimiento que se debe considerar para el sistema.



Figura 3.64 Código QR del manual de mantenimiento del proyecto

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- En base al diseño e implementación del paracaídas, se puede destacar que los parámetros más relevantes para el diseño es la cantidad de masa que debe soportar, la velocidad del descenso y el coeficiente de arrastre del plástico, los mismos que fueron considerados en este proyecto y se obtuvieron buenos resultados.
- Luego de realizar las pruebas en campo con el enlace de comunicación redundante (enlace GSM y RF) del proyecto, se concluye que estos enlaces son eficientes, pero el tiempo de respuesta y la distancia efectiva de estos equipos puede variar de acuerdo con las circunstancias que son sometidos en el exterior, por ejemplo: con línea de vista el enlace RF tiene un alcance efectivo de 340 (m) y sin línea de vista puede existir comunicación a 20 (m), con el enlace GSM se demoró 5 minutos la respuesta al estar con un nivel de intensidad de la señal recibida de -83 (dBm).
- Los datos GPS recibidos por el sistema de geolocalización del proyecto son aceptables dentro de los parámetros de fábrica, estos datos pueden tener un margen de error de 1 (m) como se evidenció en las pruebas realizadas, pero se debe a que el SIM808 no es un módulo únicamente para la aplicación GPS; también, cumple la función de la conectividad con la red celular.
- En este proyecto con el propósito de prevenir que las placas electrónicas se dañen al energizar de forma inversa, se utilizó un diodo rectificador para prevenir este problema, el cual en base a sus características técnicas permite proteger a

los circuitos electrónicos porque solo deja pasar el flujo de corriente en una dirección.

- Luego de ejecutar las pruebas con el sistema del paracaídas, se obtuvo que este sistema cumple con las expectativas de expulsar el paracaídas a una gran distancia, considerando que el diseño y la selección del grosor del alambre del resorte se lo realizó de forma empírica y también se consideró las dimensiones de la caja para que los mismos se acoplen sin dificultad.
- Es importante considerar que el sensor IMU GY-521 presenta varia información (temperatura, ángulo de rotación y velocidad del objeto), pero para este proyecto solo se consideró los ángulos de Euler, para disponer de estos ángulos se utilizó algunas fórmulas matemáticas donde se incluye el filtro complementario, por último, se puede destacar que este filtro permite reducir el ruido de las muestras del sensor.
- Con base en las pruebas en campo del paracaídas, se concluye que el diseño del paracaídas permite al UAV descender adecuadamente después de la apertura completa del plástico, en este caso el paracaídas permitió precautelar el bienestar de los equipos de la aeronave.
- En el diseño de la aplicación móvil para este proyecto se consideró el programa Android Studio en comparación al programa App inventor, esto se debe porque el programa permitió crear una aplicación móvil con un mejor acabado y, además, añadir algunas características novedosas, por ejemplo: en el mismo interfaz de la aplicación se puede observar los datos recibidos de los mensajes de texto y el uso del mapa de *Google* para la localización del sistema, estas características fueron posibles porque se puede agregar estos permisos al momento de programar la aplicación.

4.2 Recomendaciones

- El proyecto tiene un peso aproximado de 500 (g), por lo tanto, antes de montar el sistema en un UAV, se recomienda verificar cual es el peso que la aeronave puede soportar durante un vuelo de rutina, ya que existen vehículos aéreos que pueden levantar un peso superior al 1.5 (Kg) y se debe considerar que el paracaídas está diseñado para soportar el peso que se calculó en este trabajo, por lo tanto, al incrementar esta variable se debe modificar el diámetro del paracaídas.
- En el diseño y la implementación del prototipo auxiliar para la caída y localización de un UAV, se consideró que todo el sistema sea lo más ligero posible, pero al

utilizar el arreglo de baterías Li-Po se incrementó en cierta medida el peso del sistema, por lo tanto, para proyectos futuros se recomienda obtener el suministro de energía del sistema desde la misma fuente de alimentación de la aeronave.

- La comunicación redundante de este proyecto permite disponer de un enlace fiable entre el sistema con el usuario, pero se recomienda antes de utilizar el enlace de la red GSM verificar el nivel de intensidad de la señal captada por la antena del SIM808 porque actualmente en el Ecuador existen áreas donde la cobertura de la red celular es deficiente.
- El sistema para suprimir y habilitar el suministro de energía del UAV previo a la expulsión del paracaídas está destinado para aeronaves que dispongan de los conectores XT60, por lo tanto, se recomienda verificar el tipo del conector de la fuente de energía de la aeronave, en el caso que sea necesario, se debería cambiar por el conector correspondiente.
- La plataforma de programación de Arduino provee a los usuarios diferentes librerías para facilitar el uso de los módulos, pero en proyectos donde se utiliza varios módulos, se recomienda verificar las restricciones que cada librería dispone con el fin de evitar conflictos en la funcionalidad del programa.
- Con el fin de prevenir una ruptura de los hilos del paracaídas causado por la tensión que se genera al descender con la aeronave, se recomienda utilizar el hilo tipo nailon, debido a su resistencia en relación con el grosor.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] G. A. Addati y G. Pérez Lance, «ECONSTOR,» 10 2014. [En línea]. Available: <https://www.econstor.eu/bitstream/10419/130802/1/799216895.pdf>. [Último acceso: 28 12 2019].
- [2] A. L. P. Gastón, «econstor.eu,» 01 02 2014. [En línea]. Available: <https://www.econstor.eu/bitstream/10419/130802/1/799216895.pdf>. [Último acceso: 30 03 2021].
- [3] Jorge Pantoja, José Barzallo, Esteban Bustamante, «Latitude Aerospace Solutions,» 1 10 2016. [En línea]. Available: <https://www.latitudeas.com/news>. [Último acceso: 22 10 2019].
- [4] P. Hernández, «filmora,» 15 09 2017. [En línea]. Available: <https://filmora.wondershare.com/es/drones/10-must-have-drones-accessories-and-parts.html>. [Último acceso: 04 11 2019].
- [5] E. Gómez, «droneproject,» droneproject., 31 10 3019. [En línea]. Available: <https://droneproject.org/aprender-a-volar-sin-arriesgar-tu-dron/>. [Último acceso: 08 04 2021].
- [6] Dronografía, «Dronografía,» Dronografía, 31 05 2017. [En línea]. Available: <https://dronografia.es/tecnologia1/204-tu-drone-se-ha-estrellado-que-hacer-ahora.html>. [Último acceso: 08 04 2021].
- [7] E. Crespo, «Aprendiendo Arduino,» 21 12 2016. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/tag/imu/>. [Último acceso: 13 03 2020].
- [8] L. E. Ardila Pérez y J. S. Eslava Garzón, 05 02 2012. [En línea]. Available: <file:///C:/Users/Roberth/Downloads/Dialnet-Disen%C3%B3CalibracionYPruebaDeUnaUnidadDeMedidaInercia-5038483.pdf>. [Último acceso: 09 04 2021].
- [9] A. El-Rabbany, 12 05 2002. [En línea]. Available: <https://books.google.es/books?hl=es&lr=&id=U2JmgHrrB8cC&oi=fnd&pg=PR>

- 13&dq=gps+global+positioning+system&ots=9NxYsOZCGS&sig=gPMNCy9nKXA72fRILxQR1XkNZ8Q#v=onpage&q&f=false. [Último acceso: 31 03 2021].
- [10] N. C. Office, «GPS.gov,» 6 Enero 2021. [En línea]. Available: <https://www.gps.gov/systems/gps/control/>. [Último acceso: 25 03 2021].
- [11] M. L. Avilés Avilés y J. C. Bajaña Gutiérrez, «ESPOL,» 04 05 2010. [En línea]. Available: <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/11611>. [Último acceso: 09 04 2021].
- [12] J. Sanders, «GSM favorite,» 05 07 2004. [En línea]. Available: <https://www.gsmfavorites.com/articles/>. [Último acceso: 01 04 2021].
- [13] L. Harte, «TECHNICAL ARTICLES,» 2005. [En línea]. Available: <https://www.globalspec.com/reference/66227/203279/introduction-to-global-system-for-mobile-communication-gsm>. [Último acceso: 01 04 2021].
- [14] C. JH, «Móvistar,» 20 08 2013. [En línea]. Available: [https://comunidad.movistar.es/t5/Soporte-Móvil/GSM-Cómo-funciona-la-telefonía-móvil/td-p/1383296#:~:text=Funcionamiento%20de%20la%20red%20GSM,\(de%20a%20hi%20su%20nombre\).&text=se%20le%20da%20cobertura%20al,en%20qué%20celda%20se%20encuentra..](https://comunidad.movistar.es/t5/Soporte-Móvil/GSM-Cómo-funciona-la-telefonía-móvil/td-p/1383296#:~:text=Funcionamiento%20de%20la%20red%20GSM,(de%20a%20hi%20su%20nombre).&text=se%20le%20da%20cobertura%20al,en%20qué%20celda%20se%20encuentra..) [Último acceso: 10 04 2021].
- [15] D. UC, «Definición y propósito de la Investigación Aplicada,» Duoc.c Biblioteca, [En línea]. Available: <http://www.duoc.cl/biblioteca/crai/definicion-y-proposito-de-la-investigacion-aplicada>. [Último acceso: 21 05 2020].
- [16] Info.dron, «Info.dron,» Info.dron, 26 12 2018. [En línea]. Available: <http://www.infodron.es/id/2018/12/26/noticia-estrena-sistema-paracaidas-evitar-accidentes.html>. [Último acceso: 23 03 2020].
- [17] K. Ono, «PARAZERO Drone Safety Systems,» 18 04 2018. [En línea]. Available: <https://parazero.com/products/>. [Último acceso: 23 03 2020].
- [18] HispaDrones, «HispaDrones,» 15 03 2019. [En línea]. Available: <https://www.hispadrones.com/profesionales/componentes/safeair-paracaidas-drones-parazero/>. [Último acceso: 02 12 2020].

- [19] ParaZero, «ParaZero,» 20 11 2020. [En línea]. Available: <https://parazero.com/products/safeair-for-dji-phantom/>. [Último acceso: 02 12 2020].
- [20] Skycat, «Skycat,» 11 08 2019. [En línea]. Available: <https://www.skycat.pro>. [Último acceso: 02 12 2020].
- [21] DJI, «DJI,» RTK Module, 04 2020. [En línea]. Available: <https://www.dji.com/mavic-2-enterprise-advanced>. [Último acceso: 10 04 2021].
- [22] DJI, «DJI,» DJI, 15 08 2016. [En línea]. Available: <https://www.dji.com/d-rtk/info>. [Último acceso: 10 04 2021].
- [23] DJI, «DJI,» 04 2019. [En línea]. Available: <https://www.dji.com/d-rtk-2>. [Último acceso: 11 04 2021].
- [24] J. Martín, «cerem,» cerem, 29 09 2018. [En línea]. Available: <https://www.aiteco.com/matriz-de-priorizacion/>. [Último acceso: 20 04 2021].
- [25] A. T. C. Ltd, «Smart-prototyping,» 28 02 2016. [En línea]. Available: https://www.smart-prototyping.com/image/data/9_Modules/101756%20Wireless%20A6C/A6_A7_A6C_datasheet-EN.pdf. [Último acceso: 15 05 2020].
- [26] SIMCom, «Smart Machine Smart Decision,» 27 03 2014. [En línea]. Available: https://cdn-shop.adafruit.com/datasheets/SIM808_Hardware+Design_V1.00.pdf. [Último acceso: 15 05 2020].
- [27] S. M. S. Descision, «SIMCom A company of SIM Tech,» 15 05 2016. [En línea]. Available: <https://download.mikroe.com/documents/datasheets/sim868-hardware-design-v1.000.pdf>. [Último acceso: 15 05 2020].
- [28] Acoptex, «Acoptex,» 16 03 2018. [En línea]. Available: <http://acoptex.com/project/286/basics-project-064a-ai-thinker-a7-gsm-gprs-gps-module-at-acoptexcom/#sthash.3OSKGAEx.dpbs>. [Último acceso: 15 05 2020].

- [29] A. Electronics, «APM Electronics,» APM Electronics, 05 2019. [En línea]. Available: <https://apmelectronica.com/producto/a7/>. [Último acceso: 15 05 2020].
- [30] O. Impulse, «Open Impulse,» 05 2019. [En línea]. Available: <https://www.openimpulse.com/blog/products-page/product-category/2-1-gsm-gps-sim808-module/>. [Último acceso: 02 12 2020].
- [31] R. S. Ltd, «rfsolutions,» rfsolutions, 10 10 2010. [En línea]. Available: <https://www.rfsolutions.co.uk/downloads/1502290810ANT-MSTUB-3.pdf>. [Último acceso: 20 05 2020].
- [32] I. Inc., «InvenSense Inc.,» 19 08 2013. [En línea]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. [Último acceso: 05 04 2021].
- [33] E. C. D. Search, «Search, Electronic Components Datasheet,» 14 02 2021. [En línea]. Available: https://www.electronicoscaldas.com/datasheet/MG90S_Tower-Pro.pdf. [Último acceso: 06 04 2021].
- [34] B. Daniel, «APUNTES DE ELECTRONICA - INFORMATICA - TECNOLOGIA,» 27 11 2019. [En línea]. Available: <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/moduloRele.pdf>. [Último acceso: 22 04 2021].
- [35] mRobotics.io, «mRobotics.io,» Mayan Robotics LLC, 10 01 2021. [En línea]. Available: <https://store.mrobotics.io/mRo-SiK-Telemetry-Radio-V2-915Mhz-p/m10013-rk.htm>. [Último acceso: 23 02 2021].
- [36] I. Studio, «Itead Studio,» 18 06 2010. [En línea]. Available: https://components101.com/asset/sites/default/files/component_datasheet/H C-05%20Datasheet.pdf. [Último acceso: 02 04 2021].
- [37] Pro-signal, «Farnell,» 02 11 2016. [En línea]. Available: <http://www.farnell.com/datasheets/2171929.pdf>. [Último acceso: 11 04 2021].

- [38] ATmel, «ATmel,» 09 05 2009. [En línea]. Available: https://www.mouser.com/pdfdocs/gravitech_atmega328_datasheet.pdf. [Último acceso: 12 04 2021].
- [39] Atmel, «Atmega,» 02 11 2016. [En línea]. Available: <https://www.electronicaplugandplay.com/sistemas-embedidos/product/306-atmega328p-pu-dip28>. [Último acceso: 10 04 2021].
- [40] D. R. Ojeda Pereira y L. F. Ortiz Pineda, «Repositorio de la Universidad Politécnica Salesiana sede Cuenca,» 02 01 2017. [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/13637/1/UPS-CT006930.pdf>. [Último acceso: 19 04 2021].
- [41] E. Pino, «Idesia (Arica),» Universidad Nacional Jorge Basadre Grohmann, Tacna, Perú, 01 03 2019. [En línea]. Available: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-34292019000100075#:~:text=Su%20desventaja%20es%20el%20gran,y%20en%20la%20inspección%20industrial.. [Último acceso: 20 04 2021].
- [42] M. Merino, «Aereocamaras,» Aereocamaras, 16 02 2020. [En línea]. Available: <https://dronehibrido.com/es/#:~:text=Dron%20Hibrido%20AeroHyb%2C%20el%20dron%20con%20más%20autonomía%20del%20mercado&text=Con%20capacidad%20para%20trasladar%20una,el%20mercado%20profesional%20de%20drones..> [Último acceso: 28 04 2021].
- [43] Energizer, «Energizer,» 01 04 2000. [En línea]. Available: <https://docs.rs-online.com/5d92/0900766b8172fd23.pdf>. [Último acceso: 13 05 2021].
- [44] J. M. R. Andrés, «Astroeduc,» 10 2018. [En línea]. Available: <http://www.astroeduc.com.ar/ManualCohetemodelismo.pdf>. [Último acceso: 22 03 2020].
- [45] M. Rey, «Neocities,» neocities.org, 05 2019. [En línea]. Available: <https://marceloreynolds.neocities.org/teoria.htm>. [Último acceso: 23 03 2020].
- [46] J. L. Fernández, «FisicaLab,» FisicaLab, 2018. [En línea]. Available: <https://www.fisicalab.com/apartado/caida-libre>. [Último acceso: 22 03 2020].

- [47] Skycat, «Skycat,» 2019. [En línea]. Available: <https://www.skycat.pro>. [Último acceso: 02 12 2020].
- [48] Alfonso, «Antena 3,» 17 11 2013. [En línea]. Available: https://www.antena3.com/formula-1/mundial-2013/eeuu/que-fuerza_2013111758b0387a0cf28e3b3a9dd03f.html. [Último acceso: 09 05 2020].
- [49] arduino, «arduino,» 31 12 2018. [En línea]. Available: <https://playground.arduino.cc/Main/MPU-6050/>. [Último acceso: 08 07 2019].
- [50] Tr4nsduc7or, «Roboldgs Tutoriales robots y humor cibernético,» robologs, 15 10 2014. [En línea]. Available: <https://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>. [Último acceso: 20 02 2020].
- [51] ITeadStudio, «Components101,» 18 06 2010. [En línea]. Available: https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf. [Último acceso: 23 02 2021].
- [52] «Agencia de regulación y control de las telecomunicaciones,» 04 04 2018. [En línea]. Available: https://www.arcotel.gob.ec/wp-content/uploads/2015/01/BOLETIN-ESTADISTICO-Diciembre-2018-v4_4.pdf. [Último acceso: 01 04 2021].
- [53] A. G. C, «Shoptronica,» Shoptronica, 09 01 2019. [En línea]. Available: <https://www.shoptronica.com/curiosidades-tutoriales-y-gadgets/1128-que-es-las-baterias-lipo-litio-0689593937322.html>. [Último acceso: 02 04 2021].
- [54] M. Merino, «Fotografía y Drones,» Fotografía y Drones, 16 02 2020. [En línea]. Available: <https://fotografiaydrones.com/nuevo-dron-con-2-horas-de-autonomia>. [Último acceso: 28 04 2021].

ANEXOS

ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO



ESCUELA POLITECNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 25 de junio del 2021

CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, *Alex Fabricio Oña Ñacata*, docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de titulación, certifico que he constatado el funcionamiento del prototipo auxiliar para la caída y localización de un UAV, el mismo que fue implementado por el estudiante Vega Escobar Roberth David.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para ser aplicado como prototipo de un UAV de ala rotatoria.

Una firma manuscrita en tinta azul, que parece ser la del director mencionado en el texto.

DIRECTOR

Ing. Alex Fabricio Oña Ñacata, Msc.

Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 28. EXT: 2729

email: alex.ona@epn.edu.ec

Quito-Ecuador

ANEXO 2: COSTO DEL PROYECTO

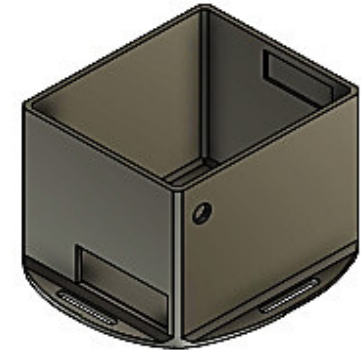
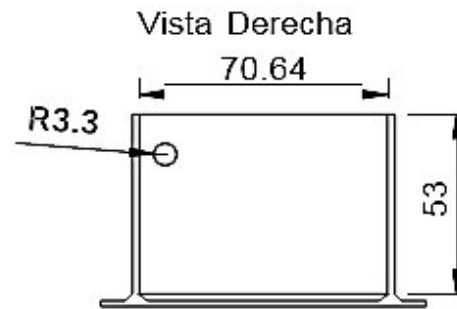
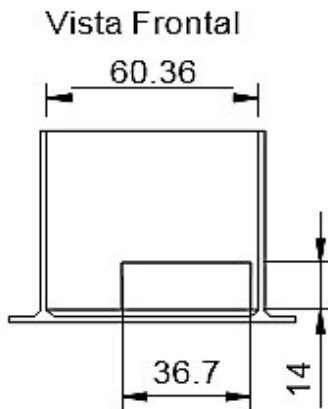
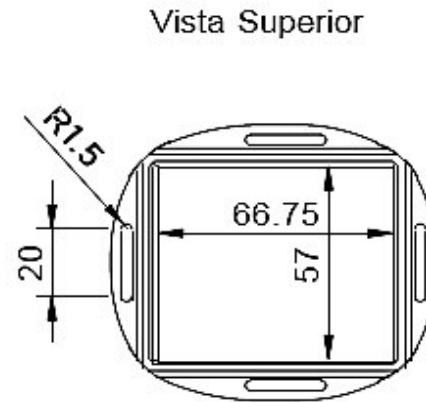
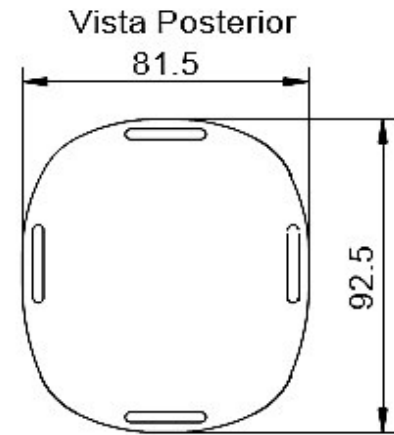
En esta sección se presenta el desglose del costo de los materiales empleados en la implementación del proyecto.

Detalle del Material	Cantidad	Precio (\$)	Total (\$)
Módulo SIM808	1	40	40
Equipos RF modelo mRo 915 (MHz)	1	60	60
Microcontrolador Atmega 328P	1	5	5
Módulo Relé de 1 canal	1	2	2
Módulo IMU GY-521	1	4	4
Micro servomotor MG-90S	1	5	5
Módulo <i>Bluetooth</i> HC-05	1	6	6
<i>Buzzer</i>	1	1	1
<i>Led</i> de 3 (mm)	3	0.1	0.3
Componentes SMD			
Resistencia 330 (Ω)	3	0.7	2.1
Resistencia 1K (Ω)	4	0.8	3.2
Capacitor 22 (pF)	2	0.5	1
Capacitor 100 (nF)	1	0.8	0.3
Capacitor 330 (nF)	1	0.8	0.3
Regulador AMS1117- 5 (V)	1	1	1
Diodo 1N4007	1	1	1
Componentes THT			
Resistencia 330 (Ω)	1	0.05	0.05
Cristal de 16 (MHz)	1	1.2	1,2
Capacitor 100 (nF)	1	0.3	0.6
Capacitor 330 (nF)	1	0.3	0.6
Regulador LM7805- 5 (V)	1	1	1
Diodo 1N4007	1	1	1
<i>Led</i> rojo	1	0.1	0.1
Conector XT-60	1	2	2
Conectores Molex de 2/3/5 pines con separación de 2.54 (mm)	6	0.3	1.8
Espadines tipo macho	1	1	1
Espadines tipo hembra	1	1	1

Cable AWG 24 de 1 (m)	1	0.4	0.4
Cable AWG 12 de 1 (m)	1	0.6	0.6
Tubo termoencogible 1 (m)	1	0.8	0.8
Batería Alcalina 9 (V)	1	4	4
Batería LI-Po 3.7 (V) 1200 (mA)	2	6	12
Cargador de baterías Li-Po TE3AC	1	30	30
Fabricación de la Baquelita			
Baquelita 6x36 (cm)	1	8.25	8.25
Baquelita 4.8x2.8 (cm)	1	7.5	7.5
Impresión en 3D			
Caja del bloque 1	1	17	17
Caja del bloque 2	1	18	18
Caja prefabricada del bloque 3	1	2.5	2.5
Resortes de 2 (mm)	2	1.5	3
Rollo de plástico de polietileno	1	2.74	2.74
Rollo de hilo nailon	1	3	3
Cinta elástica	1	1	1
Velcro adhesivo	2	3	6
Cordin de nailon (1m)	1	0.5	0.5
Micro SIM	1	5	5
Varios (estaño, pomada, tornillos, tuercas, etc.)	1	10	10
Mano de obra de 240 horas (tiempo del diseño y fabricación de las cajas, paracaídas, placas electrónicas, aplicación móvil)	1	500	500
Total			775.24

ANEXO 3: PLANOS DE LA CAJA DEL BLOQUE 1

En esta sección se presenta los planos de la caja del bloque 1 diseñados en el programa *Autodesk Fusión 360* versión 2.0



Escuela Politécnica Nacional
ESFOT

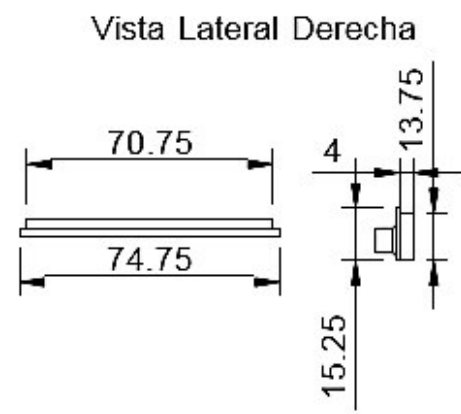
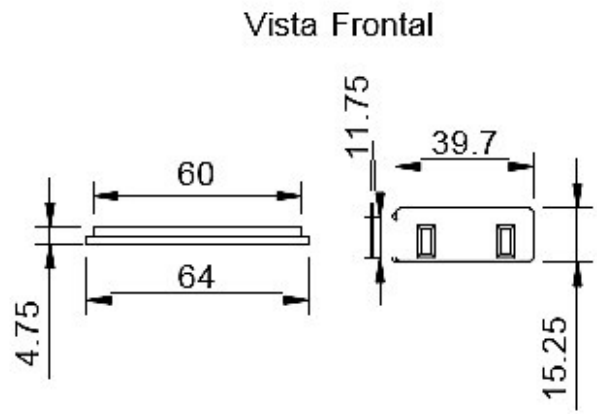
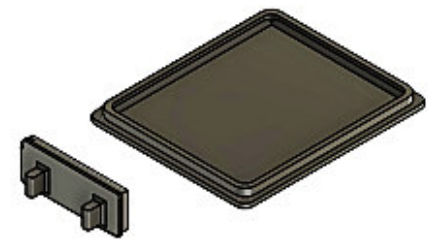
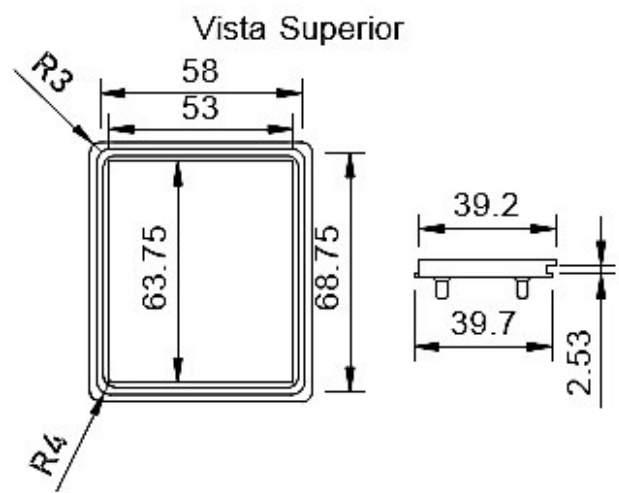
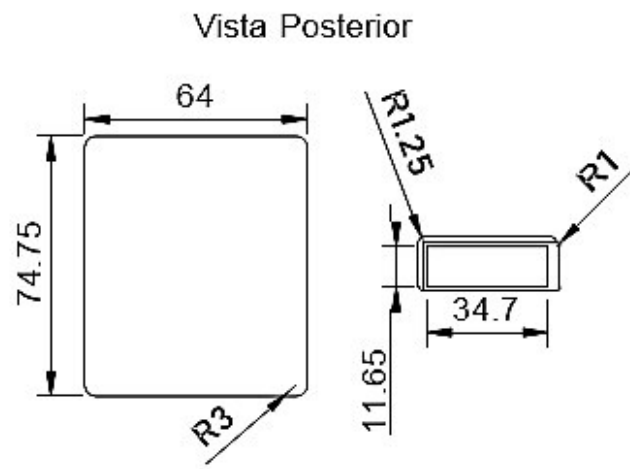
Título del modelo: Caja del bloque 1

Fecha: 04/07/2021

Autor: Roberth Vega

Escala: 1:2

No. 1/3



Escuela Politécnica Nacional
ESFOT

Título del modelo: Tapas del bloque 1

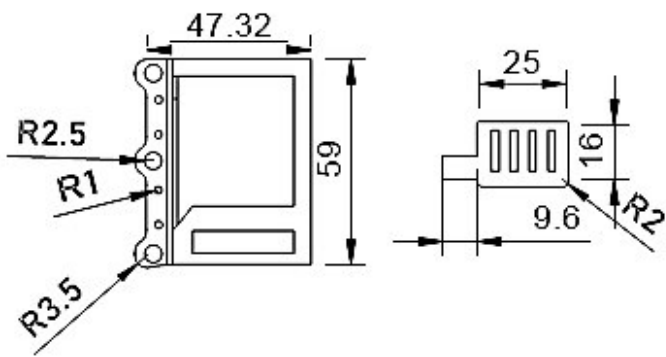
Fecha: 04/07/2021

Autor: Roberth Vega

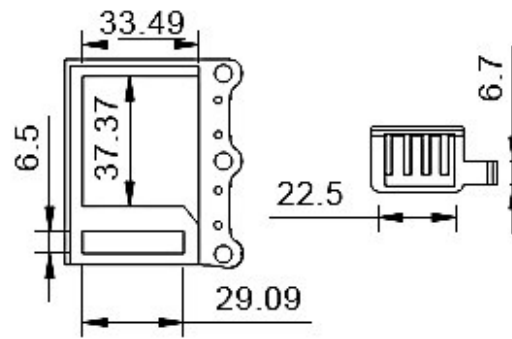
Escala: 1:2

No. 2/3

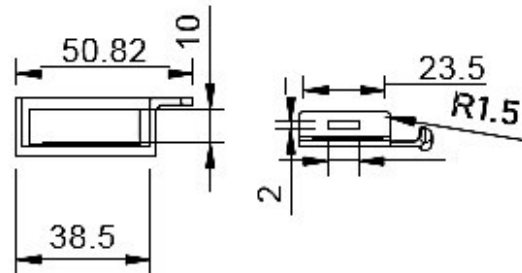
Vista Posterior



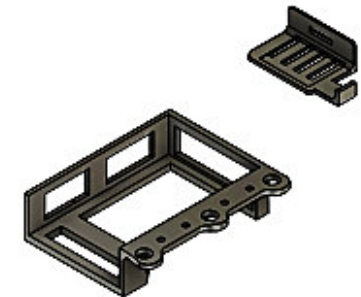
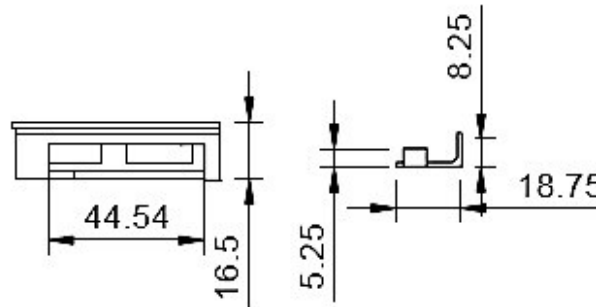
Vista Superior



Vista Frontal



Vista Lateral Derecha



Escuela Politécnica Nacional
ESFOT

Título del modelo: Bases del bloque 1

Fecha: 04/07/2021

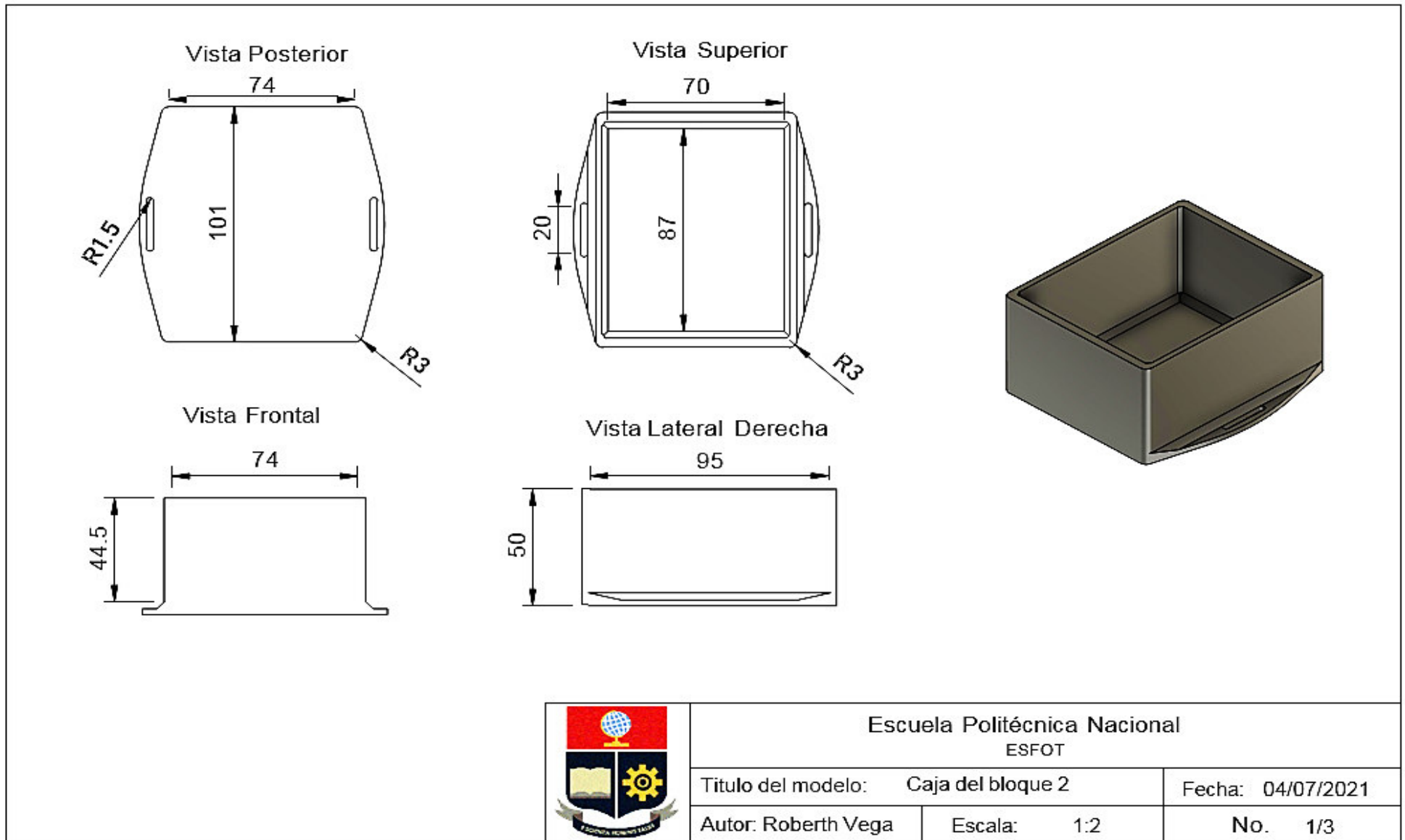
Autor: Roberth Vega

Escala: 1:2

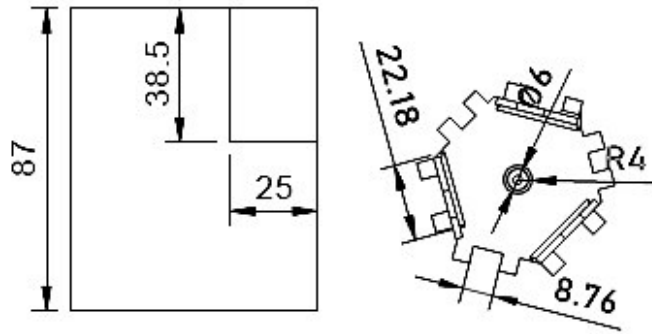
No. 3/3

ANEXO 4: PLANOS DE LA CAJA DEL BLOQUE 2

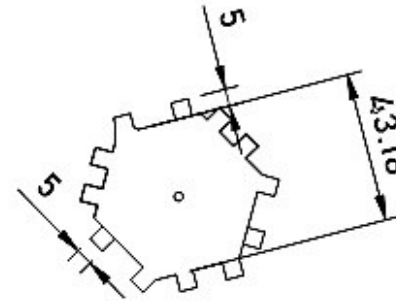
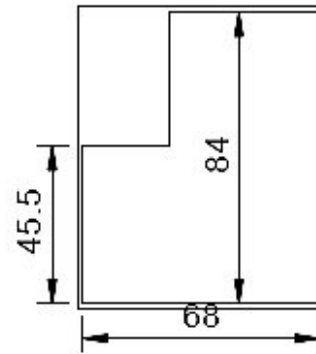
En esta sección se presenta los planos de la caja del bloque 2 diseñados en el programa *Autodesk Fusión 360* versión 2.0



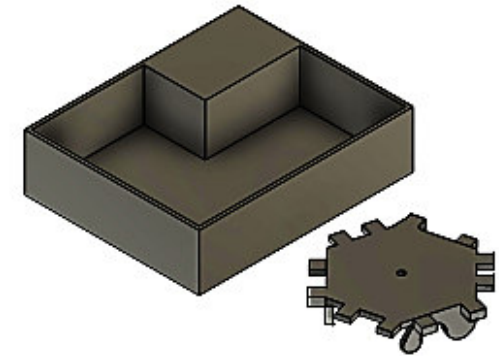
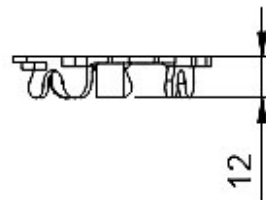
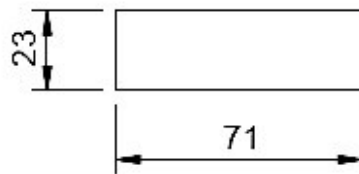
Vista Posterior



Vista Superior



Vista Frontal



Escuela Politécnica Nacional
ESFOT

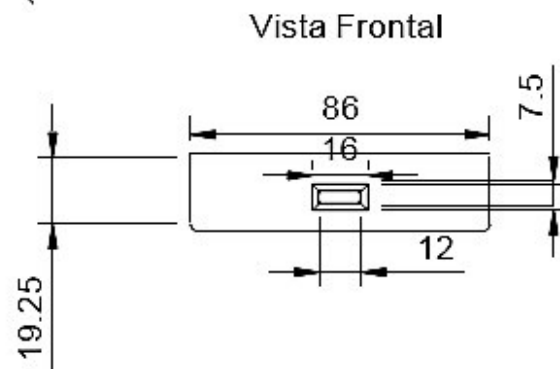
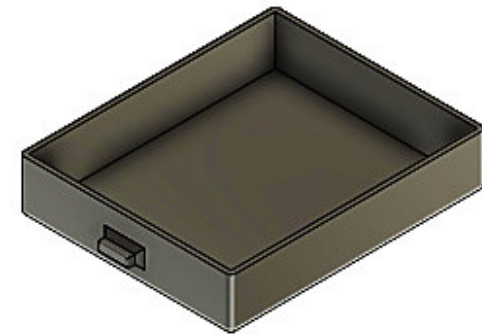
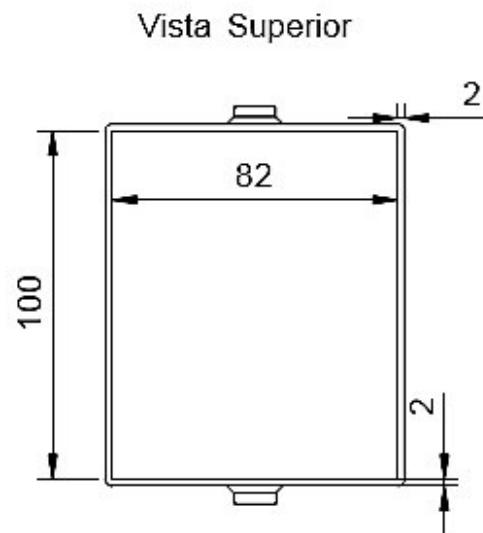
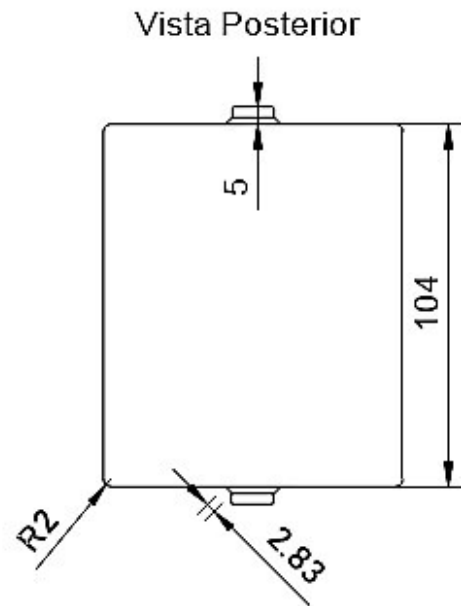
Título del modelo: Bases del bloque 2

Fecha: 04/07/2021

Autor: Roberth Vega

Escala: 1:2

No. 2/3



Escuela Politécnica Nacional
ESFOT

Título del modelo: Tapa del bloque 2

Fecha: 04/07/2021

Autor: Roberth Vega

Escala: 1:2

No. 3/3

ANEXO 5: CÓDIGO DE PROGRAMACIÓN DEL SISTEMA

En el siguiente anexo se presenta el código de programación para el funcionamiento del sistema del microcontrolador, el mismo que fue diseñado en el programa *Arduino IDE* versión 1.8.13. El código permite controlar las diferentes funciones del proyecto, por ejemplo: lectura del sensor IMU, activación y desactivación de los actuadores (relé y servomotor), solicitud de la localización GPS del SIM808, control de la recepción y envío de los mensajes de texto del SIM808 y la transferencia de datos por el módulo RF.

```
//***** Declaración de librerías *****
#include <SoftwareSerial.h> //Declaración de la librería para la comunicación serial de forma
virtual.
#include <GY521_IMU.h> //Declaración de la librería del sensor IMU modelo GY-521.
#include <DFRobot_sim808.h> //Declaración de la librería del módulo sim808.

//***** Definición de pines y de la clase de las librerías *****
#define PIN_TX 7 //Declaración del pin de transmisión.
#define PIN_RX 8 //Declaración del pin de recepción.
SoftwareSerial mySerial(PIN_TX, PIN_RX); //Establecimiento de los pines de transmisión de la
comunicación serial.
DFRobot_SIM808 sim808(&mySerial); //Conexión serial del módulo sim808.
GY521_IMU sensor; //Definición de la clase del sensor GY-521.

//***** Declaración de variables globales *****
int valoresY; //Variable para el valor del ángulo alrededor del eje (y).
int valoresX; //Variable para el valor del ángulo alrededor del eje (x).
int Rng = 60; //Ángulo predefinido para la activación del sistema del paracaídas.

unsigned long TimerA, TimerB, TimerC; //Variable para los timers
int Periodo2 = 1000, Periodo1 = 100; //Perido de tiempo (sg).

float pausa; //Variable para establecer la pausa del micro servomotor.

#define longitud_sms 160 //Definición de la longitud de los caracteres del mensaje recibidos.
char message[longitud_sms]; //Variable para almacenar los caracteres del mensaje.
int messageIndex = 0; //Variable de índice de los mensajes.
```

char phone[16]; //Variable para almacenar el número del teléfono móvil del mensaje de texto recibido.

char datetime[24]; //Variable para almacenar la fecha y hora del mensaje de texto recibido.

unsigned long leer; //Variable para guardar los datos captados por el módulo sim808.

String cadena, strDato, strDato2, strDato3, strDato4, sms, sms1; //Variable para guardar los datos captados por el módulo sim808.

String datoA, datoB, datoC; //Variables para la lectura de los datos del sim808.

unsigned long _tActual, _tEsperado = 2000; //Variable de periodos de tiempo.

char trama[80], cDato[12], cLon[11], cLat[11], DATO; //Variable para guardar los datos captados por el módulo sim808 y el módulo RF.

boolean activador = false; //Variable para reconocer la lectura de datos.

int Rele = A0; //Variable para definir el pin del relé.

int buzz = 4; //Variable para definir el pin del indicador auditivo (Buzzer).

int tx = 1; //Variable para el pin de transmisión (Tx) para el módulo RF.

int rx = 0; //Variable para el pin de recepción (Rx) para el módulo RF.

int ledA = 13; //Variable para el pin del indicador visual para la configuración global.

int ledB = 12; //Variable para el pin del indicador visual para la comunicación por RF.

int ledC = 11; //Variable para el pin del indicador visual para la comunicación por la red GSM.

//***** Configuración de los Periféricos *****

void setup() {

mySerial.begin(57600); //Definición de la velocidad de transmisión de datos en baudios.

Serial.begin(57600); //Definición de la velocidad de transmisión de datos en baudios.

pinMode (Rele, OUTPUT); //Declaración del pin digital como salida de datos.

pinMode (buzz, OUTPUT); //Declaración del pin digital como salida de datos.

pinMode (ledA, OUTPUT); //Declaración del pin digital como salida de datos.

pinMode (ledB, OUTPUT); //Declaración del pin digital como salida de datos.

pinMode (ledC, OUTPUT); //Declaración del pin digital como salida de datos.

pinMode (tx, OUTPUT); //Declaración del pin digital como salida de datos.

pinMode (rx, INPUT); //Declaración del pin digital como entrada de datos.

digitalWrite (ledA, HIGH); //Activación del led indicador de la configuración de los módulos.

mySerial.println("AT+CGNSPWR=1"); //Habilitar el suministro de energía de la antena GPS del módulo SIM808.


```

delay(2000); //Retardo de 2 (s).
mySerial.println(" AT+CGPSRST=0"); //Iniciar en modo frío la antena GPS.
delay(15000); //Retardo de 2 (s).
mySerial.println("AT+CGNSSEQ=RMC"); //Definición de la trama GPS como RCM.
delay(15000); //Retardo de 2 (s).

digitalWrite(Rele, LOW); //Activación del relé con lógica inversa.
connectionGPS (); //Función para la conexión de la antena GPS con los satélites globales.
sensor.config_imu(); //Configuración básica de la IMU.
digitalWrite (ledA, LOW); //Desactivación del led indicador de la configuración básica de los
módulos.
}

//***** Programa Principal *****
void loop() {
IMU:
digitalWrite (ledB, LOW); //Desactivación del led indicador.
digitalWrite (ledC, LOW); //Desactivación del led indicador.

//***** Detectar si hay mensajes no leídos *****
messageIndex = sim808.isSMSunread(); //Función para reconocer la existencia de los
mensajes de texto no leídos.
// Serial.print("Mensajes recibidos: ");
// Serial.println(messageIndex);

//***** Lectura de datos del sensor IMU *****
if (millis () >= TimerA + Periodo1) {
TimerA = millis ();
sensor.lectura_IMU(); //Lectura de los datos de la IMU.
valoresY = sensor.angulo_pitch(); //Obtención del ángulo alrededor del eje (y) denominado
pitch de la IMU.
valoresX = sensor.angulo_roll(); //Obtención del ángulo alrededor del eje (x) denominado roll
de la IMU.
// Serial.print("Ángulo (y):" );
// Serial.print(valoresY);
// Serial.print(" ; Ángulo (x):" );
// Serial.println(valoresX);

```

```
}
```

```
//***** Reconocimiento del ángulo de orientación y expulsión automática del  
paracaídas *****
```

```
if (millis () >= TimerB + Periodo1) {
```

```
    TimerB = millis ();
```

```
if (valoresY >= Rng || valoresX >= Rng || valoresY <= -Rng || valoresX <= -Rng ) {
```

```
    if (millis () >= TimerC + Periodo2) {
```

```
        TimerC = millis ();
```

```
        if (valoresY >= Rng || valoresX >= Rng || valoresY <= -Rng || valoresX <= -Rng ) {
```

```
            // Serial.println("Suprimir energía UAV y expulsando el paracaídas");
```

```
            delay (100); //Retardo de 100 (ms).
```

```
            digitalWrite(Rele, HIGH); //Desactivación del relé.
```

```
            delay (2000); //Retardo de 2 (s).
```

```
            moverServo (2, 115) ;
```

```
            digitalWrite (buzz, HIGH); //Activación del buzzer.
```

```
            delay(170); //Retardo de 170 (ms).
```

```
            digitalWrite (buzz, LOW); //Desactivación del buzzer.
```

```
            delay(170); //Retardo de 170 (ms).
```

```
            goto IMU;
```

```
        }
```

```
    }
```

```
}
```

```
//***** Lectura de mensajes recibidos *****
```

```
if (messageIndex > 0 ) {
```

```
    digitalWrite (ledC, HIGH);
```

```
    digitalWrite (ledB, LOW);
```

```
    sim808.readSMS(messageIndex, message, longitud_sms, phone, datetime);
```

```
    sim808.deleteSMS(messageIndex);
```

```
    // Serial.print("Mensaje: ");
```

```
    // Serial.println(message);
```

```
    String datosms = message;
```

```

if (datosms == "~EstadoGSM" ) {
  // Serial.println("Por sms solicitando el nivel de señal de la red GSM");
  String datoA = calidadGSM ();
  sms += datoA;
  // Serial.print ("sms2: ");
  send_sms (sms, phone);
  digitalWrite (ledC, LOW);
  goto IMU;
}

if (datosms == "~CloseSP" ) {
  // Serial.println("Por sms solicitando cerrar el sistema del paracaídas");
  digitalWrite(Rele, LOW);
  delay (100);
  moverServo (2, 0) ;
  digitalWrite (ledC, LOW);
  goto IMU;
}

if (datosms == "~Paracaidas" ) {
  // Serial.println("Por sms solicitando expulsar el paracaídas");
  digitalWrite(Rele, HIGH);
  delay (2000);
  moverServo (2, 115) ;
  digitalWrite (ledC, LOW);
  goto IMU;
}

if (datosms == "~CoordenadasGPS") {
  // Serial.println("Por sms solicitando las coordendas GPS");
  String datoX = dato_lat();
  String datoY = dato_lon();
  String sms = "Lat:";
  String Lon = ":Lon:";
  sms += datoX;
  sms += Lon;
  sms += datoY;
}

```

```

    send_sms (sms, phone);
    digitalWrite (ledC, LOW);
    goto IMU;
}

if (datosms == "~Buzzer") {
    // Serial.println("Por sms solicitando activar el sistema auditivo");
    buzzer();
    digitalWrite (ledC, LOW);
    goto IMU;
}
}

//***** Lectura de datos de los módulos RF *****
if (Serial.available() > 0) {
    DATO = Serial.read();
    digitalWrite (ledB, HIGH);
    digitalWrite (ledC, LOW);

    // Literal para expulsar el paracaídas.
    if (DATO == 'A') {
        delay (100);
        digitalWrite(Rele, HIGH);
        delay (2000);
        moverServo (2, 115) ;
        digitalWrite (ledB, LOW);
        goto IMU;
    }

    // Literal para obtener las coordenadas GPS.
    if (DATO == 'B') {
        String datoA = dato_lat();
        String datoB = dato_lon();
        Serial.print ("<Coor: ");
        Serial.print (datoA);
        Serial.print (":");
        Serial.print (datoB);
    }
}

```

```

Serial.println(">");
delay (90);
digitalWrite (ledB, LOW);
goto IMU;
}

// Literal para cerrar el sistema del paracaidas.
if (DATO == 'C') {
digitalWrite(Rele, LOW);
delay (100);
moverServo (2, 0) ;;
digitalWrite (ledB, LOW);
goto IMU;
}

// Literal para activar el buzzer.
if (DATO == 'E') {
buzzer();
digitalWrite (ledB, LOW);
goto IMU;
}

// Literal para obtener el nivel de la señal GSM.
if (DATO == 'F') {
datoB = calidadGSM ();
Serial.print ("<");
Serial.print(datoB);
Serial.println (">");
delay (100);
digitalWrite (ledB, LOW);
goto IMU;
}
}
}

//***** Función para la conexión de la antena GPS con los satélites globales*****
String connectionGPS () {

```

```

int cont = 0;
while (!sim808_check_with_cmd("AT+CGPSSTATUS?\r\n", "+CGPSSTATUS: Location 2D
Fix", CMD) && !sim808_check_with_cmd("AT+CGPSSTATUS?\r\n", "+CGPSSTATUS: Location
3D Fix", CMD))
{
    cont++;
    // Serial.println ("Esperando la conexión GPS");
    for (int t = 0; t < 50; t++) {
        digitalWrite (ledA, HIGH);
        delay(80);
        digitalWrite (ledA, LOW);
        delay(80);
        digitalWrite (ledB, HIGH);
        delay(80);
        digitalWrite (ledB, LOW);
        delay(80);
        digitalWrite (ledC, HIGH);
        delay(80);
        digitalWrite (ledC, LOW);
        delay(80);
    }

    if (cont == 11) {
        digitalWrite (ledA, HIGH);
        mySerial.println("AT+CGNSPWR=0"); //Deshabilitar el suministro de energía de la antena
GPS del módulo sim808.
        delay(30000); //Retrdo de 30 (s)
        mySerial.println("AT+CGNSPWR=1"); //Habilitar el suministro de energía de la antena GPS
del módulo sim808.
        delay(15000); //Retrdo de 15 (s)
        mySerial.println(" AT+CGPSRST=0"); //Iniciar en modo frío la antena GPS.
        delay(15000); //Retrdo de 15 (s)
        mySerial.println("AT+CGNSSEQ=RMC"); //Definición de la trama GPS como RCM.
        delay(15000); //Retrdo de 15 (s)
        digitalWrite (ledA, LOW);
    }
}
}

```

```

cont = 0;
for (int t = 0; t < 10 ; t++) { // repetimos la instruccion 10 veces
  digitalWrite (buzz, HIGH);
  delay(140);
  digitalWrite (buzz, LOW);
  delay(140);
  digitalWrite (ledA, HIGH);
  digitalWrite (ledB, HIGH);
  digitalWrite (ledC, HIGH);
  delay (1500);
  digitalWrite (ledA, LOW);
  digitalWrite (ledB, LOW);
  digitalWrite (ledC, LOW);
}
sms1 = "+CGPSSTATUS: Location 3D Fix";
return sms1;
}

//***** Función para obtener las coordenadas GPS *****
String dato_lat() {
  // mySerial.listen();
  int a = 0;
  String strLat, strLon, signo = "-";
  do {
    if ((unsigned long)(millis() - _tActual) > _tEsperado) {
      mySerial.println ("AT+CGNSINF\r");

      _tActual = millis (); //actualiza el tiempo
    }

    // mySerial.println ("AT+CGNSINF\r\n");
    // delay (900);
    while (mySerial.available() > 0) {
      leer = mySerial.read();
      cadena += (char)leer; //concatena en una cadena de txt.
      if (leer == 45) {

```

```

//    Serial.print("cadena: ");
//    Serial.println(cadena);

cadena.toCharArray (trama, 20); //
cadena = ""; //limpia la cadena
char *p = trama;
//    Serial.print("trama: ");
//    Serial.println(trama);

if (leer == 45 && p[8] == ',' && p[9] == '-') {
    strDato = trama;
    String strLat = signo + strDato;
    strLat.toCharArray(cLat, 10);
    //    Serial.print("trama: ");
    //    Serial.print(trama);
}
if (leer == 45 && p[9] == ',') {
    strDato2 = trama;
    String strLon = signo + strDato2;
    strLon.toCharArray(cLon, 11);
    a = a + 1;
}
}
}
} while (a == 0);
delay (1000);
activador = false;
String lat = cLat;
dato_lon();
return lat;
}
String dato_lon () {
    String lon2 = cLon;
    return lon2;
}

//***** Función para obtener el nivel de la señal GSM *****

```



```

String calidadGSM () {
do {
mySerial.print("AT+CSQ\r\n"); //enviando SMS en modo de texto
delay(100); //RETARDO PARA DE 500ms
while (mySerial.available() > 0) {
delay(100);
leer = mySerial.read();
cadena += (char)leer; //concatena en una cadena de txt.
if (leer == '\n') {
cadena.toCharArray (trama, 11); //
cadena = ""; //limpia la cadena
if (trama[0] == '+' && trama[1] == 'C' && trama[2] == 'S' && trama[3] == 'Q' && trama[4] ==
':') {
strDato = trama;
strDato.toCharArray(cDato, 11);
activador = true;
}
}
} while (activador == false);
strDato3 = cDato;
return strDato3;
}

```

//***** Función para activar el sistema auditivo *****

```

void buzzer () {
for (int t = 0; t < 20 ; t++) { // repetimos la instrucción 20 veces
digitalWrite (buzz, HIGH);
delay(170);
digitalWrite (buzz, LOW);
delay(170);
}
return;
}

```

//***** Función para enviar los datos por mensaje de texto *****

```

String send_sms (String sms, char numero[]) {

```

```

mySerial.print("AT+CMGS=");
mySerial.print((char)34);
mySerial.print(numero);
mySerial.println((char)34);
delay (1000);
mySerial.print("<");
mySerial.print(sms);
mySerial.print(">");
delay(1000); //RETARDO PARA DE 1s
mySerial.write(0x1A); //Envia el Ctrl+Z para finalizar la trama del sms.
activador = false;
// respuestaSMS = " MENSAJE ENVIADO";
return;
}

//***** Función para mover el Micro servomotor *****
void moverServo (int pin, int angulo) {
  for (int i = 0; i < 30; i++) {
    pinMode (pin, OUTPUT);
    pausa = angulo * 2000.0 / 180.0 + 500;
    digitalWrite (pin, HIGH);
    delayMicroseconds (pausa);
    digitalWrite (pin, LOW);
    delayMicroseconds (2500 - pausa);
  }
}

```

ANEXO 6: CÓDIGO DE PROGRAMACIÓN DE LA APLICACIÓN MÓVIL

En este anexo se presenta el código de programación en *Java* de la aplicación móvil, el mismo que fue elaborado en el programa *Android Studio* versión 3.6.3.

//Código java para la comunicación a través de la red GSM.

```
public class SegundaVentana extends AppCompatActivity {
    StringBuilder datosSMSbuilder = new StringBuilder();
//Declaración de variables.
    public static String parts [] ;
    public static String parts1 [] ;
    public static String datoSGSMparts [] ;
    public static String datoGPS;
    public static String flagdatoSGPS ="";
    public static String sharelon, sharelat, shareSGSM, shareSGPS;
    boolean flagdatosms1 = false;
    boolean flagdatosms2 = false;
    boolean flagdatosms3 = false;
    boolean flagdatoNum = false;
    boolean flagdatosms5 = false;
    boolean flagdatosms6 = false;
    boolean flagdatoGPS = false;
    boolean flagSGSM = false;
    private TextInputLayout inputLayoutNumero;
//Declaración de variables para la utilización de los botones de la aplicación.
    EditText nNumero, smsRx, nNumeroRx;
    FloatingActionButton btnSelecCont;
    Button btnSP, btnGSM, btnCSP, btnCooGPS, btnGPS, btnClearCoor, btnClearSGSM,
    btnClearSGPS, btnGoogMaps, btnOsmAnd, btnConfir, btnMenuP, btninfoSMS, btninfoGPS,
    btninfoOsmAnd, btnInfoBuzz2, btnBuzz2 ;
    TextView smsRxLong, smsRxLat, smsRx2, smsRx3, smsRxSGSM, smsRxSGSMRSI,
    smsRxSGPS, txtViewNota, nNombre;

    String strSGSM1;
```

```

String str31 = "31", str30 = "30",str29 = "29",str28 = "28",str27 = "27",str26 = "26",str25 =
"25",str24 = "24",str23 = "23",str22 = "22",str21 = "21",str20 = "20";
String str19 = "19", str18 = "18",str17 = "17",str16 = "16",str15 = "15",str14 = "14",str13 =
"13",str12 = "12",str11 = "11",str10 = "10",str9 = "9",str8 = "8,";
String str0 = "0,",str1 = "1,",str2 = "2,",str3 = "3,",str4 = "4,",str5 = "5,",str6 = "6,",str7 = "7,";

//Variables para obtener un resultado a través de la red GSM.
String numRx;
String inicioNumero = "+593";
String messageTxtSP = "~Paracaidas";
String messageTxtGPS = "~EstadoGPS";
String messageTxtGSM = "~EstadoGSM";
String messageTxtCooGPS = "~CoordenadasGPS";
String messageTxtCSP = "~CloseSP";
String messageTxtBuzz = "~Buzzer";
final static String DatoSGSM= "+CSQ";
final static String DatoSGPS= "+CGPSSTATUS";
final static String DatoLat= "Lat";
final static String DatoLon= "Lon";
final static String DatoIMU= "<IMU";
// final static String DatoGPS= "+GPS";
final static String DatoASP= "Abrir :";
final static String DatoCSP= "Cerrar";
final static String ClearDatos= "";

//Habilitar los permisos para acceder a los servicios de mensajes de texto.
private static final int MY_PERMISSIONS_REQUEST_RECEIVE_SMS = 0;
private static final String SMS_RECEIVED = "android.provider.Telephony.SMS_RECEIVED";
if (ContextCompat.checkSelfPermission(SegundaVentana.this,
Manifest.permission.RECEIVE_SMS) != PackageManager.PERMISSION_GRANTED) {

if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.RECEIVE_SMS)) {
} else {
ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.RECEIVE_SMS},
MY_PERMISSIONS_REQUEST_RECEIVE_SMS);

```

```
}  
}
```

//Función para reconocer el mensaje de texto recibido al dispositivo móvil.

```
MyReceiver receiver = new MyReceiver(){  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        super.onReceive(context, intent);  
        // smsRx.setText(msg);  
        String datosms= msg;  
  
        //      Toast.makeText(getApplicationContext(),"String sms: " + datosms,  
Toast.LENGTH_LONG).show();  
        boolean datoInicio = datosms.startsWith("<");  
        boolean datoFin = datosms.endsWith(">");  
        if (datoInicio== true && datoFin == true){  
            String dato1= datosms.replaceFirst("<",":");  
            String dato2= dato1.replaceFirst(";",":");  
            String dato3= dato2.replaceFirst(">",":");  
            parts= dato3.split(":");  
            parts1 = dato3.split(":");  
            flagdatosms1 = parts[1].equals(DatoLat);  
            flagdatosms2 = parts[1].equals(DatoSGSM);  
            flagdatosms3 = parts[1].equals("+CGPSSTATUS");  
            //  Toast.makeText(getApplicationContext(),"String sms: " + flagdatosms3,  
Toast.LENGTH_LONG).show();  
  
            // smsRx2.setText(parts1[1]);  
            // smsRx3.setText(parts1[2]);///  
  
//Condición para las coordenadas de ubicación recibidas por mensaje de texto.  
        if (flagdatosms1 == true){  
            Toast.makeText(getApplicationContext(),"String sms: " + parts[1],  
Toast.LENGTH_LONG).show();  
            String DatoLatitud = parts1[2];  
            String DatoLong = parts1[4];  
            boolean datoGPSInicioLat = DatoLatitud.startsWith("-");
```

```

boolean datoGPSInicioLon = DatoLong.startsWith("-");
// smsRxSGSMRSI.setText(parts1[2]);
// smsRxSGSM.setText(parts1[4]);///

if (datoGPSInicioLat== true && datoGPSInicioLon == true){
    smsRxLat.setText(parts1[2]);
    smsRxLong.setText(parts1[4]);
    flagdatoGPS = true;
    //      Toast.makeText(getApplicationContext(),"String sms: " + datoInicio,
Toast.LENGTH_LONG).show();
}
//Condición para comparar los datos de la señal GSM recibido por mensaje de texto.
flagdatosms1 = false;

if (flagdatosms2 == true){
    //Toast.makeText(getApplicationContext(),"String sms: " + parts[1],
Toast.LENGTH_LONG).show();

    String DatoSGSMRx = parts1[2];
    strSGSM1=DatoSGSMRx.substring(1,3);// Dato 31

    flagSGSM= strSGSM1.equals(str31);
    //  Toast.makeText(getApplicationContext(),"String sms: " + flagSGSM ,
Toast.LENGTH_LONG).show();
    // new Hilo1().start(); //HILO SECUNDARIO

    if (strSGSM1.equals(str31) ) {
        smsRxSGSM.setText("Execlente Señal");
        smsRxSGSMRSI.setText("-51 ");
    }
    else if (strSGSM1.equals(str30) ) {
        smsRxSGSM.setText("Execlente Señal");
        smsRxSGSMRSI.setText("-53 ");
    }else if (strSGSM1.equals(str29) ) {
        smsRxSGSM.setText("Execlente Señal");
        smsRxSGSMRSI.setText("-55 ");
    }else if (strSGSM1.equals(str28) ) {

```

```

        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-57 ");
    }else if (strSGSM1.equals(str27) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-59 ");
    }else if (strSGSM1.equals(str26) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-61 ");
    }else if (strSGSM1.equals(str25) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-63 ");
    }else if (strSGSM1.equals(str24) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-65 ");
    }else if (strSGSM1.equals(str23) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-67 ");
    }else if (strSGSM1.equals(str22) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-69 ");
    }else if (strSGSM1.equals(str21) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-71 ");
    }else if (strSGSM1.equals(str20) ) {
        smsRxSGSM.setText("Excelente Señal");
        smsRxSGSMRSI.setText("-73 ");
    }else if (strSGSM1.equals(str19) ) {
        smsRxSGSM.setText("Buena Señal");
        smsRxSGSMRSI.setText("-75 ");
    }else if (strSGSM1.equals(str18) ) {
        smsRxSGSM.setText("Buena Señal");
        smsRxSGSMRSI.setText("-53 ");
    }else if (strSGSM1.equals(str17) ) {
        smsRxSGSM.setText("Buena Señal");
        smsRxSGSMRSI.setText("-53 ");
    }else if (strSGSM1.equals(str16) ) {
        smsRxSGSM.setText("Buena Señal");
    }

```

```

        smsRxSGSMRSI.setText("-81 ");
    }else if (strSGSM1.equals(str15) ) {
        smsRxSGSM.setText("Buena Señal");
        smsRxSGSMRSI.setText("-83 ");
    }else if (strSGSM1.equals(str14) ) {
        smsRxSGSM.setText("Señal Regular");
        smsRxSGSMRSI.setText("-85 ");
    }else if (strSGSM1.equals(str13) ) {
        smsRxSGSM.setText("Señal Regular");
        smsRxSGSMRSI.setText("-87 ");
    }else if (strSGSM1.equals(str12) ) {
        smsRxSGSM.setText("Señal Regular");
        smsRxSGSMRSI.setText("-89 ");
    }else if (strSGSM1.equals(str11) ) {
        smsRxSGSM.setText("Señal Regular");
        smsRxSGSMRSI.setText("-91 ");
    }else if (strSGSM1.equals(str10) ) {
        smsRxSGSM.setText("Señal Regular");
        smsRxSGSMRSI.setText("-93 ");
    }else if (strSGSM1.equals(str9) ) {
        smsRxSGSM.setText("Pésima Señal");
        smsRxSGSMRSI.setText("-95 ");
    }else if (strSGSM1.equals(str8) ) {
        smsRxSGSM.setText("Pésima Señal");
        smsRxSGSMRSI.setText("-97 ");
    }else if (strSGSM1.equals(str7) ) {
        smsRxSGSM.setText("Pésima Señal");
        smsRxSGSMRSI.setText("-99 ");
    }else if (strSGSM1.equals(str6) ) {
        smsRxSGSM.setText("Pésima Señal");
        smsRxSGSMRSI.setText("-101 ");
    }else if (strSGSM1.equals(str5) ) {
        smsRxSGSM.setText("Pésima Señal");
        smsRxSGSMRSI.setText("-103 ");
    }else if (strSGSM1.equals(str4) ) {
        smsRxSGSM.setText("Pésima Señal");
        smsRxSGSMRSI.setText("-105 ");
    }

```



```

}else if (strSGSM1.equals(str3) ) {
    smsRxSGSM.setText("Pésima Señal");
    smsRxSGSMRSI.setText("-107 ");
}else if (strSGSM1.equals(str2) ) {
    smsRxSGSM.setText("Pésima Señal");
    smsRxSGSMRSI.setText("-109 ");
}else if (strSGSM1.equals(str1) ) {
    smsRxSGSM.setText("Pésima Señal");
    smsRxSGSMRSI.setText("-111 ");
}else if (strSGSM1.equals(str0) ) {
    smsRxSGSM.setText("Pésima Señal");
    smsRxSGSMRSI.setText("-113 ");
}

} flagdatosms2 = false;
//Condición para mostrar el estado de la conexión de los satélites GPS
if (flagdatosms3 == true){
    //Toast.makeText(getApplicationContext(),"String sms: " + parts[1],
    Toast.LENGTH_LONG).show();
    // smsRx2.setText(parts1[1]);
    String EstadoGPS="";
    EstadoGPS = parts1[2];
    // smsRxSGSM.setText(EstadoGPS);
    // smsRxSGPS.setText(parts1[2]);///

    if (EstadoGPS.contains("Location 3D Fix") ) {
        smsRxSGPS.setText("Localización fija en 3D");
        flagdatoSGPS = "SI";
    } else if (EstadoGPS.contains("Location 2D Fix")) {
        smsRxSGPS.setText("Localización fija en 2D");
        flagdatoSGPS = "SI";
    }else if (EstadoGPS.contains(" Location Not Fix")) {
        smsRxSGPS.setText("Ubicación no fija");
        flagdatoSGPS = "NO";
    }else if (EstadoGPS.contains(" Location Unknown")) {
        smsRxSGPS.setText("Ubicación desconocida");
        flagdatoSGPS = "NO";
    }
}

```

```

        }
    } flagdatosms3 = false;
    }
}
};

//Función para permitir al usuario la visualización del actividad de los sms recibidos
@Override
protected void onResume() {
    super.onResume();
    registerReceiver(receiver, new IntentFilter(SMS_RECEIVED));
}

//Función cerrar la actividad secundaria de la recepción de los sms
@Override
protected void onDestroy() {
    super.onDestroy();
    unregisterReceiver(receiver);
}

//Boton para activar el sistema del paracaídas.
btnBuzz2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(nNumero.getText().toString(), null, messageTxtBuzz, null, null);
        Toast.makeText(SegundaVentana.this, "Mensaje enviado para activar el sistema auditivo",
Toast.LENGTH_LONG).show();
    }
});

//Boton para activar el sistema del paracaídas.
btnSP.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(nNumero.getText().toString(), null, messageTxtSP,
null, null);

        Toast.makeText(SegundaVentana.this, "Mensaje enviado para activar el paracaídas",
Toast.LENGTH_LONG).show();
    }
});

```

```

    }
});

//Boton para cerrar el sistema del paracaídas.
btnCSP.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(nNumero.getText().toString(), null, messageTxtCSP,
null, null);
        Toast.makeText(SegundaVentana.this, "Mensaje enviado para cerrar el sistema",
Toast.LENGTH_LONG).show();
    }
});

//Boton para conocer el estado del GPS.
btnGPS.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(nNumero.getText().toString(), null, messageTxtGPS,
null, null);
        Toast.makeText(SegundaVentana.this, "Mensaje enviado para conocer la señal GPS",
Toast.LENGTH_LONG).show();
    }
});

//Boton para conocer el estado del GSM.
btnGSM.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(nNumero.getText().toString(), null, messageTxtGSM,
null, null);
        Toast.makeText(SegundaVentana.this, "Mensaje enviado para conocer la señal GSM",
Toast.LENGTH_LONG).show();
    }
});

```

```

});

//Boton para conocer las coordenadas GPS .
btnCooGPS.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SmsManager smsManager = SmsManager.getDefault();
        smsManager.sendTextMessage(nNumero.getText().toString(), null,
messageTxtCooGPS, null, null);
        Toast.makeText(SegundaVentana.this, "Mensaje enviado para conocer la
coordenadas GPS", Toast.LENGTH_LONG).show();
        /*
        if (flagdatoSGPS == "SI"){
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(nNumero.getText().toString(), null,
messageTxtCooGPS, null, null);
            Toast.makeText(SegundaVentana.this, "Mensaje enviado para conocer la
coordenadas GPS", Toast.LENGTH_LONG).show();
        } else{
            Config.Mensaje(SegundaVentana.this, "El GPS no está listo", true, true);
        }
        */
    }
});

//Actividad para los contactos.
btnSelecCont.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setType(ContactsContract.CommonDataKinds.Phone.CONTENT_TYPE);
        startActivityForResult(intent, 1);
    }
});

// Función del botón para acceder al mapa de Google con los datos GPS
btnGoogMaps.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

```

```

        if (flagdatoGPS==true) {
            Intent menu = new Intent(SegundaVentana.this, MapsActivity.class);
            menu.putExtra("Lat", smsRxLat.getText().toString());
            menu.putExtra("Lon", smsRxLong.getText().toString());
            startActivity(menu);
        } else {
            Config.Mensaje(SegundaVentana.this, "No hay datos GPS", true, true);
        } }
    });

//Función para validar los datos ingresados de los números de teléfono
    public boolean validardato(){
        boolean retorno = true;
        boolean compa = false;
        boolean compa2 = false;
        // nNumero.setError(null);
        String num= nNumero.getText().toString();
        numRx = nNumeroRx.getText().toString();
        Boolean compaX = num.equals("");
        Boolean compaX2 = numRx.equals("");
        Boolean flag1 = false;
        Boolean flag2 = false;
        if (compaX==true || compaX2 == true){
            // nNumero.setError("Ingresar un número");
            // nNumeroRx.setError("Ingresar un número");
            retorno= false;

            if (compaX == true){
                nNumero.setError("Ingresar un número");
                retorno= false;
            }
            if (compaX2 == true){
                nNumeroRx.setError("Ingresar un número");
                retorno= false;
            }
        }

    } else {
        String valor1 = num.substring(0,2);
    }

```

```

String valor2 = numRx.substring(0,2);
compa = valor1.equals("09");
compa2 = valor2.equals("09");
if (num.length() <= 9 || compa == false || num.length() >= 11){
    nNumero.setError("Ingresar un número válido");
    retorno= false;
}
if (numRx.length() <= 9 || compa2 == false || numRx.length() >= 11 ){
    nNumeroRx.setError("Ingresar un número válido");
    retorno= false;
}
if (flagdatoNum== true && num.length() == 10 && compa == true){
    nNumero.setError(null);
    flag1 = true;
}
if ( numRx.length() == 10 && compa2 == true){
    nNumero.setError(null);
    flag2 = true;
}
if ( flag1 == true && flag2 == true){
    nNumero.setError(null);
    nNumeroRx.setError(null);
    retorno = true;
}
/*
if ( flagdatoNum== true && num.length() == 10 && compa == true && numRx.length() ==
10 && compa2 == true ) {
    nNumero.setError(null);
    nNumeroRx.setError(null);
    retorno = true;
}
*/
}
//num.length() !=0 || | numRx.length() !=0 |
return retorno;
}

```

//Código java para la comunicación a través de la conexión BT.

```
public class TerceraVentana extends AppCompatActivity {
    //implements OnMapReadyCallback
    private MapView mMapView;
    private GoogleMap mMap;
    private static final String MAPVIEW_BUNDLE_KEY = "MapViewBundleKey";

    //Declaración de la librería del bluetooth.
    BluetoothAdapter bluetoothAdapter = null;
    BluetoothDevice myDevice = null;
    BluetoothSocket mySocket = null;
    public static String parts [] ;
    public static String parts1 [] ;
    public static String parts2 [] ;
    public static String datoGPS;
    public static String sharelon, sharelat, shareSGSM, shareSGPS ,strSGSM1;;
    public static String BanderaSGPS = "";

    //Delcaración de las variables.
    boolean conexion = false;
    private static String MAC = null;
    String MacAddressbth;
    String datosRecibidos;
    public static boolean datosGPSrx = false;
    public static boolean datosSGSM = false;
    public static boolean datosSGPS = false;
    public static boolean FlagSGPS1= false;
    public static boolean FlagSGPS2= false;
    public static boolean flagdatobth1 = false;
    public static boolean flagdatobth2 = false;
    public static boolean flagdatobth3 = false;
    public static boolean flagdatobth4 = false;
    public static boolean flagdatobth5 = false;

    UUID MY_UUID = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");

    Button btnOff, btnOn,btnSGSM,btnDispo, btnSGPS, btnCoor, btnASP, btnCSP, btnOsmAnd,
    btnGMaps, btnMenuP, btninfor, btninforOsmAnd, btninfCoorGPS, btnBuz, btnInfoBuz;
    TextView txtBTH, txtSGSM, txtSGPS, txtLat, txtLon, txtRSI, txtadver;
    FloatingActionButton infor;
    private static final int SOLICITUD_BTH = 1;
    private static final int SOLICITUD_CONEXION = 2;
    private static final int MESSAGE_READ = 3;
    private static final int STATE_MESSAGE_RECIVE = 4;

    ConnectedThread connectedThread;
    private static Handler myHandler;
    StringBuilder datosBth = new StringBuilder();

    boolean flagdato = false;

    final static String SGSM= "F";
    final static String SGPS= "D";
```

```

final static String Coor= "B";
final static String ASP= "A";
final static String CSP= "C";
final static String Buzz= "E";
String cleardato = "";

//Resultado
final static String DatoSGSM= "+CSQ:";
final static String DatoSGPS= "+CGPSSTATUS:";
final static String DatoLat= "Lat:";
final static String DatoLon= "Lon:";
final static String DatoIMU= "+IMU";
final static String DatoGPS= "+GPS";
final static String DatoASP= "Abrir :";
final static String DatoCSP= "Cerrar";

@SuppressLint("WrongViewCast")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tercera_ventana2);

    // Bundle bundle = this.getIntent().getExtras();
    // Coor = bundle.getString("clave");
    // String MacAddress = bundle.getString("MACBTH");

    // Asignación de los botones con la variable
    final Context context = this;
    SharedPreferences shareprefs = context.getSharedPreferences("ArchivoDatos",
context.MODE_PRIVATE);
    txtSGSM = findViewById(R.id.textView20);
    txtSGPS = findViewById(R.id.textView22);
    txtLon = findViewById(R.id.textView27);
    txtLat = findViewById(R.id.textView28);
    txtRSI = (TextView) findViewById(R.id.textViewRSI);
    txtadver = (TextView) findViewById(R.id.textView12);
    btnOff = findViewById(R.id.btnOff);
    btnOn = findViewById(R.id.btnOn);
    btnSGSM = findViewById(R.id.btnSGSM);
    btnSGPS = findViewById(R.id.button11);
    btnCoor = findViewById(R.id.button14);
    btnASP = findViewById(R.id.button12);
    btnCSP = findViewById(R.id.button13);
    btnOsmAnd = findViewById(R.id.button17);
    btnGMaps = findViewById(R.id.button16);
    btnMenuP = findViewById(R.id.button18);
    btnDispo = findViewById(R.id.btnDisBth);
    btninfor = findViewById(R.id.btnInfo);
    btninforOsmAnd = findViewById(R.id.btninfoOsmAnd);
    btninforCoorGPS = findViewById(R.id.btnInfoGPS);
    btnBuz = findViewById(R.id.btnBuzz);
    btnInfoBuz = (Button) findViewById(R.id.btninfoBuzz);

```



```

//Funciones para visualizar el cuadro de información de la app.
btninfor.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showinfor ();
    }
});

btninforOsmAnd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showinforOsmAnd ();
    }
});

btninfCoorGPS.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showinforCoorGPS();
    }
});

btnInfoBuz.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showinforBuzzer();
    }
});

//Función para permitir la conexión BT con otro dispositivo.
bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if (!bluetoothAdapter.isEnabled()) {
    Toast.makeText(getApplicationContext(),"El dispositivo requiere permisos de BTH",
Toast.LENGTH_LONG).show();
} else if (!bluetoothAdapter.isEnabled()) {
    Intent activaBTH = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(activaBTH, SOLICITUD_BTH);
}

btnDispo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (conexion){
            Toast.makeText(getApplicationContext(),"Ya se encuentra
conectado",Toast.LENGTH_LONG).show();
        }else {
            //proceso de conectar
            Intent abrirlista = new Intent(TerceraVentana.this, DispositivosBth.class);
            startActivityForResult(abrirlista, SOLICITUD_CONEXION);
        }
    }
});

```

```

btnOn.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        if (!bluetoothAdapter.isEnabled()) {
            Thread thread = new Thread(){
                @Override
                public void run ()
                {
                    try {
                        bluetoothAdapter.enable();

                    }catch (RuntimeException e){
                        EventBus.getDefault().post(new UiToastEvent("El permiso fue denegado",
true,false));
                    }
                }
            };
            thread.start();
        }
    }
});

//Función para desconectar la conexión BT
btnOff.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (conexion){
            // desconectar
            try {
                mySocket.close();
                conexion = false;
                Toast.makeText(getApplicationContext(),"Bluetooth
desconectado",Toast.LENGTH_LONG).show();
            }catch (IOException error){
                Toast.makeText(getApplicationContext(),"Error
desconectar",Toast.LENGTH_LONG).show();
            }
        }else {
            //proceso de conectar
            // Intent abrirlista = new Intent(TerceraVentana.this, DispositivosBth.class);
            // startActivityForResult(abrirlista, SOLICITUD_CONEXION);
        }
    }
});

//Función para enviar el dato de la señal GSM.
btnSGSM.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (conexion){

```

al

```

        connectedThread.enviar(SGSM);
        Config.Mensaje(TerceraVentana.this, "Enviando la solicitud para conocer la señal
GSM", true, false);

    }else {
        Config.Mensaje(TerceraVentana.this, "Primero establecer la conexión BT-RF", true,
true);
    }
}
});

btnCSP.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (conexion){
            connectedThread.enviar(CSP);
            Config.Mensaje(TerceraVentana.this, "El sistema del paracaídas se está cerrando",
true, false);
        }else {
            Config.Mensaje(TerceraVentana.this, "Primero establecer la conexión BT-RF", true,
true);
        }

    }
});
btnASP.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (conexion){
            connectedThread.enviar(ASP);
            Config.Mensaje(TerceraVentana.this, "Activando el sistema del paracaídas", true,
false);
        }else {
            Config.Mensaje(TerceraVentana.this, "Primero establecer la conexión BT-RF", true,
true);
        }

    }
});
//Función para activar el buzzer
btnBuz.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (conexion){
            connectedThread.enviar(Buzz);
            Config.Mensaje(TerceraVentana.this, "Activando el sistema auditivo", true, false);
        }else {

```

```

true);
        Config.Mensaje(TerceraVentana.this, "Primero establecer la conexión BT-RF", true,
true);
    }
}
});

//Función para solicitar las coordenadas GPS
btnCoor.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Toast.makeText(TerceraVentana.this, "Bander " + BanderaSGPS
        ,
Toast.LENGTH_SHORT).show();

        if (conexion){
            connectedThread.enviar(Coor);
            Config.Mensaje(TerceraVentana.this, "Enviando la solicitud para obtener la
ubicación GPS", true, false);
        } else {
            Config.Mensaje(TerceraVentana.this, "Primero establecer la conexión BT-RF", true,
true);
        }

        /*
        if (BanderaSGPS=="SI"){

            if (conexion){
                connectedThread.enviar(Coor);
                Config.Mensaje(TerceraVentana.this, "Enviando la solicitud para obtener la
ubicación GPS", true, false);
            } else {
                Config.Mensaje(TerceraVentana.this, "Primero establecer la conexión BT-RF", true,
true);
            }

        }

        }else {
            Config.Mensaje(TerceraVentana.this, "El GPS no está listo", true, true);
        }
        */

    }
});

//Función para la lectura de datos de la conexión BT
myHandler = new Handler(){
    @Override
    public void handleMessage(@NonNull Message msg) {
        if (msg.what == MESSAGE_READ)
        {
            String recibidos = (String) msg.obj;
            datosBth.append(recibidos);
        }
    }
};

```

```

Toast.makeText(TerceraVentana.this, "Datos recibidos: " + datosBth
,
Toast.LENGTH_SHORT).show();
int infinfor = datosBth.indexOf(">");
if (infinfor > 0) {
    String datoscompletos = datosBth.substring(0,infinfor);
    int supinfor = datoscompletos.length();

    if (datosBth.charAt(0) == '<'){
        String datofinal = datosBth.substring(1, supinfor);
        // Log.d("Recibido", datofinal);
        if (datofinal.contains("+CSQ")){
            parts1 = datofinal.split(":");
            String datoSGPSrx = parts1[1];
            strSGSM1= datoSGPSrx.substring(1,3);

            String str31 = "31", str30 = "30",str29 = "29",str28 = "28",str27 = "27",str26 =
"26",str25 = "25",str24 = "24",str23 = "23",str22 = "22",str21 = "21",str20 = "20";
            String str19 = "19", str18 = "18",str17 = "17",str16 = "16",str15 = "15",str14 =
"14",str13 = "13",str12 = "12",str11 = "11",str10 = "10",str9 = "9",str8 = "8,";
            String str0 = "0,",str1 = "1,",str2 = "2,",str3 = "3",str4 = "4",str5 = "5",str6 =
"6,",str7 = "7,";

            if (strSGSM1.equals(str31) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-51 ");
            } else if (strSGSM1.equals(str30) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-53 ");
            } else if (strSGSM1.equals(str29) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-55 ");
            } else if (strSGSM1.equals(str28) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-57 ");
            } else if (strSGSM1.equals(str27) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-59 ");
            } else if (strSGSM1.equals(str26) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-61 ");
            } else if (strSGSM1.equals(str25) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-63 ");
            } else if (strSGSM1.equals(str24) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-65 ");
            } else if (strSGSM1.equals(str23) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-67 ");
            } else if (strSGSM1.equals(str22) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-69 ");
            } else if (strSGSM1.equals(str21) ) {
                txtSGSM.setText("Execlente Señal");
                txtRSI.setText("-71 ");
            }

```

```

}else if (strSGSM1.equals(str20) ) {
    txtSGSM.setText("Execlente Señal");
    txtRSI.setText("-73 ");
}else if (strSGSM1.equals(str19) ) {
    txtSGSM.setText("Buena Señal");
    txtRSI.setText("-75 ");
}else if (strSGSM1.equals(str18) ) {
    txtSGSM.setText("Buena Señal");
    txtRSI.setText("-53 ");
}else if (strSGSM1.equals(str17) ) {
    txtSGSM.setText("Buena Señal");
    txtRSI.setText("-53 ");
}else if (strSGSM1.equals(str16) ) {
    txtSGSM.setText("Buena Señal");
    txtRSI.setText("-81 ");
}else if (strSGSM1.equals(str15) ) {
    txtSGSM.setText("Buena Señal");
    txtRSI.setText("-83 ");
}else if (strSGSM1.equals(str14) ) {
    txtSGSM.setText("Señal Regular");
    txtRSI.setText("-85 ");
}else if (strSGSM1.equals(str13) ) {
    txtSGSM.setText("Señal Regular");
    txtRSI.setText("-87 ");
}else if (strSGSM1.equals(str12) ) {
    txtSGSM.setText("Señal Regular");
    txtRSI.setText("-89 ");
}else if (strSGSM1.equals(str11) ) {
    txtSGSM.setText("Señal Regular");
    txtRSI.setText("-91 ");
}else if (strSGSM1.equals(str10) ) {
    txtSGSM.setText("Señal Regular");
    txtRSI.setText("-93 ");
}else if (strSGSM1.equals(str9) ) {
    txtSGSM.setText("Pésima Señal");
    txtRSI.setText("-95 ");
}else if (strSGSM1.equals(str8) ) {
    txtSGSM.setText("Pésima Señal");
    txtRSI.setText("-97 ");
}else if (strSGSM1.equals(str7) ) {
    txtSGSM.setText("Pésima Señal");
    txtRSI.setText("-99 ");
}else if (strSGSM1.equals(str6) ) {
    txtSGSM.setText("Pésima Señal");
    txtRSI.setText("-101 ");
}else if (strSGSM1.equals(str5) ) {
    txtSGSM.setText("Pésima Señal");
    txtRSI.setText("-103 ");
}else if (strSGSM1.equals(str4) ) {
    txtSGSM.setText("Pésima Señal");
    txtRSI.setText("-105 ");
}else if (strSGSM1.equals(str3) ) {
    txtSGSM.setText("Pésima Señal");
    txtRSI.setText("-107 ");
}else if (strSGSM1.equals(str2) ) {

```

```

        txtSGSM.setText("Pésima Señal");
        txtRSI.setText("-109 ");
    }else if (strSGSM1.equals(str1) ) {
        txtSGSM.setText("Pésima Señal");
        txtRSI.setText("-111 ");
    }else if (strSGSM1.equals(str0) ) {
        txtSGSM.setText("Pésima Señal");
        txtRSI.setText("-113 ");
    }
    //txtSGSM.setText(shareSGSM);

    datosSGSM = true;

}

if (datofinal.contains("+CGPSSTATUS")){
    parts = datofinal.split(":");
    //txtLat.setText(parts[1]);
    String varSGPS = parts [1];
    // txtLon.setText(varSGPS);

    flagdatobth1 = varSGPS.contains("Location 3D Fix"); //16
    flagdatobth2= varSGPS.contains(" Location 2D Fix"); //16
    flagdatobth3 = varSGPS.contains(" Location Not Fix"); //17
    flagdatobth4 = varSGPS.contains(" Location Unknown");//Location Unknown
}

// Toast.makeText(getApplicationContext(),"Localización fija en 3D" +
flagdatosms4, Toast.LENGTH_LONG).show();

if (flagdatobth1== true) {
    txtSGPS.setText("Localización fija en 3D");
    BanderaSGPS = "SI";
    Toast.makeText(TerceraVentana.this, "Datos recibidos: " + FlagSGPS1 ,
Toast.LENGTH_SHORT).show();
} else if (flagdatobth2 == true) {
    txtSGPS.setText("Localización fija en 2D");
    BanderaSGPS = "SI";
    Toast.makeText(TerceraVentana.this, "Datos recibidos: " + FlagSGPS2 ,
Toast.LENGTH_SHORT).show();
} else if (flagdatobth3 == true) {
    txtSGPS.setText("Ubicación no fija");
    BanderaSGPS = "NO";
} else if (flagdatobth4 == true) {
    txtSGPS.setText("Ubicación desconocida");
    BanderaSGPS = "NO";
}
}

if (datofinal.contains("Coor")){
    parts2 = datofinal.split(":");
    txtLat.setText(parts2[1]);
    txtLon.setText(parts2[2]);
    datosGPSrx = true;
}
}

```

```

        }
        datosBth.delete(0,datosBth.length());
    }
}
};
}
}
}
//Método notificador de salir
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == event.KEYCODE_BACK){
        AlertDialog.Builder builder = new AlertDialog.Builder(TerceraVentana.this);
        builder.setMessage("¿Desea volver al menú principal?")
            .setPositiveButton ("Si", new DialogInterface.OnClickListener() {
                @Override
                public void onClick (DialogInterface dialog, int which){
                    Intent intent = new Intent(TerceraVentana.this, MainActivity.class);
                    startActivity(intent);
                }
            })
            .setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    dialog.dismiss();
                }
            });
        builder.show();
    }
    return super.onKeyDown(keyCode, event);
}

public void MenuPrincipal (View view){
    Intent menu = new Intent(this, MainActivity.class);
    startActivity (menu);
}

public void DispositivosBTH (View view){
    Intent bth = new Intent(this, DispositivosBth.class);
    startActivityForResult(bth, Constants.CONNECT_DEVICE_SECURE);
}

// Función para poder disponer de una ventana secundaria y poder realizar otra actividad.
@Subscribe(threadMode = ThreadMode.MAIN)
public void onUiToastEvent(UiToastEvent event) {
    Config.Mensaje(TerceraVentana.this, event.getMessage(),
    event.getLongToast(),event.getIsWarning());
};

```


ANEXO 7: CÓDIGO QR PARA ACCEDER A LA APLICACIÓN MÓVIL

En esta sección del anexo, se ha considerado un código QR para acceder al archivo con la extensión APK de la aplicación móvil Geoparachute y a la aplicación OsmAnd, los mismos que son vitales para el proyecto.

