



Instituto Politécnico Nacional

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y
TECNOLOGÍAS AVANZADAS

Proyecto Terminal II

**“SISTEMA DE MONITOREO DE ESPECIES EN PELIGRO DE EXTINCIÓN MEDIANTE UNA
RED INALÁMBRICA DE SENSORES CON RECOLECCIÓN POR ENJAMBRE DE DRONES.”**

*Que para obtener el título de
“Ingeniero en Telemática”*

Presentan:

Pablo Daniel Sauer Soria
Christian Emmanuel Sánchez Jasso

Asesores:

Dr. Rodolfo Vera Amaro

Dr. Mario Eduardo Rivero Ángeles

Instituto Politécnico Nacional



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y
TECNOLOGÍAS AVANZADAS

Proyecto Terminal II

**“SISTEMA DE MONITOREO DE ESPECIES EN PELIGRO DE EXTINCIÓN MEDIANTE UNA
RED INALÁMBRICA DE SENsoRES CON RECOLECCIÓN POR ENJAMBRE DE DRONES.”**

Que para obtener el título de

“Ingeniero en Telemática”

Presenta:

Pablo Daniel Sauer Soria

Christian Emmanuel Sánchez Jasso

Asesores:

Dr. Rodolfo Vera Amaro

Dr. Mario Eduardo Rivero Ángeles

Presidente del Jurado

Profesor titular



Dra. Blanca Rosa Briseño Tepepa

Dr. Noé Torres Cruz

DEDICATORIA

A aquellos que han sido la luz en mi camino, quienes con su apoyo y aliento han hecho posible la culminación de este arduo pero gratificante viaje académico.

A mi familia, por ser mi fuente constante de inspiración y apoyo incondicional.

A mis amigos, por compartir las alegrías y los desafíos a lo largo de este camino, convirtiendo cada paso en una experiencia valiosa.

A los profesores y mentores, por su orientación sabia y su contribución a mi desarrollo académico.

A todas las personas que, de alguna manera, han dejado su huella en este proyecto, aportando su grano de arena para hacerlo posible.

Este trabajo está dedicada a todos aquellos que creyeron en mí, incluso cuando yo dudaba, y cuya presencia ha sido fundamental en esta travesía

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a todas las personas e instituciones que contribuyeron de diversas maneras a la realización de este trabajo.

Agradezco a mis asesores por su orientación y asesoramiento, que fueron fundamentales para el desarrollo de este trabajo.

Mi reconocimiento se extiende al IPN por proporcionar el entorno propicio para llevar a cabo esta investigación.

A todos los profesores, expertos y compañeros que compartieron sus conocimientos y brindaron su apoyo, gracias por enriquecer este proyecto con sus perspectivas.

Quiero expresar mi gratitud a mi familia y amigos por su constante aliento y comprensión durante todo este proceso.

Este trabajo no habría sido posible sin el apoyo general de todos aquellos que, de alguna manera, formaron parte de este viaje académico.

Contenido

Resumen	xv
Abstract	xvii
Índice de figuras	xxv
Índice de tablas	xxvii
1. Introducción	1
2. Planteamiento del Problema	5
3. Justificación	9
4. Propuesta de Solución	11
4.1. Subsistema 1: Enjambre de Drones y Estación Base	11
4.2. Subsistema 2: WSN y el Monitoreo Remoto en Especies en Peligro de Extinción	13
4.3. Subsistema 3: Integración, Esquemas y Entorno de Visualización	14



5. Objetivos	21
5.1. Objetivo General	21
5.2. Objetivos Particulares	21
5.2.1. Proyecto Terminal I	21
5.2.2. Proyecto Terminal II	22
6. Alcances	23
7. Estado del Arte	25
8. Marco Teórico	31
8.1. Especies en Peligro de Extinción	31
8.2. Métodos Tradicionales de Monitoreo Remoto	33
8.3. Enjambre de Drones	34
8.4. Protocolos de Comunicación	36
8.5. Transceptores	38
8.6. Red Inalámbrica de Sensores (WSN)	39
8.7. Esquemas de Recolección	40
8.8. Estación Base	41
8.9. Almacenamiento en la Nube	41
8.9.1. Definición y Características del Almacenamiento en la Nube	41
8.9.2. Modelos de Servicio de Almacenamiento en la Nube	42
8.9.3. Beneficios y Desafíos del Almacenamiento en la Nube	43
8.9.4. Servicios y Tecnologías Populares de Almacenamiento en la Nube	43
8.10. Bases de Datos Relacionales y No Relacionales	44
9. Análisis	49
9.1. Análisis de los Sensores	49
9.1.1. Comparación de Sensores	51
9.1.2. Selección de los Sensores	63



9.2. Análisis de la WSN (Red Inalámbrica de Sensores)	67
9.2.1. Arquitectura de la WSN:	67
9.2.2. Topología de la WSN	69
9.2.3. Microprocesador de la WSN	70
9.2.4. Fuente de Energía de la WSN	71
9.2.5. Comunicación Inalámbrica de la WSN	72
9.2.6. Módulo de Almacenamiento de la WSN	74
9.3. Análisis de los Drones.	77
9.3.1. Tipo de Dron	78
9.3.2. Controladora de Vuelo para los Drones	83
9.3.3. GPS del Dron	84
9.4. Análisis del Enjambre de Drones	87
9.4.1. Microordenador Abordo	87
9.4.2. Lenguaje de Programación del Microordenador	89
9.4.3. Comunicación del Enjambre	90
9.4.4. Tipo de Formación del Enjambre	91
9.5. Análisis del Proveedor de Servicios en la Nube.	93
9.5.1. Criterios de Selección para el Proveedor de Servicios en la Nube.	93
9.5.2. Selección del Servidor en la Nube.	95
9.6. Búsqueda y Vinculación con Otros Proyectos de Investigación y/o Organizaciones.	99
9.7. Análisis del Animal a Estudiar	103
9.7.1. Elección del Animal a Estudiar	104
9.7.1.1. Características más Importantes del Animal Seleccionado . . .	105
10. Diseño	107
10.1. Diseño de la WSN	107
10.1.1. Conexión Física de la WSN	107
10.1.2. Lenguaje de Programación de la WSN	109



10.2. Diseño de los Drones	111
10.2.1. Estructura Física	111
10.2.2. Fuente de Energía	111
10.2.3. Componentes para Volar el Dron	112
10.3. Diseño del Enjambre de Drones	114
10.3.1. Arquitectura Física	114
10.3.2. Algoritmos de Comunicación	115
10.3.3. Programación de Rutas	116
10.4. Diseño del Dominio en la Nube	119
10.4.1. Amazon Cognito	119
10.4.1.1. Grupo de Usuarios	119
10.4.1.2. Grupos de Identidades	120
10.4.1.3. Control de Acceso con Base en Roles	122
10.4.2. AWS Identity and Access Management	122
10.4.3. Amazon Lambda	123
10.4.4. Amazon Simple Storage Service (Amazon S3)	123
10.4.5. Amazon DynamoDB	124
10.4.6. Amazon QuickSight	126
10.5. Creación del Dominio en la Nube	129
11. Implementación del sistema	131
11.1. Implementación de los drones	131
11.1.1. Configuración de los Drones.	150
11.1.1.1. Calibración de la brújula. (Compass Calibration)	151
11.1.1.2. Calibración del acelerómetro (Accel Calibration).	153
11.1.1.3. Configuración del Control.	156
11.1.1.4. Verificación del giro de los motores.	158
11.1.1.5. Calibración de las controladoras de vuelo (ESC Calibration) . .	160

11.1.2. Modos de vuelo.	162
11.1.3. Planes de Vuelo.	166
11.1.4. Condiciones climatológicas.	167
11.2. Implementación de la WSN	169
11.2.1. Adquisición de los Elementos	169
11.2.2. Funcionamiento de los Elementos	172
11.2.3. Integración de los Elementos	182
11.3. Modelado del consumo energético del sistema	195
11.3.1. Variables del sistema	196
11.3.2. Estados del sistema	197
11.3.3. Implementación y resultados del sistema	201
11.4. Creación de la página Web	213
11.5. Integración de la página web.	220
11.5.1. Trabajo futuro.	222
11.6. Integración de los Elementos	222
12. Resultados	237
12.0.1. Pruebas en Simulación	237
12.0.2. Esquema de vuelo: Enjambre de Drones.	240
12.0.3. Esquema de Vuelo: Un Solo Dron.	242
13. Cronograma	245
14. Escenario de Pruebas	247
14.1. INECOL	251

Resumen

El presente proyecto terminal presenta una propuesta para implementar un sistema de monitoreo remoto para ayudar en la conservación de animales utilizando una red inalámbrica de sensores (WSN), un enjambre de drones y un servidor web con el propósito de ofrecer una alternativa a los métodos tradicionales que presentan diversas limitaciones y desafíos, proponiendo que el enjambre de drones recolecte la información obtenida por la WSN. Eligiendo la estrategia de vuelo de los drones a partir de dos esquemas de recolección de datos, los cuales serán comparados en términos de consumo energético y tiempo de vuelo. Durante la ejecución de este trabajo, se logró establecer una colaboración con investigadores del INECOL, quienes aceptaron participar activamente en la iniciativa para monitorear al venado de cola blanca. Presentando, por último, el análisis y diseño propuestos así como algunos resultados preliminares.

Palabras Clave: *Red Inalámbrica de Sensores (WSN), enjambre de drones, monitoreo de fauna, telemetría, consumo energético.*

Abstract

This terminal project presents a proposal to implement a remote monitoring system aimed at aiding animal conservation. The approach involves utilizing a wireless sensor network (WSN), a swarm of drones, and a web server to offer an alternative to traditional methods with various limitations and challenges. The proposal suggests that the drone swarm collects information gathered by the WSN. Two data collection schemes are introduced, with a focus on comparing their energy consumption and flight time. Throughout the project, a successful collaboration was established with researchers from INECOL, actively engaging in monitoring the white-tailed deer. The document concludes with the proposed analysis and design, along with some preliminary results.

Keywords: *Wireless Sensor Network (WSN), drone swarm, wildlife monitoring, telemetry, energy consumption.*

Índice de figuras

3.1. Ejemplos de especies en peligro de extinción.	10
4.1. Diseño y arquitectura del sistema de tres drones coordinados.	13
4.2. Ejemplos de tipos de collares de seguimiento de animales.	14
4.3. Funcionamiento del sistema.	15
4.4. Primer esquema de recolección (enjambre separado).	16
4.5. Segundo esquema de recolección (enjambre en conjunto).	16
4.6. Arquitectura de propuesta para el almacenamiento y procesamiento de datos. .	19
7.1. Ejemplos del uso de UAVs para el monitoreo de especies en peligro de extinción y/o de su hábitat.	27
8.1. Categorías de conservación de la IUCN.	32
9.1. Termistor LM35.	63
9.2. Módulo GPS Neo-6M.	64
9.3. Módulo GPS Neo-6M.	65
9.4. Venado de cola blanca	106



10.1. Diseño de la WSN.	109
10.2. Diseño del dron.	113
10.3. Logo de Mission Planner.	117
10.4. Diseño del enjambre de drones.	118
10.5. Grupo de Usuarios de Amazon Cognito diagrama de funcionamiento	120
10.6. Grupo de Usuarios de Amazon Cognito diagrama de funcionamiento	121
10.7. Traslado de los datos a Amazon S3, administración de los datos almacenados en Amazon S3 y análisis de los datos con otros servicios de AWS.	124
10.8. Características principales de Amazon DynamoDB y las integraciones con otros servicios de AWS	125
10.9. Diagrama de bloques de los servicios que se consideran para el desarrollo del proyecto.	127
10.10Características del dominio en la nube.	129
10.11Visualización de la página principal al iniciar sesión en AWS	130
11.1. Componentes del Kit S500.	132
11.2. Base para el tren de aterrizaje.	133
11.3. Chasis montado sobre el tren de aterrizaje.	133
11.4. Base con 4 brazos para el armado de la placa superior.	134
11.5. Sentido de giro de los motores.	135
11.6. Componentes de la parte de potencia del dron.	135
11.7. Placa superior con las ESC montadas y sin brazos.	136
11.8. Placa superior con los motores montados.	137
11.9. Montaje de la placa superior sobre la placa base (por partes).	137
11.10Montaje de la placa superior sobre la placa base (armada).	138
11.11Montaje de la computadora de vuelo sobre la placa superior.	138
11.12Montaje del GPS de vuelo sobre la placa superior.	139
11.13Diagrama de conexiones de la computadora de vuelo Pixhawk 2.4.8	140



11.14Conexiones sobre la computadora de vuelo.	141
11.15Montaje de la Raspberry Pi4 sobre la placa superior.	142
11.16Características de la telemetría NRF24L01.	143
11.17Canalización de Control en Drones Multirrotores.	144
11.18Transmisor Radio Controlado.	146
11.19Montaje de la batería 5000mAh 6 celdas 22.2v 25C-60C RC sobre la placa inferior.	147
11.20Dron listo para pruebas de vuelo.	147
11.21Transmisores de Radio Control para cada dron	148
11.22Drones listos para pruebas de vuelo.	148
11.23Diagrama de flujo para el vuelo en enjambre.	150
11.24Conexión de la Computadora devuelvo a la computadora por cable.	151
11.25Calibración de la brújula	152
11.26Ventana de Calibración del acelerómetro.	153
11.27.	154
11.28.	154
11.29.	154
11.30.	155
11.31.	155
11.32.	155
11.33Menú de calibración del Radio Control	156
11.34Calibración del Radio Control.	157
11.35Valores Rango de los controles.	158
11.36Pines de salida Pixhawk (numerados).	158
11.37Pruebas de Motor.	159
11.38Sentido y dirección de los motores con la punta como nuestro norte de nuestro GPS.	159
11.39Leyenda para diagramas de orden de motores.	160
11.40Sentido de las propelas en sentido horario y antihorario	160



11.41Representación del control con el Throttle al máximo.	161
11.42Representación del control del Throttle al mínimo.	162
11.43Inicio del primer modo de vuelo, modo “Stabilize”.	163
11.44Modo de vuelo “Auto”.	163
11.45Armado del Dron con la palanca del Throttle.	164
11.46Modo de vuelo “Land”.	165
11.47Pestaña de creación de los planes de vuelo.	166
11.48Aplicación de ayuda para condiciones de vuelo	168
11.49Elementos adquiridos para la WSN.	172
11.50Prueba del led interno de Arduino Pro Micro.	173
11.51Resultado de registrar la temperatura del termistor LM35.	174
11.52Prueba del termistor LM35 usando el Arduino Pro Micro.	174
11.53Resultado de obtener las coordenadas y la hora mediante el GPS-6M.	176
11.54Prueba del GPS-6M.	176
11.55Prueba de la cámara trampa PR-700.	177
11.56Resultado de transmitir y recibir mediante los transceptores LoRa.	178
11.57Prueba de los módulos LoRa usando Arduinos Pro Micro.	179
11.58Resultado de las funciones disponibles del adaptador MicroSD.	180
11.59Prueba del adaptador MicroSD usando el Arduino Pro Micro.	181
11.60Prueba de la batería LiPo para energizar el Arduino Pro Micro.	181
11.61Resultados de guardar la temperatura en la memoria MicroSD.	183
11.62Integración del termistor y el módulo adaptador MicroSD.	183
11.63Resultados de guardar las coordenadas GPS en la memoria MicroSD.	184
11.64Integración del GPS y el módulo adaptador MicroSD.	184
11.65Resultados de transmitir el archivo txt de la memoria MicroSD usando LoRa.	186
11.66Integración del módulo adaptador MicroSD y el transceptor LoRa.	186
11.67Resultados de guardar las coordenadas en un archivo txt en la memoria MicroSD y trasmitirlo usando LoRa.	188



11.68 Integración del GPS, el módulo adaptador MicroSD y el transceptor LoRa.	188
11.69 Resultados de recibir las coordenadas usando LoRa, guardarlos en un archivo txt en la memoria MicroSD y recolectar datos de temperatura a la par.	190
11.70 Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa.	190
11.71 Resultados de recibir las coordenadas usando LoRa, recolectar datos de tempe- ratura y transmitir estos archivos txt.	193
11.72 Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa.	193
11.73 Diagrama de estados principal del MMPP implementado.	196
11.74 Diagrama de estados secundario del MMPP implementado.	196
11.75 Resultado de una simulación de la etapa 1 de la implementación del MMPP. . . .	203
11.76 Resultado de varias simulaciones de la etapa 1 de la implementación del MMPP. . .	203
11.77 Resultados gráficos de varias simulaciones de la etapa 1 de la implementación del MMPP.	203
11.78 Resultado de una simulación de la etapa 2 de la implementación del MMPP. . . .	204
11.79 Resultado de varias simulaciones de la etapa 2 de la implementación del MMPP. . .	204
11.80 Resultados gráficos de varias simulaciones de la etapa 2 de la implementación del MMPP.	205
11.81 Resultado de una simulación de la etapa 3 de la implementación del MMPP. . . .	206
11.82 Resultado de varias simulaciones de la etapa 3 de la implementación del MMPP. . .	206
11.83 Resultados gráficos de varias simulaciones de la etapa 3 de la implementación del MMPP.	207
11.84 Resultado de una simulación de la etapa 4 de la implementación del MMPP. . . .	208
11.85 Resultado de varias simulaciones de la etapa 4 de la implementación del MMPP. . .	208
11.86 Resultados gráficos de varias simulaciones de la etapa 4 de la implementación del MMPP.	209
11.87 Resultado de una simulación de la etapa 5 de la implementación del MMPP. . . .	209



11.88	Resultado de varias simulaciones de la etapa 5 de la implementación del MMPP.	210
11.89	Resultados gráficos de varias simulaciones de la etapa 4 de la implementación del MMPP.	210
11.90	Diferentes vistas del gráfico de superficie E_{\max} vs N_{\max} .	211
11.91	Diferentes vistas del gráfico de superficie E_{on} vs E_{sleep} .	212
11.92	Diagrama de arquitectura del subsistema de almacenamiento.	213
11.93	Creación de los usuarios administradores.	214
11.94	Configuración por usuario y contraseña.	215
11.95	Configuración del servicio de almacenamiento.	215
11.96	Creación de funciones Lambda.	216
11.97	Creación del GraphQL.	216
11.98	Archivos que se crearán.	217
11.99	Opciones para creación automatizada de código.	217
11.10	Inicio de sesión.	218
11.10	Pestaña de Inicio.	219
11.10	Pestaña de Visualización de archivos.	219
11.10	Pruebas de almacenamiento de la base de datos.	221
11.10	Pruebas de almacenamiento de la base de datos.	222
11.10	Resultados de guardar la temperatura en la memoria MicroSD.	223
11.10	Integración del termistor y el módulo adaptador MicroSD.	224
11.10	Resultados de guardar la temperatura en la memoria MicroSD.	225
11.10	Integración del GPS y el módulo adaptador MicroSD.	225
11.10	Resultados de transmitir el archivo txt de la memoria MicroSD usando LoRa.	226
11.11	Integración del módulo adaptador MicroSD y el transceptor LoRa.	227
11.11	Resultados de guardar las coordenadas en un archivo txt en la memoria MicroSD y transmitirlo usando LoRa.	229
11.11	Integración del GPS, el módulo adaptador MicroSD y el transceptor LoRa.	230



11.11Resultados de recibir las coordenadas usando LoRa, guardarlos en un archivo txt en la memoria MicroSD y recolectar datos de temperatura a la par	232
11.11Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa	233
11.11Resultados de recibir las coordenadas usando LoRa, recolectar datos de temperatura y transmitir estos archivos txt.	235
11.11Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa	236
12.1. Misión de vuelo en el campo de UPIITA-UPIBI	238
12.2. Características de la misión de vuelo.	238
12.3. Pantalla de Vuelo del Dron en el Software de Mission Planner.	239
12.4. Oscilograma de la corriente del dron líder.	240
12.5. Oscilograma de la corriente del Dron 2.	241
12.6. Oscilograma de la corriente del Dron 3.	241
12.7. Misión de vuelo Dron 1.	242
12.8. Misión de vuelo Dron 2.	242
12.9. Misión de vuelo Dron 3.	242
12.10Misiones de vuelo de los Drones 1, 2 y 3.	242
12.11Oscilograma de misión de vuelo Dron 1.	243
12.12Oscilograma de misión de vuelo Dron 2.	243
12.13Oscilograma de misión de vuelo Dron 3.	244
14.1. Escenario de pruebas finales del subsistema enjambre en el campo UPIITA-UPIBI.	249
14.2. Escenario de pruebas finales del subsistema WSN en el campo UPIITA-UPIBI.	249
14.3. Escenario de pruebas finales del sistema integral en el campo UPIITA-UPIBI.	250

Índice de tablas

7.1. Comparación entre la propuesta de solución y el estado del arte.	28
8.1. Clasificación de drones según características y aplicaciones.	34
8.2. Clasificación de sensores y dispositivos que se pueden acoplar a drones.	36
9.1. Características específicas de los sensores de medición.	53
9.2. Ventajas y limitaciones de los sensores de medición.	56
9.3. Comparación entre Arquitectura Cliente-Servidor y Peer-to-Peer (P2P).	68
9.4. Comparación de proveedores de servicio en la nube (AWS y Azure).	96
9.5. Comparación de proveedores de servicio en la nube (GCP y OCI).	97
11.1. Relación de la capacidad de la batería con el tamaño de las hélices.	147
11.2. Comparación de precios de componentes electrónicos	170
13.1. Cronograma de actividades PT1.	245

CAPÍTULO 1

Introducción

La conservación de la flora y fauna en el planeta, especialmente de las especies en peligro de extinción, se ha convertido en una problemática social apremiante. Cada año, la lista de especies amenazadas o extintas sigue creciendo, lo cual plantea una preocupación apremiante para mantener el equilibrio de los ecosistemas [?]. A pesar de los esfuerzos realizados con técnicas y métodos tradicionales y actuales para el monitoreo de estas especies, existen desventajas significativas que limitan su eficacia. Por lo tanto, se requiere una innovación en el enfoque de monitoreo para mejorar la protección de estas especies vulnerables.

En la actualidad, los métodos de monitoreo tradicionales, como censos manuales y observación directa, son costosos, lentos y pueden ser invasivos ya que requieren la intervención del hábitat natural causando estrés y alterando el comportamiento natural de la especie animal [?]. Asimismo, la falta de cobertura continua y la complicada accesibilidad a áreas remotas, dificultan la recopilación de datos precisos y en tiempo real, lo cual puede dificultar la toma de decisiones informadas para su protección [?].

Ante este panorama, surge la necesidad de explorar nuevas alternativas tecnológicas que per-



mitan un monitoreo más efectivo y menos invasivo. Una solución prometedora es el uso de una red inalámbrica de sensores (WSN, por sus siglas en inglés) ubicados estratégicamente en diferentes puntos del hábitat natural de la especie en estudio [?]. Estos sensores pueden recopilar datos sobre el comportamiento, ubicación y otros parámetros de interés de la especie, sin interferir con su comportamiento natural como en el trabajo de TigerCENSE [?], donde se utiliza una WSN con un sensor infrarrojo pasivo para capturar imágenes de un tigre en peligro de extinción sin interferir con su comportamiento natural.

Algo semejante ocurre con el uso de drones como vehículos aéreos para la recolección de la información recopilada por los sensores de la WSN al ofrecer ventajas significativas en términos de acceso a áreas remotas, velocidad y precisión en la recopilación de datos [?]. Los drones pueden sobrevolar el hábitat para recolectar los datos de manera no invasiva y transportarlos a una estación base (BS), brindando a los biólogos de campo una visión detallada de la situación de la especie que desean preservar [?].

Sin embargo, los sistemas de monitoreo que utilizan un solo dron presentan desventajas significativas en términos de consumo de energía, cobertura y conexión remota. Estos sistemas pueden ser limitados en la duración de su tiempo de vuelo y alcance, lo que puede dificultar el monitoreo continuo y en tiempo real de las especies amenazadas. Agregando que en áreas remotas o de difícil acceso, la conexión remota puede ser insuficiente para transmitir los datos de manera óptima [?]. Por anterior, en este documento se abordará de manera integral la propuesta de una solución innovadora para el monitoreo de especies en peligro de extinción. Comenzando por explorar el planteamiento del problema, destacando las limitaciones de los métodos actuales, y posteriormente, justificando la urgencia de adoptar enfoques más efectivos. La propuesta de solución se desarrollará mediante la presentación de subsistemas, como el enjambres de drones, la redes inalámbricas de sensores (WSN) y su integración. Además, se delinearan los objetivos y alcances para proporcionar un marco claro de los logros esperados. Se explorará el estado del arte y el marco teórico, contextualizando la investigación en el panorama científico actual. Luego, se realizará un análisis detallado de los componentes clave, desde sensores y drones hasta los servicios en la nube. También se abordará el diseño de los

elementos vistos en el análisis, presentando las estructuras físicas y lógicas de los drones, el enjambre y el dominio en la nube. Finalmente, se examinarán los resultados de pruebas en simulación y esquemas de vuelo, preparando el terreno para las validaciones en el escenario de pruebas.

CAPÍTULO 2

Planteamiento del Problema

La protección de las especies en peligro de extinción y la preservación de la biodiversidad son temas críticos que preocupan a la comunidad científica y a la sociedad en general [?]. En este sentido, los biólogos de campo han implementado técnicas de monitoreo y seguimiento para evaluar la salud y el tamaño de la población de la especie, así como la calidad de su hábitat [?]. Una de estas técnicas es la observación directa, que consiste en la observación visual de las especies en su entorno natural [?]. Si bien, esta técnica permite obtener datos detallados sobre el comportamiento y la interacción de las especies, presenta limitaciones en términos de cobertura espacial y temporal, ya que depende de la presencia de los biólogos en el terreno. Esto dificulta su implementación en regiones remotas o de difícil acceso, conllevando riesgos y costos significativos en términos de recursos humanos y logísticos [?], y ocasionando perturbaciones que pueden generar efectos negativos en la especie y en su hábitat [?].

Otra técnica utilizada es el monitoreo acústico, que se basa en el registro y análisis de los sonidos emitidos por las especies. Permite obtener información sobre la actividad, la comunicación y la distribución espacial de las especies. Sin embargo, la identificación precisa de las



especies a través de los sonidos puede ser compleja, especialmente cuando hay varias especies presentes en un área [?]. Agregando que las condiciones ambientales pueden afectar la calidad de los datos acústicos debido al ruido de fondo.

El monitoreo por medio de rastreo y seguimiento satelital es otra técnica empleada, que permite obtener datos sobre la ubicación y los movimientos de las especies a través de dispositivos de rastreo satelital. Esto proporciona información detallada sobre los patrones de migración, la distribución espacial y la interacción de las especies con su entorno. Sin embargo, esta técnica puede resultar costosa debido al equipo y los servicios de satélite necesarios, y también puede tener limitaciones en términos de precisión y resolución espacial [?].

Aparte de estas técnicas, también se ha empleado el monitoreo con un solo dron para capturar imágenes y datos desde el aire. No obstante, esta técnica exhibe importantes desventajas técnicas. Por un lado, se encuentra restringida en cuanto a la cobertura espacial debido al alcance y autonomía limitados del dron. Además, se necesita contar con un piloto experto para operar el dron de forma segura. Asimismo, recolectar datos desde múltiples perspectivas o ángulos diferentes puede resultar complicado con un solo dron [?].

En vista de la necesidad imperante de desarrollar un sistema de monitoreo remoto para especies en peligro de extinción, se plantea la aplicación de redes inalámbricas de sensores (WSN) como una solución tecnológica con potencial prometedor. Esta propuesta busca permitir la recopilación de datos precisos sobre las especies en cuestión, al tiempo que minimiza las perturbaciones en su hábitat y reduce los riesgos asociados para los biólogos de campo. Sin embargo, la implementación de WSN en entornos caracterizados por su difícil acceso y condiciones remotas plantea desafíos particulares. Entre ellos se encuentran los problemas de comunicación en áreas aisladas, la necesidad de establecer una conectividad efectiva entre la WSN y los drones encargados de la recolección de datos, así como la garantía de una transmisión de información confiable en condiciones ambientales adversas [?]. Ante estos desafíos, el enfoque tecnológico de la telemática emerge como una solución potencialmente efectiva y viable para superar las limitaciones y maximizar los beneficios del monitoreo remoto de especies en peligro de extinción.

Basándonos en las consideraciones expuestas, la pregunta de investigación que orientará este trabajo es la siguiente: ¿Cómo se puede implementar un sistema de monitoreo remoto de especies en peligro de extinción utilizando una red de sensores inalámbrica capaz de transmitir datos a un enjambre de drones que utilice una estrategia de recolección de datos eficiente?

CAPÍTULO 3

Justificación

La protección de las especies en peligro de extinción es crucial para garantizar la supervivencia de la biodiversidad del planeta. Cada especie tiene un valor intrínseco y único, y su pérdida representa una disminución irreparable de la diversidad biológica del ecosistema [?]. La biodiversidad es esencial para el equilibrio y la estabilidad de los ecosistemas, ya que cada especie cumple un papel importante en su funcionamiento. Agregando que, el valor intrínseco de estas especies también puede tener un valor económico. Por ejemplo, pueden ser utilizadas como recursos para la investigación científica, lo que contribuye al avance del conocimiento científico y tecnológico. Del mismo modo, su conservación puede tener beneficios para el turismo ecológico, generando empleo y oportunidades económicas en las comunidades locales [?]. La protección de las especies animales que se encuentran en esta situación también tiene beneficios indirectos para el medio ambiente y la sociedad en su conjunto. La conservación de estas especies puede contribuir a la preservación de los ecosistemas y los servicios ambientales que estos proporcionan, como la regulación del clima, la polinización de cultivos y la purificación del agua. La pérdida de una especie puede tener un efecto dominó en el ecosistema, afectando



a otras especies y a la misma funcionalidad del ecosistema en su conjunto [?]. De igual forma, la pérdida de biodiversidad puede tener consecuencias negativas para la salud humana y la calidad de vida, dado que la biodiversidad está estrechamente relacionada con la salud de los ecosistemas y, por ende, con la salud de las personas [?]. La pérdida de especies puede afectar la disponibilidad de alimentos, medicinas y otros recursos naturales que son esenciales para la subsistencia y el bienestar de las comunidades humanas [?].

Por lo tanto, es fundamental contar con un sistema de monitoreo eficiente y preciso para los animales que se encuentran en riesgo de una posible extinción [?]. Un sistema de monitoreo mediante una red inalámbrica de sensores puede proporcionar datos y observaciones precisas sobre el comportamiento, la distribución y la salud de estas especies, lo que permite a los expertos tomar decisiones informadas y estratégicas para su conservación. Este tipo de sistema de monitoreo puede ser una herramienta valiosa en los esfuerzos de conservación de estas especies, garantizando su supervivencia y contribuyendo a la protección de la biodiversidad y la estabilidad de las comunidades humanas que dependen de ella.



Figura 3.1: Ejemplos de especies en peligro de extinción.

CAPÍTULO 4

Propuesta de Solución

Las etapas en las que se divide esta propuesta de solución se describen de forma general a continuación:

4.1. Subsistema 1: Enjambre de Drones y Estación Base

El uso de enjambres de drones en la biología de campo ha emergido como una innovadora y prometedora herramienta para la recolección de datos en áreas remotas y de difícil acceso, facilitando el monitoreo y estudio de la biodiversidad en estos entornos [?]. En vista de que, a menudo los biólogos de campo enfrentan largos y agotadores recorridos para acceder a áreas de estudio, implicando un gran costo en términos de tiempo, esfuerzo físico y recursos [?]. A su vez, la presencia humana en áreas naturales puede perturbar a las especies en peligro de extinción y afectar negativamente a los hábitats frágiles. Es por esto que se propone el uso de un enjambre de drones que pueda volar en formación hacia una red inalámbrica de sensores (WSN) con el fin de recolectar datos relevantes sobre la especie y su hábitat.



Las ventajas del empleo de un enjambre, en comparación con un solo dron, son diversas. En primer lugar, la robustez del enjambre es superior [?], ya que si uno de los drones falla o presenta algún problema, los demás drones pueden continuar con el proceso de recolección de datos, asegurando la continuidad de la operación sin interrupciones significativas. En segundo lugar, la capacidad de recolección de datos se ve incrementada considerablemente al utilizar un enjambre en lugar de un solo dron [?]. Cada dron del enjambre puede recolectar información simultáneamente, lo que permite cubrir un área mayor en menos tiempo y almacenar una cantidad considerablemente mayor de datos. Esto es especialmente beneficioso en estudios de biología de campo donde se requiere recolectar una gran cantidad de información para obtener resultados confiables y de forma más rápida.

Otra ventaja destacada del uso de enjambres de drones es su capacidad para enfrentar condiciones adversas [?], como ráfagas de viento. Debido a la distribución y coordinación entre los drones, el enjambre puede compensar mejor los efectos del viento, manteniendo una posición y estabilidad más robustas durante el vuelo. Esto garantiza una mayor precisión en la recolección de datos y reduce el riesgo de errores o pérdidas de información debido a las condiciones climáticas.

La arquitectura del sistema propuesto consiste en una combinación de tecnologías y componentes clave. En el enjambre de drones, cada dron estará equipado con transceptores inalámbricos que permitirán la comunicación entre ellos y con la red de sensores. Para asegurar la eficiencia en la recolección de datos, se utilizarán algoritmos de coordinación y control que permitirán la sincronización de los drones y la asignación de tareas específicas a cada uno. Este proyecto no contempla la programación del enjambre de drones, solo la implementación de drones coordinados autónomos.

En cuanto a los transceptores, se considerarán diferentes opciones tecnológicas, como WiFi o LoRa, en función de los requerimientos del sistema. Estas tecnologías permitirán la comunicación bidireccional entre los drones y la red de sensores, facilitando la transferencia de datos y comandos de control.

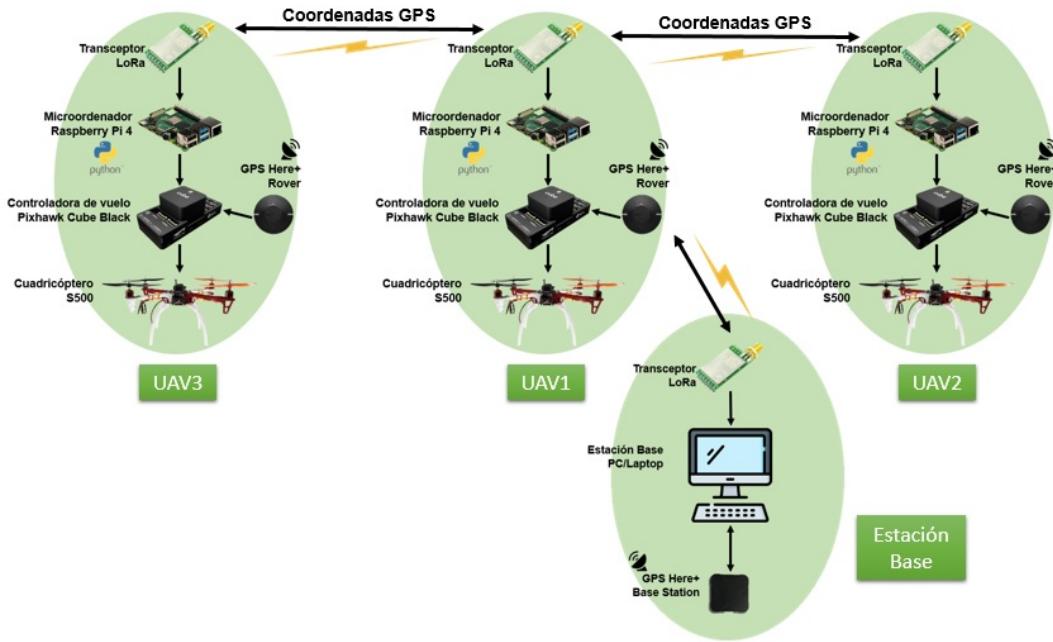


Figura 4.1: Diseño y arquitectura del sistema de tres drones coordinados.

4.2. Subsistema 2: WSN y el Monitoreo Remoto en Especies en Peligro de Extinción

En esta etapa, se propone el diseño de una red inalámbrica de sensores (WSN) de tipo clusterizada basada en un modelo matemático capaz de predecir el consumo energético de la red, determinar el número máximo de nodos y estimar la capacidad de transmisión de datos. Una vez realizado el análisis, se procederá a implementar la WSN. Para ello, se seleccionarán cuidadosamente los dispositivos y el sistema de comunicaciones más adecuado, teniendo en cuenta la compatibilidad con el subsistema de comunicación del enjambre de drones. Además, se desarrollará una interfaz temporal de adquisición de datos para monitorear, identificar y procesar la información proveniente de los animales. Por otro lado, se llevará a cabo una investigación para adquirir o desarrollar los sensores y dispositivos apropiados para la especie seleccionada. Se considerarán los tipos de datos más relevantes para el monitoreo, según la

experiencia de los expertos en el campo. Entre estos datos se encuentran las coordenadas GPS del animal durante un periodo de tiempo en caso de utilizar collares, así como las condiciones ambientales obtenidas directamente de un sensor de temperatura a la WSN. En la parte del monitoreo, se empleará exclusivamente el sistema de posicionamiento global (GPS) para geolocalizar a los animales. Las posiciones registradas se almacenarán en tarjetas de memoria SD dentro de los sensores líderes de cada clúster. Durante los vuelos del enjambre de drones para recolectar información, los datos se enviarán junto con la información de la temperatura y el video de las cámaras trampa.



Figura 4.2: Ejemplos de tipos de collares de seguimiento de animales.

La arquitectura de los clúster heads, encargados de coordinar la comunicación entre los drones y la WSN, se diseñará considerando la selección de una tecnología específica, la cual se determinará en etapas posteriores del proyecto. Algunas alternativas a considerar podrían ser el uso de dispositivos Raspberry Pi, Arduino o módulos de comunicación especializados.

4.3. Subsistema 3: Integración, Esquemas y Entorno de Visualización

En esta etapa se planea integrar los dos subsistemas anteriores mediante un funcionamiento como se puede ver en las Figuras 4.3a y 4.3b donde se pone de ejemplo el monitoreo de jaguares (el escogimiento de la especie a monitorear dependerá del análisis que se realizará con biólogos

de campo). En la Figura 4.3a, la WSN se encarga de recopilar los datos de la cámara trampa, del sensor de temperatura y de los GPS que llevará la especie animal en un dispositivo de seguimiento que dependerá del tamaño de la especie que se planea monitorear. En la Figura 4.3b, el enjambre de drones se dirigirá a la WSN para recolectar todos estos datos y ser llevados a la BS.

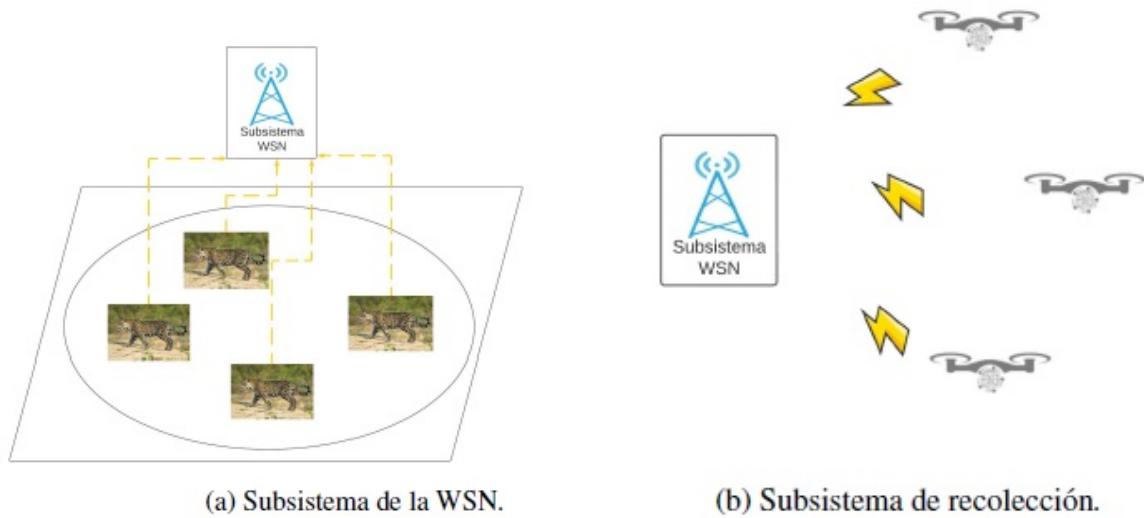


Figura 4.3: Funcionamiento del sistema.

Como parte de la propuesta de solución, se llevará a cabo una comparación energética y de tiempo de vuelo del enjambre de drones entre dos esquemas de recolección de datos.

En el primer esquema propuesto, cada uno de los drones del enjambre se dirigirá de manera individual hacia cada WSN con el propósito de recolectar los tres tipos de información almacenados: video, coordenadas GPS y temperatura ambiental. Esta configuración se representa visualmente en la Figura 4.4.

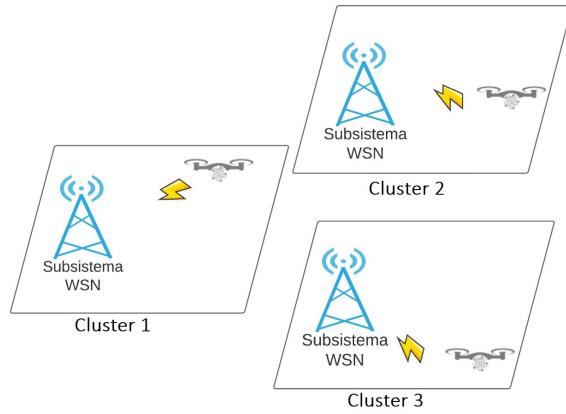


Figura 4.4: Primer esquema de recolección (enjambre separado).

En el segundo esquema, los drones del enjambre se dirigirán conjuntamente a cada WSN, sin embargo, a cada dron se le asignará la responsabilidad de recolectar un tipo específico de información. De esta forma, un dron se encargará exclusivamente de recolectar video, otro se encargará de recopilar las coordenadas GPS, y el tercer dron se focalizará en la obtención de datos relacionados con temperatura ambiental. Este enfoque de asignación de tareas se repetirá en cada WSN, tal como se ilustra en la Figura 4.5.

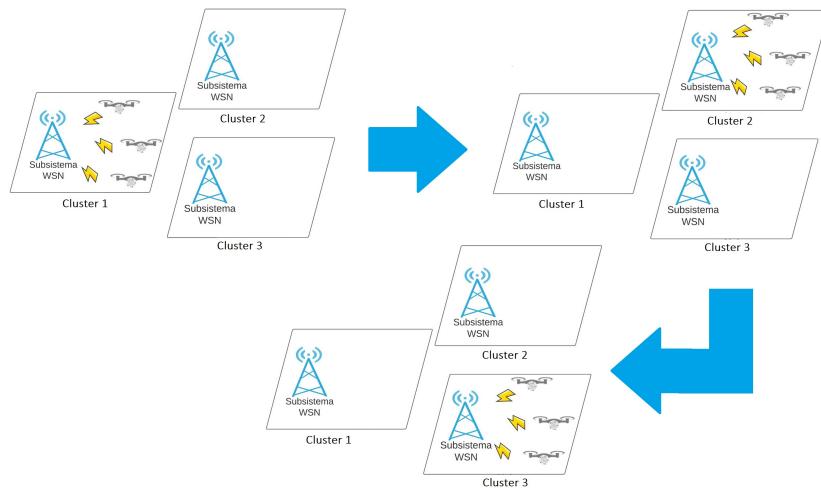


Figura 4.5: Segundo esquema de recolección (enjambre en conjunto).



La comparación entre estos dos esquemas se realizará a través de un análisis matemático que proporcionará información valiosa para determinar cuál de ellos es más eficiente y adecuado para la propuesta de solución de este trabajo, al realizar mediciones y análisis precisos de la energía consumida por cada dron durante la recolección de datos, así como el tiempo requerido para completar la tarea en cada esquema. Por todo esto, el uso de este enjambre presentará numerosas ventajas. En primer lugar, permitiría a los biólogos de campo recolectar datos de manera más rápida y efectiva, al cubrir grandes áreas en poco tiempo y acceder a lugares que serían difíciles o peligrosos para los humanos. En segunda instancia, reducirá la perturbación a los hábitats naturales, ya que los drones pueden sobrevolar el hábitat a una gran altura para diluir el ruido que sus motores generan; además, no alteran la vegetación y minimizan la presencia humana en las áreas de estudio. Esto es especialmente importante en el caso de especies que se encuentran en un alto riesgo de desaparecer, donde cualquier perturbación adicional puede tener un impacto negativo en su comportamiento y hábitat. Otra ventaja del uso de enjambres de drones es la posibilidad de programar rutas de vuelo preestablecidas, permitiendo así una mayor precisión en la recolección de datos al seguir una ruta específica para recolectar datos y eliminar la necesidad de contar con un operador de drones [?]. Una vez que el enjambre ha recopilado los datos, es necesario contar con un entorno adecuado para su análisis y almacenamiento. En este sentido, se propone implementar una solución basada en una base de datos en la nube, que permitirá disponer de un sistema distribuido de fácil acceso, el cual contará con:

- La apertura de una aplicación web que brinde la posibilidad a especialistas de visualizar los datos ya procesados.
- La creación de distintos perfiles de usuario, con acceso diferenciado a los datos almacenados y no analizados.
- Acceso a archivos multimedia como videos, imágenes y audios recabados, que estarán almacenados en una base de datos.



Esta infraestructura garantizará una gestión eficiente de los datos y la seguridad de los mismos para su posterior disponibilidad en el análisis y consulta. Para lograrlo, se propone utilizar los servicios de AWS (Amazon Web Services), los cuales ofrecen una amplia gama de herramientas y capacidades. A continuación, se mencionan algunos servicios de AWS que se plantea utilizar:

- **Amazon Cognito:** Un servicio de autenticación y autorización para las aplicaciones web y móviles [?]. Este servicio permitirá autenticar y autorizar los accesos a la información, asignando a cada usuario un nivel de acceso específico.
- **Amazon S3:** Un servicio de almacenamiento de datos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Permite almacenar y proteger cualquier cantidad de datos para diversos casos de uso, como lagos de datos, aplicaciones nativas en la nube y aplicaciones móviles [?]. Se utilizará para el almacenamiento de archivos multimedia, tales como los videos de las cámaras trampa.
- **Amazon DynamoDB:** Un servicio de base de datos NoSQL totalmente administrado que proporciona un desempeño rápido y predecible con una escalabilidad perfecta. Permite crear tablas de base de datos para almacenar y recuperar cualquier cantidad de datos y atender cualquier nivel de tráfico de solicitudes. Al distribuir automáticamente los datos y el tráfico de la tabla entre un número suficiente de servidores para gestionar la capacidad de solicitud especificada por el cliente y la cantidad de datos almacenados, al mismo tiempo que mantiene un desempeño rápido y constante [?]. Se utilizará para la administración de los archivos multimedia e información adicional sobre los datos recabados y procesados. Usando la API GraphQL, para la búsqueda de información dentro de DynamoDB, mediante AppSync; La utilización de la API GraphQL es una opción para llevar a cabo búsquedas de información dentro de DynamoDB a través de AppSync, que es un servicio proporcionado por AWS específicamente diseñado para ejecutar consultas sobre la información almacenada en la base de datos.
- **AWS Amplify:** Una solución completa que permite a los desarrolladores web y móviles crear, enviar y alojar aplicaciones de pila completa en AWS de forma sencilla [?].

Facilitará la interconexión entre las bases de datos, el Back-end y el Front-end, sin requerir experiencia en la nube. Además, se puede considerar la creación de una página web utilizando HTML5, CSS y JavaScript.

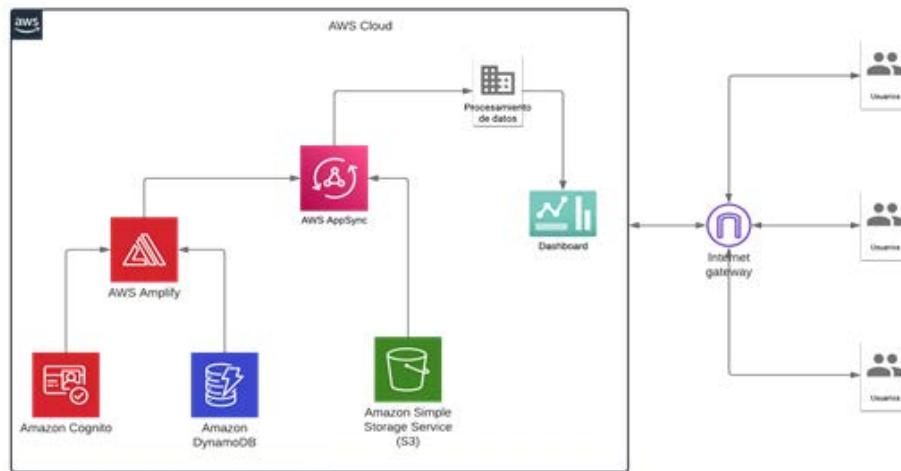


Figura 4.6: Arquitectura de propuesta para el almacenamiento y procesamiento de datos.

CAPÍTULO 5

Objetivos

5.1. Objetivo General

Implementar un sistema integral de recolección, almacenamiento y visualización de información relevante para el monitoreo de una especie en peligro de extinción, la cual se determinará en etapas posteriores del proyecto en colaboración con expertos en conservación y protección animal. Para ello, se llevará a cabo el diseño y reprogramación del enjambre de drones, así como el diseño e implementación de la WSN y del servicio en la nube, eligiendo el esquema de recolección de acuerdo al modelo energético.

5.2. Objetivos Particulares

5.2.1. Proyecto Terminal 1

- Analizar los componentes necesarios para el subsistema de la WSN, así como el tipo de red y los métodos de formación y transmisión de datos entre los sensores más adecuados



dos, así como desarrollar el modelo y simulación adecuado y para la caracterización del desempeño de la red.

- Analizar y elegir los componentes del subsistema de adquisición de datos de la WSN y enjambre, así como de la BS a enjambre basado en el modelo de comunicaciones de los enlaces y elegir el protocolo de adquisición entre enjambre y WSN que mejor convenga de acuerdo con el modelo y simulación.
- Diseñar la implementación del subsistema de enjambre, definiendo el número de drones, tiempo de vuelo y distancia de vuelo desde BS.
- Diseñar el algoritmo para programar el método de formación de clúster y transmisión de paquetes entre sensores.
- Diseñar el subsistema de adquisición de datos entre el enjambre y la WSN, así como la WSN y la BS eligiendo los componentes más adecuados para adquirir, almacenar y entregar la información de los sensores a la BS de acuerdo con el sistema de comunicación y protocolos de comunicación más adecuados.
- Diseñar el subsistema de la estación base para el enlace de comunicación entre enjambre y BS, así como diseñar la interfaz para el monitoreo, configuración, programación de rutas, y en su caso, almacenamiento, procesamiento y visualización de la información.

5.2.2. Proyecto Terminal II

- Implementar y configurar el sistema de telemetría entre los UAVs del enjambre.
- Implementar el subsistema de la WSN.
- Implementar el subsistema de recolección de datos.
- Implementar el subsistema de BS.
- Establecer y configurar un servidor web con una base de datos.

CAPÍTULO 6

Alcances

Los alcances del proyecto se centran en el desarrollo de un sistema de monitoreo remoto utilizando un enjambre de drones y una red inalámbrica de sensores (WSN). El enjambre realizará vuelos de forma autónomos y coordinados con un vuelo estable, siguiendo una trayectoria preestablecida y sin ajustes en tiempo real. Se considerará que la ruta de vuelo es libre de obstáculos y no se implementará la función de evitar obstáculos de forma automática en los drones. Es importante destacar que, para este proyecto, no se plantea la incorporación de características de robustez y resistencia en el enjambre de drones.

La elección del sistema de comunicaciones se diseñará para garantizar la transferencia de datos de manera adecuada entre los nodos de la WSN y los drones, utilizando tecnologías inalámbricas adecuadas para el entorno y las condiciones de operación, aunque estará sujeta a la disponibilidad de los transceptores proporcionados por los fabricantes y proveedores.

Los sensores utilizados para la recolección de datos se seleccionarán en función del tipo de animal que se pretenda monitorear, y se tiene previsto adquirirlos, aunque no se descarta la posibilidad de construirlos, si es necesario. También, se prevé la apertura de un servidor web



con una base de datos para almacenar toda la información recolectada por los drones, permitiendo a los especialistas acceder y analizar los datos mediante herramientas de procesamiento adecuadas.

Se considerará un escenario de pruebas acotado, determinado por el tamaño y las características del hábitat de las especies objetivo. El número de drones en el enjambre y los nodos en la WSN se establecerán en función de los recursos disponibles y las necesidades específicas del monitoreo. Y aunque se buscará la vinculación con algún programa de protección animal para realizar pruebas finales, se reconoce que estas pruebas estarán sujetas a posibles restricciones y limitaciones, por lo que no se garantiza la implementación directa del sistema en dicha especie.

CAPÍTULO 7

Estado del Arte

El uso de drones en la conservación de especies en peligro de extinción ha ido en aumento en los últimos años, ya que ofrecen una forma no invasiva y eficiente de monitorear y proteger animales salvajes y sus hábitats. A continuación, se presentan algunos de los desarrollos más recientes en este campo, junto con algunas referencias relevantes:

1. **Drones para el monitoreo de ballenas:** Los drones se han utilizado para monitorear las poblaciones de ballenas desde el aire, lo que permite a los investigadores obtener datos más precisos sobre su tamaño, comportamiento y ubicación. Los drones también se han utilizado para estudiar la salud de las ballenas y para recolectar muestras de exhalación para su análisis [?].
2. **Uso de huellas dactilares térmicas para buscar lémures:** Mediante el uso de cámaras infrarrojas montadas en drones, los investigadores pueden detectar y distinguir diferentes especies animales en función de su huella térmica única. Después se emplea el aprendizaje automático para analizar los datos térmicos e identificar las especies [?].



3. **Drones para la conservación de aves:** Los drones se han utilizado para monitorear las poblaciones de aves en peligro de extinción y para estudiar su comportamiento y hábitat. Al igual que para estudiar las aves migratorias y para monitorear la anidación de aves en áreas de difícil acceso [?].
4. **Drones para la protección de elefantes:** Los drones se han utilizado para monitorear y proteger a los elefantes de la caza furtiva y de la destrucción de su hábitat. Los drones pueden detectar a los cazadores furtivos y alertar a los guardabosques, así como pueden ser utilizados para identificar y monitorear los movimientos de los elefantes [?].
5. **Drones para la protección de rinocerontes:** Los drones se han utilizado para monitorear y proteger a los rinocerontes de la caza furtiva y de la destrucción de su hábitat. Los drones pueden ser utilizados para patrullar las áreas donde los rinocerontes viven, para detectar y rastrear a los cazadores furtivos y para recolectar datos sobre la población de rinocerontes [?].
6. **Drones para proteger a delfines Maui:** Nueva Zelanda ha lanzado un programa piloto con el objetivo de salvar a los últimos 63 delfines Maui que quedan en el mundo al utilizar un veloz y poderoso dron equipado con inteligencia artificial para obtener datos que ayuden a evitar la extinción de esta especie y proteger su hábitat [?].
7. **Protección de nidos de tortugas marinas:** Los drones se utilizan para monitorear y proteger los nidos de tortugas marinas, identificando posibles amenazas y tomando medidas de conservación [?].
8. **Monitoreo de poblaciones de aves rapaces:** Los drones se utilizan para monitorear poblaciones de aves rapaces, como águilas y halcones, de una manera no invasiva y con una alta resolución espacial para obtener datos sobre su distribución y comportamiento [?].
9. **Lucha contra la caza furtiva:** Los drones se utilizan para detectar y disuadir la caza furtiva de especies en peligro de extinción, como rinocerontes y elefantes, mediante la

vigilancia y patrullaje de áreas protegidas en tiempo real, lo que ayuda a las autoridades a tomar acciones rápidas para detener la caza ilegal [?].

10. **Detección de incendios forestales:** Los drones se utilizan para la detección temprana de incendios forestales que representan una amenaza para la vida silvestre y su hábitat. Los drones equipados con cámaras térmicas y sensores pueden volar sobre áreas forestales y detectar focos de incendio [?].
11. **Preservación de bosques:** Especialistas del Instituto Politécnico Nacional (IPN) del Laboratorio de Sistemas Autónomos Ligeros del Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (CICATA) están utilizando drones y cámaras multiespectrales para analizar la estructura de los árboles, cuantificar los cambios en la vegetación y determinar el contenido de carbono. Teniendo gran importancia debido a los compromisos internacionales para reducir las emisiones de gases de efecto invernadero [?].



Figura 7.1: Ejemplos del uso de UAVs para el monitoreo de especies en peligro de extinción y/o de su hábitat.



Tabla 7.1: Comparación entre la propuesta de solución y el estado del arte.

Proyecto	Tecnología utilizada	Resultados	Beneficios	Limitaciones
Propuesta de solución este proyecto	WSN y enjambre de drones.	Brindar una mayor comprensión de la situación actual de la especie en peligro de extinción mediante un monitoreo activo, mejorando en el conocimiento tecnológico de las técnicas actuales.	Conservación de la especie.	Los métodos de monitoreo pueden no ser 100 % precisos, lo que puede limitar la capacidad de obtener una imagen precisa de la población.
Drones para el monitoreo de ballenas	Cámaras, análisis de exhalación.	Datos precisos sobre tamaño, comportamiento y ubicación de ballenas, estudios de salud.	Datos más precisos.	Dificultad para seguir a ballenas en movimientos rápidos.
Uso de huellas dactilares térmicas para buscar lémures	Cámaras infrarrojas, aprendizaje automático.	Identificación de especies animales mediante huellas térmicas, análisis mediante aprendizaje automático.	Identificar especies.	Limitado a especies animales con huellas dactilares térmicas detectables.
Drones para la conservación de aves	Cámaras.	Monitoreo de poblaciones de aves, estudios de comportamiento y hábitat, monitoreo de anidación en áreas de difícil acceso	Estudiar comportamiento y hábitat.	Riesgo de perturbación de las aves durante el monitoreo.

Proyecto	Tecnología utilizada	Resultados	Beneficios	Limitaciones
Drones para la protección de elefantes	Cámaras, alertas, seguimiento.	Monitoreo y protección de elefantes de la caza furtiva, detección de cazadores furtivos.	Proteger a los elefantes de la caza furtiva.	Necesidad de una infraestructura de vigilancia y respuesta rápida para aprovechar al máximo la detección de cazadores furtivos.
Drones para la protección de rinocerontes	Cámaras, patrullaje, seguimiento.	Monitoreo y protección de rinocerontes de la caza furtiva, recolección de datos sobre la población de rinocerontes.	Proteger a los rinocerontes de la caza furtiva.	Limitaciones en la cobertura de áreas extensas de terreno con drones.
Protección de delfines Maui	Dron con inteligencia artificial.	Obtención de datos para evitar la extinción y proteger el hábitat de los delfines Maui.	Evitar la extinción de la especie.	Dependencia de la climatología y las condiciones del mar.
Protección de nidos de tortugas marinas	Cámaras.	Monitoreo y protección de nidos de tortugas marinas, identificación de amenazas y medidas de conservación.	Identificar posibles amenazas.	La interferencia humana en las playas puede aumentar el riesgo de perturbaciones en los nidos de las tortugas.



Proyecto	Tecnología utilizada	Resultados	Beneficios	Limitaciones
Monitoreo de poblaciones de aves rapaces	Cámaras.	Monitoreo de poblaciones de aves rapaces, datos sobre su distribución y comportamiento.	Obtener datos sobre su distribución.	Restricciones en la capacidad de volar en áreas montañosas y boscosas con drones.
Lucha contra la caza furtiva	Dron con cámaras, vigilancia.	Detección y disuasión de la caza furtiva mediante vigilancia y patrullaje de áreas protegidas.	Disuadir la caza furtiva.	Limitaciones en la capacidad de cubrir grandes áreas de terreno con drones.
Detección de incendios forestales	Dron con cámaras térmicas y sensores.	Detección temprana de incendios forestales mediante vuelos sobre áreas forestales y detección de focos de incendio.	Detectar focos de incendio forestal.	Dependencia de la disponibilidad de drones y operadores para la detección temprana.

CAPÍTULO 8

Marco Teórico

8.1. Especies en Peligro de Extinción

Una especie se considera en peligro de extinción cuando se encuentra en un alto riesgo de desaparecer completamente de su hábitat natural [?]. La Unión Internacional para la Conservación de la Naturaleza (IUCN, por sus siglas en inglés) ha establecido una serie de categorías de conservación para clasificar el estado de las especies en función de su riesgo de extinción, y se basa en criterios como el tamaño y tendencia de la población, la distribución geográfica, la calidad del hábitat, entre otros factores [?].

La IUCN utiliza las siguientes categorías de conservación:

- **Extinta:** Se refiere a una especie que ya no existe en la naturaleza ni en cautiverio.
- **Extinta en estado silvestre:** se refiere a una especie que todavía existe en cautiverio pero que ya no se encuentra en su hábitat natural.
- **En peligro crítico:** Se refiere a una especie que enfrenta un riesgo extremadamente



alto de extinción en la naturaleza.

- **En peligro:** Se refiere a una especie que enfrenta un riesgo muy alto de extinción en la naturaleza.
- **Vulnerable:** Se refiere a una especie que enfrenta un riesgo alto de extinción en la naturaleza.
- **Casi amenazada:** Se refiere a una especie que no cumple con los criterios para ser incluida en una categoría de mayor riesgo de extinción, pero que podría estar en peligro en el futuro.
- **Preocupación menor:** Se refiere a una especie que no cumple con los criterios para ser incluida en una categoría de mayor riesgo de extinción.
- **Datos insuficientes:** Se refiere a una especie que no ha sido evaluada suficientemente para determinar su estado de conservación.



Figura 8.1: Categorías de conservación de la IUCN.

De acuerdo con información proporcionada por la Unión Internacional para la Conservación de la Naturaleza, actualmente se estima que alrededor de 5,200 especies están en peligro de extinción, lo que representa un porcentaje significativo de la biodiversidad. En particular, se



estima que el 25 % de los mamíferos y anfibios, el 34 % de los peces, el 20 % de los reptiles y el 11 % de las aves se encuentran en peligro de extinción. Cabe destacar que estos datos se refieren únicamente a los animales vertebrados conocidos y que la cifra aumenta cuando se incluyen las especies vulnerables, es decir, aquellas que aunque no se consideran en peligro de extinción, han experimentado una rápida disminución en su población o han sufrido una considerable pérdida de su hábitat [?].

8.2. Métodos Tradicionales de Monitoreo Remoto

El monitoreo remoto se ha utilizado durante décadas para la recopilación de datos ambientales y de la presencia de animales. Los métodos tradicionales de monitoreo remoto incluyen técnicas como el uso de cámaras trampa, avistamiento visual, telemetría y radiotelemetría [?]. Las cámaras trampa son cámaras fotográficas que se activan mediante sensores de movimiento cuando detectan movimiento en su rango de visión. Estas cámaras se colocan en lugares estratégicos y permiten obtener imágenes de la fauna silvestre sin interferir en su comportamiento. Asimismo, las cámaras trampa pueden utilizarse para monitorear especies nocturnas o de hábitos esquivos [?].

El avistamiento visual consiste en la observación directa de las especies de interés por parte de un observador capacitado. Este método se utiliza para obtener información sobre el comportamiento, hábitat y distribución de las especies. Sin embargo, este método puede ser limitado por la capacidad del observador para detectar y registrar la presencia de las especies [?].

La telemetría es un método que utiliza dispositivos electrónicos para monitorear la ubicación y movimiento de las especies. Estos dispositivos pueden ser colocados en los animales y permiten la recolección de datos en tiempo real. La telemetría es útil para el monitoreo de especies migratorias o de comportamiento errático [?].

La radiotelemetría es un método de telemetría que utiliza ondas de radio para transmitir datos desde los dispositivos colocados en los animales hasta una antena receptora. Este método es útil para el monitoreo de animales que se mueven en áreas extensas o de difícil acceso [?].



8.3. Enjambre de Drones

Los enjambres de drones son un grupo de drones que trabajan juntos de manera coordinada con capacidad de tomar decisiones entre ellos, en lugar de estar individualmente controladas por un humano para realizar una tarea específica. Los drones de un enjambre son una tecnología se pueden clasificar según sus características y aplicaciones prácticas, tal y como se puede observar en la Tabla 8.1: Clasificación de drones según características y aplicaciones.

Tabla 8.1: Clasificación de drones según características y aplicaciones.

Tamaño							
Nano <30 mm	Micro 30–100 mm	Mini 100–300 mm	Pequeño 300–500 mm	Medio 500 mm–2 m	Grande 2 m		
Peso máximo al despegue (MTOW)							
<0.5 Kg		0.5–5 Kg		5–25 Kg			
Alcance (Distancia/Tipo de Operación)							
Corto alcance: <0.5 millas Línea De Vista Visual (VLOS)	Medio alcance: 0.5–5 millas Línea de Vista Visual Extendida (EVLOS)		Largo alcance: >5 millas Fuera de la Línea de Vista Visual (BVLOS)				
Ala							
Rotores simples y dobles	Ala giratoria Multi-rotor			Ala fija			
	Tricóptero	Cuadricóptero	Hexacóptero	Octocóptero	Ala baja Ala media Ala alta Ala delta		
Ensamblaje							
Listo para Volar (RTF)		Enlazar y Volar (BNF)		Casi Listo para Volar (ARF)			
Aplicaciones							
Logística		Ingeniería civil	Asistencia en desastres	Agricultura de precisión			
Búsqueda y rescate		Patrimonio	Recursos naturales	Cumplimiento de la ley			
Gestión de la vida silvestre		Militar	Ocio	Inspección industrial			
Pronóstico del tiempo		Asistencia en desastres	Fotografía y cine aéreo	Arqueología			

En un enjambre de drones, cada uno de los drones puede tener su propio conjunto de sensores (Tabla 8.2: Clasificación de sensores y dispositivos que se pueden acoplar a drones.), habilidades y tareas, y al estar en grupo pueden trabajar en conjunto para realizar objetivos complejos que un solo dron no podría realizar por sí solo. Los enjambres de drones pueden ser programados para seguir diferentes patrones de vuelo, como vuelos en formación, vuelos en círculos o vuelos en espiral. Sumado a esto, pueden ser programados para comunicarse entre sí y con una



estación base para coordinar su comportamiento y lograr los objetivos preestablecidos [?].



Tabla 8.2: Clasificación de sensores y dispositivos que se pueden acoplar a drones.

Instrumento		Tipo de Sensor	Resolución Espacial	Resolución Espectral	Peso	Costos
Sensores de imagen	RGB visibles	Pasivo	Muy alto 1–5 cm/píxel	Bajo(3 bandas)	Bajo < 0,5 kg	Bajo 100~1000
	Infrarrojo cercano (NIR)	Pasivo	Muy alto 1–5 cm/píxel	Bajo(3 bandas)	Bajo < 0,5 kg	Bajo 100~1000
	Multiespectral	Pasivo	Alto 5–10 cm/píxel	Medio(5–12 bandas)	Medio 0,5–1 kg	Medio 1000~10,000
	Hiperespectral	Pasivo	Alto 5–10 cm	Alto (> 50–100 bandas)	Medio 0,5–1 kg	Alto 10,000~50,000
	Térmico	Pasivo	Medio 10–50 cm/píxel	Bajo 1 banda	Medio 0,5–1 kg	Medio 1000~10,000
Sensores de alcance	Escáneres láser (LiDAR)	Activo	Muy alto 1–5 cm/píxel	Bajo 1-2 bandas	Alto 0,5–5 kg	Alto 10,000~50,000
	Radares de apertura sintética (SAR)	Activo	Medio 10–50 cm/píxel	Bajo 1 banda	Alto > 5 kg	Muy alto > \$50,000
Otros sensores y dispositivos						
Sensores atmosféricos		Temperatura, presión, viento, humedad				
Sensores químicos		Gas, geoquímica				
Sistemas de posición		Ultrasonido, infrarrojos, radiofrecuencia, GPS				
Dispositivo de grabación		Cámaras, micrófonos				
Dispositivos de muestreo		Muestreo de agua, aerobiológico, microbiológico				
Otros dispositivos		Carga, fumigación, esparcidor de semillas				

Para implementar un enjambre de drones eficaz, es importante tener en cuenta varios factores, como la capacidad de los drones para comunicarse entre sí y de coordinar sus movimientos, la duración de la batería, la capacidad de carga, la precisión de la navegación y la capacidad de evitar obstáculos o peligros potenciales.

8.4. Protocolos de Comunicación

Los protocolos de comunicación son conjuntos de reglas y estándares que definen el intercambio de información entre dispositivos en una red de comunicaciones [?]. Se pueden utilizar diversos protocolos de comunicación, según los requisitos de la aplicación y el tipo de transceptor utilizado. A continuación, se presentan algunos de los protocolos de comunicación más comunes que podrían ser utilizados en este tipo de sistema:

- **IEEE 802.11 (WiFi):** Este protocolo se utiliza en redes inalámbricas de área local (WLAN) y permite la transmisión y recepción de datos a través de una conexión inalámbrica de alta velocidad. Los dispositivos que utilizan este protocolo operan en las bandas de frecuencia de 2.4 GHz y 5 GHz, y pueden transmitir datos a velocidades de hasta varios gigabits por segundo [?].



- **Bluetooth:** Este protocolo se utiliza para la comunicación inalámbrica de corto alcance entre dispositivos, como auriculares, altavoces y dispositivos móviles. Los dispositivos Bluetooth operan en la banda de frecuencia de 2.4 GHz y pueden transmitir datos a velocidades de hasta varios megabits por segundo [?].
- **Zigbee:** Este protocolo de área personal de baja potencia y alta eficiencia energética, está diseñado para permitir la comunicación entre dispositivos de bajo consumo de energía en redes de sensores y otros dispositivos inteligentes, otorgándoles una duración de batería prolongada. Los dispositivos que utilizan este protocolo operan en la banda de frecuencia de 2.4 GHz [?].
- **LoRaWAN:** Este protocolo ofrece una larga vida útil de la batería, una alta eficiencia energética y seguridad de red integrada para dispositivos electrónicos en áreas remotas. Se utiliza en sistemas de comunicación de largo alcance y baja potencia. Los dispositivos que utilizan este protocolo operan en las bandas de frecuencia de 868 MHz o 915 MHz [?].
- **MQTT:** Este protocolo se utiliza para la comunicación de mensajes entre dispositivos en una red, y es especialmente adecuado para aplicaciones IoT (Internet de las cosas) y sistemas de sensores. MQTT es un protocolo de publicación/suscripción, lo que significa que los dispositivos pueden publicar datos en un canal específico y otros dispositivos pueden suscribirse a ese canal para recibir los datos [?].
- **ALOHA Puro:** Este protocolo es uno de los primeros protocolos de acceso múltiple desarrollados para redes de comunicación. En ALOHA Puro, los dispositivos transmiten los datos de manera asincrónica sin sincronización previa. Si ocurre una colisión, se detecta y se realiza una retransmisión posteriormente. Este protocolo es simple pero puede presentar un alto índice de colisiones, lo que afecta la eficiencia de la comunicación [?].
- **ALOHA Ranurado:** Este protocolo es una mejora del protocolo ALOHA Puro y se



basa en la división del tiempo en ranuras. Cada ranura se asigna a un dispositivo para transmitir sus datos, lo que reduce la probabilidad de colisiones. Sin embargo, puede haber un desperdicio de tiempo si una ranura está vacía [?].

La elección del protocolo de comunicación específico dependerá de los requisitos y las características del proyecto, aunque se considera la utilización del protocolo ALOHA Ranurado debido a su simplicidad y bajo consumo energético.

8.5. Transceptores

Los transceptores son dispositivos que se utilizan para la transmisión y recepción de señales en una red de comunicaciones inalámbrica. Estos dispositivos están diseñados para operar en una banda de frecuencia específica y utilizan un protocolo de comunicación específico para enviar y recibir datos [?]. En una red inalámbrica, los transceptores se utilizan para enviar y recibir datos a través del aire. Estos dispositivos convierten la señal de datos en ondas de radio que se pueden transmitir a través del aire, y a su vez, pueden recibir señales de radio y convertirlas en datos utilizable [?]. Existen varios tipos de transceptores que se utilizan comúnmente en redes inalámbricas, incluyendo:

- **Transceptores WiFi (Wireless Fidelity):** Estos dispositivos se utilizan para la conexión inalámbrica a Internet a través de una red WiFi con un radio promedio de 50 metros [?].
- **Transceptores Bluetooth:** Los transceptores Bluetooth se utilizan para conectar dispositivos electrónicos cercanos, con un radio promedio de 10 metros [?].
- **Transceptores Zigbee:** Estos dispositivos se utilizan en redes inalámbricas de sensores de baja potencia y bajo costo, contando con un radio promedio de 60 metros [?].
- **Transceptores LoRa (Long Range):** Estos dispositivos utilizan la modulación de espectro ensanchado para aumentar la distancia de transmisión y reducir el consumo de



energía de los dispositivos de la red, estos pueden transmitir a distancias de decenas de kilómetros en condiciones óptimas [?].

- **Transceptores de radiofrecuencia (RF):** Estos dispositivos se utilizan en sistemas de comunicación de largo alcance, como sistemas de radio de dos vías o sistemas de telemetría. Los transceptores de RF utilizan frecuencias de radio específicas para la transmisión y recepción de datos; en un entorno sin obstáculos y sin interferencias estos cuentan con un radio de cobertura de varios cientos de metros a varios kilómetros [?].

8.6. Red Inalámbrica de Sensores (WSN)

Las redes inalámbricas de sensores (WSN) son sistemas compuestos por nodos inalámbricos que se encargan de recolectar y transmitir datos de un entorno físico a una estación base [?]. Estos nodos están equipados con sensores que les permiten medir diferentes parámetros, como temperatura, humedad, presión, luz, sonido, etc. Las WSN se utilizan en una variedad de aplicaciones, incluyendo monitoreo ambiental, control de procesos industriales, monitoreo de infraestructuras, entre otros. En particular, las WSN son muy útiles en aplicaciones de monitoreo ambiental, ya que permiten obtener información sobre el comportamiento de las especies que se encuentran en peligro de extinción, su hábitat y otros factores relevantes para su conservación.

El funcionamiento de una WSN se basa en la comunicación inalámbrica entre los nodos y la estación base. Los nodos inalámbricos están equipados con un transceptor que les permite comunicarse con otros nodos cercanos y con la estación base. Los datos medidos por los sensores se transmiten de nodo en nodo hasta que llegan a la estación base, que es la encargada de procesarlos y almacenarlos [?].

Para que una WSN sea eficiente, es necesario que se tomen en cuenta factores como el consumo energético, la escalabilidad, la robustez y la seguridad. Por ejemplo, los nodos deben ser diseñados de manera que consuman la menor cantidad de energía posible, para prolongar la duración de las baterías y reducir los costos de mantenimiento. Al mismo tiempo, es impor-



tante que la red sea escalable con el fin de poder adaptarse a diferentes tamaños de área de monitoreo y a diferentes requerimientos de densidad de sensores [?].

8.7. Esquemas de Recolección

La recolección de datos en una WSN es un proceso crítico para asegurar la precisión y eficiencia de la recopilación de información. Existen varios esquemas de recolección de datos en WSN [?], entre los que se incluyen:

- **Esquema de recolección basado en eventos:** Este esquema de recolección se basa en la detección de eventos específicos en el entorno, como un cambio en la temperatura, la presencia de un objeto en movimiento, entre otros. Cuando se detecta un evento, el nodo correspondiente envía una alerta al nodo central para su posterior procesamiento.
- **Esquema de recolección periódica:** En este esquema, los nodos envían datos al nodo central de forma periódica. Este esquema es fácil de implementar, pero puede generar una gran cantidad de datos innecesarios si no se configura adecuadamente.
- **Esquema de recolección basado en consulta:** En este esquema, el nodo central realiza una consulta a los nodos para obtener información específica. Este esquema permite un mayor control sobre la cantidad de datos recopilados, ya que solo se obtiene la información relevante.
- **Esquema de recolección híbrido:** Este esquema combina los esquemas anteriores para obtener una mejor eficiencia en la recolección de datos. Por ejemplo, se puede utilizar el esquema basado en eventos para detectar eventos importantes y el esquema periódico para recopilar información de forma regular.

Además de estos esquemas de recolección de datos, existen técnicas para mejorar la eficiencia en la transmisión de datos, como la compresión de datos y la reducción del tamaño de los paquetes de datos enviados.



8.8. Estación Base

En un sistema de drones, la estación base es un componente crítico que permite la comunicación entre los drones y el usuario o la red central. La estación base puede ser una unidad terrestre o aérea que se encarga de controlar y supervisar el movimiento y la posición de los drones.

En términos de hardware, la estación base puede contar con una antena de alta ganancia que permite una comunicación bidireccional confiable con los drones. También puede tener un sistema de posicionamiento global (GPS) para rastrear la ubicación de los drones y un enlace de comunicación de datos para transmitir y recibir datos de los drones [?].

En cuanto a la ubicación de la estación base, depende de la aplicación específica y del área de cobertura requerida. En algunas aplicaciones, como la agricultura de precisión, la estación base puede ser una unidad fija ubicada en un lugar estratégico en el campo. En otras aplicaciones, como el monitoreo de la infraestructura de la ciudad, la estación base puede ser una unidad móvil que se desplaza por la ciudad para proporcionar cobertura en diferentes áreas [?].

8.9. Almacenamiento en la Nube

El almacenamiento en la nube, también conocido como almacenamiento en línea o almacenamiento remoto, es un modelo de almacenamiento de datos en el cual los archivos y datos se guardan en servidores remotos a los que se accede a través de Internet. Este enfoque de almacenamiento ofrece diversas ventajas y beneficios en comparación con el almacenamiento local tradicional [?]. A continuación, se presentan los conceptos y aspectos clave relacionados con el almacenamiento en la nube.

8.9.1. Definición y Características del Almacenamiento en la Nube

El almacenamiento en la nube se refiere a la práctica de guardar y gestionar datos en servidores remotos que son accesibles a través de Internet. Los datos se almacenan y mantienen



en infraestructuras de centros de datos remotos, que pueden ser operados por proveedores de servicios en la nube [?]. Algunas de las características importantes del almacenamiento en la nube incluyen:

- **Acceso remoto:** Los usuarios pueden acceder a sus datos almacenados en la nube desde cualquier lugar y en cualquier momento, siempre que tengan una conexión a Internet.
- **Escalabilidad:** Los servicios de almacenamiento en la nube ofrecen la capacidad de aumentar o reducir la cantidad de almacenamiento según las necesidades del usuario, sin limitaciones físicas.
- **Redundancia y tolerancia a fallos:** Los proveedores de servicios en la nube implementan medidas de seguridad y copias de seguridad para garantizar la integridad y disponibilidad de los datos almacenados, minimizando la posibilidad de pérdida de datos.
- **Compartición y colaboración:** El almacenamiento en la nube permite compartir y colaborar en tiempo real en documentos y archivos con otras personas, facilitando el trabajo en equipo y la colaboración a distancia.

8.9.2. Modelos de Servicio de Almacenamiento en la Nube

Existen diferentes modelos de servicio de almacenamiento en la nube, que determinan la forma en que los usuarios interactúan con el almacenamiento en la nube [?]. Los modelos más comunes son:

- **Almacenamiento en la nube público:** En este modelo, los servicios de almacenamiento en la nube son proporcionados por proveedores externos que ofrecen almacenamiento y servicios relacionados a múltiples clientes. Los datos de los usuarios se almacenan en infraestructuras compartidas, y los usuarios pagan por el espacio de almacenamiento y los servicios utilizados.
- **Almacenamiento en la nube privado:** En este caso, la infraestructura de almacenamiento en la nube es propiedad y está gestionada por una organización específica,



generalmente para uso interno. Proporciona mayor control y seguridad, ya que los datos se almacenan en servidores dedicados y no se comparten con otros usuarios externos.

- **Almacenamiento en la nube híbrido:** Este modelo combina el almacenamiento en la nube público y privado, permitiendo a las organizaciones aprovechar los beneficios de ambos enfoques. Los datos se almacenan en la nube pública y privada según su nivel de sensibilidad y requerimientos de seguridad.

8.9.3. Beneficios y Desafíos del Almacenamiento en la Nube

El almacenamiento en la nube ofrece una serie de beneficios significativos [?], como:

- **Accesibilidad y disponibilidad:** Los datos almacenados en la nube están disponibles en todo momento y desde cualquier ubicación, lo que facilita el acceso y la colaboración.
- **Escalabilidad y flexibilidad:** Los servicios de almacenamiento en la nube permiten ajustar la capacidad de almacenamiento según las necesidades del usuario, evitando la necesidad de invertir en infraestructura adicional.
- **Reducción de costos:** Al utilizar el almacenamiento en la nube, los usuarios pueden evitar gastos de hardware, mantenimiento y administración de servidores locales.
- **Copias de seguridad y recuperación de desastres:** Los proveedores de servicios en la nube implementan medidas de copias de seguridad y recuperación de datos, lo que proporciona una mayor protección contra la pérdida de datos.

Sin embargo, también existen desafíos asociados con el almacenamiento en la nube, como la seguridad de los datos, la privacidad y la dependencia de una conexión a Internet confiable.

8.9.4. Servicios y Tecnologías Populares de Almacenamiento en la Nube

Hay una amplia gama de servicios y tecnologías disponibles en el campo del almacenamiento en la nube. Algunos de los servicios más populares incluyen:



- Amazon S3 (Simple Storage Service).
- Microsoft Azure Blob Storage.
- Google Cloud Storage.
- Dropbox.
- Box.

8.10. Bases de Datos Relacionales y No Relacionales

Una base de datos relacional es un tipo de sistema de gestión de bases de datos (DataBase Management System, DBMS) que organiza los datos en tablas estructuradas llamadas "tablas" o "relaciones". Estas tablas están compuestas por filas y columnas, donde cada fila representa un registro de datos y cada columna representa un atributo o campo específico de esos registros. Algunas de las características más importantes de las bases de datos relacionales son:

- **Tablas y Columnas:** Una base de datos relacional es un conjunto de tablas que contienen datos que se ajustan a categorías predefinidas. Cada tabla contiene una o varias categorías en columnas. Cada fila contiene una instancia única de datos de las categorías definidas por las columnas [?].
- **Clave Primaria y Clave Foránea:** Cada registro es único identificado por una clave primaria. Además, las tablas también deben tener un nombre único. Las relaciones entre distintas tablas se establecen usando claves primarias y claves foráneas [?].
- **Integridad de los Datos:** Este tipo de modelos mantienen una gran integridad gracias a la correctitud y completitud de la información evitando que los datos puedan ser corrompidos y que puedan añadirse nuevas entradas no válidas dentro de la base de datos [?].



- **Relaciones entre Tablas:** Estas relaciones permiten juntar y extraer datos de distintas tablas como si fuera una de sola. Estas relaciones se llevan a cabo usando claves primarias y claves foráneas [?].
- **Índices:** Existen claves índices que permiten acceder de manera más rápida a determinados datos, pudiendo tener diferentes combinaciones para consultar algunos datos o dato en concreto [?].

Ventajas de las Bases de Datos Relacionales:

- **Facilidad de Creación y Acceso:** Este tipo de base de datos son fáciles de crear y accesibles, además de tener la ventaja de ser fáciles de ampliar [?].
- **Gestión de Datos:** Permiten hacer un seguimiento de los inventarios, procesar transacciones de comercio electrónico, gestionar grandes cantidades de información de clientes y mucho más [?].
- **SQL:** Todas las bases de datos relacionales funcionan con un gestor que permite extraer información usando lo que se conoce como SQL (Structured Query Language). Esto permite que haya una unificación y que una base de datos puede ser usada con distintos gestores [?].
- **Simplicidad:** Uno de sus puntos fuertes es su sencillez de uso. El SQL es un lenguaje que se parece mucho al lenguaje natural humano, con lo que, con poco tiempo se puede aprender a manejar este tipo de gestores [?].

Por otra parte, una base de datos no relacional, también conocida como base de datos NoSQL (No Structured Query Language o No Solo SQL), es un tipo de sistema de gestión de bases de datos diseñado para almacenar y administrar datos de manera más flexible a datos que es muy difícil crear una relación entre los datos [?]. Algunas de las características más importantes de las bases de datos NoSQL:



- **Organización de Datos:** En las bases de datos relacionales, la información se organiza de manera estructurada en tablas. Sin embargo, en las bases de datos no relacionales, la información no necesariamente se organiza en estructuras fijas como las tablas. En cambio, utilizan diversas estructuras de datos flexibles, como pares de clave-valor o grafos, para el almacenamiento y la recuperación de datos [?] [?].
- **Esquemas Flexibles:** Las bases de datos no relacionales suelen ofrecer esquemas flexibles, lo que permite un desarrollo más rápido y más iterativo. Son ideales para datos semiestructurados y no estructurados [?].
- **Arquitectura Distribuida:** Estas bases de datos ofrecen una arquitectura distribuida que permite almacenar información en casos en los que las bases de datos relacionales no son capaces de ofrecer el rendimiento y la escalabilidad necesarios [?].
- **Alto Rendimiento:** Las bases de datos NoSQL están optimizadas para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales [?].
- **Escalabilidad:** Las bases de datos NoSQL generalmente están diseñadas para escalar usando clústeres distribuidos de hardware en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado [?].
- **Tipos de Bases de Datos NoSQL:** Existen diferentes tipos de bases de datos no relacionales, en función del método que emplean para almacenar la información. Algunos ejemplos son las bases de datos de clave-valor, las bases de datos documentales, las bases de datos orientadas a grafos, las bases de datos tabulares y las bases de datos orientadas a objetos [?].
- **Compatibilidad con Lenguajes de Consulta:** Muchas bases de datos NoSQL no usan el lenguaje SQL para consultas, o lo usan como un lenguaje secundario. Esto se debe a que las bases de datos NoSQL están diseñadas para manejar grandes volúmenes

de datos y no siempre se ajustan bien a las consultas complejas que se pueden realizar con SQL [?].

- **Alto Soporte de la Comunidad:** A diferencia de las bases de datos relacionales, las bases de datos NoSQL suelen tener un soporte de la comunidad más fuerte, ya que son relativamente más nuevas y están en constante evolución [?].

CAPÍTULO 9

Análisis

9.1. Análisis de los Sensores

El análisis de los sensores constituye un paso esencial en el desarrollo de este proyecto. El objetivo principal de esta actividad es evaluar y comprender las especificaciones técnicas de los sensores disponibles en el mercado que serán utilizados en la WSN.

Durante este análisis, se examinarán aspectos clave de los sensores, como su precisión, rango de medición, resolución, consumo de energía, frecuencia de muestreo y capacidad de transmisión de datos. Además, se considerarán las condiciones ambientales en las que operarán los sensores, como la temperatura, la humedad y la resistencia a la intemperie, para asegurar su idoneidad en el entorno de monitoreo.

El resultado de esta actividad será un informe detallado que incluirá una comparación exhaustiva de los sensores analizados, resaltando sus ventajas y limitaciones. Este informe servirá como base para la selección de los sensores más apropiados en función de los requisitos específicos del proyecto, asegurando así la adquisición de equipos que se ajusten de manera óptima



a los objetivos de monitoreo y a las condiciones del entorno.

Aspectos clave a evaluar en los sensores:

1. **Precisión:** La precisión de los sensores es fundamental, ya que determina cuán cercanas son las mediciones a los valores reales. Se analizará cuán exactos son los sensores en la detección de variables específicas, como temperatura, humedad, presión, o cualquier otro parámetro relevante para el monitoreo de la especie animal.
2. **Rango de Medición:** El rango de medición define los valores mínimos y máximos que un sensor puede detectar con precisión. Se evaluará si el rango de medición de los sensores es adecuado para capturar las variaciones esperadas en los datos del entorno de monitoreo, teniendo en cuenta las características del animal y su hábitat.
3. **Resolución:** La resolución se refiere a la capacidad del sensor para distinguir pequeñas diferencias en los datos. Se examinará cuán fina es la resolución de los sensores, lo que es especialmente importante cuando se requiere una alta sensibilidad para detectar cambios sutiles en las condiciones ambientales.
4. **Consumo de Energía:** El consumo de energía de los sensores es esencial para garantizar la durabilidad de la WSN y minimizar la necesidad de cambios frecuentes de batería. Se analizará cuánta energía consumen los sensores durante la adquisición de datos y en modo de espera, y se buscarán soluciones de eficiencia energética.
5. **Frecuencia de Muestreo:** La frecuencia de muestreo indica con qué regularidad los sensores capturan y transmiten datos. Se determinará la frecuencia de muestreo óptima para obtener datos relevantes sin generar una carga innecesaria en la red y los drones.
6. **Capacidad de Transmisión de Datos:** La capacidad de transmisión de datos se relaciona con la velocidad y la distancia a la que los sensores pueden enviar información a la estación base. Se considerará si los sensores pueden transmitir datos de manera eficiente y oportuna, especialmente cuando se requiere una transmisión en tiempo real.



7. Condiciones Ambientales: Se evaluará la capacidad de los sensores para operar en las condiciones ambientales específicas del hábitat de la especie animal. Esto incluye la resistencia a la intemperie, la capacidad de funcionar en entornos húmedos o polvorientos, y la tolerancia a temperaturas extremas.

9.1.1. Comparación de Sensores

En la búsqueda de los sensores adecuados para este proyecto, es esencial comprender las capacidades y limitaciones de las diferentes opciones disponibles. A continuación, se presenta una descripción técnica detallada de los sensores que se consideran relevantes para el sistema de monitoreo de especies en peligro de extinción. Cada uno de estos sensores desempeña un papel único en la recopilación de datos. Esta comparación exhaustiva se realiza para tomar decisiones informadas al momento de seleccionar los sensores más apropiados a las necesidades del proyecto y al entorno de monitoreo en el que se operará.

- **Humedad:** Los sensores de humedad miden el contenido de humedad en el aire o el suelo. Utilizan componentes sensibles a la humedad, como higroscópicos, para convertir la humedad en señales eléctricas. Estos sensores son económicos y eficientes en cuanto al consumo de energía. Son adecuados para entornos donde la humedad ambiental es un factor relevante para el comportamiento de las especies, como en estudios de hábitats acuáticos o de humedales [?].
- **Temperatura:** Los sensores de temperatura registran la temperatura ambiente. Estos sensores están ampliamente disponibles y son económicos. Pueden utilizarse para monitorear las preferencias térmicas de las especies y detectar cambios de temperatura que puedan afectar su comportamiento [?].
- **Sonido:** Los sensores de sonido capturan las vocalizaciones y otros sonidos ambientales. Son útiles para registrar el comportamiento de las especies que emiten vocalizaciones y para analizar la actividad acústica en el entorno. Sin embargo, la identificación precisa



de las especies a menudo requiere análisis avanzados debido a la diversidad de sonidos en la naturaleza [?].

- **Presión Atmosférica:** Los sensores de presión atmosférica miden la presión del aire y, a menudo, se utilizan para estimar la altitud. Son esenciales para vuelos de drones, ya que ayudan a determinar la altitud y la presión en diferentes alturas. Aunque no proporcionan información específica sobre las especies, son cruciales para la operación segura de drones en la atmósfera [?].
- **Altitud:** Los sensores de altitud son necesarios para medir la altitud sobre el nivel del mar y la elevación en terrenos montañosos. Estos sensores son relevantes cuando se estudian especies que exhiben comportamiento altitudinal o en áreas montañosas. A menudo se utilizan en conjunto con otros sensores para obtener información precisa [?].
- **GPS (Sistema de Posicionamiento Global):** Los dispositivos GPS proporcionan datos de ubicación precisos mediante la recepción de señales de satélites. Son fundamentales para la geolocalización de las especies y la recopilación de datos de movimiento. Sin embargo, consumen más energía en comparación con otros sensores y requieren una línea de visión directa con los satélites [?].
- **Proximidad:** Los sensores de proximidad detectan la cercanía de objetos o animales a través de señales de infrarrojos, ultrasonidos u otros métodos. Son útiles para el monitoreo de trampas o la detección de la presencia de especies en áreas específicas. Sin embargo, tienen un alcance limitado y no proporcionan datos detallados sobre las especies en sí.[?]
- **Cámara Trampa:** Las cámaras trampa capturan imágenes discretas cuando se detecta movimiento. Son efectivas para la identificación visual de especies y su comportamiento. Requieren almacenamiento y procesamiento de imágenes, y pueden verse afectadas por condiciones de luz y clima [?].



- **Imagen Multiespectral:** Los sensores de imagen multiespectral capturan datos en múltiples bandas del espectro electromagnético. Son útiles para analizar características ambientales, como la vegetación y la calidad del suelo. Sin embargo, son costosos y requieren sensores específicos, así como un manejo y análisis más complejos de los datos [?].

Tabla 9.1: Características específicas de los sensores de medición.

Sensor	Características
Humedad	Precisión: $\pm 2\% \text{ HR}$ Rango de Medición: 0 % a 100 % HR Resolución: 0.1 % HR Consumo de Energía: Bajo consumo Frecuencia de Muestreo: Ajustable Capacidad de Transmisión: Hasta 1 Hz Condiciones Ambientales: -40°C a 85°C
Temperatura	Precisión: $\pm 0.5^\circ\text{C}$ Rango de Medición: -40°C a 125°C 0.1 % HR: 0.1°C Consumo de Energía: Bajo consumo Frecuencia de Muestreo: Ajustable Capacidad de Transmisión: Hasta 1 Hz Condiciones Ambientales: -55°C a 125°C
Sonido	Precisión: Variable Rango de Medición: Varía 0.1°C: Variable Consumo de Energía: Bajo consumo Frecuencia de Muestreo: Ajustable



	<p>Capacidad de Transmisión: En tiempo real</p> <p>Condiciones Ambientales: Sensible a la humedad y temperatura</p>
Presión Atmosférica	<p>Precisión: ± 0.01 hPa</p> <p>Rango de Medición: 300 hPa a 1100 hPa</p> <p>Variable: 0.01 hPa</p> <p>Consumo de Energía: Bajo consumo</p> <p>Frecuencia de Muestreo: Ajustable</p> <p>Capacidad de Transmisión: Hasta 1 Hz</p> <p>Condiciones Ambientales: -40°C a 85°C</p>
Altitud	<p>Precisión: ± 1 metro</p> <p>Rango de Medición: -500 m a 9,000 m</p> <p>0.01 hPa: 0.1 metro</p> <p>Consumo de Energía: Bajo consumo</p> <p>Frecuencia de Muestreo: Ajustable</p> <p>Capacidad de Transmisión: Hasta 1 Hz</p> <p>Condiciones Ambientales: -40°C a 85°C</p>
GPS	<p>Precisión: Variable</p> <p>Rango de Medición: Coordenadas geográficas</p> <p>0.1 metro: Variable</p> <p>Consumo de Energía: Moderado a alto</p> <p>Frecuencia de Muestreo: Variable</p> <p>Capacidad de Transmisión: En tiempo real</p> <p>Condiciones Ambientales: Amplio rango de temperaturas</p>
Proximidad	<p>Precisión: Milímetros</p> <p>Rango de Medición: 2 cm a 200cm</p> <p>Variable: Milímetros</p> <p>Consumo de Energía: Bajo consumo</p>



	<p>Frecuencia de Muestreo: Alta frecuencia en tiempo real</p> <p>Capacidad de Transmisión: En tiempo real</p> <p>Condiciones Ambientales: Sensible a la humedad y temperatura</p>
Cámara Trampa	<p>Precisión: Alta resolución</p> <p>Rango de Medición: Depende del campo de visión</p> <p>Milímetros: Alta resolución</p> <p>Consumo de Energía: Bajo consumo</p> <p>Frecuencia de Muestreo: Disparo por detección de movimiento</p> <p>Capacidad de Transmisión: Almacenamiento local</p> <p>Condiciones Ambientales: Resistente a condiciones climáticas</p>
Imagen espectral	<p>Precisión: Variable</p> <p>Rango de Medición: Múltiples bandas</p> <p>Alta resolución: Variable</p> <p>Consumo de Energía: Moderado a alto</p> <p>Frecuencia de Muestreo: Configurable</p> <p>Capacidad de Transmisión: Variable</p> <p>Condiciones Ambientales: Sensible a la humedad</p>



Tabla 9.2: Ventajas y limitaciones de los sensores de medición.

Sensor	Ventajas	Limitaciones
Humedad	- Información precisa sobre la humedad ambiental. - Sensores económicos y de bajo consumo.	- No es relevante para todas las especies. - Requiere calibración periódica.
Temperatura	- Relevante para la mayoría de las especies. - Sensores económicos y ampliamente disponibles.	- Rango de medición insuficiente en extremos. - Respuesta lenta en algunos sensores.
Sonido	- Permite la detección de sonidos relacionados con la fauna. - Útil para monitorear actividad e interacción.	- Identificación precisa desafiante. - Mayor procesamiento de datos requerido.
Presión Atmosférica	- Estimación de altitud y presión atmosférica. - Proporciona información útil sobre cambios climáticos.	- No proporciona información específica sobre especies. - Requiere calibración periódica.
Altitud	- Importante para especies con comportamiento altitudinal. - Responde a cambios rápidos.	- No relevante para todas las especies o hábitats. - Requiere sensores adicionales para precisión.
GPS	- Datos de ubicación precisos. - Geolocalización esencial.	- Consumo de energía moderado. - Necesita visión directa con satélites.
Proximidad	- Detecta la cercanía de objetos o animales. - Bajo consumo de energía.	- Alcance limitado requiere disposición cuidadosa. - No proporciona datos detallados sobre especies.
Cámara Trampa	- Captura imágenes de forma discreta. - Útil para identificación visual de especies.	- Almacenamiento y procesamiento de imágenes. - Vulnerable a condiciones de luz y clima.
Imagen Multiespectral	- Permite análisis detallado de la vegetación y el terreno. - Útil para estudios ecológicos y agrícolas.	- Costosa y requiere sensores específicos. - No es adecuado para la detección directa de animales.

A continuación, se presentan opciones de sensores para los tipos de mediciones mencionadas anteriormente, junto con algunas características relevantes que las diferencian entre sí.

1. Humedad:

- Sensor de Humedad Capacitivo (DHT22) [?]:
 - a) Precisión en la medición.
 - b) Rango de medición: 0-100 % de humedad relativa.
 - c) Alta estabilidad en la medición a lo largo del tiempo.
 - d) Costo: Aproximadamente \$5-10 USD.
 - e) Tamaño: Alrededor de 4 cm.
 - f) Peso: 3 gramos.
 - g) Consumo de energía: 1-1.5 mA a 3-5 volts.
- Sensor de Humedad Resistivo (HR202) [?]:



- a) Adecuado para aplicaciones de bajo costo.
- b) Rango de medición: 0-100 % de humedad relativa.
- c) Buena resistencia ante la contaminación.
- d) Costo: Aproximadamente \$2-5 USD.
- e) Tamaño: Cerca de 2 cm.
- f) Peso: 1 gramo.
- g) Consumo de energía: 0.2-0.5 mA a 3-5 volts.

2. Temperatura

- Termistor (LM35) [?]:
 - a) Alta precisión.
 - b) Rango de medición: -55°C a 150°C.
 - c) Respuesta rápida a cambios de temperatura.
 - d) Costo: Alrededor de \$1-3 USD.
 - e) Tamaño: Cerca de 1 cm.
 - f) Peso: 0.1 gramos.
 - g) Consumo de energía: 50-60 μ A a 5 volts.
- Sensor de Temperatura de Platino (PT100) [?]:
 - a) Muy alta precisión.
 - b) Amplio rango de medición: -200°C a 850°C.
 - c) Robusto y duradero.
 - d) Costo: Aproximadamente \$20-30 USD.
 - e) Tamaño: Alrededor de 5 cm.
 - f) Peso: 5 gramos.
 - g) Consumo de energía: Depende del circuito de interfaz, generalmente bajo.

3. Sonido:



- Micrófono de Condensador Electret (MAX4466) [?]:
 - a) Sensibilidad en la detección de sonidos.
 - b) Amplio rango de frecuencia.
 - c) Bajo nivel de ruido en la señal.
 - d) Costo: Aproximadamente \$5-10 USD.
 - e) Tamaño: Alrededor de 1.5 cm.
 - f) Peso: 1 gramo.
 - g) Consumo de energía: 0.5-1 mA a 3-10 volts.
- Sensor de Sonido de Efecto Piezoeléctrico (KY-038) [?]:
 - a) Buena respuesta a sonidos de alta frecuencia.
 - b) Amplio rango de frecuencia.
 - c) Alta sensibilidad.
 - d) Costo: Aproximadamente \$2-5 USD.
 - e) Tamaño: Cerca de 2 cm.
 - f) Peso: 1 gramo.
 - g) Consumo de energía: 0.5-1 mA a 3-5 volts.

4. Presión Atmosférica:

- Sensor de Presión Piezoeléctrico (MPX5700AP) [?]:
 - a) Respuesta rápida a cambios de presión.
 - b) Rango de medición: 0-10 bar.
 - c) Robusto y duradero en entornos hostiles.
 - d) Costo: Aproximadamente \$15-20 USD.
 - e) Tamaño: Alrededor de 3 cm.
 - f) Peso: 2 gramos.
 - g) Consumo de energía: 1-10 mA a 3-5 volts.



- Sensor de Presión Capacitivo (MS5611) [?]:
 - a) Alta precisión en la medición.
 - b) Rango de medición: 10 mbar a 1200 mbar.
 - c) Bajo consumo de energía para aplicaciones portátiles.
 - d) Costo: Aproximadamente \$20-30 USD.
 - e) Tamaño: Cerca de 2 cm.
 - f) Peso: 1.5 gramos.
 - g) Consumo de energía: 0.6 mA a 1.2 mA a 3-5 volts.

5. Altitud:

- Sensor de Altitud Barométrico (BMP180) [?]:
 - a) Mide altitud basada en la presión atmosférica.
 - b) Precisión adecuada para aplicaciones de navegación.
 - c) Bajo consumo de energía.
 - d) Costo: Aproximadamente \$5-10 USD.
 - e) Tamaño: Alrededor de 1.5 cm.
 - f) Peso: 1 gramo.
 - g) Consumo de energía: 0.5-3 mA a 1.8-3.6 volts.
- Sensor de Ultrasonidos (HC-SR04) [?]:
 - a) Mide la distancia desde el sensor al objeto mediante ultrasonidos.
 - b) Amplio rango de detección.
 - c) Alta velocidad de respuesta en la medición.
 - d) Costo: Aproximadamente \$2-5 USD.
 - e) Tamaño: Cerca de 4 cm.
 - f) Peso: 3 gramos.
 - g) Consumo de energía: 2 mA a 5 volts.



6. GPS:

- Módulo GPS de un solo canal (NEO-6M) [?]:
 - a) Precisión estándar en geolocalización.
 - b) Bajo consumo de energía.
 - c) Pequeño y liviano, ideal para dispositivos compactos.
 - d) Costo: Aproximadamente \$10-20 USD.
 - e) Tamaño: Alrededor de 4 cm.
 - f) Peso: 10 gramos.
 - g) Consumo de energía: 30-40 mA a 3-5 volts.
- Módulo GPS de varios canales (u-blox NEO-M8N) [?]:
 - a) Mayor precisión y capacidad de adquisición rápida de señales GPS.
 - b) Soporte para múltiples sistemas de navegación, como GPS, GLONASS y Galileo.
 - c) Alta velocidad de actualización.
 - d) Costo: Aproximadamente \$20-30 USD.
 - e) Tamaño: Cerca de 5 cm.
 - f) Peso: 15 gramos.
 - g) Consumo de energía: 30-40 mA a 3-5 volts.

7. Proximidad:

- Sensor de Infrarrojos de Reflexión (TCRT5000) [?]:
 - a) Detecta objetos cercanos por reflexión de luz infrarroja.
 - b) Alta velocidad de respuesta en la detección.
 - c) Costo: Aproximadamente \$1-5 USD.
 - d) Tamaño: Cerca de 1 cm.
 - e) Peso: 2 gramos.



- f) Consumo de energía: 1-5 mA a 3-5 volts.
- Sensor Ultrasónico de Proximidad (HC-SR04) [?]:
 - a) Mide la distancia entre el sensor y el objeto mediante ultrasonidos.
 - b) Amplio rango de detección y bajo costo.
 - c) Alta sensibilidad.
 - d) Costo: Aproximadamente \$1-5 USD.
 - e) Tamaño: Alrededor de 4 cm.
 - f) Peso: 3 gramos.
 - g) Consumo de energía: 15 mA a 5 volts.

8. Cámara Trampa:

- Cámara de Infrarrojos (Browning Strike Force) [?]:
 - a) Captura imágenes en condiciones de poca luz.
 - b) Función de visión nocturna por infrarrojos para imágenes nítidas en la oscuridad.
 - c) Alta velocidad de disparo para capturar acciones rápidas.
 - d) Costo: Aproximadamente \$150-200 USD.
 - e) Tamaño: Alrededor de 12 cm x 8 cm.
 - f) Peso: 160 gramos.
 - g) Consumo de energía: Variable, en modo de espera consume muy poca energía.
- Cámara de Alta Resolución (PR700) [?]:
 - a) Captura imágenes detalladas de alta calidad.
 - b) Diseñada para resistir condiciones climáticas adversas en entornos de monitoreo de vida silvestre.
 - c) Amplio rango de temperatura de funcionamiento.
 - d) Costo: Aproximadamente \$500-800 USD.



- e) Tamaño: Cerca de 15 cm x 8 cm.
- f) Peso: 400 gramos.
- g) Consumo de energía: Variable, pero bajo en modo de espera.

9. Imagen Multiespectral:

- Cámara Multiespectral de Detección Remota (Parrot Sequoia) [?]:
 - a) Captura información en múltiples bandas espectrales.
 - b) Permite la generación de índices vegetativos de alta precisión.
 - c) Ligera y fácil de montar en drones.
 - d) Costo: Aproximadamente \$3,000-4,000 USD.
 - e) Tamaño: Alrededor de 10 cm x 6 cm.
 - f) Peso: 110 gramos.
 - g) Consumo de energía: 8 W.
- Cámara Multiespectral Hiperespectral (Headwall Nano-Hyperspec) [?]:
 - a) Captura información en una amplia gama de bandas espectrales.
 - b) Mayor capacidad para la detección de diferencias sutiles en la reflectancia espectral de los objetos
 - c) Utilizada en aplicaciones de investigación científica avanzada.
 - d) Costo: Varía según las especificaciones, generalmente en el rango de \$10,000-30,000 USD.
 - e) Tamaño: Varía entre 10 cm x 10 cm a 20 cm x 20 cm , según la configuración.
 - f) Peso: Varía entre 200 gramos a 1 kilogramo, según la configuración.
 - g) Consumo de energía: Varía según la configuración, típicamente más alto que otros sensores.



9.1.2. Selección de los Sensores

La selección de sensores desempeña un papel fundamental en el diseño de nuestro sistema de monitoreo. Cada sensor cumple un papel específico en la recopilación de datos críticos para el éxito de este proyecto. Por lo que se ha optado por medir la temperatura, utilizar un módulo GPS y la captura de imagen con una cámara trampa. A continuación, se justifica esta selección:

- **Medición de la Temperatura (seleccionando Termistor LM35):**

La temperatura es un factor ambiental crítico que puede afectar la distribución y el comportamiento de las especies. Los cambios en la temperatura pueden tener un impacto directo en la actividad de los animales, su reproducción y su búsqueda de alimento. Monitorear la temperatura es esencial para comprender cómo los cambios climáticos pueden estar afectando a las especies en peligro de extinción.

El termistor LM35 es una elección adecuada para medir la temperatura debido a su alta precisión, su amplio rango de medición y su facilidad de integración. Además, su bajo consumo de energía lo hace adecuado para ser situado en lugares remotos [?]. En la Figura 9.1 se puede ver un ejemplo del termistor LM35.

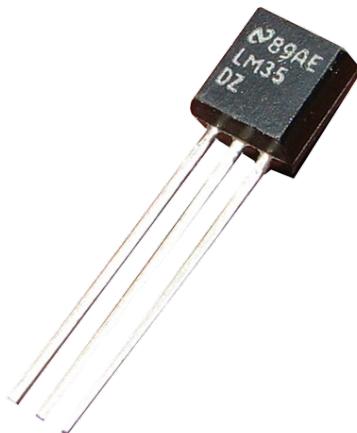


Figura 9.1: Termistor LM35.

- **Geolocalización (seleccionando Módulo GPS Neo-6M):**

La capacidad de rastrear y registrar con precisión la ubicación geográfica de las observaciones es fundamental en proyectos de monitoreo de especies. Ayuda a determinar la distribución espacial de las especies, sus patrones de migración y su relación con el hábitat.

El módulo GPS Neo-6M es elegido debido a su precisión y su capacidad para rastrear múltiples satélites de diferentes sistemas de navegación, lo que mejora la precisión de la geolocalización. Su bajo consumo de energía lo hace adecuado para drones [?]. En la Figura 9.2 se puede ver un ejemplo del módulo GPS Neo-6M.



Figura 9.2: Módulo GPS Neo-6M.

- **Captura de Imágenes (seleccionando Cámara Trampa PR-700):**

Las imágenes visuales son una fuente invaluable de datos en proyectos de monitoreo de vida silvestre. Permiten la identificación de especies, el seguimiento de la actividad y el análisis de la salud de los individuos.

La cámara trampa PR-700 se selecciona debido a su capacidad para capturar imágenes de alta resolución y calidad en condiciones diversas, incluyendo situaciones de poca luz o nocturnas. Su resistencia a condiciones climáticas adversas y su capacidad para capturar imágenes de vida silvestre de manera discreta la hacen ideal para este proyecto [?]. En la Figura 9.3 se puede ver un ejemplo de la cámara trampa PR-700.



Figura 9.3: Módulo GPS Neo-6M.

La elección de medir la temperatura, utilizar un módulo GPS y una cámara trampa se basa en la relevancia de estas mediciones para el propósito de este proyecto. El termistor LM35 proporcionará datos precisos sobre las condiciones ambientales que pueden influir en el comportamiento de las especies, mientras que el módulo GPS Neo-6M permitirá geolocalizar con exactitud nuestras observaciones y rastrear los movimientos de las especies. Por último, la cámara trampa PR-700 brindará imágenes de alta calidad que serán esenciales para la identificación y el análisis de las especies.

Esta elección de sensores se basa en un análisis detallado de diversos factores, entre los cuales se destaca la consideración de costos, tamaño, peso y consumo energético. En primer lugar, el costo de los sensores es un factor muy importante en este proyecto, por lo que se han seleccionado sensores que ofrecen una excelente relación calidad-precio, permitiendo mantener el proyecto dentro de un presupuesto razonable y alcanzable.

En cuanto al tamaño y el peso, es esencial que los sensores sean compactos y livianos para ser instalados sin dificultad en las ubicaciones terrestres identificadas, como árboles o estructuras diseñadas para este propósito.

Además, la eficiencia energética sigue siendo esencial, ya que estos sensores deberán funcionar de manera autónoma durante períodos prolongados. Los sensores seleccionados son conocidos por su eficiencia energética, lo que minimiza el drenaje de la batería y garantiza un tiempo



de operación adecuado. Esto es esencial para recopilar datos de monitoreo durante períodos prolongados sin requerir frecuentes cambios de batería o recarga.

Concluyendo en que los sensores seleccionados no sólo son relevantes desde el punto de vista técnico, sino que también se ajustan de manera óptima a los aspectos económicos, logísticos y operativos del proyecto de monitoreo de especies en peligro de extinción.

9.2. Análisis de la WSN (Red Inalámbrica de Sensores)

El diseño y la operación de la red inalámbrica de sensores (WSN) debe estar meticulosamente planificado para garantizar la recopilación precisa de datos en tiempo real. Por lo que hay que examinar detenidamente los aspectos clave de la WSN que se piensan implementar en este proyecto, destacando sus características técnicas y consideraciones críticas.

9.2.1. Arquitectura de la WSN:

Existen dos arquitecturas principales para la implementación de redes inalámbrica de sensores: la arquitectura cliente-servidor y la arquitectura peer-to-peer (P2P). Las cuales se describen en la Tabla 9.3: Comparación entre Arquitectura Cliente-Servidor y Peer-to-Peer (P2P).

Se selecciona la arquitectura Cliente-Servidor para la WSN debido a la necesidad de una gestión centralizada que permita un control preciso de la red y optimice el procesamiento de datos. Dada la importancia de coordinar eficientemente las operaciones de monitoreo, así como de gestionar y distribuir recursos de manera controlada, la arquitectura Cliente-Servidor ofrece un marco robusto para lograr estos objetivos. Además, la posibilidad de realizar mantenimiento y actualizaciones de manera remota resulta crucial para asegurar un funcionamiento continuo y adaptativo de la red. La arquitectura Cliente-Servidor proporciona una estructura organizativa sólida que se alinea con los requisitos específicos del monitoreo a realizar.



Tabla 9.3: Comparación entre Arquitectura Cliente-Servidor y Peer-to-Peer (P2P).

Arquitectura	Cliente-Servidor	Peer-to-Peer (P2P)
Descripción	<p>La red está organizada en torno a un nodo central denominado "servidor" y múltiples nodos periféricos llamados "clientes". El servidor coordina y gestiona las operaciones, mientras que los clientes recopilan datos y los envían al servidor para su procesamiento [?].</p>	<p>Todos los nodos actúan como iguales, compartiendo y procesando datos de manera descentralizada. No hay un nodo central que coordine las operaciones; en cambio, cada nodo puede comunicarse directamente con otros nodos en la red. Esta arquitectura es más distribuida y autónoma[?].</p>
Ventajas	<ul style="list-style-type: none"> -Gestión de Recursos: Permite una gestión eficaz de recursos, ya que el servidor puede asignar tareas y controlar el flujo de datos. -Control Centralizado: Facilita la administración y coordinación eficientes de la red. -Mantenimiento Remoto: Facilita actualizaciones y mantenimiento remotos. 	<ul style="list-style-type: none"> -Descentralización: Elimina la dependencia de un servidor central, mejorando la redundancia y robustez de la red. -Escalabilidad: Puede ser más escalable, ya que la adición de nuevos nodos no requiere cambios significativos en la infraestructura existente.
Desventajas	<ul style="list-style-type: none"> -Dependencia del Servidor: Si el servidor falla, la red puede volverse inoperable. -Posible Cuello de Botella: El servidor puede convertirse en un cuello de botella. 	<ul style="list-style-type: none"> -Coordinación Desafinante: La coordinación y gestión pueden ser desafinantes debido a la falta de un punto centralizado de control. -Problemas de Seguridad: La seguridad puede ser más compleja de implementar en una red P2P debido a la naturaleza distribuida de la arquitectura.



9.2.2. Topología de la WSN

La topología de la red inalámbrica de sensores se organizará de manera que garantice una cobertura eficiente de la zona de monitoreo. Se evaluarán opciones como topología en malla, árbol o estrella, ya que cada una ofrece enfoques distintos para gestionar la conectividad entre los nodos de la red, como se puede ver a continuación:

- **Topología en Malla:** En esta topología, cada nodo de la red se conecta con múltiples nodos vecinos, formando una estructura de malla interconectada. Esto crea múltiples rutas de comunicación entre los nodos, aumentando la redundancia y la fiabilidad de la red. Si un nodo falla o se desconecta, la red puede encontrar una ruta alternativa para mantener la comunicación [?]. Usualmente se utiliza en aplicaciones donde la redundancia y la tolerancia a fallos son críticas, como en entornos industriales o militares.

Ventaja: Redundancia y tolerancia a fallos. La presencia de múltiples rutas de comunicación permite que la red se adapte a fallos individuales sin perder conectividad. Ideal para entornos críticos donde la fiabilidad es esencial.

Desventaja: Mayor complejidad de gestión y configuración. La necesidad de gestionar varias conexiones y rutas puede aumentar la complejidad del diseño y la administración de la red.

- **Topología en Árbol:** En este tipo de topología, los nodos se organizan jerárquicamente, con un nodo central (raíz) que se comunica con nodos secundarios, y estos, a su vez, pueden tener nodos hijos adicionales. La comunicación sigue una estructura descendente desde la raíz hasta los nodos periféricos [?]. Usualmente se utiliza en aplicaciones donde hay un nodo central de control y los nodos periféricos recopilan datos, como en sistemas de monitorización de edificios.

Ventaja: Jerarquía y escalabilidad. La estructura jerárquica facilita la gestión y permite escalar la red añadiendo más nodos periféricos. Es eficiente para aplicaciones que requieren una estructura de control centralizado.

Desventaja: Vulnerabilidad ante la falla del nodo central. Si el nodo raíz falla, toda la



rama de la red podría perder conectividad, lo que podría ser problemático en situaciones críticas.

- **Topología en Estrella:** En esta topología, todos los nodos de la red se conectan directamente a un nodo central (hub o coordinador), pero no se conectan entre sí. Todas las comunicaciones pasan a través del nodo central. Esto simplifica la gestión y la configuración de la red [?]. Usualmente se utiliza en aplicaciones donde la simplicidad, la facilidad de configuración y el bajo consumo energético es prioritario, como en sistemas de monitoreo y control remoto.

Ventaja: Sencillez y facilidad de gestión. La conexión directa de todos los nodos al centro simplifica la gestión y la configuración. Es fácil de entender y mantener.

Desventaja: Vulnerabilidad del nodo central. Si el nodo central (hub) falla, todos los nodos quedan desconectados. Puede ser menos apropiada en aplicaciones críticas que requieren alta disponibilidad.

Se elige una topología en estrella para la WSN por su facilidad de implementación, gestión centralizada y capacidad para identificar rápidamente problemas. La sencillez en la instalación y configuración de los sensores permite una rápida implementación en el terreno, mientras que la estructura en estrella facilita la administración centralizada desde un nodo principal, optimizando la eficiencia operativa. Además, la rápida identificación de problemas individuales sin afectar a otros nodos simplifica el mantenimiento y garantiza una operación continua. Aunque se reconoce la vulnerabilidad potencial del nodo central, medidas de seguridad y redundancia pueden mitigar este riesgo, respaldando así la elección de la topología en estrella para el monitoreo efectivo de especies en peligro de extinción.

9.2.3. Microprocesador de la WSN

En la selección del microprocesador para la WSN y para los collares de los animales, se evaluaron diversas opciones considerando factores clave como consumo energético, capacidad de procesamiento y facilidad de integración, como se muestra a continuación:



- **Arduino Pro Micro:** Es una placa de desarrollo basada en el microcontrolador ATmega32U4. Destaca por su tamaño compacto y bajo consumo de energía, haciéndolo ideal para dispositivos portátiles y con restricciones de energía. Aunque tiene menos capacidad de procesamiento que algunas alternativas, es suficiente para aplicaciones específicas y cuenta con una comunidad activa de desarrolladores que brindan soporte y una amplia gama de bibliotecas [?].
- **Raspberry Pi Zero W:** Es una mini computadora de bajo costo con conectividad inalámbrica. Aunque es más potente que Arduino Pro Micro y puede ejecutar un sistema operativo completo, tiene un consumo de energía relativamente mayor. Su versatilidad lo hace adecuado para proyectos más complejos que requieren un sistema operativo, pero puede ser excesivo para dispositivos pequeños y de bajo consumo [?].
- **ESP32:** Es un microcontrolador de bajo costo y bajo consumo de energía con conectividad Wi-Fi y Bluetooth. Ofrece un equilibrio entre capacidad de procesamiento y eficiencia energética, lo que lo hace atractivo para diversas aplicaciones [?]. Sin embargo, su tamaño y complejidad pueden no ser ideales para dispositivos extremadamente pequeños como collares de monitoreo.

La elección de Arduino Pro Micro para la WSN y los collares se fundamenta en su bajo consumo energético, capacidad suficiente para tareas específicas de monitoreo, y su tamaño compacto. Estas características son esenciales para garantizar una operación de larga duración de los collares, mientras que la simplicidad y confiabilidad hacen de Arduino Pro Micro una opción óptima para la WSN, donde la eficiencia y la estabilidad son prioritarias. La consistencia en la elección de Arduino Pro Micro simplifica el desarrollo, la programación y el mantenimiento de la red de monitoreo.

9.2.4. Fuente de Energía de la WSN

La elección de la fuente de energía para el microprocesador de la WSN es un aspecto crítico en el diseño de dispositivos de monitoreo, ya que afecta directamente la autonomía y



la eficiencia operativa. Entre las opciones comunes se encuentran las baterías recargables de iones de litio (Li-Ion) y las baterías alcalinas.

- **Baterías Recargables de Iones de Litio (Li-Ion):** Estas baterías son conocidas por su alta densidad de energía y su capacidad de ser recargadas, lo que las hace ideales para dispositivos de larga duración. Ofrecen una tensión constante y tienen una vida útil prolongada. Sin embargo, suelen ser más costosas y requieren circuitos de protección para evitar sobrecargas o descargas excesivas [?].
- **Baterías Alcalinas:** Las baterías alcalinas son una opción tradicional y fácilmente disponibles en el mercado. Aunque son más económicas, su vida útil es limitada, y una vez agotadas, deben ser reemplazadas. También pueden experimentar caídas de voltaje a medida que se descargan, lo que podría afectar el rendimiento del microprocesador [?].

La elección de baterías recargables de iones de litio se fundamenta en la necesidad de una fuente de energía eficiente y de larga duración. Para dispositivos de monitoreo que requieren una autonomía prolongada, las baterías Li-Ion ofrecen una solución confiable. Aunque su costo inicial puede ser superior, su capacidad de recarga y su vida útil más extensa reducen los costos a largo plazo y minimizan la necesidad de reemplazo frecuente. Además, los circuitos de protección pueden integrarse para garantizar un uso seguro y optimizar la eficiencia de la batería durante la operación continua del microprocesador.

9.2.5. Comunicación Inalámbrica de la WSN

La eficacia de la comunicación inalámbrica es esencial para el éxito de nuestro sistema de monitoreo, ya que se debe permitir la transferencia de archivos de datos, así como de archivos multimedia. Por lo que se consideran las siguientes opciones:

- **LoRa UART 868/915 MHz:** LoRa (Long Range) en la banda de frecuencia UART 868/915 MHz es una elección ideal para la transmisión de archivos de texto en entornos donde se requiere un alcance extenso y una baja tasa de transferencia de datos. Su



capacidad para penetrar obstáculos y brindar una conectividad confiable a larga distancia lo hace adecuado para la comunicación eficiente de datos de texto entre los nodos de la red. La baja demanda de energía de LoRa contribuye a la eficiencia operativa, crucial para prolongar la duración de la batería de los dispositivos de monitoreo [?].

- **Antenas WiFi:** Las antenas WiFi son una elección apropiada cuando se busca una transmisión rápida y eficiente de archivos de video. La tecnología WiFi ofrece un ancho de banda significativamente mayor en comparación con LoRa, permitiendo la transmisión de datos de video con calidad adecuada. Aunque su alcance puede ser más limitado en comparación con LoRa, la prioridad aquí es la velocidad de transmisión para garantizar la entrega oportuna de archivos de video desde los drones a la estación base. El uso de antenas WiFi se justifica por su capacidad para manejar el alto volumen de datos asociado con la transmisión de video en tiempo real [?].
- **Bluetooth:** Es una tecnología inalámbrica adecuada para la comunicación a corta distancia. Aunque su alcance es más limitado en comparación con LoRa, es ideal para la comunicación entre dispositivos cercanos, como la interacción entre drones y sensores terrestres en un entorno localizado. Su baja demanda de energía también lo hace adecuado para aplicaciones de corto alcance [?].
- **RFID (Identificación por Radiofrecuencia):** Es una tecnología que se utiliza para la identificación sin contacto a corta distancia. Puede ser útil para el seguimiento y la identificación de individuos o elementos específicos en un área cercana. Sin embargo, su aplicación se limita a distancias cortas y a la identificación más que a la transmisión de datos sustanciales [?].

Se selecciona LoRa UART 868/915 MHz para la transmisión de archivos de texto entre los collares gps y la WSN, así como entre la WSN y los drones, debido a su capacidad única para proporcionar una conectividad confiable a larga distancia con un bajo consumo de energía. Dada la naturaleza extensa del área de monitoreo y la necesidad de establecer una red robusta



entre los nodos de drones y WSN, LoRa se destaca por su excepcional alcance y capacidad para penetrar obstáculos, garantizando una comunicación estable incluso en entornos desafiantes. Su baja tasa de transferencia de datos es adecuada para la transmisión de archivos de texto críticos para el monitoreo de especies en peligro de extinción, contribuyendo así a la eficiencia operativa y a la duración de la batería.

Por otra parte, se selecciona la antena WiFi para la transmisión de archivos de video por la necesidad de un ancho de banda significativo y una velocidad de transmisión rápida. Aunque su alcance puede ser más limitado en comparación con LoRa, la prioridad aquí es la velocidad y calidad de transmisión para garantizar la entrega oportuna de archivos de video. La capacidad de las antenas WiFi para manejar grandes volúmenes de datos los hace esenciales para la recopilación eficiente de información visual en el contexto del monitoreo de especies.

9.2.6. Módulo de Almacenamiento de la WSN

Existen diversas opciones de módulos de almacenamiento para integrar en nuestra red inalámbrica de sensores (WSN). Entre ellas, consideramos módulos USB y módulo MicroSD, como se puede observar a continuación:

- **Módulos USB:** Los módulos USB para almacenamiento ofrecen una solución conveniente para la transferencia y almacenamiento de datos. Estos dispositivos suelen ser compactos y fáciles de usar, conectándose directamente a los puertos USB de los dispositivos compatibles. Además, muchos de ellos admiten una variedad de formatos de tarjetas de memoria y unidades USB [?].

Ventajas:

- **Portabilidad:** Los módulos USB son pequeños y ligeros, lo que facilita su transporte y conexión a diferentes dispositivos.
- **Interfaz estándar:** La interfaz USB es universal, lo que garantiza la compatibilidad con una amplia gama de dispositivos.



- o **Capacidades variables:** Disponibles en diversas capacidades de almacenamiento para adaptarse a las necesidades específicas.

Desventajas:

- o **Mayor consumo de energía:** Algunos módulos USB pueden consumir más energía en comparación con otras soluciones, lo que puede ser relevante en aplicaciones con restricciones energéticas.
- o **Exposición física:** Al estar conectados externamente, los módulos USB pueden estar expuestos a condiciones climáticas adversas y riesgos físicos.
- **Adaptadores de tarjetas MicroSD:** Estos adaptadores permiten la conexión de tarjetas de memoria MicroSD a dispositivos que no tienen un puerto MicroSD integrado. Suelen ser simples y compactos, proporcionando una interfaz eficiente entre la tarjeta de memoria y el dispositivo anfitrión [?]. Ventajas:

- o **Tamaño compacto:** Los adaptadores MicroSD son pequeños y livianos, ocupando poco espacio y siendo fáciles de integrar en dispositivos.
- o **Compatibilidad:** Al admitir tarjetas MicroSD estándar, ofrecen una gran flexibilidad y variedad de opciones de almacenamiento.
- o **Bajo consumo energético:** Consumen menos energía en comparación con algunas soluciones USB.

Desventaja:

- o **Manipulación Sensible:** Dada su naturaleza pequeña, los adaptadores de tarjetas MicroSD pueden ser delicados y requerir manejo cuidadoso.

El módulo MicroSD Card Adapter (modelo MLMSD) destaca por su compatibilidad con tarjetas MicroSD, que ofrecen capacidades de almacenamiento significativas y una amplia disponibilidad en el mercado. La elección de este adaptador se basa en su simplicidad, bajo consumo de energía y facilidad de integración con microcontroladores, como el Arduino Pro Micro que



cuentan collares y la WSN. Además, la tecnología MicroSD proporciona un acceso rápido y eficiente a los datos almacenados, permitiendo una gestión ágil y local de la información reco-
pilada por los sensores distribuidos en el terreno de monitoreo. Con esta elección, aseguramos
un sistema de almacenamiento robusto y confiable que se alinea perfectamente con los requi-
sitos operativos de nuestra WSN. La elección de este módulo implica la implementación de
memorias MicroSD como solución de almacenamiento para los archivos de datos generados
tanto por el microprocesador en los collares y en la WSN, como por los archivos de video las
cámaras trampa.



9.3. Análisis de los Drones.

El empleo de drones en este proyecto presenta varias ventajas significativas que justifican su elección como plataforma para la recolección de datos. A continuación, se detallan las razones clave:

1. **Acceso a Áreas Remotas y de Difícil Acceso:** Los drones permiten llegar a áreas remotas y de difícil acceso para los humanos. Esto es crucial en estudios de biología de campo, donde algunas especies habitan en entornos inexplorados o de difícil acceso terrestre.
2. **Reducción del Impacto Ambiental:** La presencia humana en áreas naturales puede perturbar a las especies y afectar negativamente a los hábitats frágiles. Los drones reducen este impacto al volar a altitudes que minimizan la interferencia con la vida silvestre y al evitar la alteración de la vegetación.
3. **Eficiencia en la Recolección de Datos:** Los enjambres de drones pueden cubrir grandes áreas en poco tiempo y recolectar datos simultáneamente, mejorando significativamente la eficiencia en la recolección de datos en comparación con métodos tradicionales de campo.
4. **Menor Riesgo para los Investigadores:** Al utilizar drones, se reducen los riesgos asociados con largos y agotadores recorridos a pie en terrenos complicados, mejorando la seguridad de los investigadores y permitiendo la exploración de áreas potencialmente peligrosas.
5. **Programación de Rutas Precisas:** Los drones pueden ser programados para seguir rutas preestablecidas, garantizando una cobertura completa de las áreas de interés, lo cual es valioso en estudios de biología de campo donde la precisión en la recolección de datos es fundamental.



9.3.1. Tipo de Dron

Como se pudo ver en la sección 8.3 Enjambre de drones del Marco Teórico de este documento, existen varios tipos de drones, como cuadricópteros, hexacópteros, etc. Por lo que para seleccionar el tipo de dron a utilizar en este proyecto, se realizará un análisis comparativo considerando diferentes aspectos.

- **Tamaño:** Al analizar las categorías de tamaño de drones, que abarcan desde nano hasta grande, cada clasificación presenta ventajas y limitaciones particulares.

Los drones de tamaño nano y micro son extremadamente compactos y ligeros, ideales para maniobras precisas y operaciones en espacios reducidos. Sin embargo, su capacidad de carga, autonomía y estabilidad en condiciones climáticas adversas pueden ser limitadas.

En el otro extremo, los drones de tamaño grande, con una envergadura superior a 2 metros, ofrecen una mayor capacidad de carga y resistencia a condiciones climáticas desafiantes. Sin embargo, suelen ser más costosos, difíciles de transportar y pueden requerir infraestructuras especializadas para su despliegue [?].

Por lo que se considera más adecuado el tamaño medio, que abarca desde 500 mm hasta 2 metros, esto se justifica considerando un equilibrio entre las ventajas de las categorías más pequeñas y grandes. Los drones de tamaño medio ofrecen una combinación óptima de capacidad de carga, autonomía y estabilidad. Presentan la versatilidad necesaria para adaptarse a diferentes entornos de monitoreo, desde áreas densamente arboladas hasta terrenos abiertos, sin comprometer significativamente la maniobrabilidad.

Además, los drones de tamaño medio son más fáciles de transportar y desplegar en comparación con sus contrapartes más grandes, lo que facilita su movilidad entre ubicaciones de monitoreo. Esta característica es esencial en escenarios de campo donde la accesibilidad y la rapidez en la implementación del monitoreo son factores críticos.

- **Peso máximo al despegue (MTOW):** La categoría de peso máximo al despegue (MTOW) de los drones, que varía desde menos de 0.5 kg hasta más de 25 kg, presenta



consideraciones críticas en términos de capacidad de carga, autonomía y restricciones regulatorias. Cada rango de peso tiene implicaciones específicas para el monitoreo de especies en peligro de extinción.

En la categoría de menos de 0.5 kg, los drones son ligeros y compactos, lo que facilita la portabilidad y el despliegue rápido. Sin embargo, su capacidad de carga es limitada, lo que podría restringir la instalación de sensores avanzados y dispositivos de recolección de datos más robustos. Además, pueden verse afectados por condiciones climáticas adversas debido a su menor estabilidad.

En la categoría de 0.5–5 kg, los drones ofrecen un equilibrio crucial entre portabilidad y capacidad de carga. Este rango de peso permite la instalación de sensores especializados, cámaras de alta resolución y sistemas de telemetría más avanzados. La autonomía también tiende a ser mayor en comparación con drones más livianos, lo que es fundamental para la eficiencia operativa en misiones extensas de monitoreo.

En la categoría de 5–25 kg, los drones tienen una mayor capacidad de carga, lo que podría permitir la integración de equipos más pesados y potentes. Esto sería beneficioso para misiones que requieran sensores de mayor tamaño, baterías de mayor capacidad o equipos especializados. Sin embargo, la portabilidad y la agilidad podrían verse comprometidas debido al mayor tamaño y peso del dron. Además, la autonomía podría ser menor en comparación con drones más ligeros.

Para drones con un MTOW superior a 25 kg, se abre la posibilidad de utilizar sistemas aún más avanzados y pesados. Estos drones podrían ser ideales para aplicaciones que requieran equipos especializados de alta gama, como sensores de imagen multiespectral de mayor tamaño, sistemas de muestreo de aire o equipos científicos avanzados. No obstante, su uso puede estar sujeto a regulaciones más estrictas y requerir permisos especiales [?].

La elección de drones con un MTOW de 0.5–5 kg es la más adecuada para el monitoreo de especies en peligro de extinción. Aunque los drones más pesados ofrecen capacidades avanzadas, suelen ser más adecuados para aplicaciones especializadas y pueden estar



sujetos a restricciones regulatorias más estrictas. El rango de 0.5–5 kg proporciona el equilibrio adecuado entre capacidad de carga, autonomía y flexibilidad operativa, lo que resulta fundamental para abordar los desafíos específicos del monitoreo de la biodiversidad en entornos variados. Además, en términos de regulaciones, los drones de esta categoría suelen estar sujetos a menos restricciones que los de mayor MTOW, facilitando la obtención de permisos y cumplimiento normativo.

- **Alcance (Distancia/Tipo de Operación):** Al comparar los drones según el alcance de operación, se encuentran tres categorías clave: corto alcance (<0.5 millas), medio alcance (0.5–5 millas) y largo alcance (>5 millas).

Los drones de corto alcance son ideales para misiones en entornos cercanos y confinados, donde la distancia entre la estación base y el dron es relativamente corta. Estos drones son adecuados para operaciones de baja altitud y vuelos a corta distancia, pero pueden no ser eficientes en la cobertura de áreas extensas.

Los drones de largo alcance, por otro lado, ofrecen la capacidad de cubrir distancias significativas, pero suelen estar diseñados para aplicaciones más especializadas, como la monitorización de oleoductos o la exploración de grandes extensiones de terreno. Pero para el monitoreo de especies en peligro de extinción, donde la atención puede centrarse en áreas específicas, los drones de largo alcance pueden no ser la opción más eficiente [?].

Por lo que, debido a su versatilidad en términos de cobertura de área y eficiencia operativa, los drones de medio alcance (0.5–5 millas) se consideran la mejor opción. Estos drones proporcionan un equilibrio adecuado entre la capacidad de cubrir distancias moderadas y la flexibilidad para operar en entornos diversos. Además, se alinean con las regulaciones de vuelo visual lineal (VLOS) y visual extendida (EVLOS), lo que garantiza operaciones seguras y dentro de los límites reglamentarios.

- **Ala:** Los drones de ala fija son eficientes para cubrir grandes áreas y realizar misiones de largo alcance. Sin embargo, suelen requerir pistas de despegue y aterrizaje, lo que



limita su utilidad en áreas con acceso limitado. Estos drones son más adecuados para aplicaciones como la cartografía y la vigilancia aérea extensiva, pero pueden no ser la opción más práctica para el monitoreo detallado de especies en hábitats complejos.

Los drones de ala giratoria, también conocidos como multi-rotores, utilizan rotores para generar sustentación y control, lo que les proporciona una capacidad excepcional de despegue y aterrizaje vertical (VTOL). Esta característica es crucial al operar en áreas con limitaciones espaciales, como bosques densos o áreas montañosas. Los multi-rotores ofrecen una excelente maniobrabilidad, lo que facilita el vuelo a baja altitud y la captura de datos detallados en entornos complejos. Su capacidad para mantenerse estacionarios es esencial al realizar tareas de monitoreo que requieren observación precisa, como el seguimiento de especies animales [?].

Los drones híbridos ofrecen una combinación de las ventajas de las alas giratorias y fijas. Pueden despegar y aterrizar verticalmente, pero también pueden cambiar a vuelo de ala fija para mayor eficiencia en distancias largas. Aunque son versátiles, la complejidad de diseño y el costo adicional pueden superar los beneficios específicos requeridos para el monitoreo de especies en peligro de extinción.

Es por ello que la elección de drones de ala giratoria, especialmente en la configuración multi-rotor, se justifica por su capacidad VTOL, maniobrabilidad precisa y eficiencia en la cobertura de áreas diversas, características esenciales para el monitoreo detallado de especies en hábitats desafiantes. De este tipo de drones multi-rotor, existen varios tipos por lo que también hay que realizar una comparación entre ellos:

- o El tricóptero, con sus tres rotores, destaca por su eficiencia energética, consumiendo menos energía en comparación con otros diseños. Sin embargo, su estabilidad se ve comprometida debido al menor número de rotores, y su maniobrabilidad es buena, pero no alcanza los estándares de cuadricópteros y hexacópteros.
- o El cuadricóptero, con cuatro rotores, equilibra adecuadamente estabilidad y maniobrabilidad. Su diseño simple facilita el control y mantenimiento, lo que lo convierte



en una opción versátil para operar en entornos desafiantes y variables. Aunque es menos eficiente que el tricóptero en términos de consumo de energía.

- o El hexacóptero, con seis motores, ofrece una mayor estabilidad, especialmente en condiciones adversas. Su capacidad para maniobrar y adaptarse a entornos variables es destacable, y su eficiencia energética supera la de los cuadricópteros, proporcionando una mayor autonomía.
- o El octocóptero, con ocho motores, destaca por su máxima estabilidad, incluso en condiciones climáticas desafiantes. Aunque su eficiencia energética es menor debido al mayor número de motores, su estabilidad compensa este aspecto, haciendo que sea una opción adecuada para misiones que requieren una alta estabilidad.

Se considera al cuadricóptero como la mejor opción para este proyecto, destacando por su equilibrio entre estabilidad y maniobrabilidad, su facilidad de control y mantenimiento, así como la disponibilidad generalizada de componentes y repuestos en el mercado. Su capacidad para adaptarse eficazmente a condiciones ambientales cambiantes garantiza un desplazamiento preciso en entornos desafiantes, mientras que su diseño simple y fácil acceso a conocimientos técnicos simplifican las operaciones diarias y reducen los tiempos de inactividad por posibles problemas técnicos. La amplia disponibilidad de componentes esenciales asegura la continuidad operativa, contribuyendo a la sostenibilidad y longevidad del proyecto.

- **Ensamblaje:** Los drones RTF (Listo para Volar) vienen completamente ensamblados y listos para operar. Este tipo de configuración minimiza la intervención del usuario en términos de ensamblaje y ajustes, lo que resulta crucial, especialmente en un proyecto con múltiples facetas como el propuesto. La simplicidad y rapidez de implementación de los drones RTF son factores clave, permitiendo centrar los esfuerzos en la programación, configuración de sensores y aspectos más específicos del proyecto.

Los drones BNF (Enlazar y Volar) ofrecen una alternativa donde el ensamblaje básico ya está realizado, pero se requiere que el usuario complete ciertos pasos antes de volar,



como vincular el dron con su control remoto. Aunque representan un paso intermedio entre RTF y ARF, en este proyecto, donde la eficiencia y la rapidez son esenciales, la opción BNF no proporciona una ventaja significativa sobre RTF.

Los drones ARF (Casi Listo para Volar) ofrecen la máxima flexibilidad, ya que el usuario debe ensamblar varios componentes antes de que el dron esté listo para operar. Aunque esta opción puede ser atractiva para usuarios con experiencia técnica avanzada y requisitos de personalización específicos, la complejidad adicional y el tiempo requerido para el ensamblaje hacen que no sea la elección más eficiente para este proyecto [?].

Es por esto que la opción RTF se alinea mejor con los requisitos de simplicidad y rapidez de implementación, maximizando así el tiempo dedicado a aspectos específicos del proyecto y minimizando las complejidades asociadas con el ensamblaje de drones.

Concluyendo esta evaluación con que los cuadricópteros de tamaño medio, con un peso máximo al despegue de 0.5–5 kg y configuración RTF, se perfilan como la opción más adecuada para cumplir con los objetivos de este proyecto de manera precisa y oportuna.

9.3.2. Controladora de Vuelo para los Drones

Las controladoras de vuelo gestionan la estabilidad y el vuelo de los drones, por lo que su selección es un aspecto crítico para un sistema que utilice drones. Se han evaluado varias opciones, incluyendo la Pixhawk Black Cube, la DJI N3, la Vector Flight Controller y la Naze32. A continuación, se presenta una descripción de cada una:

- **Pixhawk Black Cube:** La Pixhawk Black Cube es conocida por su robustez y confiabilidad, siendo una opción ampliamente utilizada en aplicaciones profesionales y comerciales. Su diseño modular y su capacidad para conectarse con una variedad de sensores y periféricos facilitan la expansión y la adaptación a diferentes configuraciones de drones. La Pixhawk Black Cube se destaca por su procesador potente, lo que la hace capaz de manejar complejas tareas de control de vuelo y navegación. Su algoritmo de control de úl-



tima generación proporciona estabilidad y precisión en diversas condiciones ambientales, lo que es crucial para el éxito de las misiones de monitoreo [?].

- **DJI N3:** La DJI N3 es una controladora de vuelo confiable, especialmente diseñada para sistemas de drones DJI. Si bien es una opción sólida para aplicaciones de fotografía aérea y cinematografía, su integración con componentes no DJI podría presentar desafíos y limitaciones en términos de flexibilidad y expansión [?].
- **Vector Flight Controller:** La Vector Flight Controller es conocida por su facilidad de uso y configuración. Sin embargo, puede carecer de algunas características avanzadas necesarias para aplicaciones más complejas de monitoreo de especies, como la capacidad de integrar una amplia variedad de sensores [?].
- **Naze32:** El Naze32 es una opción popular en drones de carreras, pero puede resultar limitado en términos de capacidad de procesamiento y capacidad de expansión. No está optimizado para aplicaciones que requieren un control preciso y una amplia integración de sensores [?].

La Pixhawk Black Cube destaca por su combinación de potencia de procesamiento, capacidad de expansión y confiabilidad comprobada. Su versatilidad para integrarse con una variedad de sensores y actuadores, junto con su sólido rendimiento en términos de control de vuelo, la convierten en la opción ideal para este proyecto. Además, la comunidad activa de usuarios respalda el desarrollo continuo y la resolución eficiente de problemas, lo que es esencial para el éxito a largo plazo del sistema de drones.

9.3.3. GPS del Dron

En la búsqueda del sistema GPS más adecuado para este proyecto, se evaluaron diferentes opciones disponibles en el mercado. La comparación se centró en aspectos clave como precisión, conectividad, y capacidad de corrección en tiempo real. Entre las alternativas consideradas se



encuentran el GPS Here+, así como otros modelos ampliamente utilizados en aplicaciones similares:

- **DJI Phantom GPS Module:** El módulo GPS DJI Phantom ofrece una buena precisión para la ubicación de drones, aunque su rendimiento puede disminuir en áreas con interferencias significativas. La conectividad está integrada en el dron, limitándose a la señal GPS estándar. Sin embargo, es esencial tener en cuenta que su capacidad de corrección se basa únicamente en las señales satelitales, sin la opción de correcciones diferenciales en tiempo real [?]. Esto lo hace adecuado para operaciones en entornos menos desafiantes y áreas con menos interferencias.
- **Trimble ZX5 Multirotor:** Este módulo se destaca por ofrecer una alta precisión centimétrica en la ubicación del dron. Su conectividad está integrada, pero para aprovechar al máximo su precisión, se requiere una suscripción a servicios de corrección. Este modelo proporciona correcciones en tiempo real, lo que mejora significativamente la precisión de la ubicación en diversas condiciones. Su enfoque en la precisión centimétrica lo convierte en una opción ideal para aplicaciones que demandan datos geoespaciales altamente precisos, aunque la suscripción a servicios adicionales debe considerarse en términos de costos operativos [?].
- **GPS Here+ con Rover y Base Station:** El módulo GPS Here+ es una opción de alta calidad que ofrece una precisión excepcional en la ubicación de drones. Diseñado con capacidades de corrección diferencial en tiempo real, este módulo utiliza tecnología RTK (Real-Time Kinematic) para proporcionar datos de posición con una precisión centimétrica. La conectividad del GPS Here+ va más allá de las señales satelitales estándar, ya que integra una estación base para transmitir correcciones en tiempo real al dron. La comunicación entre el Rover y la Base Station permite una mayor redundancia y adaptabilidad en entornos desafiantes, como áreas boscosas o montañosas, donde la señal GPS puede ser irregular, con esta configuración se ofrece una solución más robusta y adaptable para garantizar la integridad de los datos geoespaciales. Además, la capa-



ciudad de almacenar datos históricos en la Base Station facilita el análisis posterior y la generación de rutas de vuelo más eficientes [?]. Cabe mencionar que, aunque la inversión inicial puede ser mayor, la precisión y la fiabilidad del GPS Here+ lo convierten en una elección valiosa para operaciones que demandan la máxima exactitud en la información de ubicación.

Se selecciona al GPS Here+ como la mejor opción debido a la necesidad crítica de posicionar con una gran precisión a los drones durante las operaciones de vuelo. Y como gracias a que este módulo integra tecnología RTK y correcciones diferenciales en tiempo real, garantiza una precisión centimétrica en la ubicación de los drones. Esta alta precisión es esencial para llevar a cabo vuelos controlados y dirigidos, permitiendo la recolección específica de datos en áreas predefinidas. Además, la inclusión de una estación base para correcciones en tiempo real refuerza aún más su idoneidad, asegurando que los drones se ubiquen de manera precisa en relación con el entorno. A pesar de representar una inversión inicial más significativa, la fiabilidad y exactitud proporcionadas por el GPS Here+ son esenciales para garantizar el éxito de los vuelos de este proyecto.



9.4. Análisis del Enjambre de Drones

El análisis del enjambre de drones representa un componente esencial en el diseño integral de un sistema de monitoreo como el de este proyecto. En esta sección, se abordarán distintos diferentes aspectos para el funcionamiento colectivo de los drones. Se iniciará con el análisis del microordenador abordo, evaluando las algunas opciones disponibles, sus capacidades y su idoneidad para el procesamiento de datos y control de vuelo. Posteriormente, se examinará el lenguaje de programación seleccionado para el microordenador, destacando su pertinencia para el desarrollo y la coordinación de las funciones del enjambre. Seguido de ello, se explorará en detalle la comunicación entre los drones, al analizar diversas opciones de transceptores. Finalmente, se abordará el tipo de formación adoptado por el enjambre, considerando la eficiencia en vuelo, la coordinación y el consumo de energía en misiones de monitoreo prolongadas. Este análisis integral contribuirá a la implementación de un enjambre de drones altamente funcional y coordinado, esencial para el éxito del proyecto de monitoreo de especies en situación crítica.

9.4.1. Microordenador Abordo

En la selección del microordenador para integrar en los drones del enjambre, se han considerado varias opciones, incluyendo el Arduino Nano, la Raspberry Pi 4, el NVIDIA Jetson Nano y el BeagleBone Black.

Arduino Nano: En la selección del microordenador para integrar en los drones del enjambre, se han considerado varias opciones, incluyendo el Arduino Nano, la Raspberry Pi 4, el NVIDIA Jetson Nano y el BeagleBone Black. Cada una de estas plataformas tiene sus propias características y capacidades, a continuación se presenta una comparación entre ellas:

- **Arduino Nano:** El Arduino Nano es conocido por su simplicidad y eficiencia energética. Sin embargo, su capacidad de procesamiento y almacenamiento es limitada en comparación con las otras opciones consideradas, lo que podría afectar la capacidad del sistema para manejar grandes volúmenes de datos y ejecutar algoritmos complejos [?].



- **Raspberry Pi 4:** La Raspberry Pi 4 destaca por su potencia de procesamiento, ya que cuenta con un procesador de cuatro núcleos ARM Cortex-A72 a 1.5 GHz y opciones de memoria RAM de hasta 8 GB. Esto proporciona un rendimiento robusto para manejar tareas computacionales intensivas, como el procesamiento de datos de sensores y la ejecución de algoritmos de control de drones. Además, posee puertos USB 3.0 para una rápida transferencia de datos, lo cual es crucial para la eficiencia en la comunicación entre los componentes y el sistema central [?]. La conectividad también es una fortaleza, con opciones de conexión Ethernet y Wi-Fi, permitiendo la integración fluida con la red de sensores y la transmisión de datos. Su versatilidad se ve respaldada por la amplia comunidad de desarrolladores y el soporte continuo, asegurando actualizaciones y mejoras a lo largo del tiempo.
- **NVIDIA Jetson Nano:** El NVIDIA Jetson Nano se destaca por su capacidad de procesamiento de inteligencia artificial (IA) y su GPU integrada. Si bien es poderoso para tareas de aprendizaje profundo, puede resultar más especializado de lo necesario para este proyecto, donde la prioridad está en la adquisición, procesamiento y transmisión eficientes de datos de sensores [?].
- **BeagleBone Black:** El BeagleBone Black es una opción que combina potencia y versatilidad. Con un procesador AM3358 a 1 GHz y 512 MB de memoria RAM, ofrece un equilibrio adecuado para tareas de control y procesamiento de datos. Su comunidad de usuarios y el soporte extenso proporcionan una base sólida para el desarrollo continuo [?].

La elección del Raspberry Pi 4 se basa en su equilibrio entre potencia de procesamiento, conectividad y versatilidad. Su capacidad para manejar diversas tareas computacionales, junto con la sólida comunidad de desarrollo, asegura que sea una elección sólida para la integración en este proyecto. Además, la potencia de procesamiento adicional de la Raspberry Pi 4 proporciona margen para futuras expansiones y mejoras en el sistema.



9.4.2. Lenguaje de Programación del Microordenador

Por otra parte se debe analizar el lenguaje programación del enjambre, ya que debe adaptarse a las necesidades del proyecto y a la compatibilidad con el hardware y el software de los drones. Aquí se comparan algunos de los lenguajes de programación comúnmente utilizados en drones:

- **Python:** Es conocido por su legibilidad y versatilidad. Es ampliamente utilizado en el desarrollo de software para drones, ya que ofrece una amplia variedad de bibliotecas y marcos de trabajo que facilitan el desarrollo [?]. Su sintaxis clara y concisa es beneficiosa para proyectos complejos, como el diseño y la programación de enjambres de drones.
- **C++:** Es un lenguaje de programación de bajo nivel que brinda un control cercano sobre el hardware, lo que puede ser crucial en aplicaciones que requieren un rendimiento optimizado, como el control de vuelo [?]. Sin embargo, su curva de aprendizaje puede ser más pronunciada en comparación con Python.
- **ROS (Robot Operating System):** ROS no es un lenguaje de programación en sí mismo, sino un marco de trabajo que utiliza varios lenguajes, incluidos C++ y Python. Está diseñado específicamente para robots, facilitando la programación de drones al proporcionar abstracciones y herramientas para el desarrollo de software [?].
- **MATLAB:** MATLAB es una opción común en entornos académicos e industriales. Ofrece una amplia variedad de herramientas para el análisis y diseño de sistemas, así como capacidades para la programación de drones. Puede ser especialmente útil en proyectos que involucran simulaciones y análisis complejos [?].

Python destaca por ofrecer la combinación ideal de legibilidad, versatilidad y soporte comunitario para abordar los diversos aspectos del proyecto. Además, cuenta con una amplia variedad de bibliotecas y marcos de trabajo que son relevantes para el desarrollo de sistemas de drones, en particular para el diseño y programación de enjambres de drones.



9.4.3. Comunicación del Enjambre

Para la selección del transceptor para la comunicación entre drones, se han considerado varias opciones, evaluando diferentes aspectos, como se puede ver a continuación:

- **Transceptor RF de 2.4 GHz:** El transceptor RF de 2.4 GHz (NRF24L01) es una opción común que ofrece velocidades de transmisión de datos razonables, generalmente en el rango de Mbps, también es ultra compacto y de muy bajo consumo energético. Sin embargo, su alcance es limitado y puede experimentar interferencias en entornos urbanos congestionados. Además, la penetración a través de obstáculos es menos eficiente en comparación con frecuencias más bajas [?]. Haciéndolo adecuado para aplicaciones de corto alcance y sin obstáculos significativos.
- **Módulo XBee:** Los módulos XBee se caracterizan por su facilidad de uso y configuración. Ofrecen velocidades de transmisión moderadas, generalmente hasta varios cientos de Kbps, y son adecuados para aplicaciones que no requieren transmisión de datos de alta velocidad. El alcance varía según el modelo, pero tiende a ser adecuado para distancias moderadas. Sin embargo, su costo puede ser más alto en comparación con otras opciones, y el consumo de energía depende del modelo específico [?].
- **Módulo LoRa:** Los módulos LoRa operan en frecuencias más bajas (868MHz/915MHz) y son reconocidos por su capacidad para lograr un mayor alcance, incluso superando varios kilómetros en condiciones ideales. La modulación LoRa permite una comunicación robusta a través de obstáculos y un menor consumo de energía, lo que los hace ideales para aplicaciones de drones que requieren una larga duración de la batería. Las velocidades de transmisión son más bajas en comparación con algunos transceptores, pero suficientes para datos de telemetría y comandos de control [?].
- **Transceptor RF de 5.8 GHz:** Los transceptores de 5.8 GHz (PAT 530) ofrecen velocidades de transmisión más altas, generalmente en el rango de varios Mbps, lo que los hace adecuados para aplicaciones que requieren transmisión de video en tiempo real.



Sin embargo, su alcance es más limitado en comparación con frecuencias más bajas, y la penetración a través de obstáculos puede ser un desafío. El consumo de energía también tiende a ser más alto, lo que puede afectar la duración de la batería en misiones prolongadas [?].

Se ha elegido el Módulo LoRa UART 868MHz/915MHz debido a las necesidades específicas del proyecto, ya que puede alcanzar distancias significativas, atravesar obstáculos de manera efectiva y logra maximizar el tiempo de vuelo de los drones gracias a su bajo consumo energético. Además, la velocidad de transmisión adecuada para telemetría y comandos de control, junto con la disponibilidad de estos módulos en el mercado, respaldan la elección de LoRa como la solución óptima para la comunicación entre drones en este proyecto.

9.4.4. Tipo de Formación del Enjambre

La elección del tipo de formación del enjambre de drones también es importante para optimizar el rendimiento del enjambre y minimizar el consumo energético en las misiones a realizar. Se han evaluado varias configuraciones, incluyendo formación en V, en línea y en red [?]. A continuación, se presenta una descripción de las formaciones:

- **Formación en V:** La formación en V implica que los drones vuelen en una disposición en forma de "V", con un dron líder en la parte delantera y los demás siguiendo sus trayectorias. Esta formación es conocida por su eficiencia en términos de resistencia al viento y consumo de energía, ya que los drones siguientes aprovechan la estela del dron líder.
- **Formación en Línea:** La formación en línea consiste en que los drones sigan una trayectoria recta uno detrás del otro. Aunque es simple y fácil de implementar, puede resultar menos eficiente en términos de resistencia al viento y consumo de energía, ya que cada dron no se beneficia de la estela del anterior.
- **Formación en Red:** La formación en red implica una comunicación constante entre



todos los drones, permitiéndoles ajustar su posición en tiempo real. Aunque puede ofrecer flexibilidad y adaptabilidad, la complejidad de la comunicación entre drones podría aumentar, y la eficiencia podría verse afectada.

Se selecciona la formación en V debido a su equilibrio entre eficiencia y simplicidad. Esta configuración aprovecha la eficiencia aerodinámica al tiempo que mantiene una estructura relativamente simple. Dado que la resistencia al viento y la eficiencia energética son factores críticos en misiones de monitoreo prolongadas, la formación en V ofrece una solución óptima para el enjambre de drones en este proyecto. Además, la disposición en V facilita el seguimiento del líder y la coordinación, elementos esenciales para el éxito de las misiones de monitoreo.



9.5. Análisis del Proveedor de Servicios en la Nube.

Es fundamental tener una infraestructura sólida y escalable para el almacenamiento y procesamiento de datos. En este sentido, la elección del proveedor de servicios en la nube, conocido como DBaaS (Database as a Service) [?], es de suma importancia, ya que proporciona diversos servicios para el procesamiento de datos y desempeña un papel fundamental en este proyecto.

En general, la recopilación de datos para este tipo de proyectos involucra una variedad de sensores, desde mediciones de la temperatura hasta obtención de coordenadas GPS e imágenes de cámaras trampa. La gestión eficiente y segura de esta diversa gama de datos es esencial para garantizar el éxito de este proyecto y contribuir a la preservación de las especies en peligro de extinción.

En este análisis del proveedor de servicios en la nube, se examinarán los aspectos clave a considerar al analizar y seleccionar un proveedor en la nube.

Cuando hablamos de servicios en la nube, también hablamos de beneficios que conllevan, los cuales describiremos a continuación y serán un punto de partida para nuestra selección de nuestro proveedor en la nube.

9.5.1. Criterios de Selección para el Proveedor de Servicios en la Nube.

Al seleccionar un proveedor de servicios en la nube, existen varios criterios que se consideran. Los siguientes son una lista de los más relevantes:

- **Servicios en la nube necesarios:** Algunos proveedores de servicios en la nube solo cubren aspectos individuales, mientras que otros ofrecen una cartera completa de servicios, incluyendo servicios de infraestructura de TI como servicios de red o servicios de servidores virtuales [?]. Al conocer los problemas a los que nos enfrentamos, es fácil poder llegar a seleccionar algunos de los servicios que necesitamos para resolverlos.
- **Seguridad de los servicios en la nube:** La seguridad es un aspecto crucial al elegir



un proveedor de servicios en la nube. Se deben considerar las capacidades del proveedor para garantizar la seguridad de los datos y las infraestructuras de TI (Tecnologías de la Información) [?].

- **Tiempo de funcionamiento de los servicios en la nube:** La disponibilidad y el rendimiento del proveedor son factores importantes a considerar. Se deben analizar las métricas de tiempo de actividad y rendimiento para determinar con qué frecuencia el proveedor de la nube experimenta interrupciones y el tiempo de resolución promedio [?]. Esto nos ayuda para saber cuánto habría que pagar por usar los servicios, ya que cada servicio funciona diferente, puede ser por cantidad de información que se mueve, por ejemplo en Gigabytes o Terabytes, o por la disponibilidad en linea.
- **Portabilidad de datos entre diferentes nubes:** La capacidad de mover datos entre diferentes plataformas de nube es un factor importante a considerar. Esto puede ser especialmente relevante si se considera el uso de múltiples proveedores de servicios en la nube [?]. Esta característica es importante para una empresa en la que se va a estar contando con los servicios durante demasiado tiempo, aunque esto no es tan relevante para este proyecto ya que únicamente se hará uso de los servicios el tiempo de la realización del mismo.
- **Escalabilidad de los servicios en la nube:** La capacidad de escalar los servicios según las necesidades de la empresa es crucial. Algunos proveedores ofrecen escalado manual, automático o programado [?]. La escalabilidad en servicios de la nube se refiere a la capacidad de un sistema o aplicación para manejar un aumento o disminución en la carga de trabajo de manera adecuada y sin comprometer el rendimiento.
- **Soporte al cliente:** Es importante que el proveedor ofrezca soporte al cliente las 24 horas del día y los 365 días del año. También se deben considerar cómo se presta el soporte, si está incluido en el contrato y cuántas veces puede ser utilizado [?].
- **Capacidades y procesos técnicos:** El proveedor debe tener mecanismos que faciliten



la implementación, administración y actualización de su software y sus aplicaciones. Además, debe utilizar transformaciones de datos y API estándares para que la organización, en este caso nosotros como proyecto, se pueda crear conexiones con la nube fácilmente [?].

- **Planificación de salida:** Es posible que se necesite abandonar el proveedor de la nube debido a la seguridad, los costos, el rendimiento o un cambio de estrategia. Por lo tanto, se debe revisar la planificación de salida y evitar el bloqueo del proveedor de la nube [?].
- **Cartera de clientes consolidada:** Conocer cuáles son los clientes del proveedor también es importante, puesto que lo que funciona para otros negocios, aplicaciones o proyectos puede que también funcionen para este proyecto [?].

9.5.2. Selección del Servidor en la Nube.

Para la selección de servidor en la nube, se evaluarán los aspectos mencionados en los beneficios de selección de bases de datos y nos centraremos principalmente en los costes por volumen de almacenamiento y en las tecnologías usadas adyacentemente, como pueden ser otros servicios requeridos posteriormente para el proyecto.

Se analizarán principalmente 4 proveedores de servicios los cuales son, AWS (Amazon Web Services), Microsoft Azure, GCP (Google Cloud Platform) y OCI (Oracle Cloud Infrastructure) [?].



Tabla 9.4: Comparación de proveedores de servicio en la nube (AWS y Azure).

Servicio	AWS	Azure
Almacenamiento de Objetos	-Simple Storage Services (S3)	Blob Storage
Almacenamiento para archivado	-S3 Infrequent Access (IA) -Glacier	-Storage (Cool) -Storage (Archive)
Backup	-Glacier -Storage Gateway	-Azure Backup
Almacenamiento híbrido	-Storage Gateway	-StorSimple
Bases de datos relacionales	-RDS -Amazon Aurora	SQL Database
NoSQL: Almacenamiento de documentos	-DynamoDB	-Azure Cosmos DB
NoSQL: Almacenamiento de clave-valor	-DynamoDB -SimpleDB	-Table Storage
Migración de bases de datos	-Database Migration Service	-Azure Database Migration Service
Bases de datos basada en grafos	-Neptune	-Azure Cosmos DB
Facturación	-Usage and Billing	-Azure Billing API
Autenticación y autorización	-Identity and Access Management -Organizations	-Active Directory -Active Directory Premium
Administración de identidades	-Cognito	-Azure Active Directory B2C
Autenticación multifactor	-Multi-Factor Authentication	-Multi-Factor Authentication
Visualización	-QuickSight	-PowerBI
Servicio gestionado para Machine Learning	-SageMaker	-Azure Machine Learning Studio -Servicio Azure Machine Learning
Reconocimiento de voz e interfaz de conversación	-Amazon Lex	-Bing Speech API -Speaker Recognition API



Tabla 9.5: Comparación de proveedores de servicio en la nube (GCP y OCI).

Servicio	GCP	OCI
Almacenamiento de Objetos	-Google Cloud Storage	-OCI Object Storage
Almacenamiento para archivado	-Nearline -Coldline	-Archive Storage
Backup	-Coldline	-Oracle Database -Backup Cloud Service
Almacenamiento híbrido	-No aplica	-No aplica
Bases de datos relacionales	-Cloud SQL -Cloud Spanner	-Base Database
NoSQL: Almacenamiento de documentos	-Cloud Datastore -Cloud Bigtable	-NoSQL Database Cloud Service
NoSQL: almacenamiento de clave-valor	-Cloud Datastore	-NoSQL Database Cloud Service
Migración de bases de datos	-No aplica	-Oracle Database Cloud Service usando Data Pump Export e Import
Bases de datos basada en grafos	-No aplica	-Base de datos orientada a grafos y analítica de grafos
Facturación	-Stackdriver Monitoring	-No aplica
Autenticación y autorización	-Cloud IAM -Cloud Identity-Aware Proxy	-Identity and Access Management. -Oracle Security Zones
Administración de identidades	-No aplica	-Identity and Access Management
Autenticación multifactor	-Multi-Factor Authentication	-Oracle Database Vault
Visualización	-Data Studio	-Data Visualization
Servicio gestionado para Machine Learning	-Cloud Machine Learning Engine	-OCI Data Science -Machine Learning in Oracle Database -OCI Data Labeling
Reconocimiento de voz interfaz de conversación	-Dialogflow Enterprise Edition	-OCI Speech



Por la naturaleza del proyecto y la recopilación de diferentes tipos de archivos de los diferentes sensores seleccionados, se ha determinado que la base de datos será NoSQL la cual será capaz de almacenar toda la información de nuestros sensores.

Comparando los precios y capacidad de almacenamiento de las calculadoras de los servicios de Azure [?], DataStore de Google [?], OCI en la selección de almacenamiento de Objetos [?] y el nivel gratis de AWS [?]; se ha seleccionado a AWS como el proveedor de servicios.

En caso de que el proveedor de servicios AWS no cumpla las expectativas del proyecto, se optará como segunda opción usar los servicios de OCI.



9.6. Búsqueda y Vinculación con Otros Proyectos de Investigación y/o Organizaciones.

La búsqueda de proyectos u organizaciones dedicadas a la conservación de la biodiversidad y el monitoreo de la vida silvestre es importante para poder ampliar la base de conocimientos y recursos disponibles para el proyecto.

En esta etapa, se llevará a cabo una búsqueda de organizaciones que se dedican a la protección y preservación de especies en peligro de extinción. Estas organizaciones pueden incluir ONGs (Organizaciones No Gubernamentales) ambientales, instituciones gubernamentales de conservación, grupos de investigación dedicados a la biodiversidad y otras entidades relacionadas. El propósito principal de esta actividad es establecer conexiones y colaboraciones que puedan aportar experiencia, datos, acceso a áreas de conservación y recursos adicionales al proyecto. La vinculación con otras entidades en el campo de la conservación con la telemática permitirá al proyecto beneficiarse de la experiencia y el conocimiento acumulado en la conservación de la biodiversidad, al tiempo que fortalece su impacto en la protección de las especies en peligro. En México, existen diversas organizaciones y entidades dedicadas a la conservación de la biodiversidad y la protección de especies en peligro de extinción. Este proyecto podría beneficiarse significativamente de la colaboración con las siguientes organizaciones:

1. **Instituto Politécnico Nacional (IPN):** Dado que este proyecto está siendo realizado dentro del IPN, una colaboración interna con otras unidades o facultades de este prestigioso instituto podría brindar acceso a recursos y experiencia técnica adicional [?].
2. **Comisión Nacional de Áreas Naturales Protegidas (CONANP):** La CONANP es una entidad gubernamental encargada de la administración y conservación de áreas naturales protegidas en México [?]. Colaborar con ellos podría brindarnos acceso a áreas de alto valor ecológico y permitirnos realizar pruebas en entornos de conservación.
3. **Universidad Nacional Autónoma de México (UNAM):** Como una de las universidades más grandes de México, la UNAM alberga una amplia gama de expertos



en conservación y biodiversidad [?]. Colaborar con investigadores de la UNAM podría enriquecer nuestra comprensión y enfoque en la protección de especies en peligro de extinción.

4. **ONGs de Conservación:** Organizaciones no gubernamentales como Pronatura México [?], WWF México [?] y México Tierra de Jaguar [?] tienen un fuerte compromiso con la conservación de la fauna y flora mexicana. Colaborar con estas ONGs podría brindarnos apoyo técnico y financiero.
5. **Universidad de Veracruz:** Esta institución también tiene un interés significativo en la conservación de la biodiversidad, especialmente por encontrarse en la región de Veracruz, que alberga una gran diversidad de especies animales [?]. Esta colaboración podría permitir pruebas en entornos específicos de la región.
6. **Institutos de Ecología y Medio Ambiente:** Instituciones como el Instituto de Ecología de la UNAM [?] y el Instituto Nacional de Ecología y Cambio Climático (INECC) [?] cuentan con expertos en monitoreo ambiental. Podrían proporcionar orientación técnica y acceso a datos relevantes.
7. **Secretaría de Medio Ambiente y Recursos Naturales (SEMARNAT):** Esta entidad gubernamental está involucrada en la regulación y protección del medio ambiente en México [?]. La colaboración con SEMARNAT podría facilitar la obtención de permisos y la realización de pruebas en áreas protegidas.
8. **Instituto de Ecología, A.C. (INECOL):** INECOL es una institución de investigación dedicada a la ecología y la conservación de la biodiversidad [?]. Su experiencia en investigación ecológica y monitoreo ambiental podría proporcionar orientación técnica y brindar acceso a expertos en diversas áreas de la ecología.
9. **Parques Nacionales:** México cuenta con numerosos parques nacionales que albergan especies en peligro de extinción. Trabajar en conjunto con la administración de estos parques permitiría realizar pruebas en entornos naturales.



10. Centros de Rehabilitación de Fauna Silvestre: Organizaciones como Centros de Atención y Rehabilitación para la Fauna Silvestre (CARE) se dedican a la rehabilitación de animales silvestres. Colaborar con ellos podría proporcionar la oportunidad de probar nuestro sistema en especies en proceso de re-introducción.

La colaboración con estas instituciones se considera relevante para poder enriquecer nuestro conocimiento y experiencia en la conservación de especies en peligro de extinción. Mediante el intercambio de ideas, recursos y experiencias, se busca fortalecer el enfoque tecnológico y científico, con la esperanza de lograr un impacto significativo en la protección de la biodiversidad. Por lo que, aspirando a contribuir a la conservación de especies animales cuya preservación se encuentra en una situación crítica, se ha buscado establecer contactos sólidos con algunas de estas organizaciones y universidades antes mencionadas.

Por parte de la Universidad de Veracruz, nos pusimos en contacto con la Dra. Denise Spaan [?], quien generosamente compartió detalles de su investigación actual. La Dra. Spaan está realizando estudios sobre el mono araña en el sur de México, centrándose en particular en el conteo de estos primates. Inicialmente, llevaba a cabo monitoreo de forma tradicional, pero encontró que esta metodología era muy agotadora.

Para mejorar su enfoque, se asoció con la *Liverpool John Moores University* para trabajar conjuntamente en la implementación de nuevas tecnologías. En un intento por facilitar el conteo de primates, se utilizaron drones equipados con cámaras térmicas [?]. Sin embargo, el uso de drones grandes resultó ser un desafío para ella y su equipo de biólogos de campo debido a su tamaño y dificultades de maniobrabilidad.

Para abordar estas limitaciones, decidieron cambiar a drones DJI, que eran más maniobrables pero solo permitían grabar video. Luego, experimentaron combinando el monitoreo tradicional con el uso de drones, ya que desde el aire no podían ver claramente lo que ocurría debajo de la densa vegetación forestal.

Finalmente, exploraron técnicas acústicas, en las cuales colocaban dispositivos de grabación de sonido en el hábitat de los primates. Después de un período de tiempo, recogían los dispositivos y, utilizando un programa de filtrado de sonidos, identificaban los sonidos de interés



emitidos por los animales.

Cuando le presentamos nuestra propuesta de solución, la Dra. Spaan la consideró como una excelente opción y muy útil para biólogos de campo en ciertos contextos. Sin embargo, señaló que para su proyecto actual, con los monos araña que pasan la mayor parte de su tiempo en árboles, las cámaras trampa no serían la opción más adecuada. En su lugar, sugirió que nuestro prototipo sería más adecuado para animales que se desenvuelven principalmente en entornos terrestres, en lugar de aquellos que se encuentran principalmente en entornos arbóreos, lo cual fue una recomendación valiosa para considerar en este proyecto.

Por otra parte, se ha establecido contacto con el Dr. Salvador Mandujano Rodríguez [?] y el Maestro en Ciencias Alberto Rísquez Valdepeña [?], del INECOL, con el propósito de explorar la posibilidad de una colaboración en nuestro proyecto. La razón principal de esta comunicación se basa en la similitud de enfoques entre su proyecto actual, titulado 'Drones: observación de fauna y de hábitats desde el aire' [?], y nuestro proyecto. El trabajo en ambos proyectos involucra el monitoreo de la fauna y los hábitats, con un énfasis en la utilización de tecnologías de drones. Quienes muy amablemente han manifestado un significativo interés en nuestra iniciativa y han aceptado colaborar con nuestro proyecto. Estamos entusiasmados con la perspectiva de aprovechar su experiencia y conocimientos, lo cual contribuirá de manera sustancial al éxito de la propuesta.



9.7. Análisis del Animal a Estudiar

La selección y el análisis del animal a estudiar es esencial para el diseño efectivo de la red inalámbrica de sensores (WSN) y el enjambre de drones. En esta sección, se detallarán los criterios utilizados para seleccionar al animal objeto de estudio y se describirá su relevancia en el contexto de la conservación de especies en peligro de extinción. La elección del animal a estudiar se basará en varios criterios clave:

1. **Estado de Amenaza:** Se dará prioridad a especies clasificadas como en peligro de extinción o con una población críticamente reducida. Esto garantiza que nuestro proyecto tenga un impacto significativo en la conservación.
2. **Viabilidad Técnica:** Se considerará la viabilidad técnica para la monitorización de la especie. Esto implica evaluar si es posible recopilar datos relevantes sin perturbar el hábitat natural del animal.
3. **Acceso a Especies:** La disponibilidad de acceso a las áreas donde habita la especie es un factor crítico. Se priorizarán especies que se encuentren en áreas accesibles para el despliegue de drones y sensores.
4. **Importancia Ecológica:** Se dará preferencia a especies con un papel significativo en el ecosistema local. Su monitoreo puede ayudar a comprender y proteger no solo a la especie en sí, sino también a su entorno. **Interés Científico:** Se considerará el interés científico general en la especie, incluyendo el potencial de contribuir al conocimiento existente sobre su biología y comportamiento.
5. **Interés Científico:** Se considerará el interés científico general en la especie, incluyendo el potencial de contribuir al conocimiento existente sobre su biología y comportamiento.
6. **Consideración específica:** La experiencia compartida por la Dra. Denise Spaan en el capítulo Búsqueda y Vinculación con otros proyectos de investigación y/o organizaciones, centrada en el estudio de monos araña, subraya la importancia de incorporar las



preferencias y comportamientos específicos de las especies al seleccionar el sujeto de investigación. En el contexto de los monos araña, que predominantemente habitan en los árboles, se plantea la posibilidad de que la eficacia de las cámaras trampa pueda verse comprometida. Esta observación subraya la imperiosa necesidad de no solo evaluar el estado de amenaza y la viabilidad técnica al planificar un proyecto, sino también considerar los hábitos naturales del animal en su entorno, y cómo estos podrían incidir en el rendimiento global de este proyecto.

9.7.1. Elección del Animal a Estudiar

Al considerar diversos criterios, se han identificado cinco posibles candidatos para el estudio de conservación mediante el empleo de la WSN y el enjambre de drones. Cada candidato se evalúa según los criterios mencionados anteriormente:

1. **Venado de Cola Blanca (*Odocoileus virginianus*):** Es una especie de cérvido nativa de América. Aunque no está clasificado como en peligro de extinción, enfrenta diversas amenazas. Su viabilidad técnica como sujeto de estudio radica en su actividad en áreas abiertas, siendo común en lugares accesibles para drones. Este venado ejerce una influencia significativa en los ecosistemas locales y ha sido objeto de estudio por su comportamiento y reproducción [?].
2. **Lince Rojo (*Lynx rufus*):** Es un felino nativo de América del Norte. Aunque no se encuentra en peligro de extinción, su conservación es de preocupación menor. Su hábitat y tamaño de territorio están vinculados a la densidad de presas, siendo un regulador natural de poblaciones de roedores. La investigación se ha centrado en su comportamiento de caza y hábitos migratorios [?].
3. **Jaguar (*Panthera onca*):** Es el mayor felino de América y se encuentra casi amenazado. Su conservación se ha destacado como modelo para la planificación de la conservación de especies. A pesar de los desafíos técnicos en áreas remotas, el jaguar desempeña un pa-



pel crucial como principal depredador en su ecosistema. La investigación se ha enfocado en patrones de caza y migración [?].

4. **Oso Pardo (*Ursus arctos*):** Es una especie de oso que habita en diversas regiones del hemisferio norte. Aunque no está en peligro de extinción, su conservación es de preocupación menor. La estructura social de los osos pardos en Alberta ha sido objeto de estudio, destacando comportamientos como la hibernación y los movimientos estacionales [?].
5. **Tapir de Baird (*Tapirus bairdii*):** Es una especie de tapir que habita en áreas húmedas de América Central y América del Sur. Clasificado como vulnerable, enfrenta desafíos técnicos en su conservación en áreas húmedas. La investigación se ha centrado en su influencia en la regeneración del bosque, así como en patrones de alimentación y comportamiento en hábitats acuáticos [?].

Tras la evaluación, el venado de cola blanca se destaca como una elección sólida. Su presencia en áreas accesibles, comportamiento en ambientes abiertos y su importancia ecológica hacen que sea un candidato ideal para este proyecto de monitoreo y conservación.

9.7.1.1. Características más Importantes del Animal Seleccionado

El venado de cola blanca (*Odocoileus virginianus*) [?] es una especie formada por cerca de 38 subespecies que difieren notoriamente en tamaño, de las cuales 14 se distribuyen en territorio mexicano.

- **Características:** En áreas tropicales tienden a ser pequeños; los adultos miden entre 60 y 114 cm. de altura. El color generalmente es de pardo rojizo o pardo grisáceo. La garganta tiene un área blanquecina, que continúa bajo la mandíbula, excepto por un área negra localizada debajo de la comisura de la boca; el mentón es blanco. Los ojos están bordeados de blanco y el rostro es blanco en el extremo, atrás de la nariz. El vientre es blanquecino. La cola es parda por arriba, pero muy blanca abajo. Los machos presentan astas durante la época apareamiento.



- **Descripción de su hábitat:** Habitán generalmente zonas de montes que ofrezcan protección y la mayor parte de su alimento, ya que a pesar de que se alimentan temporalmente de pastos verdes y hierbas, son principalmente ramoneadores (forma de alimentación en que un herbívoro se alimenta de hojas, brotes tiernos o frutos de plantas de alto crecimiento).
- **Comportamiento:** Los venados de cola blanca son animales crepusculares, lo que significa que son más activos al amanecer y al atardecer. Son conocidos por ser cautelosos y tener un sentido agudo del oído y el olfato, lo que les ayuda a detectar depredadores.



Figura 9.4: Venado de cola blanca

CAPÍTULO 10

Diseño

10.1. Diseño de la WSN

10.1.1. Conexión Física de la WSN

La interconexión de los componentes clave en cada nodo sensor de la red inalámbrica de sensores (WSN) se llevará a cabo de manera precisa y adecuada para garantizar la operación coherente del sistema, como se muestra a continuación:

- **Microprocesador Arduino Pro Micro:**

El microprocesador Arduino Pro Micro, actuando como el cerebro de cada nodo sensor, establecerá conexiones directas a través de sus pines GPIO con cada uno de los sensores y módulos para la recopilación y procesamiento de datos. Además, se interconectará con el módulo LoRa UART para la transmisión inalámbrica de información.

- **Sensores Específicos:**

El termistor LM35, destinado a medir la temperatura, será conectado a los pines analó-



gicos del Arduino Pro Micro, garantizando una lectura precisa y oportuna de las condiciones ambientales.

El módulo GPS Neo-6M, crucial para la geolocalización, se integrará a través de conexiones seriales, permitiendo al Arduino interpretar y registrar las coordenadas geográficas.

■ **Comunicación Inalámbrica:**

El módulo LoRa UART 868/915 MHz establecerá una conexión serial con el Arduino Pro Micro, permitiendo la comunicación bidireccional entre los nodos sensores y la estación base. Esto facilitará la transmisión de datos y comandos, asegurando una coordinación efectiva dentro de la red.

La antena WiFi estará conectada directamente a la cámara trampa PR-700. Esta conexión permitirá que la cámara trampa se una a la red local a través de WiFi para la transmisión eficiente de imágenes y videos. La cámara trampa estará equipada con la capacidad de conectarse a una red WiFi, y la antena facilitará esta conexión para garantizar una transferencia de datos rápida y confiable.

■ **Almacenamiento:**

La memoria MicroSD a través del módulo MicroSD Card Adapter (modelo MLMSD), implementada como soluciones de almacenamiento local, se conectarán al Arduino Pro Micro a través de interfaces SPI (Serial Peripheral Interface). Esto posibilitará un acceso rápido y eficiente para escribir y leer datos, garantizando la efectividad en la gestión de la información recopilada por los sensores distribuidos.

■ **Fuente de Energía:**

Las baterías recargables, conectadas a través de sistemas de gestión de energía, suministrarán la potencia necesaria al Arduino Pro Micro y sus componentes asociados. La eficiencia energética será clave para mantener la autonomía de los nodos sensores durante operaciones prolongadas.

También se incorporarán baterías a la cámara trampa PR-700 para garantizar su operación autónoma. Estas baterías proporcionarán la energía necesaria para el funciona-



miento continuo de la cámara, asegurando la captura de imágenes y videos en diversas condiciones y entornos.

Esta configuración física(Figura 10.1) garantizará una interconexión robusta y eficiente, permitiendo que cada nodo sensor desempeñe sus funciones específicas de manera coordinada y efectiva dentro de la WSN.

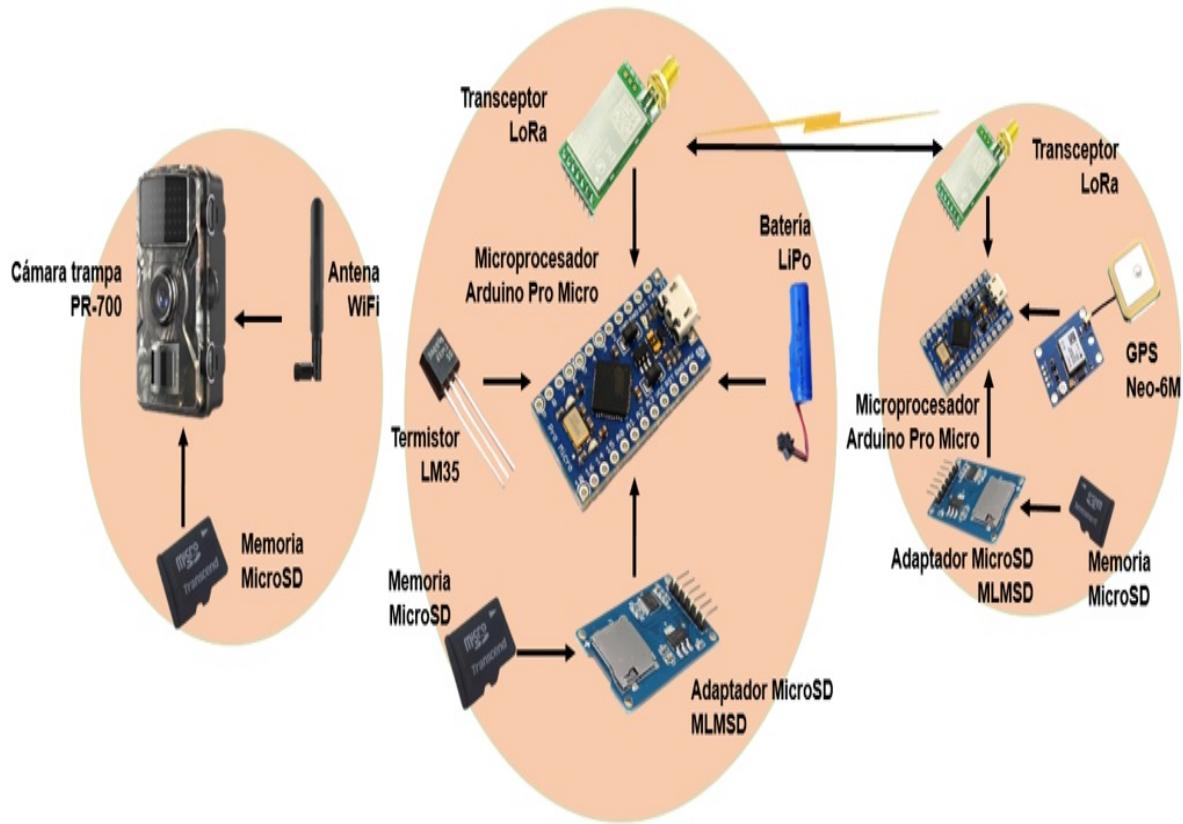


Figura 10.1: Diseño de la WSN.

10.1.2. Lenguaje de Programación de la WSN

El lenguaje de programación seleccionado para el desarrollo de la WSN será Arduino IDE, dada su accesibilidad y versatilidad en la programación de microcontroladores. Esta elección facilita la implementación de algoritmos y la gestión de la comunicación entre los nodos de la



red. A continuación, se detallan las principales librerías que se utilizarán en el desarrollo:

- **SoftwareSerial:** Será utilizada para establecer comunicación serial con componentes que requieran un puerto de comunicación adicional. Facilitará la conexión con módulos y dispositivos que no estén directamente conectados a los pines de hardware de comunicación serial [?].
- **LoRa by Sandeep Mistry:** Se empleará para la implementación de la comunicación inalámbrica mediante la tecnología LoRa. Permitirá la transmisión de datos de manera eficiente y de largo alcance entre los nodos de la WSN, contribuyendo a la formación de un enjambre coordinado [?].
- **TinyGPS:** Se aplicará para la interfaz con los módulos GPS. Facilitará la decodificación de datos de ubicación y simplificará el proceso de geolocalización en los nodos de la red [?].
- **SD:** La librería SD se utilizará para la gestión de la tarjeta MicroSD en los nodos que integran almacenamiento. Facilitará la lectura y escritura de datos en las tarjetas, permitiendo el almacenamiento local de información recolectada por los sensores [?].



10.2. Diseño de los Drones

El diseño de los drones desempeña un papel fundamental en el éxito de este proyecto. Cada elemento, desde el chasis hasta los componentes de propulsión y la fuente de energía, se utilizarán para garantizar un rendimiento óptimo y una adaptabilidad a las condiciones específicas del entorno de monitoreo.

10.2.1. Estructura Física

La estructura física de los drones se basará en un cuadricóptero, caracterizado por su configuración de cuatro rotores en disposición diagonal. Se ha seleccionado el chasis del kit S500 de la marca Readytosky, conocido como "engranaje". Este chasis, fabricado en fibra de carbono, ofrece una combinación ideal de resistencia y ligereza para optimizar el rendimiento del dron en operaciones de vuelo.

Con dimensiones de 11.42 pulgadas de longitud, 7.09 pulgadas de ancho y 2.36 pulgadas de altura, el chasis proporciona un equilibrio adecuado entre tamaño compacto y capacidad para albergar los componentes esenciales. La distancia entre ejes de 19.685 pulgadas contribuye a la estabilidad del dron.

La configuración sugerida del kit S500 incluye un peso total con tren de aterrizaje de 15.87 oz. Para la propulsión, se recomiendan motores sin escobillas en el rango de tamaños 2212-2216, junto con hélices de 10"-12". La batería sugerida abarca desde 3S a 4S con capacidades de 2200mAh a 5200mAh, y se sugiere un controlador de velocidad (ESC) en el rango de 20A-40A. Estas especificaciones brindan flexibilidad para adaptar el dron a los requisitos específicos del proyecto. Además, que este fabricado en fibra de carbono garantiza durabilidad y resistencia estructural, elementos esenciales para operaciones en entornos desafiantes.

10.2.2. Fuente de Energía

La energía necesaria para alimentar el dron se suministrará a través de una batería de polímero de litio (LiPo) 3S con una capacidad de 5000mAh y una clasificación de voltaje de



11.1V. Este tipo de batería proporciona una combinación óptima de potencia y peso para garantizar un rendimiento eficiente durante las operaciones de monitoreo.

La capacidad de 5000mAh ofrece una duración de vuelo considerable, lo que es esencial para misiones de monitoreo prolongadas. El voltaje de 11.1V se adapta a la configuración del sistema, brindando la potencia necesaria para los motores y otros componentes electrónicos del dron.

El conector XT-60 asegura una conexión segura y confiable entre la batería y el sistema eléctrico del dron. Este conector es conocido por su eficiencia en la transmisión de corriente, minimizando pérdidas y garantizando un suministro estable de energía durante la operación.

La elección de esta batería LiPo específica se basa en su equilibrio entre capacidad, voltaje y compatibilidad con el sistema eléctrico del dron. Esto contribuye a la eficiencia general del dron, asegurando una fuente de energía confiable para las diversas funciones del sistema durante las misiones de vuelo.

10.2.3. Componentes para Volar el Dron

El sistema de propulsión del dron estará compuesto por componentes de alta calidad que garantizan un rendimiento confiable y eficiente durante las operaciones de vuelo. A continuación se detallan los elementos clave de este sistema:

- **Par de Hélices 1045 de Carbono y Nylon para F450/F550:** Cada ala del dron estará equipada con un par de hélices 1045 fabricadas con una combinación de carbono y nylon. Estas hélices proporcionan un equilibrio óptimo entre resistencia y ligereza, contribuyendo a la eficiencia del vuelo y asegurando una generación de empuje adecuada.
- **Motor sin Escobillas 2212 920KV (actualización CCW):** El sistema de propulsión contará con motores sin escobillas de alta calidad, específicamente el modelo 2212 con una clasificación de 920KV y configuración de rotación antihoraria (CCW). Estos motores ofrecen un rendimiento potente y eficiente, adecuado para las demandas de misiones de monitoreo y vuelo estable.



- **Pixhawk 2.4.8 Autopiloto Controlador PM PPM PX4 Ardupilot:** El corazón del sistema de vuelo será el controlador Pixhawk 2.4.8, que opera bajo los sistemas PX4 y Ardupilot. Este autopiloto proporciona funciones avanzadas de control de vuelo, navegación y estabilización. Su capacidad para integrarse con sistemas de posicionamiento, como el GPS Here+, garantiza una precisión y confiabilidad superiores durante las misiones de monitoreo.

La combinación de estas propelas, motores y el sistema de control autopiloto como se puede ver en la Figura 10.2 proporciona una configuración integral que respalda un vuelo controlado y preciso del dron durante las misiones de vuelo.

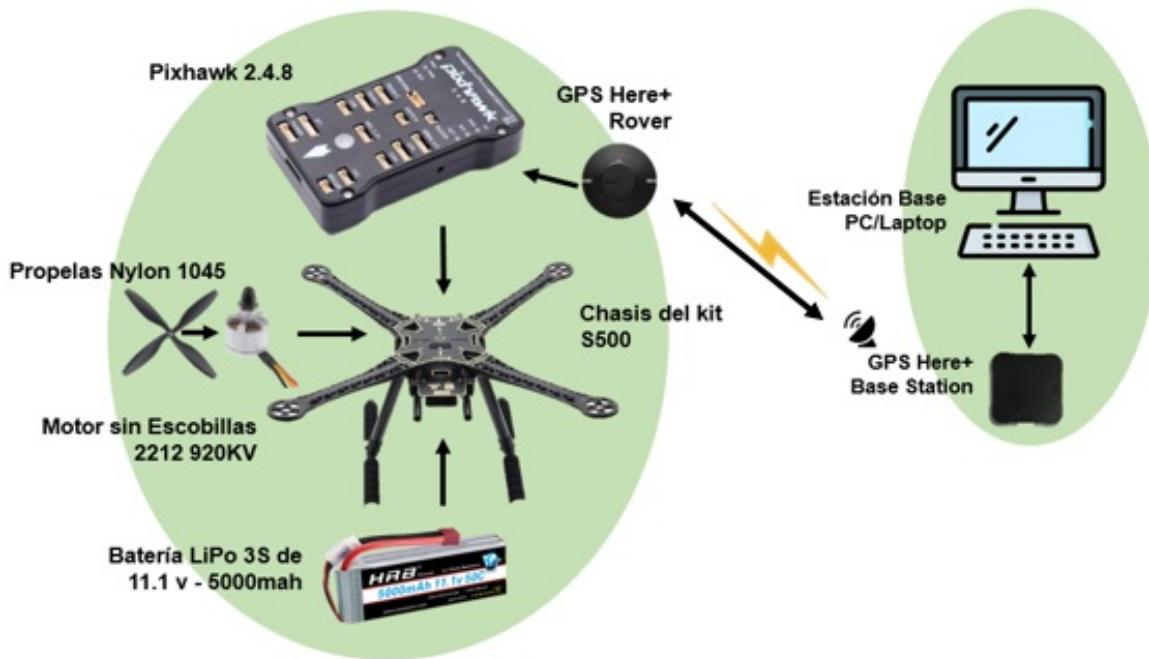


Figura 10.2: Diseño del dron.



10.3. Diseño del Enjambre de Drones

El diseño del enjambre de drones para el monitoreo de especies en peligro de extinción representa la culminación de un análisis exhaustivo y una planificación meticulosa. Este concepto innovador fusiona la eficiencia individual de los drones con una coordinación inteligente y descentralizada para abordar los desafíos críticos de la conservación biológica. En este diseño, cada dron individual se convierte en un nodo esencial en una red autónoma y colaborativa. La integración de microordenadores y transceptores de comunicación, potenciará la capacidad de procesamiento y la conexión inalámbrica entre los drones, respectivamente. Este enfoque permitirá una ejecución de algoritmos avanzados y una coordinación en tiempo real entre los miembros del enjambre.

10.3.1. Arquitectura Física

La arquitectura física del enjambre de drones aprovechará la sólida configuración de cuadricópteros, heredada del diseño individual de drones descrito previamente. Que se conforma por el chasis del kit S500 de fibra de carbono, baterías LiPo 3S, etc.

La evolución hacia un enjambre inteligente implica la incorporación estratégica de componentes clave. A la estructura existente se le añadirán un microordenador Raspberry Pi 4 y un transceptor LoRa UART 868MHz/915MHz. Estos elementos, cuidadosamente seleccionados durante el análisis, elevan la funcionalidad de cada dron a una capacidad de procesamiento superior y a una comunicación eficiente entre ellos.

En el contexto del diseño del enjambre de drones, la integración del microordenador Raspberry Pi 4 representa un avance significativo en la capacidad de procesamiento del sistema. Este componente actúa como el cerebro central de cada dron, brindando la potencia computacional necesaria para ejecutar algoritmos avanzados. La inclusión de Raspberry Pi 4 no solo permite un procesamiento más eficiente de los datos recopilados por los sensores del dron, sino que también habilita la implementación de algoritmos complejos para tareas como la navegación autónoma y la toma de decisiones en tiempo real. Además, la introducción del transceptor



LoRa UART 868MHz/915MHz mejora significativamente la capacidad de comunicación entre los drones del enjambre. La frecuencia LoRa seleccionada (868MHz/915MHz) garantiza una cobertura adecuada y una transmisión confiable de datos, elementos cruciales para el éxito de las misiones de monitoreo en entornos diversos y desafiantes.

En conjunto, la combinación de Raspberry Pi 4 y el transceptor LoRa fortalece la inteligencia colectiva del enjambre, mejorando su capacidad para ejecutar misiones de manera colectiva. Así, la estructura física del enjambre fusiona la confiabilidad estructural del diseño individual del dron con la potencia adicional de estos nuevos componentes, creando una plataforma integral que maximiza la inteligencia colectiva del enjambre para afrontar con eficacia las misiones de este proyecto.

10.3.2. Algoritmos de Comunicación

En el diseño del sistema de comunicación para el enjambre de drones, se implementarán algoritmos que permitan una transmisión eficiente de instrucciones desde la estación base hasta los drones individuales. Este enfoque optimiza el proceso al evitar la necesidad de enviar instrucciones directamente a cada dron, delegando esta tarea al dron líder. A continuación, se describen los algoritmos clave para facilitar esta comunicación jerárquica:

1. **Instrucciones de la estación base al dron líder:** La estación base enviará instrucciones y comandos generales al dron líder a través de la red de comunicación establecida. Estas instrucciones pueden incluir cambios en la ruta de vuelo, ajustes en la altitud, solicitudes de recopilación de datos específicos, entre otros.
2. **Transmisión de instrucciones del dron líder a los drones subordinados:** El dron líder, al recibir las instrucciones de la estación base, actúa como el punto central de coordinación dentro del enjambre. Utilizando el transceptor LoRa, el dron líder transmite las instrucciones a los demás drones de manera eficiente, asegurando una comunicación rápida y confiable. Este proceso se realiza de manera iterativa y continua, permitiendo una actualización constante de las instrucciones a medida que la misión avanza.



3. **Confirmación de recepción y ejecución:** Cada dron, al recibir las instrucciones, envía una confirmación al dron líder para indicar la recepción exitosa. Una vez que el dron líder recibe confirmaciones de todos los drones subordinados, puede informar a la estación base sobre el estado de la transmisión y la ejecución de las instrucciones.
4. **Gestión de conflictos y replanificación:** En caso de que un dron subordinado encuentre un conflicto o imprevisto durante la ejecución de las instrucciones, informará al dron líder. El dron líder puede, entonces, replanificar la misión, ajustando las instrucciones y transmitiendo las actualizaciones correspondientes al enjambre.

Este enfoque jerárquico en la comunicación permite una mayor eficiencia operativa, ya que centraliza la gestión de instrucciones en el dron líder, reduciendo la carga de comunicación desde la estación base y mejorando la sincronización del enjambre durante las misiones de monitoreo. Además, la implementación de confirmaciones y mecanismos de gestión de conflictos contribuye a la robustez y adaptabilidad del sistema en entornos dinámicos.

10.3.3. Programación de Rutas

La programación de rutas es un componente crítico en el despliegue eficiente del enjambre de drones para el monitoreo de especies en peligro de extinción. Para esta tarea, se empleará Mission Planner 1.3.81, una plataforma de planificación de vuelo robusta y versátil. A continuación, se detallan los pasos clave y consideraciones para la programación de rutas utilizando esta herramienta:

1. **Configuración inicial:** Antes de iniciar la programación de rutas, se realizará una configuración inicial en Mission Planner. Esto incluirá la conexión de la estación base con el dron líder a través de la interfaz de comunicación establecida.
2. **Definición de puntos de interés:** Se identificarán y marcarán los puntos de interés críticos en el área de monitoreo, como hábitats de especies en peligro o zonas específicas que requieran mayor atención. Estos puntos servirán como referencias para la planificación de rutas y la asignación de tareas específicas a los drones.



3. **Creación de rutas autónomas:** Utilizando Mission Planner, se diseñarán rutas autónomas para cada dron en el enjambre. Estas rutas se basarán en la ubicación de los puntos de interés y se optimizarán para cubrir eficientemente el área de monitoreo. Se programarán altitudes específicas, velocidades de vuelo y patrones de mapeo para garantizar una cobertura completa y detallada del terreno.
4. **Implementación de puntos de control:** Se establecerán puntos de control a lo largo de las rutas para garantizar la precisión y estabilidad del vuelo. Estos puntos permitirán ajustes en tiempo real durante la misión. Los puntos de control también facilitarán la coordinación entre los drones, asegurando una distribución equitativa de las tareas y evitando superposiciones no deseadas.
5. **Verificación y simulación:** Antes de la ejecución en campo, se verificarán las rutas programadas y se realizarán simulaciones en Mission Planner. Esto permite identificar posibles problemas y optimizar la eficiencia de las rutas antes del despliegue real.
6. **Carga de misiones a los drones:** Una vez finalizada la programación y verificación, las misiones planificadas se cargarán en cada dron del enjambre. Esto se realizará de manera centralizada desde la estación base, asegurando consistencia en las instrucciones.
7. **Monitoreo en tiempo real:** Durante la ejecución de la misión, Mission Planner proporcionará herramientas para monitorear en tiempo real la posición de cada dron, la cobertura del área y cualquier desviación con respecto a la ruta planificada.

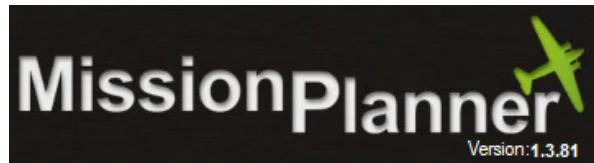


Figura 10.3: Logo de Mission Planner.

La adición de estos componentes al diseño de un dron individual, como se puede ver en la Figura 10.4, les permitirá a los drones comportarse de una forma colectiva simulando el comportamiento de un enjambre.

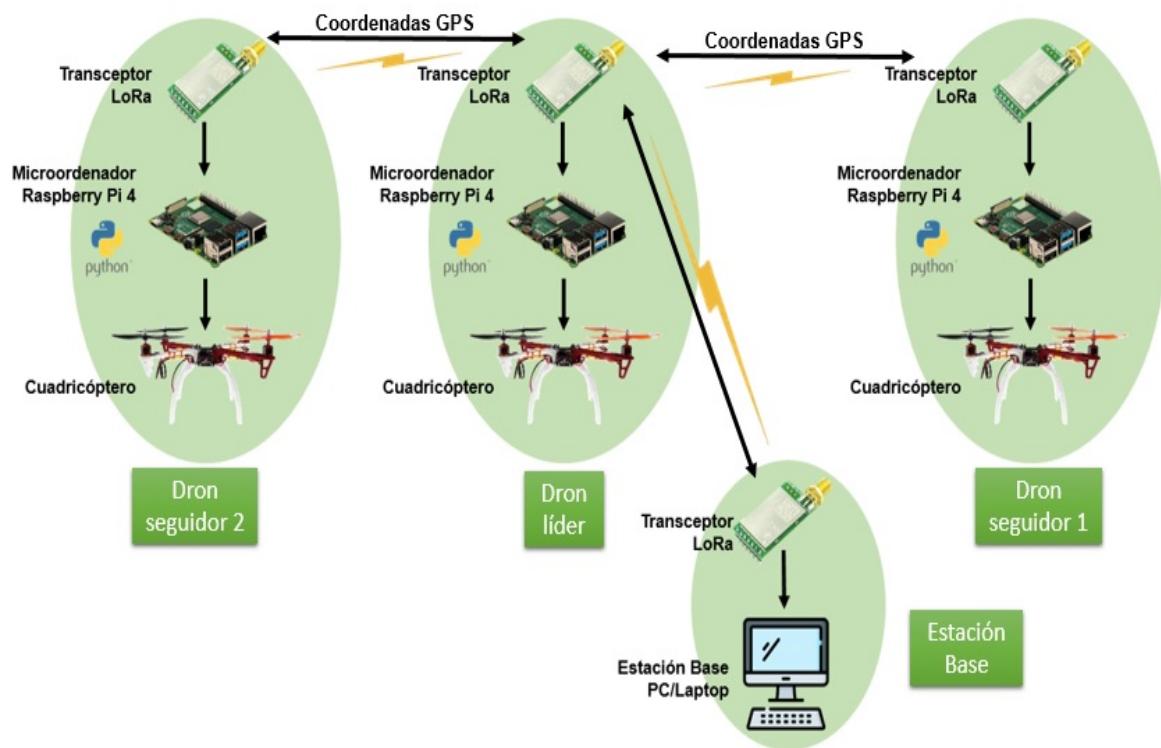


Figura 10.4: Diseño del enjambre de drones.



10.4. Diseño del Dominio en la Nube

Para el diseño en la nube, debemos de tener en cuenta varios servicios que nos interesan adquirir, serán una mezcla de varios servicios que pueden interactuar unos con otros para automatizar ciertos procesos como por ejemplo, el ordenamiento de los datos por fechas y que se clasifiquen dependiendo del tipo de archivo que sea, como puede ser audio, vídeo, imágenes o simplemente datos; también podremos programar ciertas visualizaciones de los datos obtenidos y procesarlos de manera que tengamos datos limpios (nos referimos a datos limpios como los datos que nos son útiles para procesar en visualizaciones, análisis o similar, sin presentar inconsistencias, valores repetidos o datos incorrectos). Uno de los servicios que nos ofrece AWS es Cognito, que lo usaremos para autenticación de los usuarios que tendrán acceso a la base de datos, por lo que vamos a explicarlo a continuación.

10.4.1. Amazon Cognito

Amazon Cognito es una plataforma de identidad para aplicaciones web y móviles. Es un directorio de usuarios, un servidor de autenticación y un servicio de autorización para los tokens y credenciales de AWS de acceso de OAuth 2.0. Con Amazon Cognito, puede autenticar y autorizar a los usuarios desde el directorio de usuarios integrado, desde el directorio empresarial y desde proveedores de identidad de consumidores como Google y Facebook. OAuth 2.0 es un protocolo de autorización y NO un protocolo de autenticación, utiliza tokens de acceso[?]. Los dos componentes siguientes componen Amazon Cognito. Funcionan de forma independiente o en conjunto, en función de las necesidades de acceso de los usuarios.

10.4.1.1. Grupo de Usuarios

Sirve para crear un grupo de usuarios cuando quiera autenticar y autorizar a los usuarios a la aplicación o la API. Los grupos de usuarios son un directorio de usuarios con funciones de creación, administración y autenticación de usuarios automáticas e impulsadas por el administrador. Los grupos de usuarios no requieren la integración con un grupo de identidades. Desde

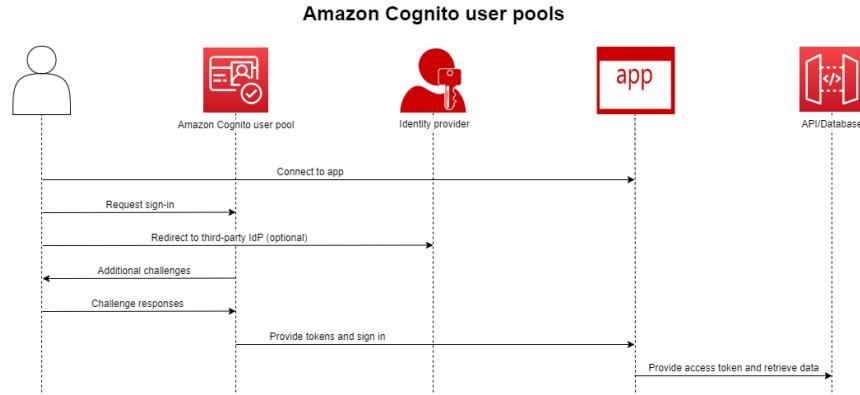


Figura 10.5: Grupo de Usuarios de Amazon Cognito diagrama de funcionamiento

un grupo de usuarios, puede emitir JSON Web Token (JWT) autenticados directamente a una aplicación, un servidor web o una API.

Un grupo de usuarios de Amazon Cognito es un directorio de usuarios. Con un grupo de usuarios, los usuarios pueden iniciar sesión en su aplicación web o móvil por medio de Amazon Cognito o federarse mediante un IdP de terceros. Los usuarios federados y locales tienen un perfil de usuario en el grupo de usuarios. Los grupos de usuarios de Amazon Cognito aceptan tokens y afirmaciones de IdP de terceros y recopilan los atributos de los usuarios en un JWT(JSON Web Token) que emite a la aplicación. Puede estandarizar la aplicación en un conjunto de JWT mientras Amazon Cognito gestiona las interacciones con los IdP y asigna las reclamaciones a un formato de token central.

10.4.1.2. Grupos de Identidades

Configurar un grupo de identidades de Amazon Cognito cuando deseé autorizar a los usuarios autenticados o anónimos a acceder a los recursos de AWS. Un grupo de identidades emite credenciales de AWS para que la aplicación proporcione recursos a los usuarios. Puede autenticar a los usuarios con un proveedor de identidades de confianza, como un grupo de usuarios. También puede emitir, opcionalmente, credenciales para los usuarios invitados. Los grupos de identidades utilizan un control de acceso basado en roles y atributos para administrar

Amazon Cognito federated identities (identity pools)

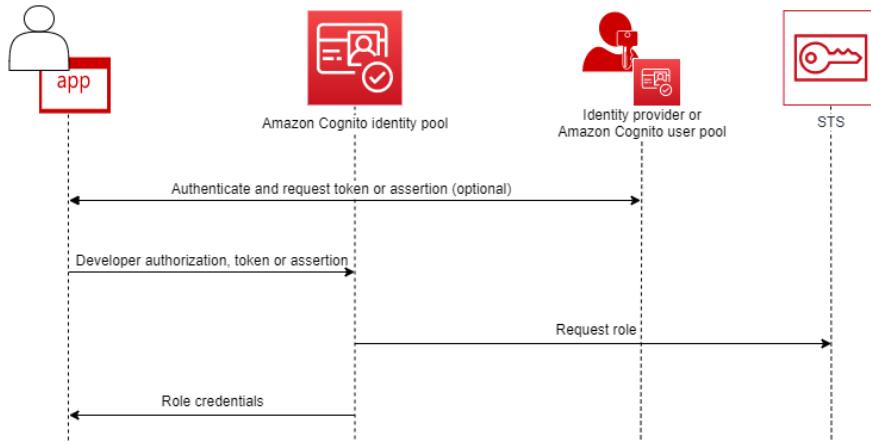


Figura 10.6: Grupo de Usuarios de Amazon Cognito diagrama de funcionamiento

la autorización de los usuarios para acceder a los recursos de AWS.

Se utiliza Amazon Cognito para autenticar al usuario y, a continuación, concederle acceso a un Servicio de AWS.

1. El usuario de la aplicación inicia sesión a través de un grupo de usuarios y recibe los tokens de OAuth 2.0.
2. La aplicación intercambia un token de grupo de usuarios por un grupo de identidades para obtener credenciales de AWS temporales que puede usar con las API de AWS y la AWS Command Line Interface (AWS CLI).
3. La aplicación asigna la sesión de credenciales al usuario y proporciona acceso autorizado a Servicios de AWS como Amazon S3 y Amazon DynamoDB.

En Amazon Cognito, la obligación de seguridad de la nube del modelo de responsabilidad compartida cumple con SOC 1-3, PCI DSS, ISO 27001 e HIPAA-BAA. Puede diseñar la seguridad en la nube en Amazon Cognito para que cumpla con SOC1-3, ISO 27001 e HIPAA-BAA, pero no con DSS de PCI.



Un grupo de identidades es un conjunto de identificadores o identidades únicas que se asignan a los usuarios o invitados y se autoriza a recibir credenciales temporales de AWS. Al presentar una prueba de autenticación en un grupo de identidades en forma de afirmaciones fiables de un proveedor de identidades sociales (IdP) de SAML 2.0, OpenID Connect (OIDC) u OAuth 2.0, se asocia al usuario con una identidad del grupo de identidades. El token que el grupo de identidades crea para la identidad puede recuperar las credenciales de sesión temporales de AWS Security Token Service (AWS STS).

Para complementar las identidades autenticadas, también se puede configurar un grupo de identidades para autorizar el acceso de AWS sin autenticación de IdP. Puede ofrecer su propia prueba de autenticación personalizada o no tener autenticación. Puede conceder credenciales de AWS temporales a cualquier usuario de la aplicación que las solicite, con identidades no autenticadas.

10.4.1.3. Control de Acceso con Base en Roles

Cuando el usuario pasa las reclamaciones al grupo de identidades, Amazon Cognito elige el rol de IAM que solicita. Para personalizar los permisos del rol según las necesidades, se aplican las políticas de IAM a cada rol. Amazon Cognito puede solicitar un rol predeterminado, un rol basado en reglas que consultan las reclamaciones del usuario o un rol basado en la suscripción al grupo del usuario en un grupo de usuarios. También puede configurar la política de confianza de roles para que IAM confíe solo en el grupo de identidades para generar sesiones temporales[?].

10.4.2. AWS Identity and Access Management

AWS Identity and Access Management (IAM) es un servicio web que ayuda a controlar de forma segura el acceso a los recursos de AWS. Con IAM, se puede administrar de forma centralizada los permisos que controlan a qué recursos de AWS pueden acceder los usuarios. Se utiliza IAM; para controlar quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos.

Cuando se crea una Cuenta de AWS, se comienza con una identidad de inicio de sesión que



tiene acceso completo a todos los recursos y Servicios de AWS de la cuenta. Esta identidad recibe el nombre de usuario raíz de la Cuenta de AWS y se accede a ella iniciando sesión con el email y la contraseña que utilizó para crear la cuenta. Se recomienda no utilizar la cuenta raíz para hacer actividades cotidianas[?].

Los 2 servicios anteriores únicamente son para autenticación lo cual es una herramienta muy importante ya que los datos que vamos a almacenar, aunque no sean confidenciales, sí son muy importantes y servirán para los análisis profesionales de los biólogos; y es destacable la forma de autenticación y verificación de las personas que se van a encargar del almacenamiento de los datos (nosotros mismos contaremos con los roles de administradores), ya que nosotros mismos nos encargaremos de darle seguridad e integridad a los datos para que no tengan vulnerabilidades los accesos y no entren intrusos a la información.

10.4.3. Amazon Lambda

AWS Lambda es un servicio informático que permite ejecutar código sin aprovisionar ni administrar servidores. Con Lambda, lo único que tiene que hacer es suministrar el código en uno de los tiempos de ejecución de lenguaje compatibles con Lambda [?].

Con el servicio de lambda podemos automatizar el proceso de limpieza de datos y el propio almacenamiento en la base de datos que vamos a utilizar, y podemos utilizar diferentes lenguajes de programación como Java o Python para procesar los datos.

10.4.4. Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento, se utiliza Amazon S3 para almacenar y proteger cualquier cantidad de datos para diversos casos de uso, tales como lagos de datos, sitios web, aplicaciones móviles, copia de seguridad y restauración, archivado, aplicaciones empresariales, dispositivos IoT y análisis de big data.

Amazon S3 proporciona funciones de gestión para que pueda optimizar, organizar y configurar el acceso a los datos almacenados.



Figura 10.7: Traslado de los datos a Amazon S3, administración de los datos almacenados en Amazon S3 y análisis de los datos con otros servicios de AWS.

El diagrama muestra cómo trasladar los datos a Amazon S3, administrar los datos almacenados en Amazon S3 y analizar los datos con otros servicios. Se muestran tres secciones de izquierda a derecha [?].

Amazon S3 ofrece varios tipos de almacenamiento diseñados para distintos casos de uso. Por ejemplo, puede almacenar datos de producción críticos en S3 Standard para obtener acceso frecuente, ahorrar costos al almacenar datos a los que se accede con poca frecuencia en S3 Standard-IA o S3 One Zone-IA, y archivar datos con los costos más bajos en S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval y S3 Glacier Deep Archive.

También, se pueden instanciar páginas web en el servicio de S3, por lo que nos será útil para cuando diseñemos la página web para visualización y acceso a los datos de usuarios que tengan permiso y acceso a ellos.

10.4.5. Amazon DynamoDB

Amazon DynamoDB es un servicio de base de datos NoSQL totalmente administrado que ofrece un rendimiento rápido y predecible, así como una perfecta escalabilidad. Permite delegar las cargas administrativas que supone tener que utilizar y escalar bases de datos



distribuidas, para que no tengamos que preocuparnos del aprovisionamiento, la instalación ni la configuración del hardware, ni tampoco de las tareas de replicación, aplicación de parches de software o escalado de clústeres. DynamoDB también ofrece el cifrado en reposo, que elimina la carga y la complejidad operativa que conlleva la protección de información confidencial.

Podemos crear tablas de base de datos capaces de almacenar y recuperar cualquier cantidad de datos, así como de atender cualquier nivel de tráfico de solicitudes. Se puede escalar la capacidad de rendimiento de las tablas para aumentarla o reducirla sin tiempos de inactividad ni reducción del desempeño.

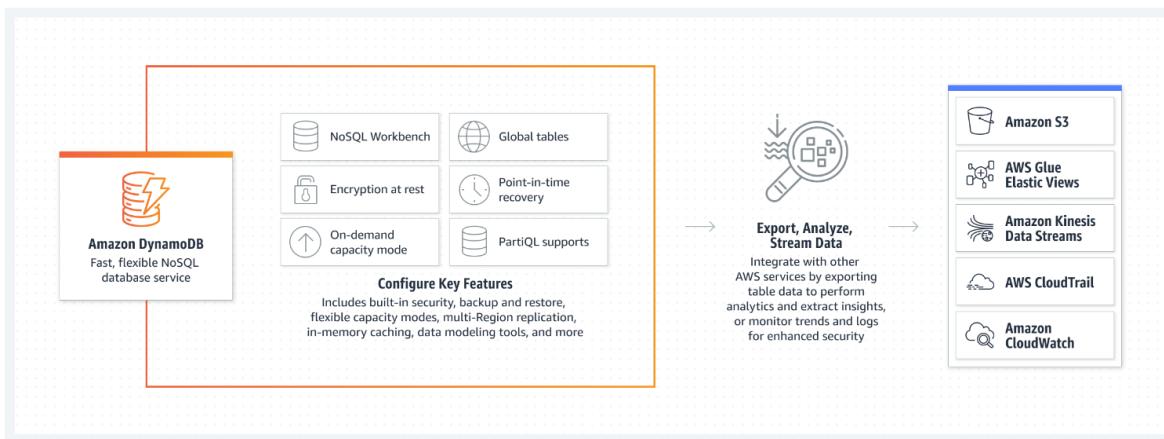


Figura 10.8: Características principales de Amazon DynamoDB y las integraciones con otros servicios de AWS

En el diagrama se muestran las características principales de Amazon DynamoDB y las integraciones con otros servicios de AWS. Se muestran tres secciones de izquierda a derecha.

Las características de importación y exportación de DynamoDB ayudan a mover, transformar y copiar cuentas de tablas de DynamoDB, o AWS. Se puede importar desde sus orígenes de S3 y puede exportar los datos de sus tablas de DynamoDB a Amazon S3 y utilizar servicios de AWS como Athena, Amazon SageMaker y AWS Lake Formation que son servicios que se pueden considerar introducir en algún momento para analizar los datos y extraer información procesable. También se puede importar datos directamente a nuevas tablas de DynamoDB.



para crear nuevas aplicaciones con un rendimiento de un milisegundo a escala, facilitar el uso compartido de datos entre tablas y cuentas, y simplificar sus planes de recuperación de desastres y continuidad empresarial [?].

10.4.6. Amazon QuickSight

Amazon QuickSight es una herramienta que nos ayudará a crear presentaciones gráficas sobre los datos que obtengamos, podremos utilizar diversas gráficas que nos ayudarán a una mejor visualización de los datos, serán más entendibles que simple texto plano y serán más amigables de entregar en reportes de análisis de datos.

Lo que estaremos usando serán estos servicios de almacenamiento, gestión y seguridad de datos. Es muy importante la conexión e integración de estos servicios ya que cada uno realizará una tarea en específico para lograr los objetivos de análisis de la especie a analizar.

El siguiente esquema es una representación gráfica de lo que vamos a hacer en el proyecto.

Habrá 2 administradores que serán los alumnos que desarrollarán el proyecto y tendrán todos los permisos, contaremos con 2 usuarios (depende como evolucione el proyecto se irán incluyendo más usuarios) los cuales no contarán con todos los permisos de escritura y únicamente contarán con los permisos de visualización de los datos, y estos 2 usuarios serán los asesores del proyecto. Usaremos el servicio de Amazon Cognito para la validación de identidades de los usuarios (se aplicará a los usuarios administradores y los usuarios normales), posteriormente usaremos 2 servicios de almacenamiento, el servicio de S3 lo usaremos como almacenamiento de todos los datos que involucran al proyecto, tales son archivos de video, imagen, audio, texto plano, gráficos, entre otros tipos de archivos, el servicio de DynamoDB será ocupado como la base de datos NoSQL en la cuál vamos a crear nuestras tablas NoSQL donde organizaremos nuestros archivos.

Es muy importante hacer la diferencia entre los servicios DynamoDB y S3, DynamoDB es una base de datos que se almacenan en caché y únicamente crearemos tablas NoSQL para identificación de archivos, por lo que para almacenar los datos necesitamos el servicio de S3 que será el lugar de almacenamiento de todos los archivos descritos anteriormente y también

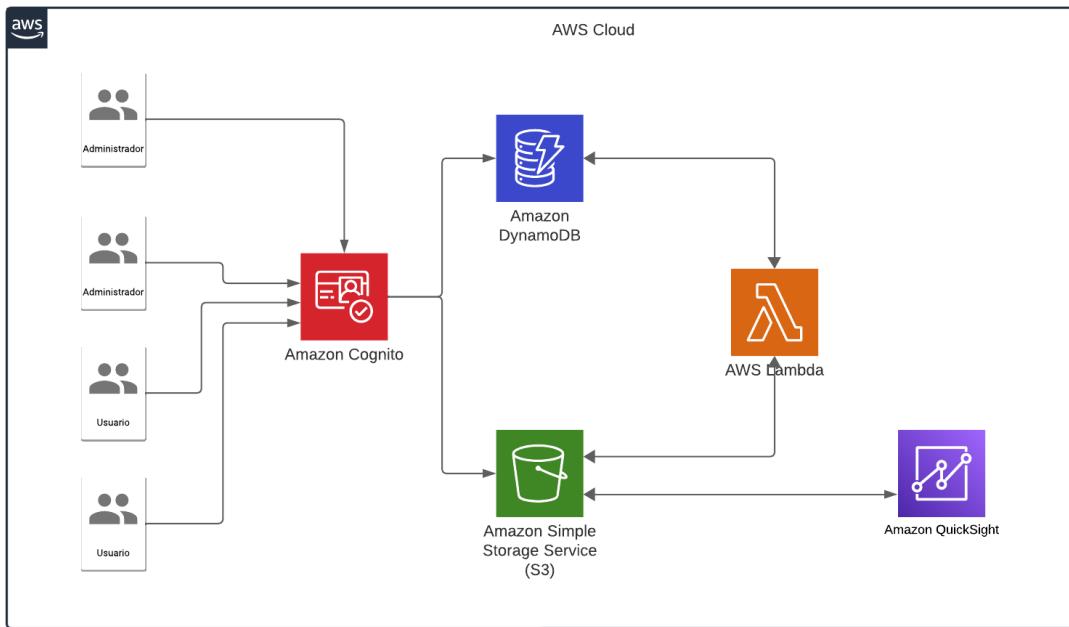


Figura 10.9: Diagrama de bloques de los servicios que se consideran para el desarrollo del proyecto.

almacenaremos copias de seguridad y las propias tablas que generamos en DynamoDB.

Usaremos el servicio de Lambda que estará vinculado a los servicios de S3 y DynamoDB para procesar archivos. Con lo de procesar los datos nos referimos a la limpieza de los datos y almacenamiento automático de los datos. Ya que el servicio de Lambda nos ayuda a crear funciones que nos permitirán procesar los datos con diferentes lenguajes de programación, entre ellos podemos usar Java y Python.

El servicio de QuickSight únicamente lo tomamos para visualización de los datos, ya que es muy necesario interpretar de forma correcta los datos obtenidos por nosotros y crear gráficas o recursos visuales que nos ayuden a entender de forma más sencilla los datos limpios, aunque este servicio lo podemos reemplazar por el propio AWS Lambda o algún programa de manipulación de datos local como PowerBI u otros en línea como Tableau que se pueden colocar nuestros resultados en línea para consulta de terceros (esto es opcional y dependerá de los permisos de



publicación de datos con las vinculaciones que tengamos con los biólogos).



10.5. Creación del Dominio en la Nube

El dominio en la nube se elaboró con una cuenta de los alumnos que están desarrollando el proyecto y contiene la siguiente información.

Configuración de la cuenta		Editar
ID de cuenta 311681021224	Proveedor de servicios Amazon Web Services, Inc.	
Nombre de la cuenta PT_TTT_2023_2_14	Contraseña *****	

Figura 10.10: Características del dominio en la nube.



Ya creado el dominio, podemos observar la página de inicio y podemos empezar a usar los servicios; Es importante estar inspeccionando los precios de los servicios para estar informados sobre los movimientos que vamos haciendo y saber cuánto dinero es necesario pagar al final de la fecha de facturación.

The screenshot shows the AWS Home Page with the following sections:

- Visitados recientemente:** Billing, AWS Cost Explorer, Amazon Location Service, Lambda, S3, DynamoDB, Security Hub, Cloud9.
- Le damos la bienvenida a AWS:**
 - Introducción a AWS: Conozca los aspectos fundamentales y encuentre información valiosa para sacar el máximo provecho de AWS.
 - Formación y certificación: Aprenda de expertos de AWS, mejore sus habilidades y aumente sus conocimientos.
 - ¿Cuáles son las novedades de AWS? (Link)
- AWS Health:**
 - Problemas abiertos: 0 (Últimos 7 días)
 - Cambios programados: 0 (Próximos 7 días y últimos 7 días)
 - Otras notificaciones: 0 (Últimos 7 días)
- Costo y uso:**
 - Costos del mes actual: 0,00 US\$
 - Costos finales de mes previstos: 0,00 US\$
 - Costos del mes pasado: 0,00 US\$
 - Costos principales del mes actual
 - Sin desgloses de costos principales: Los desgloses de costos se muestran cuando usa los servicios.
- Crear una solución:**
 - Lance una máquina virtual: Con EC2 (2 minutos)
 - Registre un dominio: Con Route 53 (5 minutos)
 - Comience un proyecto de desarrollo: Con CodeDeploy (5 minutos)
 - Cree una aplicación web: Con AWS App Runner (5 minutos)
 - Implemente un microservicio sin servidor: Con Lambda (2 minutos)
 - Comience a migrar a AWS: Con AWS MGN (2 minutos)
 - Aloje una aplicación web estática: Con AWS Amplify Console (2 minutos)
 - Cree SQL Server en AWS: Con alta disponibilidad (HA) y FCI (2 minutos...)
 - Desplegar SAP en AWS: Con NetWeaver y HANA (con HA) (10 min...)

Figura 10.11: Visualización de la página principal al iniciar sesión en AWS

CAPÍTULO 11

Implementación del sistema

11.1. Implementación de los drones

Para el proceso de documentación del armado de los drones, se tiene en cuenta los siguientes elementos :

- Chasis de Kit: S500
- Propelas: Nylon 1045
- Motor sin escobillas: 2212 920KV
- Controladoras de Velocidad: RC Brushless (ESC)
- Power Distribution Board (PDB) y Power Module: Apm2.8 2.6 2.52
- Batería LiPo 3S de 11.1V - 5000mAh
- Controladora de vuelo: Pixhawk 2.4.8
- Sistema de GPS: HERE+
- Sistema de Telemetría NRF24L01
- Transceptor De Telemetría Personalizado: NRF24L01



- Raspberry Pi4
- Xiangtat Flysky FS-i6X 2.4GHz 10CH AFHDS 2A RC Transmisor TX con receptor iA6B (Radio transmisor para control manual)

Comenzando con el armado del dron, se utilizará el Chasis S500, el cual viene con los siguientes elementos:

- Placa central PCB con punto de soldadura
- Tubos de fibra de carbono de 16 mm de diámetro con una altura de 250 mm
- Tubos de fibra de carbono de 10 mm de diámetro con longitud de 250 mm para la base
- Tubos de esponja anti salto
- Brazos DJI S800 EVO con 6 agujeros para colocar motores



Figura 11.1: Componentes del Kit S500.

El primer paso para el armado del chasis S500 es para el tren de aterrizaje. Se arma tal como se muestra en la siguiente imagen:



Figura 11.2: Base para el tren de aterrizaje.

Se colocan los tubos de fibra de carbono para que formen la base del cuadricóptero. Se van a montar sobre la placa central para formar la base.

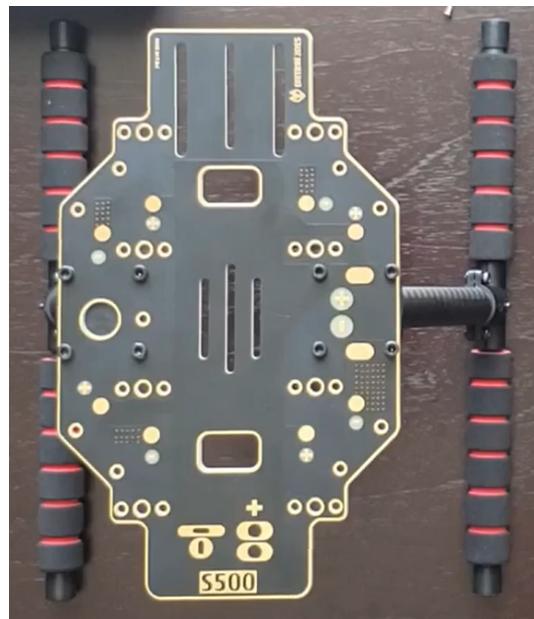


Figura 11.3: Chasis montado sobre el tren de aterrizaje.

Cuando ya está la base armada, se procede al armado de los brazos de la placa superior que

consta de los 4 brazos y la placa superior.

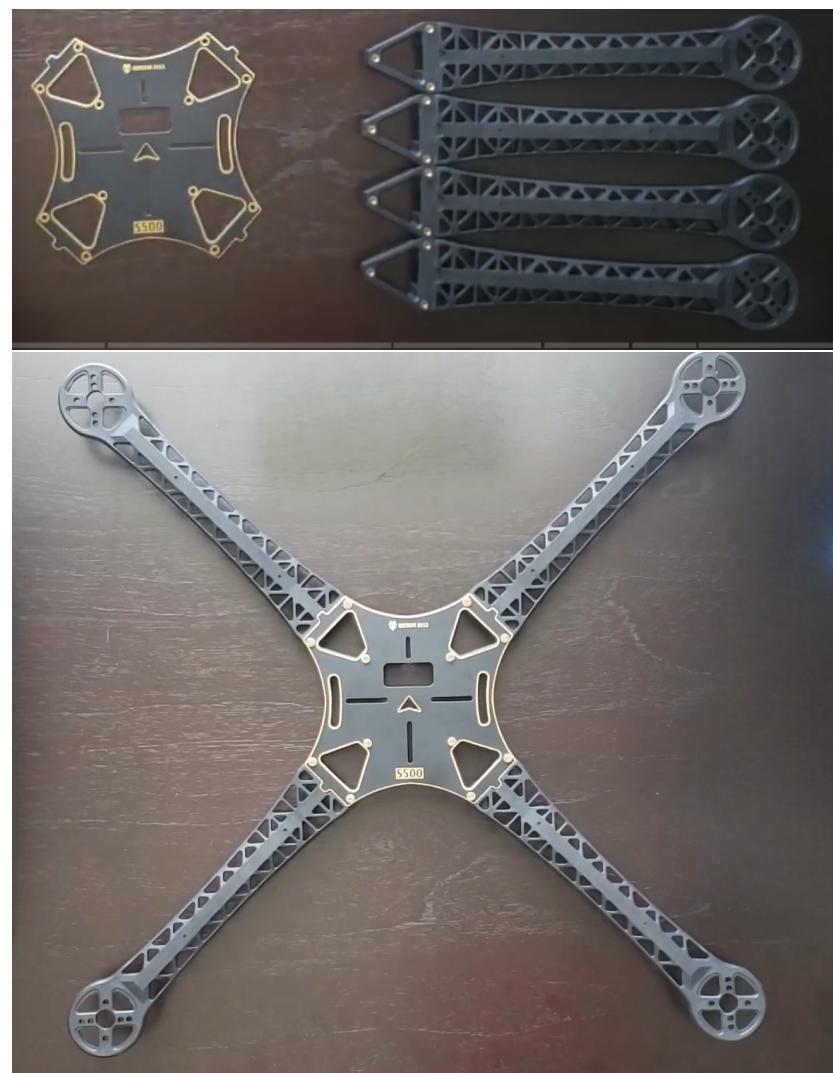


Figura 11.4: Base con 4 brazos para el armado de la placa superior.

Cuando la placa superior esté armada, se colocan los controladores de velocidad (ESC), los cuales debe tener en cuenta el sentido de giro de los motores y el diseño de las propelas como se muestra en la siguiente figura:

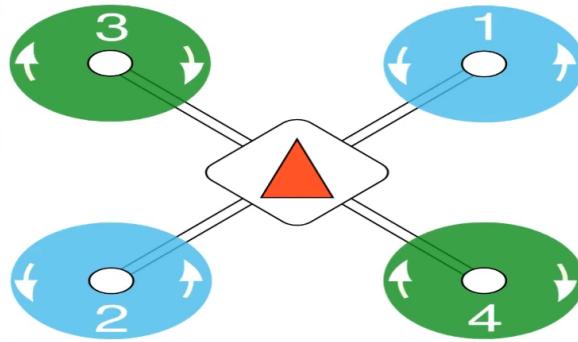


Figura 11.5: Sentido de giro de los motores.

Con esta consideración en los extremos se soldan los controladores de velocidad ESC con el tablero de distribución de energía (PDB) y el módulo de potencia.. El amperaje del ESC depende de la corriente que consumen los motores, siendo de 20 % más amperaje como mínimo (si el motor consume 10A, el ESC debe ser de 12A como mínimo).

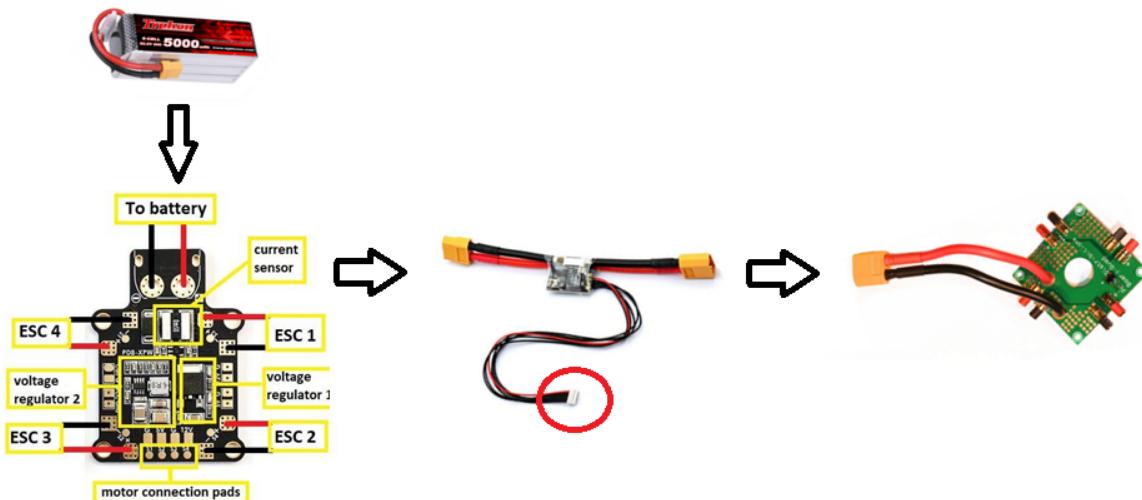


Figura 11.6: Componentes de la parte de potencia del dron.

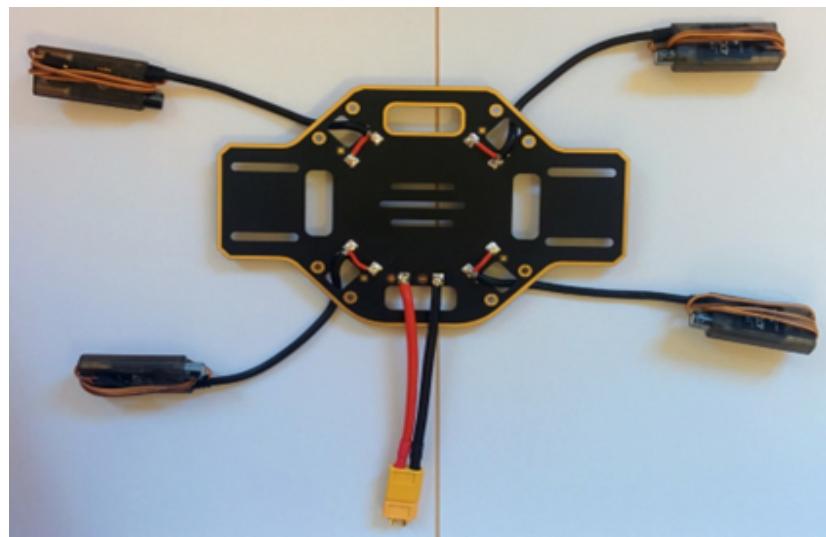


Figura 11.7: Placa superior con las ESC montadas y sin brazos.

Una vez teniendo el conjunto de las ESC y los módulos de distribución de potencia soldados, se procede a instalar los motores. Los motores utilizados son los motores sin escobillas: 2212 920KV que cuentan con las siguientes características.

- Eficiencia máxima: 80 %
- Eficiencia de la corriente máxima: 4-10A (> 75 %)
- Capacidad de corriente: 12A/60s
- Consumo de corriente sin carga a 10 V: 0.5A
- Eficiencia -MAX real: 4-10A (> 75 %)
- RPM/V: 930 KV
- Diámetro del eje: 3.17mm
- Dimensiones: 28 x 28 x 39 mm
- Peso: 67g



Figura 11.8: Placa superior con los motores montados.

Una vez ensambladas las componentes de ambas partes (la placa principal y la placa superior), únicamente hay que unir estos 2 conjuntos. Colocando la placa superior sobre la placa principal, utilizando los tornillos que ya vienen incluidos sobre el KIT S500. No hay que perder en cuenta las conexiones que se consideraron del sentido de rotación de los motores.

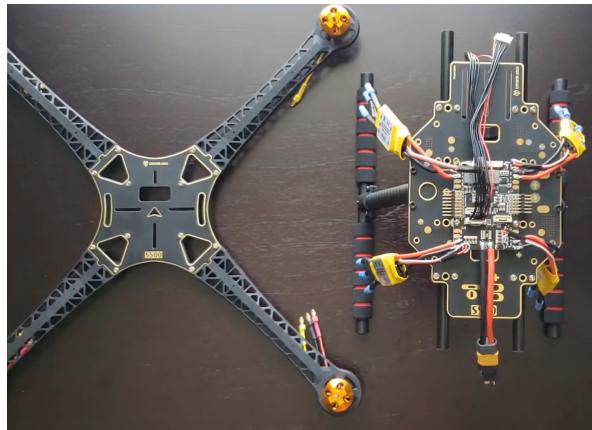


Figura 11.9: Montaje de la placa superior sobre la placa base (por partes).

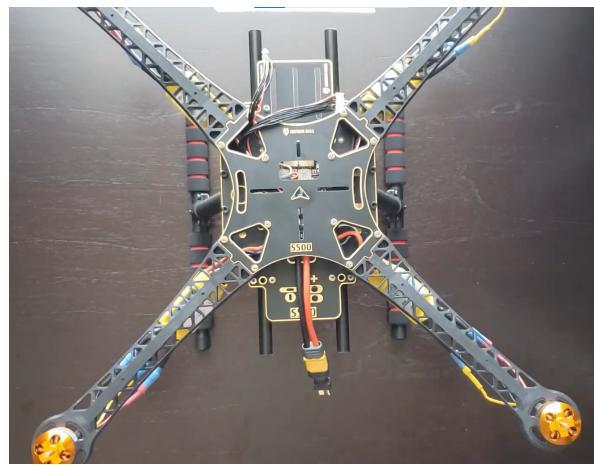


Figura 11.10: Montaje de la placa superior sobre la placa base (armada).

Para finalizar el ensamblado básico del dron, se añade la computadora de vuelo Pixhawk 2.4.8 y el módulo GPS para el posicionamiento del dron. Para el módulo GPS, debe estar separado mínimo 15 cm de la computadora de vuelo para no causar interferencia, por lo que se le coloca en un mástil que irá en la parte superior.

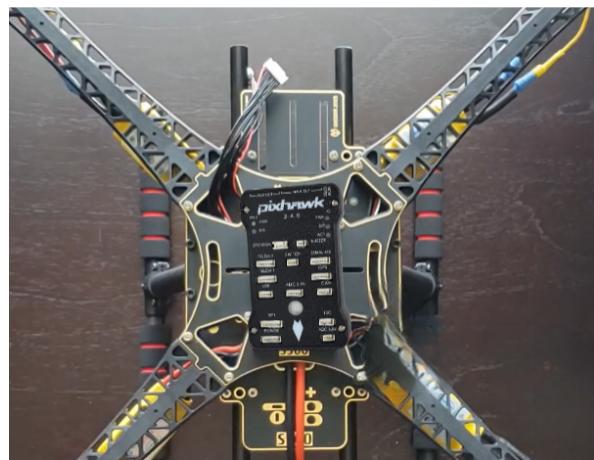
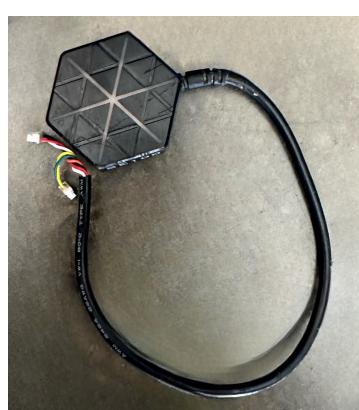


Figura 11.11: Montaje de la computadora de vuelo sobre la placa superior.



a) Individualmente.



b) Mástil.



c) Montado.

Figura 11.12: Montaje del GPS de vuelo sobre la placa superior.

La computadora de vuelo Pixhawk 2.4.8 cuenta con las siguientes características:

■ **Microporcesador**

- Frecuencia: 168 MHZ, 256K RAM



- o Coprocesador de copia de seguridad 32 STM32F103

■ Sensor

- o Giroscopio digital de 3 ejes L3GD20 16
- o Acelerómetro de 3 ejes LSM303D 14/Magnetómetro
- o Acelerómetro / magnetómetro MPU6000 de 6 ejes
- o Barómetro de precisión MS5607

■ Interfaz

- o UART 1, 2 compatible con alto voltaje con control de flujo de hardware
- o Entrada compatible con receptor de satélite Spektrum DSM / DSM2 / DSM-X
- o Entradas y salidas compatibles con Futaba SBUS
- o Entrada de señal PPM
- o Entrada RSSI (PWM o voltaje)
- o I2C
- o SPI
- o 3.3 y entrada 6.6VADC
- o Interfaz micro USB externa



Figura 11.13: Diagrama de conexiones de la computadora de vuelo Pixhawk 2.4.8



Se utilizan las conexiones de POWER, las 2 de telemetría, el módulo GPS, el buzzer, el safety switch y la comunicación I2C, como se muestra en la siguiente figura:

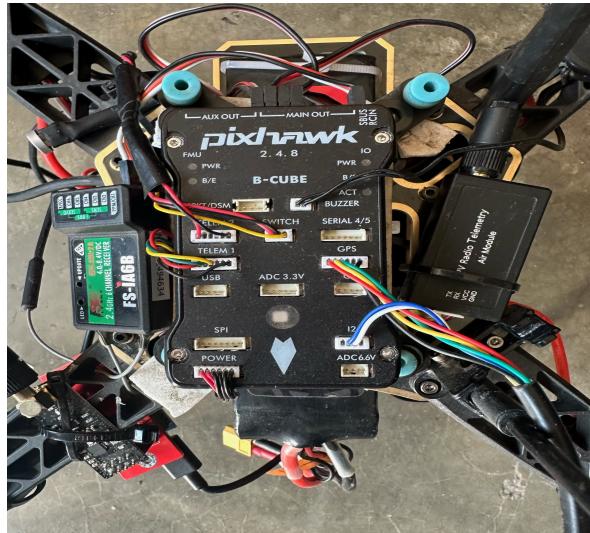
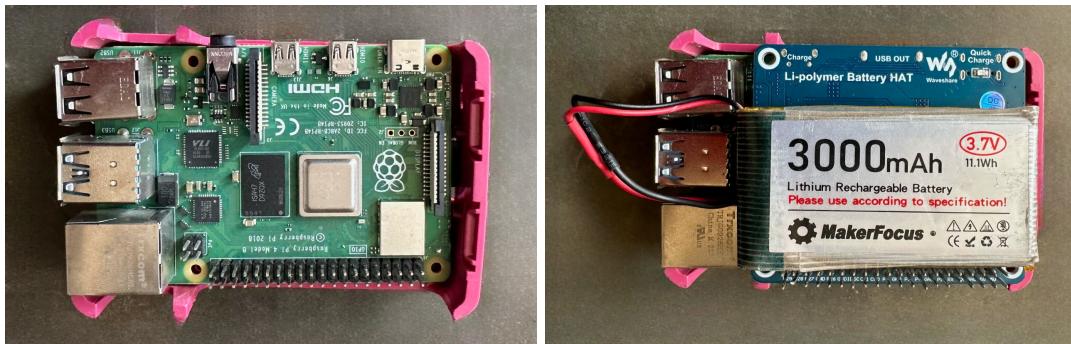


Figura 11.14: Conexiones sobre la computadora de vuelo.

Para la parte específica de control del dron, se le añade otra computadora, la Raspberry Pi4, ya que contiene las librerías para el protocolo MavLink, cuenta con los puertos suficientes para conectar dispositivos y tiene la capacidad de procesar señales digitales (DSP) en caso de ser necesario. Algunos de los usos de la Raspberry Pi 4 para este proyecto podrían incluir:

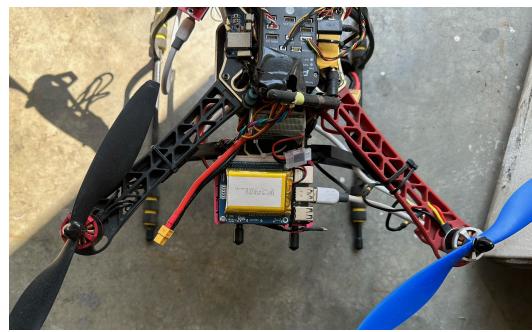
- **Control de vuelo:** Puede ejecutar algoritmos de control de vuelo y software de navegación para dirigir los drones (enjambre de drones) de manera autónoma o semi-autónoma.
- **Interfaz de usuario y monitoreo:** Se puede utilizar Python para desarrollar una interfaz de usuario en la Raspberry Pi 4 que permita supervisar y controlar los drones de manera intuitiva.

Esta computadora externa al ser montada en el dron, se alimenta independientemente de la batería principal utilizando una batería externa.



a) Individualmente.

b) Con batería de 3000mAh.



c) Montada con su batería.

Figura 11.15: Montaje de la Raspberry Pi4 sobre la placa superior.

Por otra parte, los sistemas de telemetría también son importantes, por lo que para ello se utilizará el módulo NRF24L01. Las características de la telemetría se muestran en la Figura 11.17: Este módulo se puede agregar fácilmente a cualquier sistema MCU/ARM/PIC/AVR/STM32. Además, está diseñado con un amplificador de potencia y antena SMA, lo que le permite utilizar la comunicación inalámbrica hasta a 800 metros (con línea de vista).

- **Rango:** 800m de LOS
- **Frecuencia:** 2.4 - 2.5GHz
- **Voltaje de operación:** 3 - 3.6V Max
- **Corriente máxima:** 115mA

Un microcontrolador de anfitrión puede comunicarse y configurar el NRF24L01 a través de

una interfaz periférica serial de 4 pines (SPI). Los registros de configuración son accesibles a través de la conexión SPI. Los parámetros configurables incluyen el canal de frecuencia (125 canales seleccionables), la potencia de salida y la velocidad de datos (velocidades de datos: 250kbps, 1Mbps y 2Mbps).

El regulador de voltaje en el chip acepta voltajes de suministro de 1.9 a 3.6V. El módulo tiene entradas tolerantes a 5V que permiten la conexión directa de los pines SPI.

El filtrado interno resulta en márgenes altos para cumplir con los estándares regulatorios de RF. La radio del módulo utiliza modulación de desplazamiento de frecuencia gaussiana (GFSK) así como control automático de ganancia rápida (AGC).

El módulo incluye un pin de solicitud de interrupción (IRQ) que se puede utilizar para despertar al microcontrolador principal del modo de suspensión cuando el módulo recibe una transmisión, lo que proporciona un gran ahorro de energía en dispositivos con batería.

Parámetros del producto



Voltaje de operación	3~3.6V
Potencia de salida	+20dBm
Corriente de trabajo en modo de transmisión (valor pico)	115mA
Corriente de trabajo en modo de recepción (valor pico)	45mA
Corriente en modo de apagado	4.2uA
Temperatura de trabajo	-20~70°C
Agilidad de aceptación en modo 2Mbps	-92dBm
Agilidad de aceptación en modo 1Mbps	-95dBm
Agilidad de aceptación en modo 250kbps	-104dBm
Ganancia del PA	20dB
Ganancia del LNA	10dB

Figura 11.16: Características de la telemetría NRF24L01.

Para cada acción controlable a distancia, se necesita un canal único para transmitir la entrada. El mínimo necesario para pilotar un multirrotor es de cuatro canales: throttle (aceleración), yaw (rotación), pitch (timón) y roll (alerón). Para cada interruptor de modo de vuelo, control



de cardán o control de iluminación, se necesita un canal adicional.

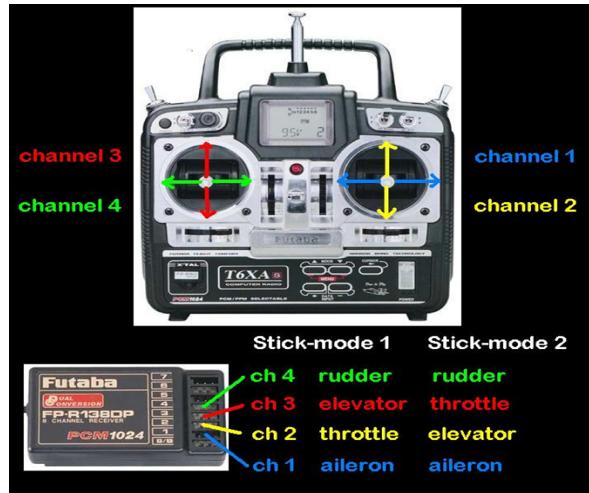


Figura 11.17: Canalización de Control en Drones Multirrotores.

El Xiangtat Flysky FS-i6X Transmisor con receptor iA6B ayudará a tener un control manual de vuelo para poder controlar el dron desde tierra. El transmisor FS-i6X y el receptor FS-iA6B constituyen un sistema RC computarizado proporcional digital AFHDS 2A de 6 canales. Este sistema ofrece una protección superior contra interferencias mientras mantiene un menor consumo de energía y una alta sensibilidad fiable del receptor. Está especialmente desarrollado para todos los modelos de control de radio. [?]

Especificaciones del transmisor FS-i6X:

- Canales: 6-10 (por defecto 6)
- Tipo de modelo: Ala fija/planeador/cuadricóptero
- Rango de RF: 2.408 - 2.475 GHz
- Potencia RF: <20dBm
- Canal RF: 135
- Ancho de banda: 500kHz.
- Sistema de 2.4GHz: AFHDS 2A / AFHDS.
- Tipo de modulación: GFSK



- Resolución del barril: 4096
- Advertencia de baja tensión: <4.2V
- Puerto DSC: puerto PS/2
- Longitud de la antena: 26mm (antena dual)
- Peso: alrededor de 400g
- Visualización: visualización transreflectiva STN, red LCD de 128 x 64, VA 73 x 39 mm, LCD con retroiluminación blanca
- Tamaño: 190 x 174 x 89 mm
- Actualización en línea: sí
- Certificado: CE0678, FCC ID: N4ZFLYSKYI6X

Especificaciones del receptor FS-iA6B:

- Canales: 6
- Rango de RF: 2.4055 - 2.475GHz
- Canal RF: 140
- Sensibilidad del receptor RF: -105dBm
- Ancho de banda: 500kHz
- Sistema de 2.4GHz: AFHDS 2A
- Tipo de modulación: GFSK
- Potencia: 4.0 - 6.5V
- Transmisor compatible: FS-i6X, FS-i4, FS-i6, FS-i10, FS-GT2E, FS-GT2G
- Longitud de la antena: 26mm (antena dual)
- Peso: 16.3g
- Tamaño: 47 x 26.2 x 15 mm
- Puerto i-bus: sí
- Puerto de adquisición de datos: sí



Figura 11.18: Transmisor Radio Controlado.

En cuanto a la batería del dron, esta se coloca ajustadamente entre la placa inferior y la placa superior para evitar su deslizamiento. Tomando en cuenta que para no reducir su tiempo de vida, no se debe permitir que sus celdas se descarguen menos del 80 % (para una batería de 12V no se debe descargar a menos de 9.6V). Su capacidad y corriente por hora dependen del consumo energético del UAV:

- Entre más grande más consumo
- Entre más propelas más consumo
- Entre más dispositivos conectados, más consumo
- Entre más velocidad y movimiento, más consumo

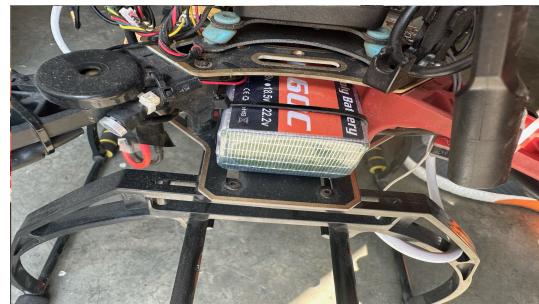
Tabla 11.1: Relación de la capacidad de la batería con el tamaño de las hélices.

Tamaño de las hélices	Capacidad de la batería
6"	1500 mAh a 2200 mAh
5"	1300 mAh a 1800 mAh
4"	850 mAh a 1300 mAh
3"	650 mAh a 1000 mAh

El tiempo de vuelo está relacionado al consumo energético de la batería y se puede calcular a partir de la corriente que consume el dron, la batería, el peso, la mínima descarga del dron, etc. Existen modelos matemáticos (e.g. Backstepping) y simuladores (e.g. ROS Gazebo) que pueden predecir el consumo energético de un UAV [?].



a) Individualmente.



b) Montada.

Figura 11.19: Montaje de la batería 5000mAh 6 celdas 22.2v 25C-60C RC sobre la placa inferior.

Ya por último, podemos ver en la Figura 11.20 el dron completamente armado con todos los componentes ensamblados y montados sobre la estructura y las placas del dron.

Figura 11.20: Dron listo para pruebas de vuelo.

Estos mismos procesos se hacen para el ensamblado de los otros dos prototipos, más adelante



se seguirán haciendo pruebas y documentación sobre las demás actividades para modelar el consumo energético y los esquemas de vuelo para la recolección de datos.



Figura 11.21: Transmisores de Radio Control para cada dron .



Figura 11.22: Drones listos para pruebas de vuelo.

Por último, se describe el proceso de vuelo coordinado del enjambre de drones en el diagrama de flujo de la Figura 11.23. Comenzando con la inicialización de los dispositivos involucrados, donde, se encienden los tres drones, que consisten en el dron líder y los dos drones seguidores. Simultáneamente, la estación base se conecta a un monitor serial, lo que permite la comunicación con el dron líder desde la estación base.



Una vez encendidos los drones, cada uno de ellos obtiene sus coordenadas GPS. Los drones seguidores envían sus coordenadas al dron líder, quien recopila y envía estas coordenadas junto con las propias a la estación base. Este ciclo de envío de coordenadas se repite continuamente hasta que todas se reciban en la estación base y se muestren en el monitor serial.

La estación base verifica entonces si ha recibido todas las coordenadas necesarias. No se puede continuar el proceso si no se han recibido las coordenadas de todos los drones en la estación base.

Una vez que la estación base ha recibido todas las coordenadas, se envía un comando de inicio al dron líder. Este comando indica al dron líder que debe prepararse para iniciar el vuelo coordinado. Si el comando ha sido recibido correctamente, el dron líder envía un comando de despegue a los drones seguidores para que todos los drones despeguen simultáneamente.

Una vez en vuelo, cada dron sigue su ruta programada. Este seguimiento de ruta se mantiene mientras el vuelo esté en progreso, asegurando que los drones se desplacen de acuerdo con sus planes de vuelo configurados. Finalmente, el proceso concluye cuando los drones regresan a la estación base, finalizando el procedimiento de vuelo en enjambre o coordinado.

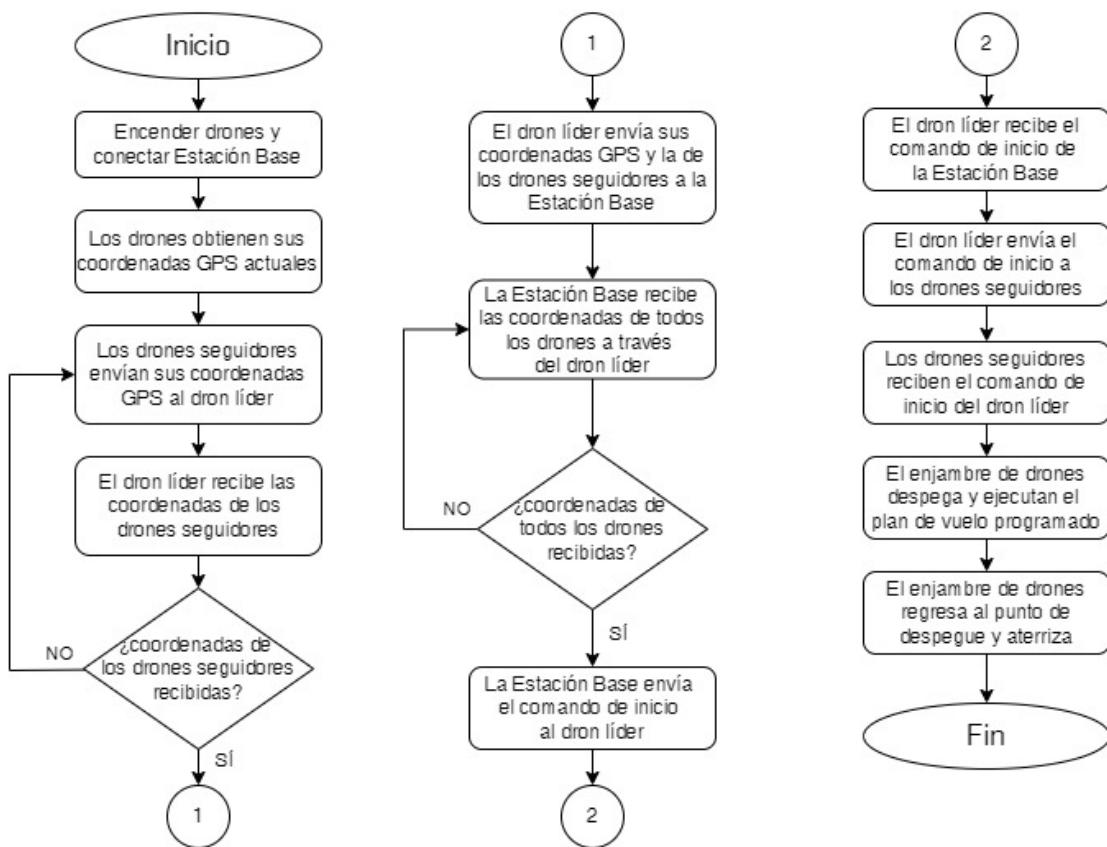


Figura 11.23: Diagrama de flujo para el vuelo en enjambre.

11.1.1. Configuración de los Drones.

Para configurar los drones se hacen una serie de pasos para calibrar y corroborar que los componentes funcionen. Para comenzar, conectaremos nuestra computadora de vuelo directamente a nuestra computadora o por telemetría.



Figura 11.24: Conexión de la Computadora devuelvo a la computadora por cable.

11.1.1.1. Calibración de la brújula. (Compass Calibration)

Dentro de las configuraciones más importantes y que siempre hacemos antes de empezar un vuelo, calibramos la brújula. Una vez conectada nuestra telemetría a nuestra computadora, abriremos el software “Mission Planner” para configurar la brújula siguiendo los siguientes pasos:

En la pestaña *Setup* → *Mandatory Hardware* → *Compass*, nos aparecerá la siguiente ventana.

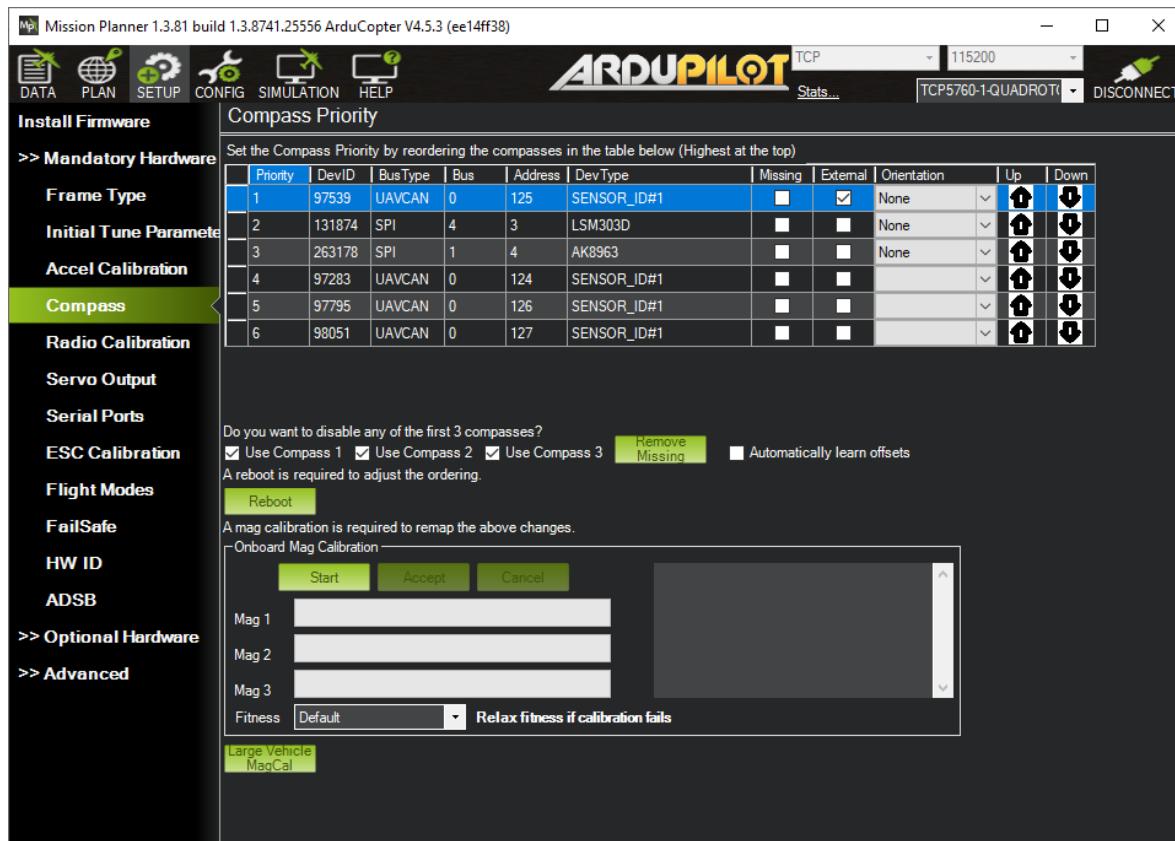


Figura 11.25: Calibración de la brújula

Observaremos que contamos con barrios GPS en nuestra ventana, es importante tener en cuenta que le debemos dar prioridad a nuestro GPS externo que lo marcaremos con la casilla de “External”, nuestra computadora de vuelo también incluye un GPS, así que ese dispositivo también se calibrará con estos pasos. Como nota, el Dron debe no estar armado y no se debe calibrar cerca de algún dispositivo electrónico ni metálico, ya que puede afectar la forma de calibración del GPS.

Continuando con la calibración, al dar click en el botón Start dentro de la sección de “On board Mag Calibration” y se debe mover el Dron en los 360°, dando vueltas, moviéndolo de izquierda a derecha, poniéndolo de cabeza y en todas las direcciones posibles. A medida que se gira el vehículo, las barras verdes deben extenderse cada vez más hacia la derecha hasta que



se complete la calibración. Una vez finalizado con éxito, se emitirán tres tonos ascendentes y aparecerá una ventana de Reinicie el piloto automáticoz deberá reiniciar el auto Pilot antes de que sea posible armar el vehículo

11.1.1.2. Calibración del acelerómetro (Accel Calibration).

Los acelerómetros del Dron deben calibrarse para corregir sus desplazamientos de polarización en los tres ejes (Pitch, Yaw y Roll). La calibración del acelerómetro es obligatoria en ArduPilot y la calibración no se puede realizar mientras el vehículo está armado. Nos dirigimos a la calibración del acelerómetro siguiendo las siguientes rutas:

En la pestaña *Setup* → *Mandatory Hardware* → *Accel Calibration*, nos aparecerá la siguiente ventana.

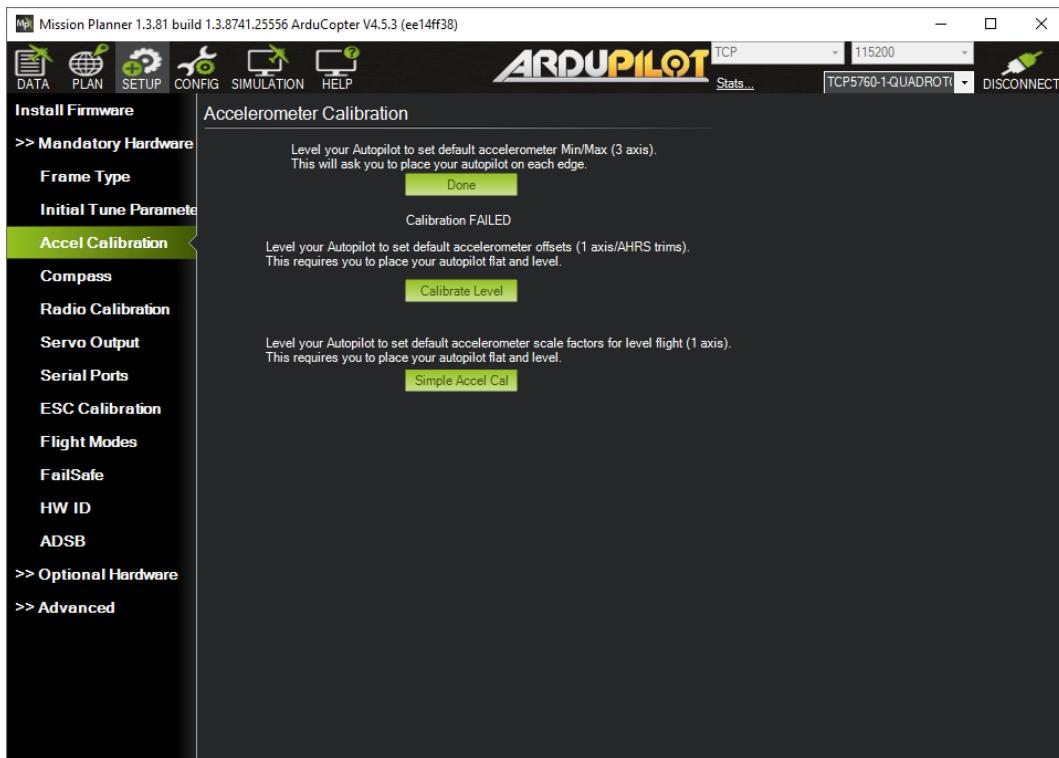


Figura 11.26: Ventana de Calibración del acelerómetro.



Deberemos hacer click en la primera opción, seguido de eso, el propio software, nos indicará las posiciones en las que debemos colocar el Dron, las posiciones son:

- Level. (*En el nivel del suelo*).

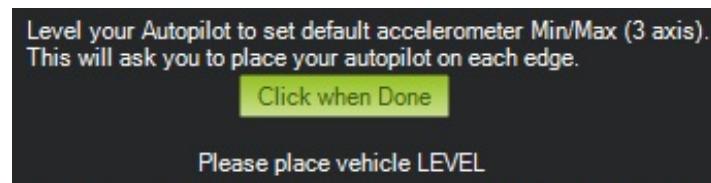


Figura 11.27

- Left. (*Izquierda*).

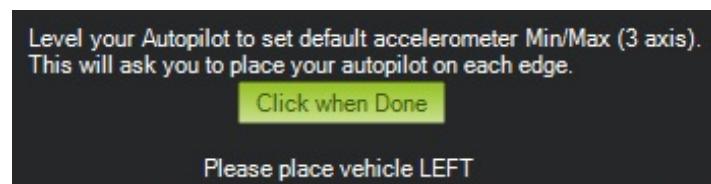


Figura 11.28

- Right. (*Derecha*).

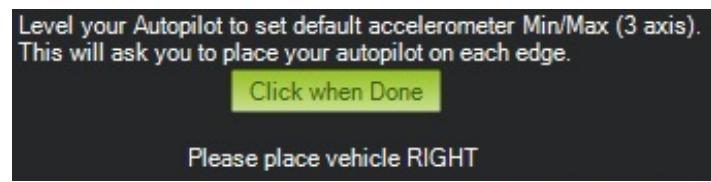


Figura 11.29

- Nose Down. (*Con la parte frontal del Dron mirando hacia abajo*).

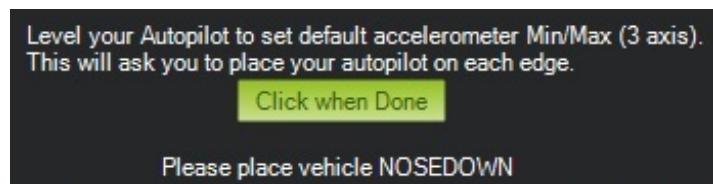


Figura 11.30

- Nose Up. (*Con la parte frontal del Dron mirando hacia arriba*).

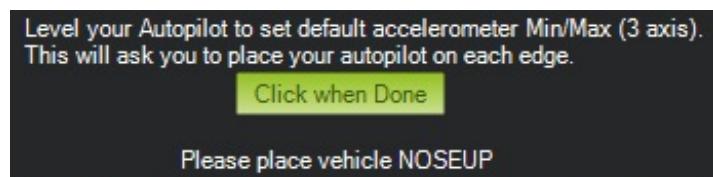


Figura 11.31

- Back. (*Con la parte superior del Dron mirando hacia abajo*).

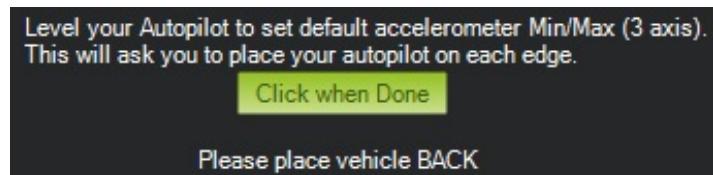


Figura 11.32

El Dron debe mantenerse quieto inmediatamente después de presionar la tecla para cada posición. Esto es más importante que conseguir el ángulo exacto, es decir a la izquierda a 90 grados a horizontal exactamente, etc. A excepción del primer "NIVEL", las posiciones pueden estar dentro de los 20 grados (por inclinación del suelo) de ser exactas. Se debe estar quieto en cada posición mientras presionamos la tecla de aceptar. La posición de nivel es la más importante de todas para la calibración, ya que esta será la posición que el Dron considere nivelada mientras vuela.



11.1.1.3. Configuración del Control.

Los controles transmisores nos permiten configurar el modo de vuelo, controlar el movimiento y la orientación del vehículo y también activar y desactivar las funciones auxiliares (es decir, subir y bajar el tren de aterrizaje, etc.) de las cuales hablaremos más adelante.

La calibración del control implica la captura de los valores mínimos, máximos y de "ajuste" de cada canal para que ArduPilot pueda interpretar correctamente la entrada.

Antes de comenzar, hay que asegurarnos de que el control tenga una batería con carga, al momento de encender el control verificar que esté en "Auto" y que las lengüetas del control se encuentren totalmente levantadas..

Para abrir el menú de configuración, nos dirigiremos a: *Setup* → *Mandatory Hardware* → *Radio Calibration*, nos aparecerá la siguiente ventana.

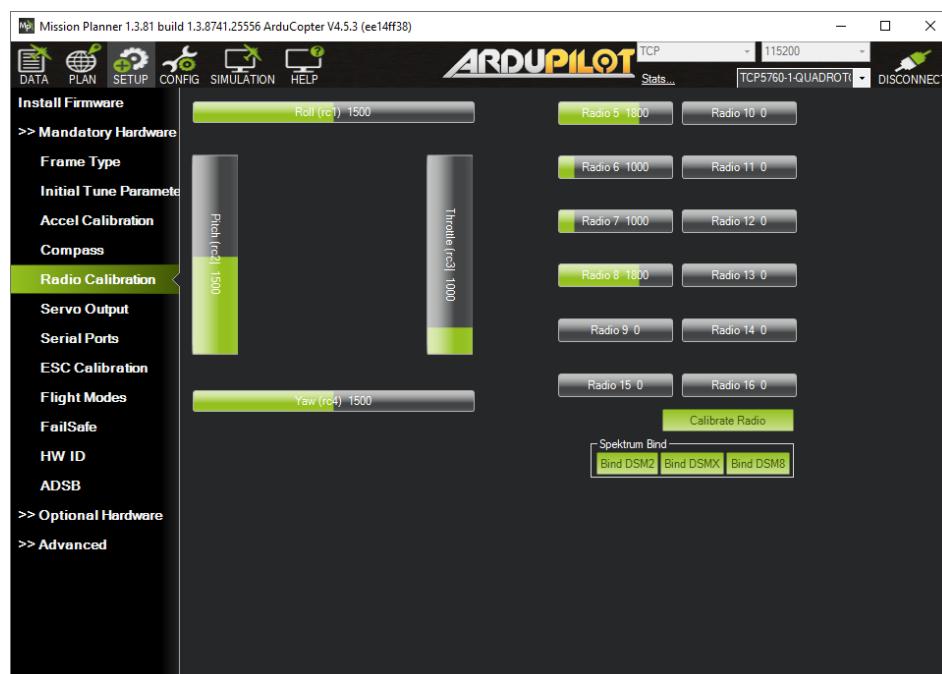


Figura 11.33: Menú de calibración del Radio Control

Al colocarnos en esta pestaña, debemos ver que hay barras verdes, si no se encuentran



esas barras verdes, hay que verificar el Radio Control. En esta ventana comprobaremos el mapeo de los canales de transmisión, también las perillas, palancas e interruptores de nuestro control, lo comprobaremos al presionar los botones y mover las palancas, las barras verdes se empezarán a desplazar. Para la calibración del control, pulsaremos el botón de “Calibrate Radio”, presionamos el botón de “Ok” cuando se solicite verificar que el equipo de control del radio esté encendido y las hélices no están conectadas.



Figura 11.34: Calibración del Radio Control.

Hacer click cuando hayamos terminado. Acabando de configurar, Mission Planner nos mostrará un resumen de los datos calibrados, los valores comunes rondan entre los 1100 para los mínimos y 1900 para los valores máximos.

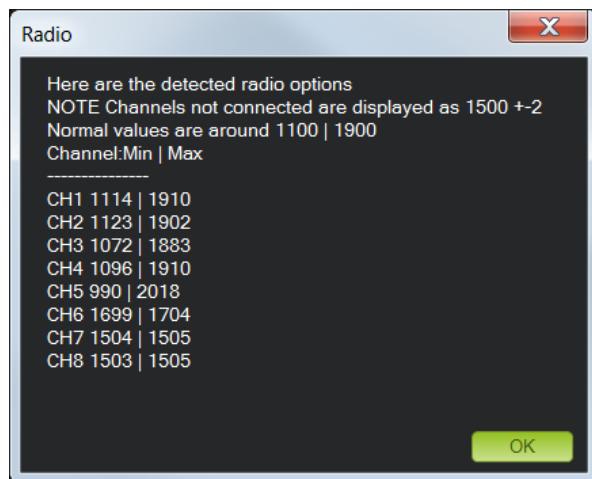


Figura 11.35: Valores Rango de los controles.

11.1.1.4. Verificación del giro de los motores.

Es importante verificar la dirección de los motores, porque en caso de que algún motor y propela estén en el sentido contrario, el Dron será incapaz de volar y puede provocar algún tipo de accidente.

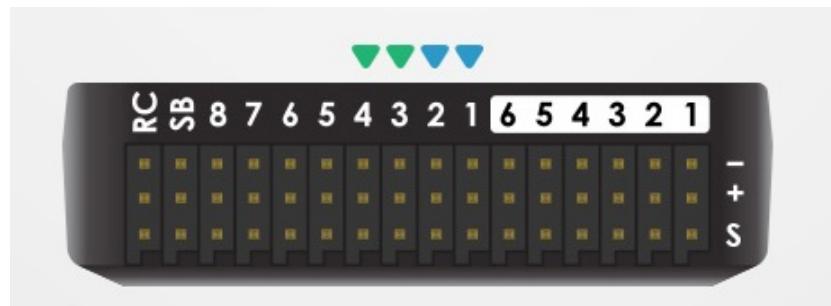


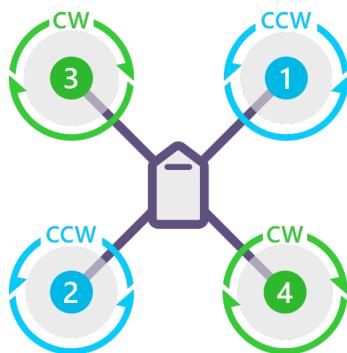
Figura 11.36: Pines de salida Pixhawk (numerados).

Conectamos los cables de alimentación (+), tierra (-) y señal (s) de cada ESC a los pines de salida principales del Dron por número de motor.

Para dirigirnos a la pestaña de: *Option→ Optional Hardware→Motor test*.



Figura 11.37: Pruebas de Motor.



QUAD X

Figura 11.38: Sentido y dirección de los motores con la punta como nuestro norte de nuestro GPS.



Figura 11.39: Leyenda para diagramas de orden de motores.

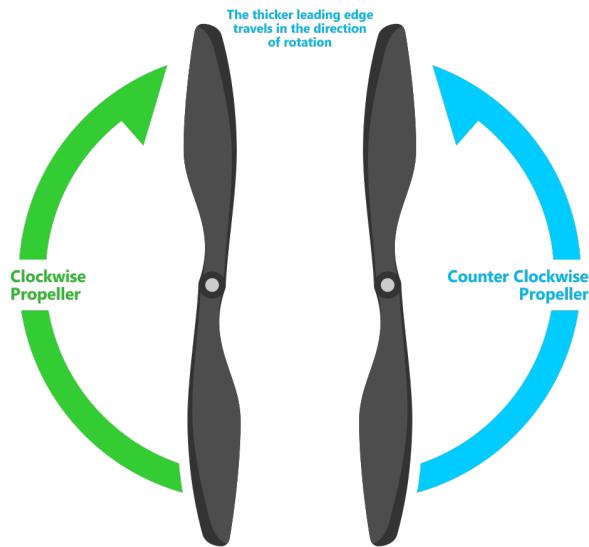


Figura 11.40: Sentido de las propelas en sentido horario y antihorario

La forma de comprobar que todo esté correcto es dirigirnos la pestaña de pruebas de motores y probar los 4 motores, sabremos que está bien el sentido porque el aire de impulso se sentirá en la parte de abajo del Dron, en caso de que el aire se sienta en la parte de arriba, habrá que corregir el sentido. Puede que al momento de probar el motor se pueda “atascar” o no poder girar, para solucionar el problema es de ayuda subirle el porcentaje de potencia en la misma ventana (*Throttle*).

11.1.1.5. Calibración de las controladoras de vuelo (ESC Calibration)

Las controladoras de vuelo son las encargadas de hacer girar a los motores a la velocidad solicitada para el Dron, la mayoría de las ESC necesitan ser calibradas para que reconozcan



los valores mínimos y máximos de los PWM.

Para la correcta calibración, se debe quitar las propelas del Dron, a continuación, pulsamos el botón de “Calibrate ESCs”, después, y colocar el control en su punto máximo de Pitch y el Dron debe estar con la batería conectada.



Figura 11.41: Representación del control con el Throttle al máximo.

Después de presionar el botón que da comienzo a la calibración, desconectamos la batería y volveremos a conectar inmediatamente hasta escuchar los pitidos de confirmación de calibración, cuando escuchemos los pitidos de confirmación, bajaremos la palanca y se procede a la captura del punto mínimo.

Se escucharán los pitidos de confirmación de la calibración. Al terminar, se escuchará un pitido largo. Para salir del modo de calibración de la controladora de vuelo, únicamente se desconecta la batería y se vuelve a conectar, esto hará que se salga del modo de configuración.



Figura 11.42: Representación del control con el Throttle al mínimo.

11.1.2. Modos de vuelo.

Los modos de vuelo programados en el Radio Control son los que utilizamos en las pruebas de vuelo. Como comentario inicial, la palanca del Throttle (Palanca izquierda del control) siempre debe coincidir que esté completamente hacia abajo, la otra palanca se mantiene en su lugar.



Figura 11.43: Inicio del primer modo de vuelo, modo “Stabilize”.

En el primer modo de vuelo, lo llamaremos “Stabilize”, es el modo con las lengüetas (A, B, C y D) están totalmente hacia arriba, es el modo en que podemos controlar al Dron de una forma completamente manual y la primera parte para empezar el vuelo automático. El modo de vuelo en automático se establece cuando la lengüeta B se baja en su totalidad.



Figura 11.44: Modo de vuelo “Auto”.



Para empezar el modo de vuelo automático, se sigue el siguiente procedimiento:

1. Primero, subimos el plan de vuelo al Dron.
2. Poner el control en modo Stabilize.
3. Se arma el Dron, este paso movemos la palanca del Throttle totalmente hacia abajo y la movemos a la derecha, es importante notar que las calibraciones y el botón de seguridad del Dron debe estar desactivado.



Figura 11.45: Armado del Dron con la palanca del Throttle.

4. Una vez armado el Dron, se pone en modo Auto y se levanta ligeramente la palanca del Throttle y el plan de vuelo comienza a ejecutarse.

Por último, contamos con el modo de vuelo de “*Land*”, este modo es únicamente para el aterrizaje del Dron. Para colocar el Dron en este modo, pasamos del modo Auto al modo Land, bajamos la lengüeta B y seguido subimos la lengüeta A.



Figura 11.46: Modo de vuelo “Land”.

El Dron empezará a aterrizar en la posición en que se encuentra. Para las pruebas que hicimos, se presentaban problemas de lectura con el GPS externo del Dron, por lo que perdía su posición y en muchos casos el Dron hacía un aterrizaje forzado y algunos componentes presentaban descomposturas, por eso, al momento de hacer las pruebas de vuelo, el piloto debe estar atento a los movimientos del Dron en caso de alguna pérdida de lectura del GPS o cambios de velocidad del viento bruscos.

Para controlar que el Dron tenga un buen aterrizaje en caso de alguna falla sin causar daños, tenemos dos opciones de aterrizaje.

1. **Opción de aterrizaje por modo Land:** Es mover el modo de vuelo de *Stabilize* → *Land* o de *Auto* → *Land*, ajustando la palanca del movimiento (Palanca de la Derecha del Control) para evitar la colisión con algún objeto.
2. **Opción de vuelo Manual:** Esta opción es poco recomendada, la opción aplica del modo *Land* → *Stabilize* o del modo *Auto* → *Stabilize*, el piloto toma el control total del Dron, lo que implica tener un excelente control con las palancas de Throttle y de Movimiento, ya que si la palanca de Throttle está totalmente hacia abajo, los motores se apagan y



el Dron caerá, por lo que se tiene que tener en una posición adecuada la palanca del Throttle para hacer un descenso lento y controlado.

En cualquier caso, el piloto debe contar con experiencia con el control manual y el control automático del Dron.

11.1.3. Planes de Vuelo.

Los planes de vuelo se personalizan desde la pestaña en Plan.

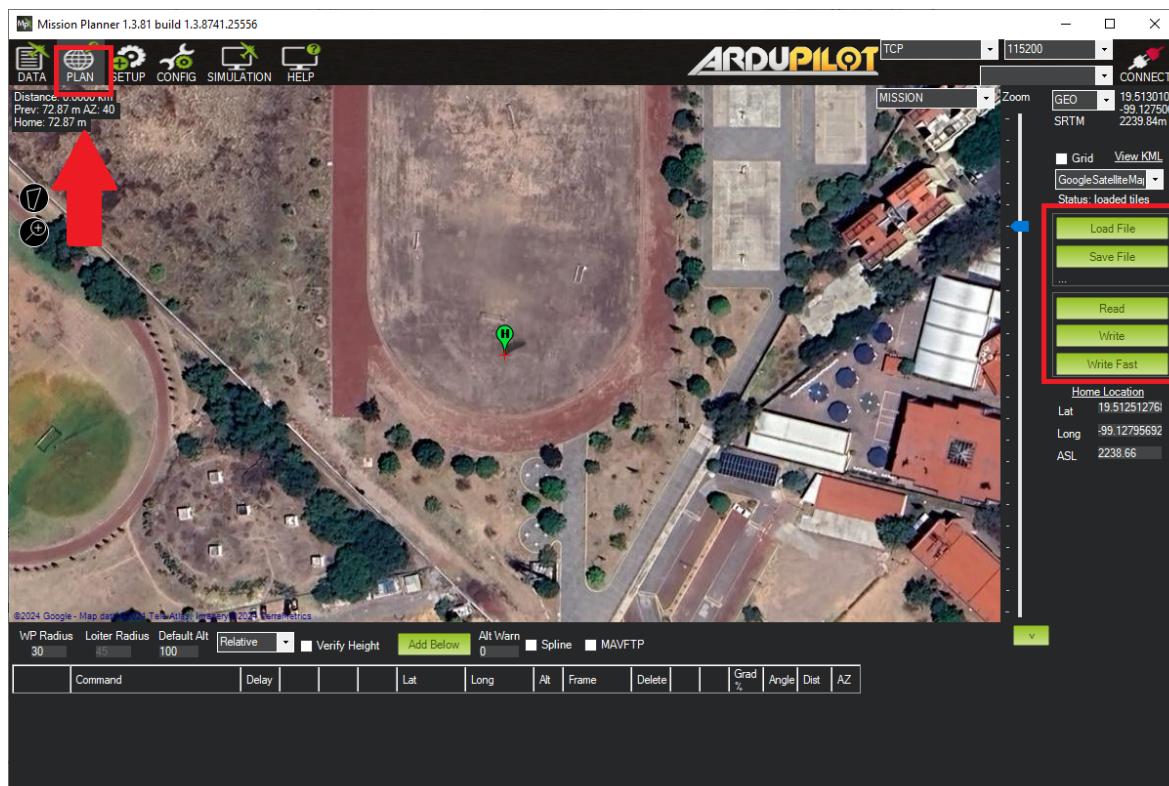


Figura 11.47: Pestaña de creación de los planes de vuelo.

En la derecha del panel encontramos las opciones de guardar y cargar los planes de vuelo, así como de leer el plan que tiene el Dron y mandarlo a la Memoria del Dron.

Las órdenes que más comúnmente usamos son las siguientes:



- Land: Para aterrizaje.
- Takeoff: Para despegue del Dron, contiene el parámetro de altura.
- Waypoint: El waypoint es un punto al cual queremos que se dirija el Dron, contiene un parámetro para el tiempo de espera dentro del waypoint y la altitud que debe tomar el Dron llegado al punto.
- Conditional Yaw: Esta instrucción hace que el Dron gire y se profile con respecto al norte geográfico, esta opción la colocamos siempre después de la instrucción de takeoff y waypoints para mantener el Dron apuntando hacia el norte.

Hay más acciones que se pueden tomar para los planes de vuelo para poder hacer más experimentos y observar su comportamiento. Una vez creado el plan de vuelo, el Dron debe estar conectado a través de la telemetría al software de Mission Planner, pulsamos el botón de Write, y el plan estará listo para ejecutarse.

11.1.4. Condiciones climatológicas.

Para tener en cuenta un plan de vuelo, se deben tener las siguientes consideraciones climáticas:

- Viento: Se recomienda según blogs [3] tener una velocidad de entre 10 mph a 15 mph, lo que es equivalente de 16 km/h a 24 km/h.
- Precipitación: Debemos planear el vuelo con nula precipitación o una probabilidad menor o igual al 40 %, según recomendaciones de la aplicación “*UAV Forecast*”.
- Cantidad de satélites disponibles: Como recomendación por la misma aplicación *UAV Forecast*, se recomiendan un mínimo de 8 satélites GPS, incluyendo los satélites GNSS.

Una vez comprobadas las condiciones climáticas apropiadas para el vuelo, se puede proceder a crear el plan de vuelo y a ejecutarlo.

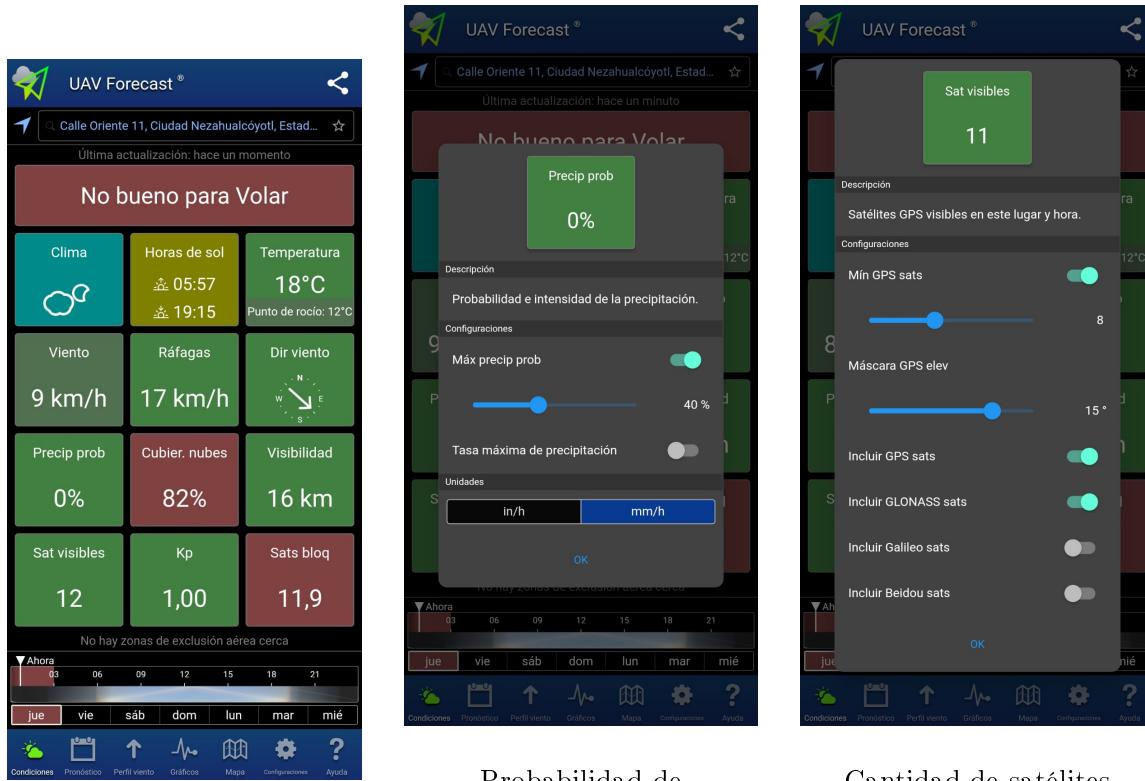


Figura 11.48: Aplicación de ayuda para condiciones de vuelo



11.2. Implementación de la WSN

11.2.1. Adquisición de los Elementos

En esta etapa, se procederá con la adquisición de los elementos esenciales para la implementación de la WSN, basándonos en las selecciones detalladas durante el análisis y diseño del proyecto:

1. **Termistor LM35 (temperatura):** Utilizado para monitorear la temperatura ambiental y su análisis podría indicar algún impacto en el comportamiento de las especies.
2. **Módulo GPS Neo-6M (geolocalización):** Necesario para rastrear y registrar la ubicación geográfica precisa de las especies, proporcionando información vital sobre la distribución espacial y migración de las especies.
3. **Cámara Trampa PR-700 (video):** Esencial para obtener imágenes de alta resolución que facilitarán la identificación de especies y el análisis de su actividad y comportamientos.
4. **Arduino Pro Micro (microcontrolador):** Elegido por su bajo consumo energético, capacidad suficiente para tareas específicas del proyecto y tamaño compacto.
5. **Módulos LoRa UART 868/915 MHz (transmisión de datos):** Seleccionados por su capacidad única para proporcionar conectividad confiable a larga distancia con bajo consumo de energía.
6. **Baterías LiPo recargables (fuente de energía):** Seleccionadas por su buen funcionamiento y larga duración, ofreciendo una solución confiable para dispositivos de monitoreo autónomos.
7. **Módulo MicroSD Card Adapter MLMSD (adaptador):** Elegido por su compatibilidad con tarjetas MicroSD, proporcionando capacidades de almacenamiento significativas y fácil integración con microcontroladores.



8. **Memorias MicroSD (almacenaje de datos):** Utilizadas para el almacenamiento de datos generados por los sensores y archivos de video de las cámaras trampa.

A continuación, se presenta la tabla comparativa 11.2 que resume los precios y proveedores de cada elemento necesario, con su respectivo hipervínculo:

Tabla 11.2: Comparación de precios de componentes electrónicos

Elemento	Amazon	Mercado Libre	AliExpress
Arduino Pro Micro	\$243.00 [Amazon]	\$175.50 [Mercado Libre]	\$65.09 [AliExpress]
Termistor LM35	\$40.00 [Amazon]	\$63.50 [Mercado Libre]	\$25.86 [AliExpress]
Módulo GPS Neo-6M	\$119.00 [Amazon]	\$149 [Mercado Libre]	\$72.62 [AliExpress]
Cámara Trampa PR-700	\$1,309.64 [Amazon]	\$770.65 [Mercado Libre]	\$234.37 [AliExpress]
Módulos LoRa	\$353.50 [Amazon]	-----	\$59.09 [AliExpress]
Módulo adaptador MicroSD	\$22.99 [Amazon]	\$43.60 [Mercado Libre]	\$6.17 [AliExpress]
Memorias MicroSD	\$54.50 [Amazon]	\$72.00 [Mercado Libre]	\$63.54 [AliExpress]
Batería (1200mAh 3.7v 103040)	\$110.02 [Amazon]	\$110.02 [Mercado Libre]	\$71.60 [AliExpress]

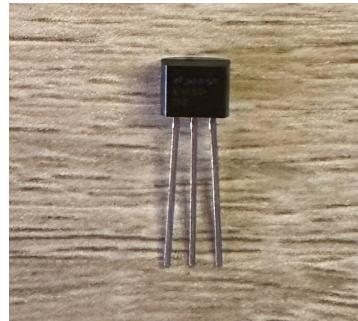
Nota: Los precios indicados son en la moneda mexicana, los cuales junto con la disponibilidad de los productos pueden variar con el tiempo.

A continuación, se muestran las imágenes de los elementos adquiridos para la implementación de la WSN. Estas imágenes proporcionarán una visualización clara de los componentes seleccionados y facilitarán la identificación y verificación de los elementos.

☞☞



a) Arduino Pro Micro.



b) Termistor LM35.



c) GPS Neo-6M.

d) Transceptor LoRa.



e) Batería Li-Po de 3.7V.
f) Cámara PR-700.

Figura 11.49: Elementos adquiridos para la WSN.

11.2.2. Funcionamiento de los Elementos

Una vez adquiridos todos los elementos necesarios para la implementación de la WSN, se verificará su funcionamiento. Esta etapa garantizará que cada componente funcione correctamente y esté listo para su integración en el sistema. A continuación, se describen los pasos para verificar el funcionamiento de cada elemento:



- **Arduino Pro Micro:**

Se conecta el Arduino Pro Micro a un puerto USB disponible en el ordenador mediante un cable USB adecuado y se verifica que el LED de alimentación del Arduino Pro Micro se encienda, lo que indica que está recibiendo energía correctamente.

Se carga un programa de prueba simple en el Arduino Pro Micro utilizando el entorno de desarrollo Arduino IDE. Este programa puede ser tan básico como hacer parpadear el LED interno del Arduino cada segundo.

Se observa el comportamiento del Arduino Pro Micro para confirmar que responde correctamente al programa cargado. Se espera que el LED conectado parpadee según lo programado, lo que indica que el Arduino está funcionando correctamente.



Figura 11.50: Prueba del led interno de Arduino Pro Micro.

- **Termistor LM35:**

El termistor LM35 se conecta a la placa Arduino según las especificaciones del fabricante. Se conecta el pin de salida del termistor al pin analógico de lectura de la placa Arduino (por ejemplo, A0), el pin VCC del termistor al pin de 5V de la placa Arduino, y el pin GND del termistor al pin de tierra (GND) de la placa Arduino.

Para probar su funcionamiento, se desarrolla un programa en el entorno de desarrollo Arduino IDE para leer la temperatura ambiente mediante el termistor LM35. Este programa realiza la lectura del valor analógico del pin conectado al termistor y lo convierte



a temperatura en grados Celsius utilizando la fórmula proporcionada por el datasheet del LM35, la cual es: $T[^\circ\text{C}] = \left(\frac{\text{valorSensor} \times 5}{1023}\right) \times 100$.

Una vez cargado el programa en la placa Arduino, se puede verificar la correcta lectura de la temperatura ambiente. Si se coloca el termistor en un entorno con una temperatura conocida y se observa la salida del monitor serial en el Arduino IDE. Se pueden comparar las lecturas obtenidas con la temperatura ambiente medida independientemente para verificar la precisión del termistor en la medición de la temperatura.

The screenshot shows the Arduino Serial Monitor window. The title bar says "Output" and "Serial Monitor". The main area is titled "Message (Enter to send message to 'Arduino Lear'")". Below that, several lines of text output are displayed, each showing a timestamp followed by a temperature reading: "01:51:51.594 -> Temperatura: 21.99 °C", "01:51:52.612 -> Temperatura: 21.99 °C", "01:51:53.609 -> Temperatura: 21.99 °C", "01:51:54.621 -> Temperatura: 21.51 °C", "01:51:55.598 -> Temperatura: 21.99 °C", and "01:51:56.618 -> Temperatura: 21.99 °C".

Figura 11.51: Resultado de registrar la temperatura del termistor LM35.



Figura 11.52: Prueba del termistor LM35 usando el Arduino Pro Micro.

■ Módulo GPS Neo-6M:

El módulo GPS Neo-6M se conecta a la placa Arduino siguiendo las especificaciones proporcionadas por el fabricante. Se conecta el pin de datos del módulo GPS (RX) al pin TX de la placa Arduino y el pin de datos (TX) al pin RX de la placa Arduino.



mediante cables jumper. De igual modo, se conectan los pines VCC y GND del módulo GPS a los pines de alimentación de 5V (VCC) y tierra (GND) de la placa Arduino, respectivamente.

Se desarrolla un programa en el entorno de desarrollo Arduino IDE para configurar el módulo GPS Neo-6M y recibir datos de los satélites para obtener la ubicación actual. El programa establece la comunicación serial con el módulo GPS y envía los comandos AT necesarios para configurar el módulo y solicitar datos de ubicación.

Para la realización de este programa, primeramente se desarrolló y probó en una placa Arduino Uno, utilizando la librería TinyGPS++, esto debido a que se presentaban problemas de comunicación al utilizarlo con un Arduino Pro Micro. Una vez verificado que el programa funcione correctamente en la placa Arduino Uno, que el módulo GPS adquiera señales de satélite y que proporcione datos precisos de ubicación, se procedió a ajustar la velocidad de comunicación del módulo para que fuera compatible con el Arduino Pro Micro.

Esto mediante un nuevo programa encargado de configurar la velocidad de comunicación del módulo GPS Neo-6M a través del envío de un comando específico (`$PMTK251,9600*17`). Luego, en el bucle principal, se lee la información proporcionada por el módulo GPS y se procesa el mensaje GPRMC para obtener la latitud, la longitud y la hora.

Las coordenadas obtenidas se convierten a formato decimal y se ajustan según el hemisferio y el huso horario para obtener la ubicación exacta y la hora local de la Ciudad de México (GMT-6). Finalmente, se imprimen las coordenadas y la hora en el formato deseado.

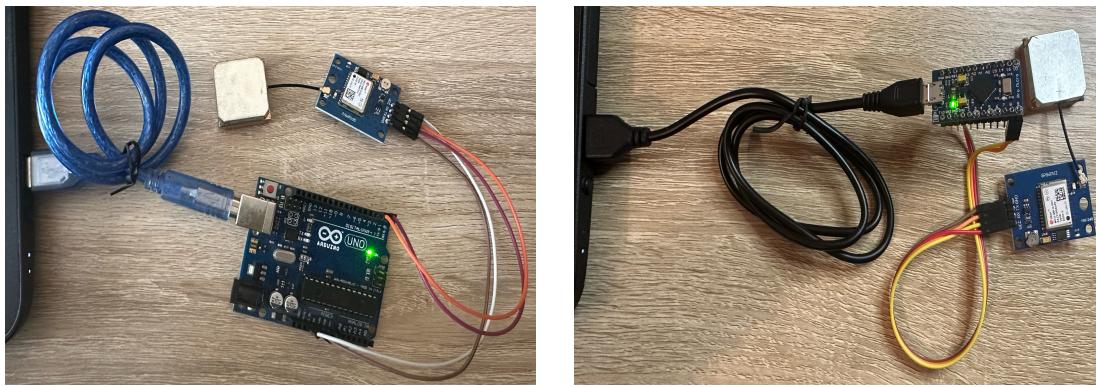
Una vez cargado el programa en la placa Arduino, se procede a verificar que el módulo GPS pueda adquirir señales de satélite y proporcionar datos precisos de ubicación. Se coloca el módulo GPS en un lugar abierto y se observa la salida del monitor serial en el Arduino IDE. Se verifican los datos de ubicación proporcionados por el módulo GPS y se comparan con la ubicación conocida para asegurar su precisión.



```
Output Serial Monitor X
Message (Enter to send message to 'Arduino Leonardo' on 'COM 13')

14:47:36.454 -> Coordenadas: 19.454799, -99.095840 ; Hora: 14:47:37
14:47:37.446 -> Coordenadas: 19.454799, -99.095840 ; Hora: 14:47:38
14:47:38.446 -> Coordenadas: 19.454799, -99.095832 ; Hora: 14:47:39
14:47:39.430 -> Coordenadas: 19.454799, -99.095832 ; Hora: 14:47:40
14:47:40.433 -> Coordenadas: 19.454797, -99.095832 ; Hora: 14:47:41
```

Figura 11.53: Resultado de obtener las coordenadas y la hora mediante el GPS-6M.



a) Usando el Arduino UNO.

b) Usando el Arduino Pro Micro.

Figura 11.54: Prueba del GPS-6M.

■ **Cámara Trampa PR-700:**

La cámara trampa se alimenta con 4 baterías AA (aunque tiene espacio para 8 baterías AA, para utilizar las otras 4 baterías si se llegara a acabar la carga de las primeras 4) o mediante el cable tipo Micro USB que viene incluido, utilizando una fuente de energía adecuada según las especificaciones del fabricante. Asegurándose de que la alimentación sea estable y suficiente para el funcionamiento óptimo de la cámara trampa durante el período de monitoreo.

Se ajustan los parámetros de la cámara trampa, como la resolución de imagen y la



frecuencia de disparo, de acuerdo con los requisitos del proyecto, esto mediante el menú de configuración de la cámara trampa.

Después de la configuración, se procede a verificar que la cámara trampa pueda capturar imágenes de alta calidad y almacenarlas en una tarjeta de memoria de tipo MicroSD. Se coloca la cámara trampa en el lugar deseado y se activa para que comience a capturar imágenes. Se revisan las imágenes capturadas para evaluar su calidad y claridad. Se verifica que las imágenes sean nítidas y que capturen adecuadamente archivos multimedia cuando ocurre un movimiento en la zona durante el tiempo preestablecido.



Figura 11.55: Prueba de la cámara trampa PR-700.

- **Módulo LoRa UART 868/915 MHz:**

Se conecta el módulo LoRa a la placa Arduino siguiendo las especificaciones del fabricante. Es crucial asegurar una conexión adecuada para garantizar la comunicación efectiva entre los dispositivos.

Se configura el módulo LoRa para establecer la comunicación con otro módulo LoRa. Esto implica ajustar parámetros como la frecuencia de operación, el factor de propagación y la potencia de transmisión para garantizar una transmisión confiable de datos, utilizando



la librería "LoRa_E22.h", que proporcionan las funciones y definiciones necesarias para la comunicación y el control del módulo LoRa E22. También se configuran los pines utilizados para conectar el Arduino Pro Micro con el módulo LoRa E22, esto incluye los pines de recepción (Rx), transmisión (Tx), AUX (para indicaciones), y los pines M0 y M1 para configurar los modos de funcionamiento del módulo LoRa.

Se realizan pruebas de comunicación entre los módulos LoRa para confirmar la conectividad y la transmisión de datos sin errores. Enviando mensajes desde un módulo LoRa utilizando la función 'sendMessage()' y verificando que sean recibidos correctamente por el otro módulo con la función 'receiveMessage()'. Se realizan múltiples pruebas en diferentes condiciones para garantizar la robustez del sistema de comunicación.

Serial Monitor X Output

```
Message (Enter to send message to 'Arduino Leonardo')
23:02:19.014 -> Success
23:02:19.014 -> Mensaje enviado.
23:02:20.524 -> Success
23:02:20.524 -> Mensaje enviado.
23:02:22.027 -> Success
23:02:22.027 -> Mensaje enviado.
```

Serial Monitor X

```
Message (Enter to send message to 'Arduino Leonardo' on 'COM11')
23:02:19.122 -> Success
23:02:19.122 -> ¡Hola receptor, este es un mensaje del transmisor!
23:02:20.683 -> Success
23:02:20.683 -> ¡Hola receptor, este es un mensaje del transmisor!
23:02:22.158 -> Success
23:02:22.158 -> ¡Hola receptor, este es un mensaje del transmisor!
23:02:23.687 -> Success
```

a Transmisor.

b Receptor.

Figura 11.56: Resultado de transmitir y recibir mediante los transceptores LoRa.



a) Transmisor.

b) Receptor.

Figura 11.57: Prueba de los módulos LoRa usando Arduinos Pro Micro.

■ **Módulo Adaptador MicroSD y Memorias MicroSD:**

Se inserta una tarjeta MicroSD en el módulo adaptador y se conecta este último a la placa de Arduino Pro Micro. Es importante asegurarse de que la tarjeta esté correctamente insertada.

Después, se conecta físicamente el módulo adaptador MicroSD a la placa Arduino, siguiendo las siguientes conexiones de acuerdo a la librería utilizada llamada SD: el pin SCK se conecta al pin 15 de la placa (SCK del adaptador), MOSI al pin 16 (MOSI del adaptador), MISO al pin 14 (MISO del adaptador) y el ChipSelect del adaptador a un pin de elección de la placa Arduino Pro Micro(en nuestro caso se configuró el pin 10).

Posteriormente, se realizan pruebas de escritura y lectura de datos desde la tarjeta MicroSD para asegurarse de que la comunicación y el almacenamiento funcionen correctamente. Esto, implementa un menú interactivo en el monitor serial que permite al usuario seleccionar entre las opciones de escritura, lectura o borrado del archivo en la tarjeta MicroSD. Este menú proporciona una forma conveniente de probar las diferentes funciones que se pueden utilizar con la librería SD.

Una vez seleccionada una opción, se ejecuta el proceso correspondiente y se realizan las



pruebas necesarias para verificar el correcto funcionamiento de cada función. Durante la escritura, se ingresan datos en el archivo en la tarjeta MicroSD. En el caso de la lectura, se recuperan los datos almacenados y se muestran en el monitor serial para su visualización. Para la opción de borrado, se elimina el archivo de la tarjeta MicroSD. El funcionamiento de este menú se puede observar en la Figura 11.58.

Después, se retira la tarjeta MicroSD y se coloca en otro dispositivo capaz de leer la tarjeta para así poder verificar la integridad de los datos almacenados. Se comparan estos datos leídos con los datos originales para asegurarse de que no haya errores de lectura o corrupción de datos.

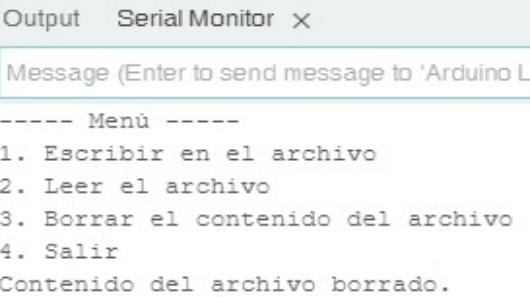
 <p>Output Serial Monitor ×</p> <p>Message (Enter to send message to 'Arduino Leonardo' on 'COM13')</p> <pre>Inicializando tarjeta SD...Inicialización completa. ----- Menú ----- 1. Escribir en el archivo 2. Leer el archivo 3. Borrar el contenido del archivo 4. Salir Contenido añadido al archivo.</pre>	 <p>Output Serial Monitor ×</p> <p>Message (Enter to send message to 'Arduino L')</p> <pre>----- Menú ----- 1. Escribir en el archivo 2. Leer el archivo 3. Borrar el contenido del archivo 4. Salir Contenido de Archivo.txt: Hola Mundo</pre>
a) Escribir en el archivo.	b) Leer el archivo.
 <p>Output Serial Monitor ×</p> <p>Message (Enter to send message to 'Arduino L')</p> <pre>----- Menú ----- 1. Escribir en el archivo 2. Leer el archivo 3. Borrar el contenido del archivo 4. Salir Contenido del archivo borrado.</pre>	 <p>Output Serial Monitor ×</p> <p>Message (Enter to send message to 'Arduino L')</p> <pre>----- Menú ----- 1. Escribir en el archivo 2. Leer el archivo 3. Borrar el contenido del archivo 4. Salir Saliendo del programa.</pre>
c) Borrar el archivo.	d) Salir del programa.

Figura 11.58: Resultado de las funciones disponibles del adaptador MicroSD.

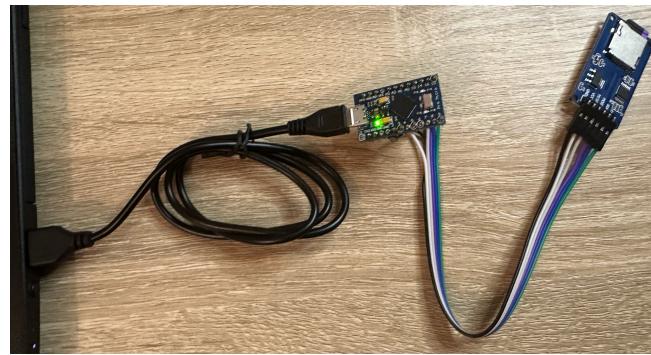


Figura 11.59: Prueba del adaptador MicroSD usando el Arduino Pro Micro.

■ Baterías LiPo:

Se utilizan cargadores compatibles para cargar completamente las baterías LiPo. Durante este proceso, se verifica que las baterías acepten la carga y que la tensión de salida sea la esperada mediante el uso de un multímetro.

Una vez que las baterías están completamente cargadas, se mide la tensión de salida para garantizar que esté dentro del rango especificado por el fabricante. Esta medida es crucial para asegurar un suministro de energía estable y confiable al sistema.

Después, se conectan las baterías LiPo al Arduino Pro Micro, asegurándose de que el positivo de la batería se conecte al pin RAW del Arduino y que la tierra se conecte al pin GND. Esta configuración garantiza un suministro de energía adecuado al Arduino y proporciona la conexión eléctrica necesaria para su funcionamiento dentro de la WSN.



Figura 11.60: Prueba de la batería LiPo para energizar el Arduino Pro Micro.



Una vez completada la verificación de cada elemento individual, se procederá a la integración de los componentes en la red de sensores inalámbricos. Por lo que es fundamental que cada componente funcione correctamente para garantizar la fiabilidad del sistema en su conjunto.

11.2.3. Integración de los Elementos

Después de haber verificado el funcionamiento de los componentes de manera individual se llevará a cabo su integración. Explorando cómo almacenar los datos de temperatura y GPS en una tarjeta MicroSD para su posterior análisis y cómo utilizar la tecnología LoRa para la transmisión eficiente de estos datos entre nodos de la red. A través de estos procesos de integración, se espera obtener una red de sensores completamente funcional y lista para su despliegue en el campo.

- **Almacenamiento de datos de temperatura en la tarjeta MicroSD:** Se almacenan los datos recopilados por el sensor de temperatura en la tarjeta MicroSD, mediante la conversión de los valores analógicos obtenidos del termistor a grados Celsius y guardándolos junto con una marca de tiempo en un archivo de texto en la tarjeta MicroSD, además se muestran estos datos en el Monitor Serial.

El código realizado comienza inicializando la comunicación serial y configurando un pin para alimentar el termistor directamente, ya que el pin de VCC será utilizado para alimentar el módulo adaptador MicroSD, también se le configura el chip select y se conectan los demás pines como se vio en el Funcionamiento de los Elementos. Después, se inicializa la tarjeta MicroSD y se borra el archivo de temperatura en caso de ya existir.

En el bucle principal, se lee el valor analógico del termistor, se convierte a temperatura en grados Celsius y se obtiene la marca de tiempo en formato HH:MM:SS iniciando desde el encendido del Arduino utilizando la función millis() de la librería TimeLib. Esta información se muestra en el Monitor Serial y se guarda en un archivo de texto en la tarjeta MicroSD en el siguiente formato: "temperatura; hora".



```
Output Serial Monitor X
Message (Enter to send message to 'Arduino Leonardo' on 'COM13')
15:25:43.253 -> Temperatura: 28.84 °C; Hora: 00:01:31
15:25:43.253 -> Temperatura y hora guardadas en SD.
15:25:44.270 -> Temperatura: 28.84 °C; Hora: 00:01:32
15:25:44.270 -> Temperatura y hora guardadas en SD.
15:25:45.273 -> Temperatura: 29.81 °C; Hora: 00:01:33
15:25:45.315 -> Temperatura y hora guardadas en SD.
15:25:46.277 -> Temperatura: 29.81 °C; Hora: 00:01:34
15:25:46.361 -> Temperatura y hora guardadas en SD.
15:25:47.314 -> Temperatura: 29.81 °C; Hora: 00:01:35
```

a) Monitor Serial.

ARCHIVO: Bloc de notas

Archivo Edición Formato Ver Ayuda

Temperatura: 28.84 °C; Hora: 00:01:31
 Temperatura: 28.84 °C; Hora: 00:01:32
 Temperatura: 29.81 °C; Hora: 00:01:33
 Temperatura: 29.81 °C; Hora: 00:01:34
 Temperatura: 29.81 °C; Hora: 00:01:35

b) Archivo de texto.

Figura 11.61: Resultados de guardar la temperatura en la memoria MicroSD.

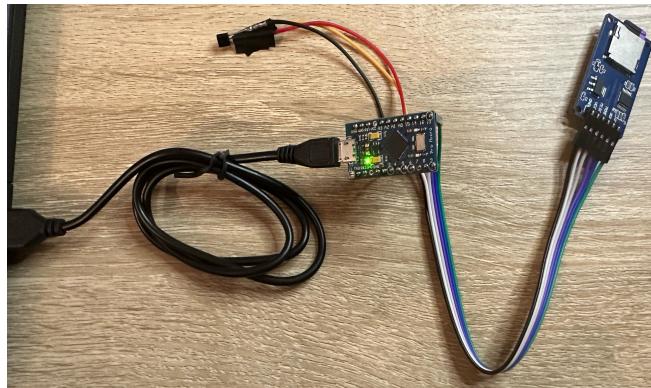


Figura 11.62: Integración del termistor y el módulo adaptador MicroSD.

- **Almacenamiento de datos GPS en la tarjeta MicroSD:** Se registran las coordenadas GPS en un archivo de texto en la tarjeta MicroSD utilizando un Arduino y un módulo GPS. Las coordenadas se registran junto con la hora actual en formato HH:MM:SS ajustada para la Ciudad de México (GMT-6).

El código comienza inicializando la comunicación serial con el módulo GPS utilizando una comunicación de software serial a través de los pines RX y TX del Arduino. Luego, se configura la velocidad de comunicación del módulo GPS y se inicializa la tarjeta MicroSD.



En el bucle principal, se lee continuamente la información del módulo GPS. Cuando se recibe un mensaje que comienza con "\$GPRMC" (el mensaje de datos mínimos recomendados de GPS), se procesa para extraer la hora actual, la latitud y la longitud. Estos datos se convierten a formato decimal y se ajusta la hora al huso horario de la Ciudad de México (GMT-6).

Luego, se crea una cadena con los datos de coordenadas y hora en el formato "latitud, longitud ; hora". Esta cadena se imprime en el Monitor Serial y se guarda en el archivo de texto en la tarjeta MicroSD.

```
Serial Monitor x
Message (Enter to send message to 'Arduino Leon'
Datos guardados en la tarjeta SD.
19.454790, -99.095749 ; 23:24:37
Datos guardados en la tarjeta SD.
19.454792, -99.095749 ; 23:24:38
Datos guardados en la tarjeta SD.
19.454792, -99.095749 ; 23:24:39
Datos guardados en la tarjeta SD.
19.454794, -99.095749 ; 23:24:40
```

a) Monitor Serial.

GPS1: Bloc de notas				
Archivo	Edición	Formato	Ver	Ayuda
19.454790, -99.095749 ; 23:24:37				
19.454792, -99.095749 ; 23:24:38				
19.454792, -99.095749 ; 23:24:39				
19.454794, -99.095749 ; 23:24:40				

b) Archivo de texto.

Figura 11.63: Resultados de guardar las coordenadas GPS en la memoria MicroSD.

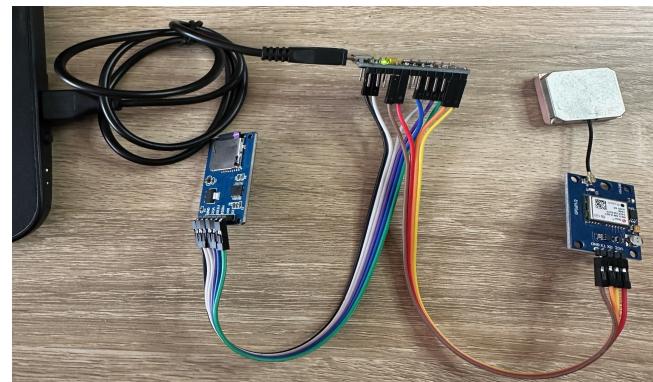


Figura 11.64: Integración del GPS y el módulo adaptador MicroSD.



- **Transmisión de archivos desde la tarjeta MicroSD mediante LoRa:** Este proyecto realiza un intercambio de archivos de texto entre dispositivos usando un Arduino Pro Micro, un módulo adaptador MicroSD y un módulo LoRa E22. El archivo de texto recibido se almacena en una tarjeta MicroSD y el usuario puede interactuar con el sistema mediante un menú en el monitor serial.

El menú permite al usuario seleccionar entre diferentes opciones: escribir en el archivo, leer el archivo, borrar el contenido del archivo y enviar el archivo por LoRa. Dependiendo de la opción seleccionada, se ejecuta la función correspondiente.

Durante la ejecución, el programa verifica continuamente si hay datos disponibles en el módulo LoRa. Si hay datos disponibles, estos se reciben y se almacenan en una variable. Los datos recibidos se muestran en el Monitor Serial para que el usuario pueda ver lo que se ha recibido y se guardan en un archivo txt en la tarjeta MicroSD. Si ocurre un error al intentar abrir el archivo, se notifica al usuario.

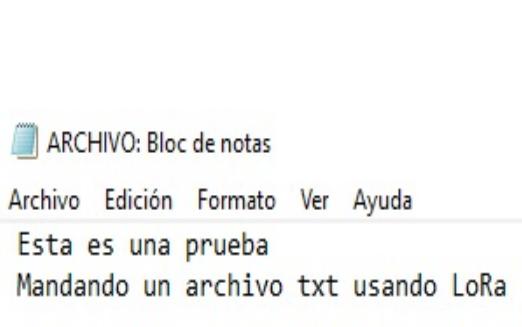
Por otra parte, para enviar el contenido del archivo, primeramente, se abre el archivo, después se guardan los datos en una variable y se envían a través del módulo LoRa. De igual forma , se notifica al usuario sobre el éxito o el fracaso del envío.

Serial Monitor X	Serial Monitor X
Message (Enter to send message to 'Arduino Leonardo' on 'COM11')	Message (Enter to send message to 'Arduino Leonardo' on 'COM11')
----- Menú -----	----- Menú -----
1. Escribir en el archivo	1. Escribir en el archivo
2. Leer el archivo	2. Leer el archivo
3. Borrar el contenido del archivo	3. Borrar el contenido del archivo
4. Enviar archivo por LoRa	4. Enviar archivo por LoRa
5. Salir	5. Salir
Contenido añadido al archivo.	Envioando contenido de Archivo.txt por LoRa...
	Success
a) Escribir en el archivo.	b) Transmitir el archivo.





c) Leer el archivo.



d) Archivo de texto recibido.

Figura 11.65: Resultados de transmitir el archivo txt de la memoria MicroSD usando LoRa.

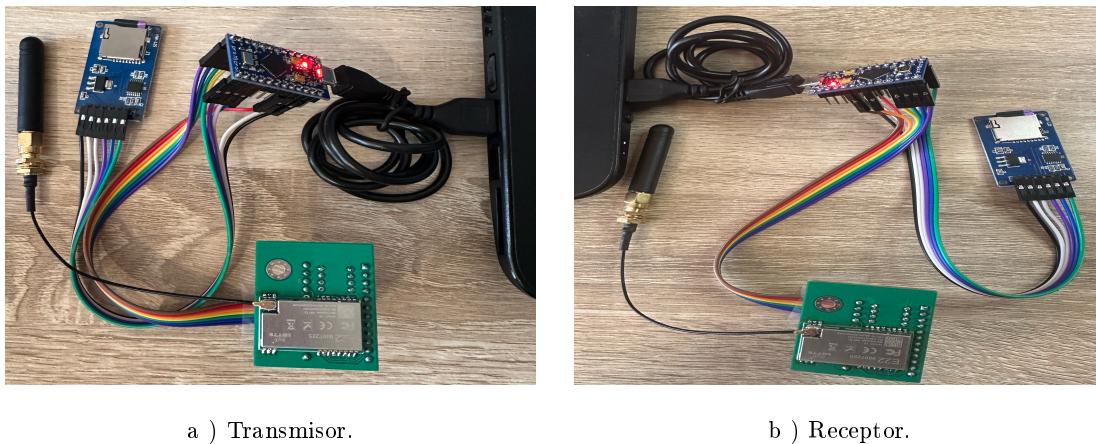


Figura 11.66: Integración del módulo adaptador MicroSD y el transceptor LoRa.

- **Almacenamiento de datos GPS en la tarjeta MicroSD con transmisión mediante LoRa:** Este proyecto permite registrar coordenadas GPS en una tarjeta MicroSD y transmitir estos datos a través de una transceptores LoRa. Utiliza un Arduino, un módulo GPS, un módulo adaptador MicroSD, un módulo LoRa E22 y una tarjeta MicroSD para realizar estas tareas.

El sistema comienza con la inicialización de la tarjeta MicroSD. Si la tarjeta no se



inicializa correctamente, se notifica al usuario y el programa se detiene. A continuación, se configura el módulo GPS para comunicarse a través de una conexión serial, ajustando la velocidad de comunicación del módulo GPS NEO-6M. Posteriormente, se inicializa el módulo LoRa E22, preparándolo para la transmisión de datos.

El módulo GPS se utiliza para recibir datos de ubicación en formato NMEA, que incluye información sobre la latitud, longitud y hora. Estos datos se procesan para extraer las coordenadas y la hora actual, convirtiendo las coordenadas a un formato decimal y ajustando la hora al huso horario de la Ciudad de México (GMT-6).

Las coordenadas GPS y la hora se guardan en un archivo de texto en la tarjeta MicroSD, actualizándose cada vez que se reciben nuevos datos GPS. El sistema abre el archivo en modo de escritura, guarda los datos y luego cierra el archivo para asegurar que la información se almacene correctamente.

Después, se define un intervalo de tiempo en el que se estarán guardando las coordenadas en la tarjeta y concluyendo este tiempo se iniciará la transmisión del archivo. Donde se lee el contenido del archivo y se envía a través del módulo LoRa. La transmisión se realiza en varios pasos: primero se envía un mensaje de inicio, seguido del nombre del archivo y luego cada línea de datos. Finalmente, se envía un mensaje de finalización para indicar que la transmisión ha terminado. Si la transmisión es exitosa, el archivo en la tarjeta MicroSD se borra para preparar el sistema para la próxima serie de datos.

Aunque el sistema opera principalmente de forma autónoma, la comunicación con el usuario se realiza a través del Monitor Serial. El usuario puede ver mensajes que indican el estado de la inicialización, la lectura de datos, el almacenamiento y la transmisión. Ilustrando así cómo integrar múltiples tecnologías (GPS, MicroSD y LoRa) para crear un sistema de registro y transmisión de datos eficiente, ideal para aplicaciones que requieren seguimiento de ubicación y comunicación inalámbrica.



Figura 11.67: Resultados de guardar las coordenadas en un archivo txt en la memoria MicroSD y trasmitirlo usando LoRa.

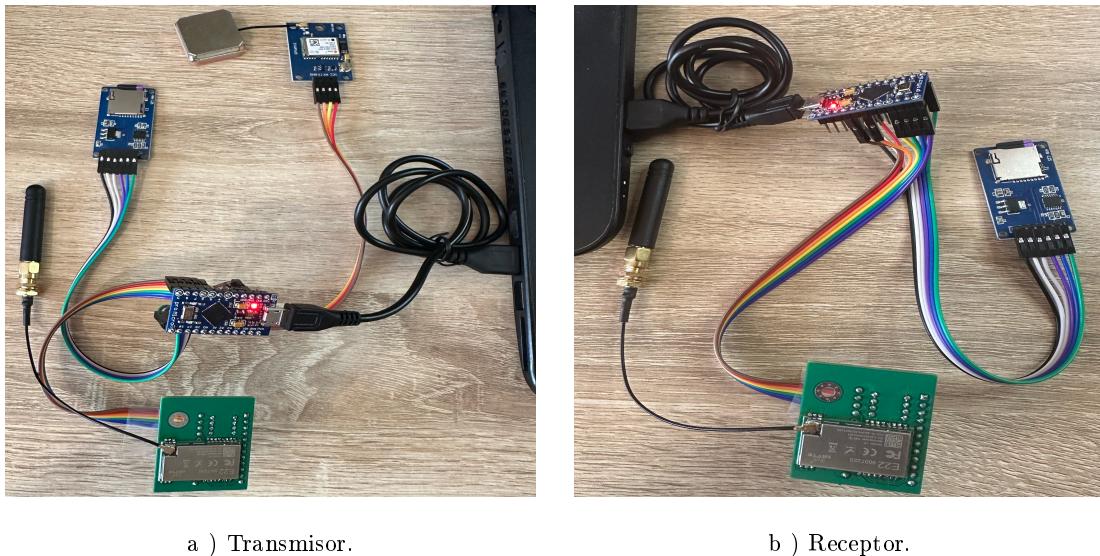


Figura 11.68: Integración del GPS, el módulo adaptador MicroSD y el transceptor LoRa.

- **Recepción y almacenamiento de datos GPS mediante LoRa junto con datos de temperatura en la tarjeta MicroSD:** En este proyecto, se establece el sistema de recepción de datos GPS mediante LoRa, el cual almacena estos datos recibidos en



una tarjeta MicroSD con el nombre de archivo recibido, a la par que almacena datos de temperatura. Utilizando un Arduino Pro Micro, un módulo GPS, un termistor, un módulo LoRa E22 y una tarjeta MicroSD.

Al iniciar, el sistema configura y alimenta el termistor, el módulo LoRa, y la tarjeta MicroSD. Si la inicialización del dispositivo de almacenamiento falla, se notifica al usuario y se detiene la ejecución para evitar errores posteriores. Una vez que todo está configurado correctamente, el sistema entra en el bucle principal donde se ejecutan dos tareas principales: la lectura y almacenamiento de la temperatura, y la recepción de archivos a través de LoRa.

La primera tarea es la lectura de la temperatura. El sistema lee el valor analógico del termistor, convierte este valor a grados Celsius, y obtiene la hora actual utilizando la librería TimeLib. La temperatura y la hora se imprimen en el Monitor Serial y se almacenan en un archivo en la tarjeta MicroSD. Este proceso se repite cada intervalo de tiempo establecido, proporcionando un registro continuo de la temperatura ambiental.

La segunda tarea es la recepción de archivos a través de LoRa. El sistema espera mensajes del módulo LoRa, y cuando recibe un mensaje de inicio (“START”), procede a recibir el nombre del archivo. Una vez recibido el nombre, se crea (en caso de no existir aún) y abre un archivo con ese nombre en la tarjeta MicroSD y se empiezan a recibir los datos del archivo. Los datos se almacenan línea por línea hasta que se recibe un mensaje de finalización (“END”). Si hay algún error en la recepción de los mensajes o al abrir el archivo, se notifica al transmisor para que no borre el archivo transmitido.

Combinando de esta forma, la recepción de datos GPS mediante LoRa con el almacenamiento de datos de temperatura en una tarjeta MicroSD. Utilizando tecnologías de comunicación inalámbrica y almacenamiento digital, se crea una solución robusta para la recopilación y registro de datos ambientales y de ubicación, adecuada para diversas aplicaciones en monitoreo remoto y seguimiento de activos.



```
Serial Monitor X
Message (Enter to send message to 'Arduino Leonardo' on 'COM13')

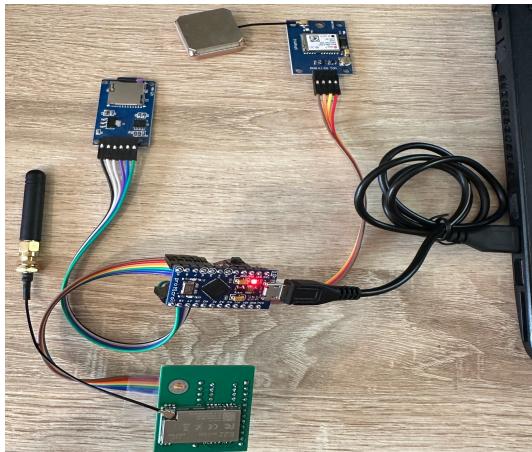
Temperatura: 33.72 °C; Hora: 00:05:52
Temperatura y hora guardadas en SD.
Temperatura: 33.72 °C; Hora: 00:05:54
Temperatura y hora guardadas en SD.
Inicio de recepción de archivo.
Nombre de archivo recibido: gps1.txt
Guardando archivo...
Fin de recepción de archivo.
Temperatura: 32.26 °C; Hora: 00:06:24
Temperatura y hora guardadas en SD.
```

a) Monitor serial.

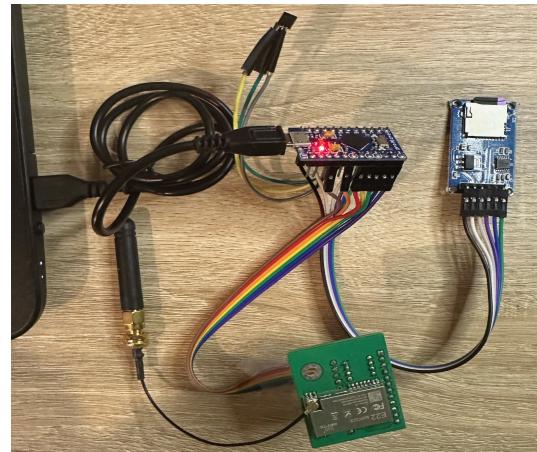
SD > SDHC (G:)				
Nombre	Fecha de modificación	Tipo	Tamaño	
GPS1	01/01/2000 01:00 a. m.	Documento de te...	21 KB	
TEMP	01/01/2000 01:00 a. m.	Documento de te...	35 KB	

b) Archivos generados.

Figura 11.69: Resultados de recibir las coordenadas usando LoRa, guardarlos en un archivo txt en la memoria MicroSD y recolectar datos de temperatura a la par.



a) Transmisor.



b) Receptor.

Figura 11.70: Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa.



- **Transmisión de datos GPS y de temperatura de la tarjeta MicroSD mediante LoRa:**

En este proyecto, se establece un sistema para la transmisión de datos GPS y de temperatura contenidos en archivos de texto en una tarjeta MicroSD utilizando la tecnología LoRa. El sistema está compuesto por un Arduino Pro Micro, un termistor, un módulo LoRa E22, y una tarjeta MicroSD. La función principal es enviar los datos almacenados en la tarjeta MicroSD a través del módulo LoRa a un receptor remoto.

El programa comienza con la configuración inicial de los componentes. Se configuran los pines del módulo LoRa y de la tarjeta MicroSD. El módulo LoRa es configurado utilizando la biblioteca LoRa_E22, y la tarjeta MicroSD es inicializada utilizando la biblioteca SD. Si la tarjeta no se inicializa correctamente, se notifica al usuario y el programa se detiene para evitar errores posteriores.

En el bucle principal, el sistema opera en tres modos principales de operación: recepción de archivos GPS por LoRa, medición y almacenamiento de temperatura, y transmisión de archivos GPS y de temperatura. Respecto a la recepción por LoRa, el Arduino monitorea continuamente la disponibilidad de mensajes mediante el módulo LoRa. Cuando se detecta un mensaje de inicio (“START”), se inicia la recepción del archivo correspondiente. Esto implica la espera y recepción del nombre del archivo a través de LoRa, seguido por la recepción secuencial de cada línea de datos. Cada línea de datos recibida se guarda directamente en un archivo con su correspondiente nombre en la tarjeta MicroSD, repitiendo esto hasta recibir un mensaje de finalización (“END”). Notificando al transmisor si ocurrió algún error en la recepción de los datos para anular la eliminación del archivo en el transmisor.

Por otro lado, el Arduino también realiza mediciones periódicas de temperatura a intervalos regulares establecidos. Durante cada ciclo de medición, el termistor conectado se activa para obtener una lectura analógica que representa la temperatura ambiental. Esta lectura se convierte a grados Celsius y se complementa con la hora actual en formato HH:MM. Los datos de temperatura junto con la marca de tiempo se registran de manera



sistemática en un archivo específico (temp.txt) en la tarjeta SD.

Además de la recepción y almacenamiento local de datos, lo agregado en este sistema es que también está diseñado para la transmisión periódica de sus archivos por LoRa cada cierto periodo de tiempo (mayor al de la medición de temperatura). Durante este proceso, el Arduino verifica la existencia de archivos predefinidos (gps1.txt, gps2.txt, gps3.txt y temp.txt) en la tarjeta MicroSD. Para cada archivo existente, se inicia una secuencia de transmisión por LoRa, comenzando con el envío del nombre del archivo y seguido por el envío de cada línea de datos de manera ordenada. Un mensaje “START” indica el inicio de la transmisión y un mensaje “END” señala su finalización exitosa. Después de completar una transmisión satisfactoria (recibiendo un mensaje de confirmación del receptor), el archivo correspondiente se elimina automáticamente de la tarjeta MicroSD para gestionar eficientemente el espacio de almacenamiento y garantizar la continuidad en la recolección de datos.

```
Serial Monitor x Output
Message (Enter to send message to 'Arduino Leonardo' on 'COM13')
Inicio de recepción de archivo.
Nombre de archivo recibido: gps3.txt
Guardando archivo...
Fin de recepción de archivo.
Temperatura: 32.51 °C; Hora: 00:12:36
Temperatura y hora guardadas en SD.
Temperatura: 32.68 °C; Hora: 00:12:37
Temperatura y hora guardadas en SD.
Enviando contenido de gps1.txt por LoRa...
Success
Datos enviados por LoRa.
Enviando contenido de gps2.txt por LoRa...
Success
Datos enviados por LoRa.
Enviando contenido de gps3.txt por LoRa...
Success
Datos enviados por LoRa.
Enviando contenido de temp.txt por LoRa...
Success
Datos enviados por LoRa.
Temperatura: 33.12 °C; Hora: 00:17:09
Temperatura y hora guardadas en SD.
```

a) Monitor serial.



Nombre	Fecha de modificación	Tipo	Tamaño
GPS1	01/01/2000 01:00 a. m.	Documento de te...	106 KB
GPS2	01/01/2000 01:00 a. m.	Documento de te...	105 KB
GPS3	01/01/2000 01:00 a. m.	Documento de te...	104 KB
TEMP	01/01/2000 01:00 a. m.	Documento de te...	75 KB

b) Archivos generados.

Figura 11.71: Resultados de recibir las coordenadas usando LoRa, recolectar datos de temperatura y transmitir estos archivos txt.

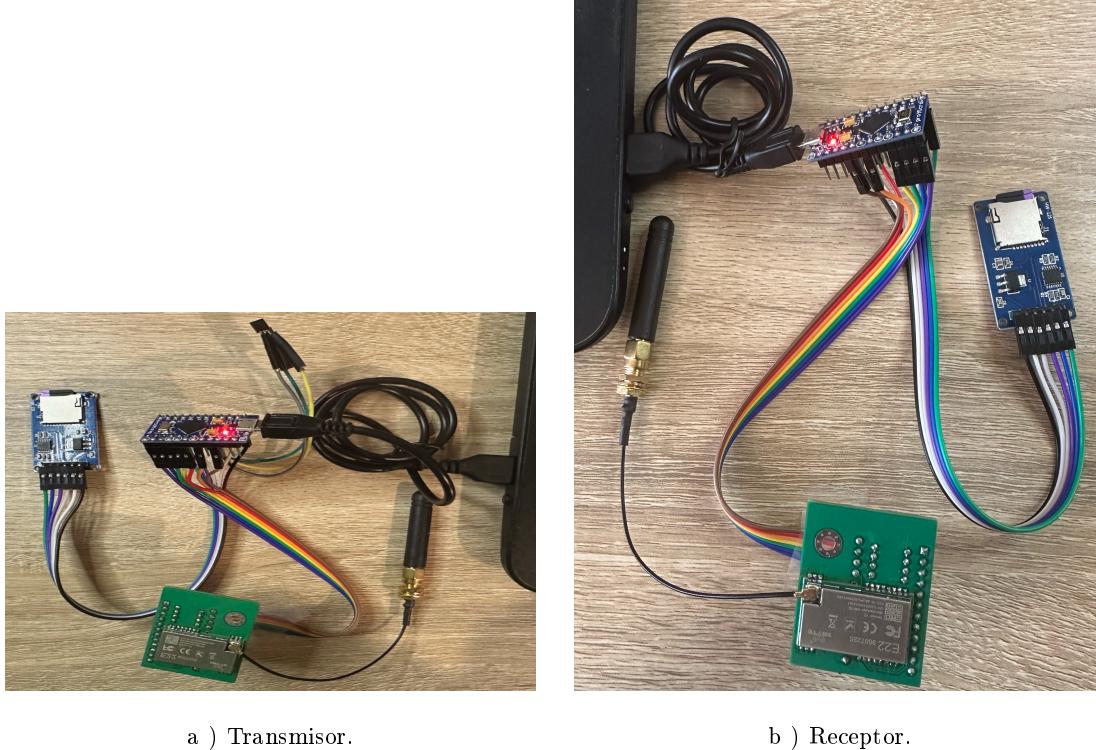


Figura 11.72: Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa.

Con esta implementación integral de captura, almacenamiento local y transmisión remota



de datos utilizando sensores, Arduinos, transceptores LoRa y tarjetas MicroSD, se concluye la configuración de la Red de Sensores Inalámbricos (WSN) propuesta para el monitoreo de especies en peligro de extinción. Este sistema no solo permite el registro en tiempo real de la temperatura ambiental y de coordenadas GPS, sino que también facilita la transmisión remota de datos a largas distancias. Esto garantiza el cumplimiento de las necesidades de monitoreo y conservación animal en entornos remotos y de difícil acceso.



11.3. Modelado del consumo energético del sistema

El modelado del consumo energético es un aspecto clave en el diseño y la optimización de sistemas de redes de sensores inalámbricos (WSN), especialmente en aplicaciones donde los dispositivos deben operar con una fuente de energía limitada, como baterías. Por lo que, se propone utilizar el Proceso de Poisson Modulado por Markov o MMPP (Markov Modulated Poisson Process) para modelar el consumo energético de nuestro proyecto. Este enfoque permitirá capturar la naturaleza estocástica y dinámica del consumo de energía, tomando en cuenta la variabilidad en la actividad de la red.

Este proceso se basa en la combinación de un proceso de Poisson, que representa la llegada de eventos, y una cadena de Markov, que modula la tasa de llegada de estos eventos. Para una WSN, los estados de la cadena de Markov representan diferentes niveles de actividad de la red, mientras que los eventos de Poisson corresponden a la ocurrencia de transmisiones de datos, procesamiento de información, sensado de datos, entre otros.

En el MMPP, se define una cadena de Markov a tiempo continuo con un conjunto finito de estados, donde cada estado representa una condición específica del sistema. La transición entre estados se rige por probabilidades de transición, que dependen del estado actual del sistema y de parámetros característicos del proceso.

La tasa de llegada de eventos en un estado dado se modula por el estado actual de la cadena de Markov. Esto significa que tanto la frecuencia con la que ocurren eventos, como el almacenamiento de los datos, su transmisión o el encendido de nodos, varía dependiendo del nivel de actividad de la red en ese momento.

Para nuestro sistema se planea utilizar un MMPP descrito con las siguientes características que se explicarán a continuación:

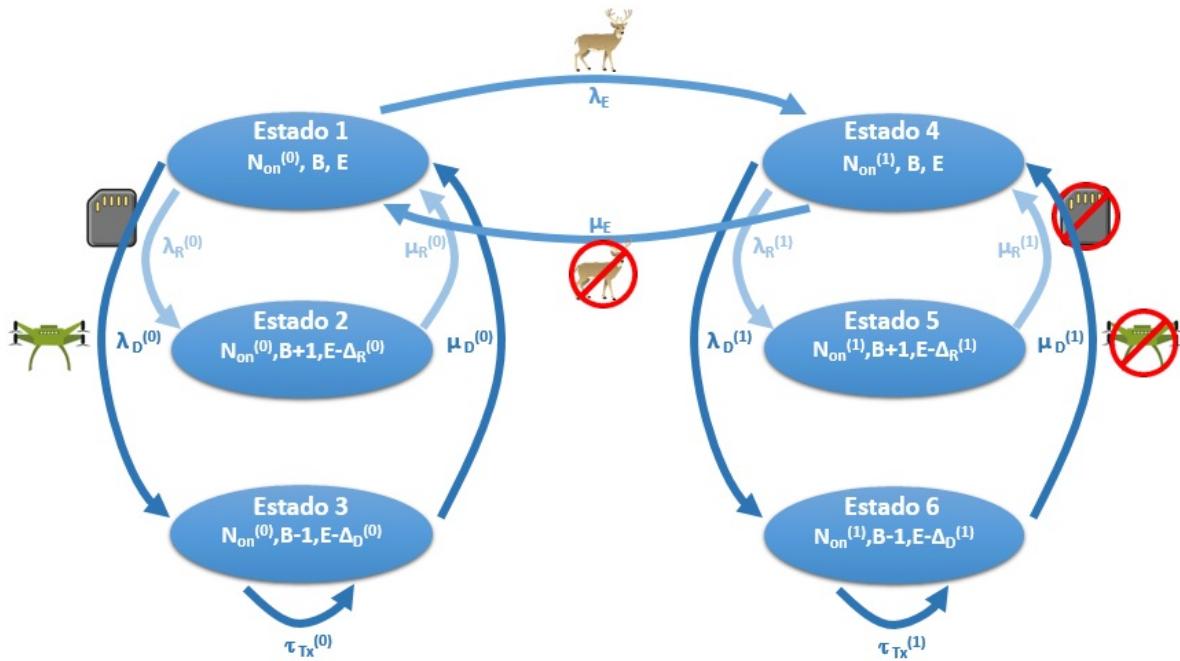


Figura 11.73: Diagrama de estados principal del MMPP implementado.

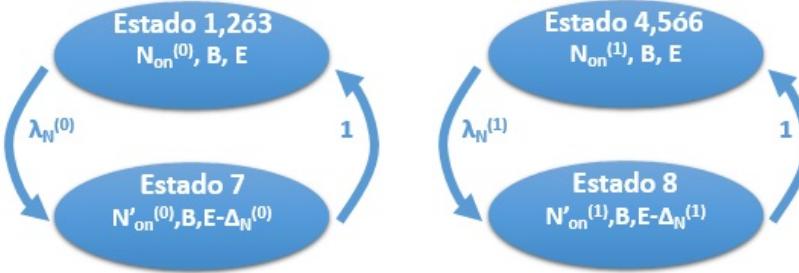


Figura 11.74: Diagrama de estados secundario del MMPP implementado.

11.3.1. Variables del sistema

El modelo de consumo energético de nuestro sistema considera tres variables fundamentales que influyen en el comportamiento y tiempo de vida de la red:

- **Número de nodos activos (Non):** Representa la cantidad de nodos que están activos y consumiendo energía en un momento dado. La variación en el número de nodos activos



afecta directamente el consumo energético de la red, ya que cada nodo requiere energía para su funcionamiento y participación en las tareas de recolección, procesamiento y transmisión de datos. Cada nodo puede estar en uno de cuatro estados: encendido, transmitiendo a un nodo, transmitiendo a un dron o en modo de reposo (sleep). En el estado encendido, el nodo consume una cantidad de energía E_{on} , mientras que transmitiendo datos a otro nodo consume E_{Tx} , transmitiendo a un dron consume $E_{\text{Tx_D}}$ y en el estado de reposo consume E_{sleep} . Teniendo en cuenta que: $E_{\text{Tx_D}} > E_{\text{Tx}} > E_{\text{on}} \gg E_{\text{sleep}}$.

- **Tamaño del buffer (B):** El tamaño del buffer se refiere a la capacidad de almacenamiento de datos en cada nodo de la red. Los datos recolectados por los sensores se almacenan temporalmente en el buffer antes de ser procesados o transmitidos. El tamaño del buffer influye en la eficiencia operativa de la red, ya que determina la cantidad de datos que pueden ser retenidos antes de su procesamiento, lo que a su vez impacta en el consumo energético.
- **Energía total del sistema (E):** Es la energía total disponible en la red en un instante específico. Esta energía se distribuye entre los nodos activos y se consume durante las distintas actividades de la red. El tiempo de vida del sistema depende directamente de la energía disponible (E), ya que cuando se reduce hasta cero, el proceso se termina y la red deja de funcionar. Por lo tanto, la gestión eficiente de la energía es esencial para maximizar la vida útil de la red y garantizar su máximo funcionamiento.

11.3.2. Estados del sistema

El modelo se basa en la definición de distintos estados que representan las diferentes etapas del ciclo de operación de la red. Estos estados incluyen las actividades típicas realizadas por los nodos y los eventos que ocurren en la red. A continuación, se describen los estados principales del sistema, junto con las modificaciones en las variables asociadas a cada estado, los cuales también se pueden ver en la Figura 11.73:

1. **Estado inicial:** En este estado, se inicializan los parámetros del sistema, incluyendo el número de nodos activos ($N_{\text{on}}^{(0)}$), el tamaño del buffer (B), y la energía inicial disponible en la red (E_i). Es el punto de partida desde el cual comienza el proceso de monitoreo y recolección de datos.
2. **Almacenamiento de datos:** Los nodos activos de la red comienzan a recolectar los datos de la temperatura, GPS o de interés y los almacenan temporalmente en sus buffers. En este estado, el tamaño del buffer aumenta en una unidad ($B + 1$), mientras que la energía disponible en la red disminuye en una cantidad específica ($E_i - \Delta_R^{(0)}$), donde $\Delta_R^{(0)} = (N - N_{\text{on}}^{(0)}) \times E_{\text{sleep}} + (N_{\text{on}}^{(0)} - 1) \times E_{\text{on}} + E_{\text{Tx}}$.
3. **Llegada del dron:** En este estado, el dron o enjambre llega a la zona de cobertura de los nodos activos para recolectar los datos almacenados en sus buffers. Esta etapa implica la transferencia de datos desde los nodos hacia el dron para su procesamiento posterior. El tamaño del buffer disminuye en una unidad ($B - 1$), mientras que la energía disponible en la red se reduce ($E_i - \Delta_D^{(0)}$), donde $\Delta_D^{(0)} = (N - N_{\text{on}}^{(0)}) \times E_{\text{sleep}} + (N_{\text{on}}^{(0)} - 1) \times E_{\text{on}} + E_{\text{Tx_D}}$.
4. **Llegada del animal:** Cuando el animal de interés entra en el área cubierta pero no se están recolectando datos, se produce este estado. En este estado, el número de nodos activos es $N_{\text{on}}^{(1)}$, mientras que el tamaño del buffer y la energía disponible permanecen sin cambios.
5. **Almacenamiento de datos con el animal presente:** Durante este estado, los nodos recolectan y almacenan datos, aprovechando la presencia del animal en el área de cobertura. El tamaño del buffer aumenta en una unidad ($B + 1$), mientras que la energía disponible en la red disminuye ($E_i - \Delta_R^{(1)}$), donde $\Delta_R^{(1)} = (N - N_{\text{on}}^{(1)}) \times E_{\text{sleep}} + (N_{\text{on}}^{(1)} - 1) \times E_{\text{on}} + E_{\text{Tx}}$.
6. **Llegada del dron con el animal presente:** Similar al estado 3, este estado implica la llegada del dron o enjambre para recolectar datos de los nodos mientras el animal está presente en el área de cobertura. En este estado, el tamaño del buffer disminuye en una



unidad ($B - 1$), y la energía disponible en la red también se reduce ($E_i - \Delta_D^{(1)}$), donde $\Delta_D^{(1)} = (N - N_{\text{on}}^{(1)}) \times E_{\text{sleep}} + (N_{\text{on}}^{(1)} - 1) \times E_{\text{on}} + E_{\text{Tx_D}}$.

7. y 8. **Cálculo de los nuevos nodos encendidos:** Estos estados ocurren a la par de la transición de los otros seis estados y representan la determinación del número de nodos que estarán activos en el próximo ciclo de operación. Aquí, se calcula el nuevo número de nodos activos ($N_{\text{on}}^{(0)}$ o $N_{\text{on}}^{(1)}$, según sea el caso), esto en función de las condiciones actuales de la red, mientras que el tamaño del buffer se mantiene igual porque no está almacenando datos nuevos, pero la energía disponible sí se reduce ($E_i - \Delta_N^{(0)}$ o $E_i - \Delta_N^{(1)}$), donde $\Delta_N^{(0)} = (N - N_{\text{on}}^{(0)}) \times E_{\text{sleep}} + (N_{\text{on}}^{(0)}) \times E_{\text{on}}$ y $\Delta_N^{(1)} = (N - N_{\text{on}}^{(1)}) \times E_{\text{sleep}} + (N_{\text{on}}^{(1)}) \times E_{\text{on}}$, según corresponda.

Probabilidades de transición del sistema

Dado que las probabilidades de transición están definidas por tasas de transición en el modelo, podemos asociar cada transición a una tasa específica que determina la probabilidad de pasar de un estado a otro en un intervalo de tiempo dado. A continuación, se detallan las tasas de transición asociadas a cada estado, las cuales también se pueden ver en la Figura 11.73:

1. Estado inicial:

- λ_E : Tasa de transición que representa la probabilidad de transición de pasar del estado inicial (estado 1) al estado donde llega el animal (estado 4).
- $\lambda_R^{(0)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado inicial (estado 1) al estado de almacenamiento de datos (estado 2).
- $\lambda_D^{(0)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado inicial (estado 1) al estado donde llega el dron (estado 3).

2. Almacenamiento de datos:

- $\mu_R^{(0)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado de almacenamiento de datos (estado 2) al estado inicial (estado 1).
- $\tau_B^{(0)}$: Tasa de transición que representa la probabilidad de transición de permanecer en el estado de almacenamiento de datos (estado 2).

3. Llegada del dron:

- $\mu_D^{(0)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado donde el dron está recopilando los datos (estado 3) al estado inicial (estado 1).
- $\tau_{Tx}^{(0)}$: Tasa de transición que representa la probabilidad de transición de permanecer en el estado donde el dron está recopilando los datos (estado 3).

4. Llegada del animal:

- μ_E : Tasa de transición que representa la probabilidad de transición de pasar del estado donde está el animal presente (estado 4) al estado donde no se encuentra el animal presente (estado 1).
- $\lambda_R^{(1)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado donde está el animal presente (estado 4) al almacenamiento en el buffer con el animal presente (estado 5).
- $\lambda_D^{(1)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado donde está el animal presente (estado 4) al estado donde llega el dron con el animal presente (estado 6).

5. Almacenamiento de datos con el animal presente:

- $\mu_R^{(1)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado de almacenamiento de datos con el animal presente (estado 5) al estado con el animal presente pero sin almacenar datos (estado 4).



- $\tau_B^{(1)}$: Tasa de transición que representa la probabilidad de transición de permanecer en el estado de almacenamiento de datos con el animal presente (estado 5).

6. Llegada del dron con el animal presente:

- $\lambda_D^{(1)}$: Tasa de transición que representa la probabilidad de transición de pasar del estado donde el dron está recopilando los datos con el animal presente (estado 5) al estado con el animal presente pero sin estar el dron (estado 4).
- $\tau_{Tx}^{(1)}$: Tasa de transición que representa la probabilidad de transición de permanecer en el estado donde el dron está recopilando los datos con el animal presente (estado 5).

7. Cálculo de los nuevos nodos encendidos:

- $\lambda_N^{(0)}$: Tasa de transición que representa la probabilidad de transición de pasar a un estado de cálculo de los nuevos nodos encendidos (estado 7). Esta tasa está definida por $\lambda_N^{(0)} = \binom{N_{on}(0)}{N'_{on}(0)} \times (P_{on}^{(0)})^{N'_{on}(0)} \times (1 - P_{on}^{(0)})^{(N_{on}^{(0)} - N'_{on}(0))}$ donde $P_{on}^{(0)}$ es la probabilidad de que prendan nuevos nodos y $N'_{on}(0)$ son los nuevos nodos prendidos.

8. Cálculo de los nuevos nodos encendidos con el animal presente:

- $\lambda_N^{(1)}$: Tasa de transición que representa la probabilidad de transición de pasar a un estado de cálculo de los nuevos nodos encendidos estando el animal presente (estado 8). Esta tasa está definida por $\lambda_N^{(1)} = \binom{N_{on}(1)}{N'_{on}(1)} \times (P_{on}^{(1)})^{N'_{on}(1)} \times (1 - P_{on}^{(1)})^{(N_{on}^{(1)} - N'_{on}(1))}$ donde $P_{on}^{(1)}$ es la probabilidad de que prendan nuevos nodos y $N'_{on}(1)$ son los nuevos nodos prendidos cuando el animal está presente.

11.3.3. Implementación y resultados del sistema

En esta sección, se describe la implementación del modelo de consumo energético del sistema utilizando el lenguaje de programación Python. La implementación se divide en varias



partes, cada una correspondiente a una etapa específica del ciclo de operación de la red. Además, se presentan los resultados obtenidos de la simulación para validar el modelo y analizar el comportamiento del sistema.

1. Simulación del tiempo de vida sin energía ni actualización del número de nodos encendidos:

Para esta primera etapa de implementación, se desarrolló un programa que simula el tiempo de vida del sistema sin tener en cuenta la actualización del número de nodos encendidos ni la variación en el nivel de energía al realizar una transición de un estado a otro, es decir, sin considerar las Δ vistas en *Estados del sistema*, considerando el consumo de todas las transiciones de estados como 1 unidad de energía. Utilizando las siguientes funciones:

- **Función para generar tiempos exponenciales:** Esta función genera tiempos exponenciales basados en la distribución exponencial, con el fin de generar los tiempos de permanencia en cada estado del sistema. Utilizando: $t = -\frac{1}{\lambda} \cdot \ln(1-u)$, donde λ es la probabilidad de transición a ese estado y u es un número aleatorio uniformemente distribuido en el intervalo $[0, 1]$.
- **Función para simular el tiempo de vida del sistema:** Esta función utiliza un Proceso de Poisson Modulado por Markov para calcular cuántas unidades de tiempo duraría el sistema hasta agotar su energía. Por lo tanto, esta función se realiza iterativamente hasta que la energía del sistema llega a cero. Durante cada iteración, para cada estado se generan tiempos exponenciales mediante la función anterior para determinar que el próximo estado del sistema será aquel cuyo tiempo sea el mínimo entre los tiempos exponenciales generados de todos los posibles siguientes estados a partir del estado actual, y se ajusta la energía del sistema (en este caso, solamente se le resta una unidad de energía). Estos tiempos mínimos se van acumulando hasta que la energía se ha agotado para obtener el tiempo total de vida del sistema.



El resultado de una simulación de este programa se puede ver en la siguiente figura:

Tiempo total de vida del sistema: 784.36 unidades de tiempo

Figura 11.75: Resultado de una simulación de la etapa 1 de la implementación del MMPP.

Para obtener un promedio del tiempo de vida, se procedió a realizar múltiples simulaciones del programa, mostrando un gráfico de dispersión que muestra los tiempos de vida en cada simulación con una línea punteada roja que representa el tiempo promedio de vida del sistema. También se muestra un histograma que muestra la distribución de los tiempos de vida y la frecuencia de aparición de diferentes rangos de tiempo, como se puede ver a continuación:

Tiempo promedio de vida del sistema en 1000 simulaciones: 977.23 unidades de tiempo

Figura 11.76: Resultado de varias simulaciones de la etapa 1 de la implementación del MMPP.

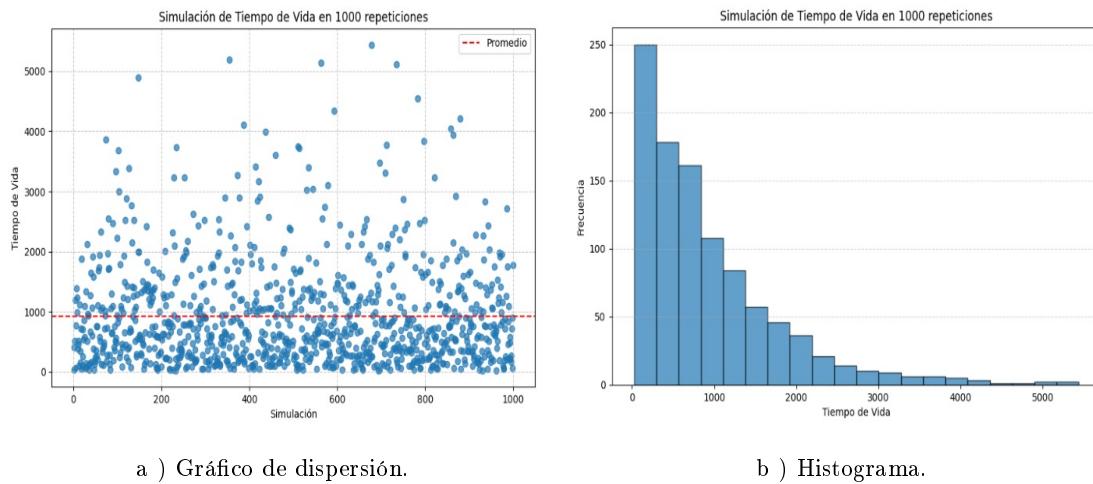


Figura 11.77: Resultados gráficos de varias simulaciones de la etapa 1 de la implementación del MMPP.



2. Simulación del tiempo de vida con energía sin actualización de nodos encendidos:

En esta etapa, se incorpora el cálculo de la energía que consume la transición a los diferentes estados, es decir, las Δ vistas en *Estados del sistema*. Pero todavía no se considera la actualización dinámica del número de nodos encendidos. Obteniendo una estimación un poco más realista del tiempo de vida del sistema.

Por lo que, en vez de restarle una unidad a la energía cada vez que se realiza una transición de estado como se hizo en la etapa anterior, ahora se le resta la respectiva Δ que consumirá esa transición, considerando las fórmulas de cada Δ vistas en *Estados del sistema*, obteniendo el siguiente resultado para una iteración:

Tiempo total de vida del sistema: 18.92 unidades de tiempo

Figura 11.78: Resultado de una simulación de la etapa 2 de la implementación del MMPP.

Realizando el mismo proceso en un mayor número de simulaciones para obtener un tiempo de vida promedio, se obtuvieron los siguientes resultados:

Tiempo promedio de vida del sistema en 10000 simulaciones: 46.31 unidades de tiempo

Figura 11.79: Resultado de varias simulaciones de la etapa 2 de la implementación del MMPP.

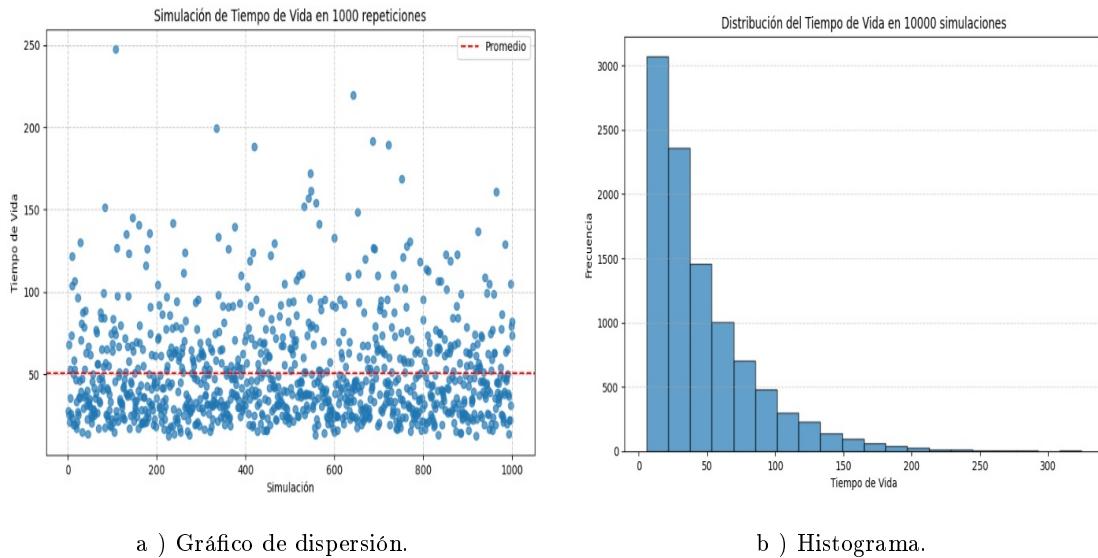


Figura 11.80: Resultados gráficos de varias simulaciones de la etapa 2 de la implementación del MMPP.

Pudiendo observar que al considerar un consumo energético real en el sistema, el tiempo de vida es menor en comparación con el escenario anterior donde se asume un consumo de energía uniforme de 1 unidad para cada estado. Esto se debe a que un consumo de energía más alto provocará una disminución más rápida en el nivel de energía total del sistema, lo que resultará en un tiempo de vida total más corto.

3. Actualización de nodos encendidos y su nueva probabilidad:

En esta etapa, se diseñó un programa para calcular el número de nodos a encender y la probabilidad de transición asociada en el siguiente paso del sistema. Este cálculo se realiza considerando un número máximo de nodos N_{\max} y una probabilidad inicial $P_{\text{on}0}$, esto mediante las siguientes funciones:

- Función para calcular la probabilidad de transición: Esta función calcula la probabilidad de transición P para encender N nodos. Esta función utiliza la distribución

binomial para calcular P para todas las posibles cantidades de nodos encendidos dentro del rango de 0 a N .

- Función para calcular el siguiente número de nodos encendidos: Esta función determina el número de nodos a encender en el siguiente paso del sistema. Dentro de esta función, se genera un número aleatorio v entre 0 y 1. A continuación, se calcula la probabilidad acumulada sumando las probabilidades generadas en el paso anterior hasta que esta sea mayor que v . En ese punto, se toma la cantidad de nodos N correspondiente a esa probabilidad acumulada como la nueva cantidad de nodos encendidos. Finalmente, se recalcula P para la cantidad específica de nodos encendidos determinada anteriormente.

El resultado de una iteración de este programa utilizando un número máximo de nodos igual a 5 es el siguiente:

```
Se prenden 2 nodos
El valor de P es: 0.3087
```

Figura 11.81: Resultado de una simulación de la etapa 3 de la implementación del MMPP.

Realizando múltiples simulaciones del cálculo del siguiente número de nodos y las probabilidades de transición correspondientes, se obtienen las siguientes gráficas de dispersión y distribución:

```
En 1000 simulaciones mayormente se suele prender : 1 nodo
Con su respectiva P de: 0.2701
```

Figura 11.82: Resultado de varias simulaciones de la etapa 3 de la implementación del MMPP.

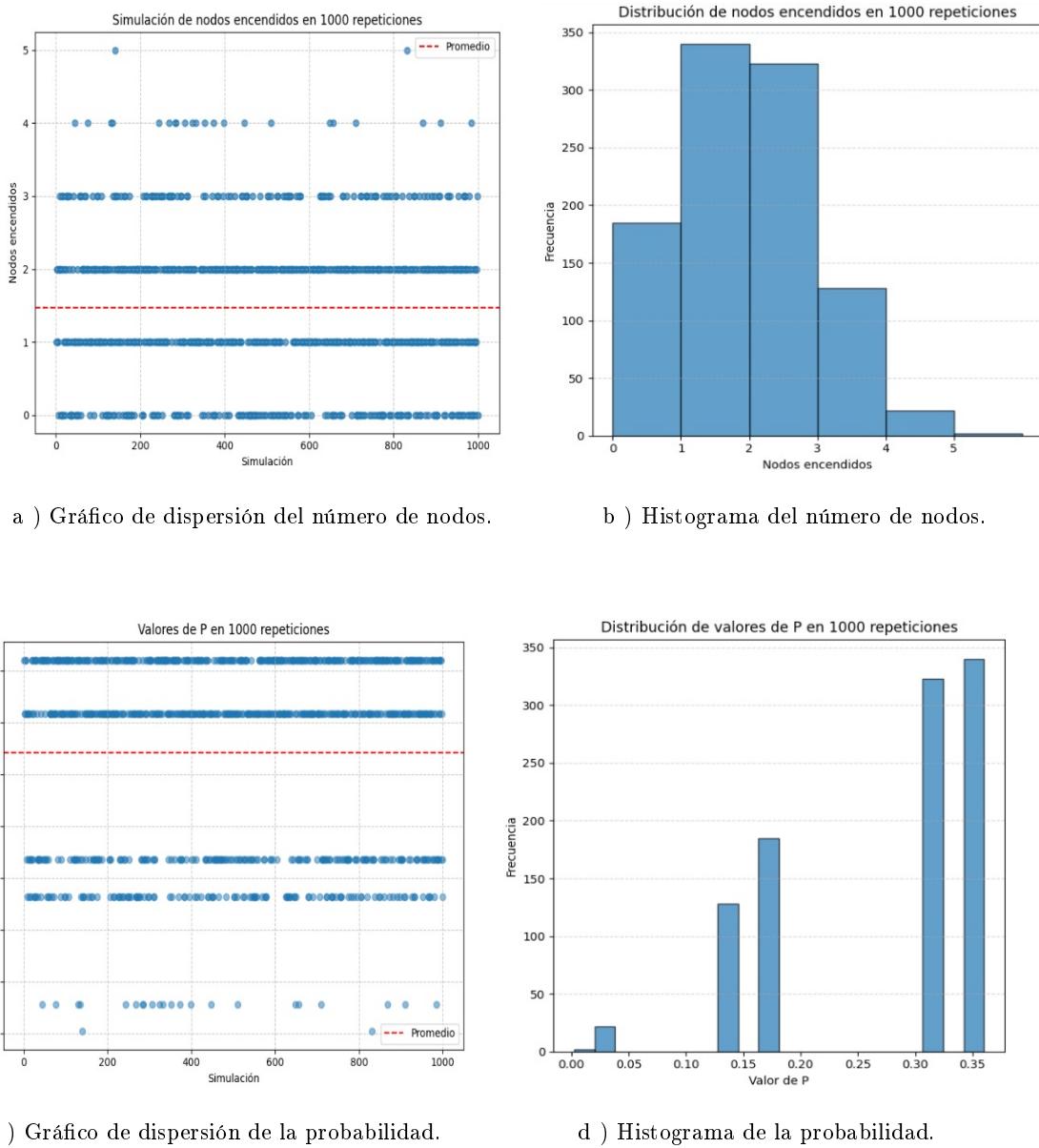


Figura 11.83: Resultados gráficos de varias simulaciones de la etapa 3 de la implementación del MMPP.

Pudiendo observar que es posible caer en cualquier número de nodos dentro del número máximo de nodos establecido. Sin embargo, se destaca que existe una baja probabilidad

de que el número de nodos encendidos sea grande. Esta observación sugiere que, aunque es posible que el sistema tenga un alto número de nodos encendidos, es menos probable en comparación con tener un número más bajo de nodos encendidos.

4. Consumo energético del nuevo número de nodos:

En esta etapa de implementación, se extiende el programa desarrollado en la sección anterior agregando el cálculo de cuánta energía se consumiría para realizar la transición al nuevo número de nodos. Esto se logra mediante la inclusión de la variable de energía en el cálculo del consumo total de energía, utilizando la fórmula de Δ de los estados 7 y 8 vistos en los *Estados del sistema*. Teniendo el siguiente resultado para una simulación:

```
Se prenden 3 nodos
El valor de P es: 0.1323
La energía consumida es: 3.2 unidades de energía
```

Figura 11.84: Resultado de una simulación de la etapa 4 de la implementación del MMPP.

Realizando un mayor número de simulaciones del cálculo de la energía consumida y obteniendo su promedio se presentan los resultados mostrados en las siguientes gráficas de dispersión y distribución:

```
En 1000 simulaciones mayormente se suele prender : 1 nodo
Con su respectiva P de: 0.2716
Consumiendo: 1.8347 unidades de energía
```

Figura 11.85: Resultado de varias simulaciones de la etapa 4 de la implementación del MMPP.

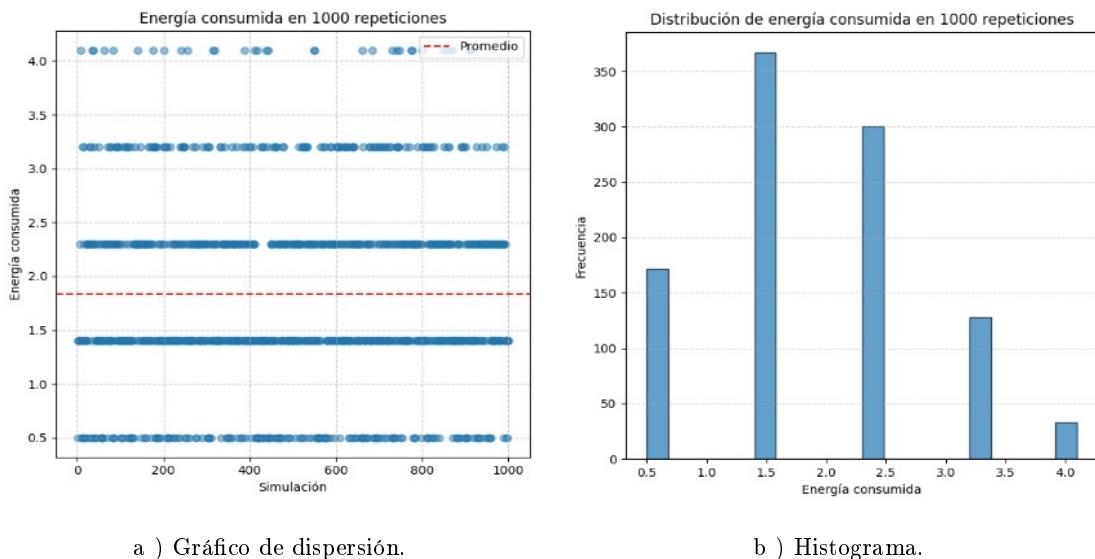


Figura 11.86: Resultados gráficos de varias simulaciones de la etapa 4 de la implementación del MMPP.

5. Simulación del tiempo de vida con energía y actualización de nodos encendidos:

En esta etapa, se juntan las etapas 4.2 y 4.4 para simular el tiempo de vida del sistema considerando tanto la variación en la energía por realizar la transición a los diferentes estados del sistema como por el nuevo número de nodos encendidos.

En esta etapa, se fusionan los conceptos de la sección 4.2, que trata sobre la simulación del tiempo de vida considerando los estados 1 a 6, con los de la sección 4.4, que se enfoca en el cálculo del siguiente número de nodos, su probabilidad de transición y su consumo de energía. Teniendo el siguiente resultado para una simulación:

Tiempo total de vida del sistema: 29.7 unidades de tiempo

Figura 11.87: Resultado de una simulación de la etapa 5 de la implementación del MMPP.

Realizando el mismo proceso un mayor número de simulaciones para obtener un tiempo



de vida promedio, se obtuvieron los siguientes resultados:

Tiempo promedio de vida del sistema en 1000 simulaciones: 39.97 unidades de tiempo

Figura 11.88: Resultado de varias simulaciones de la etapa 5 de la implementación del MMPP.

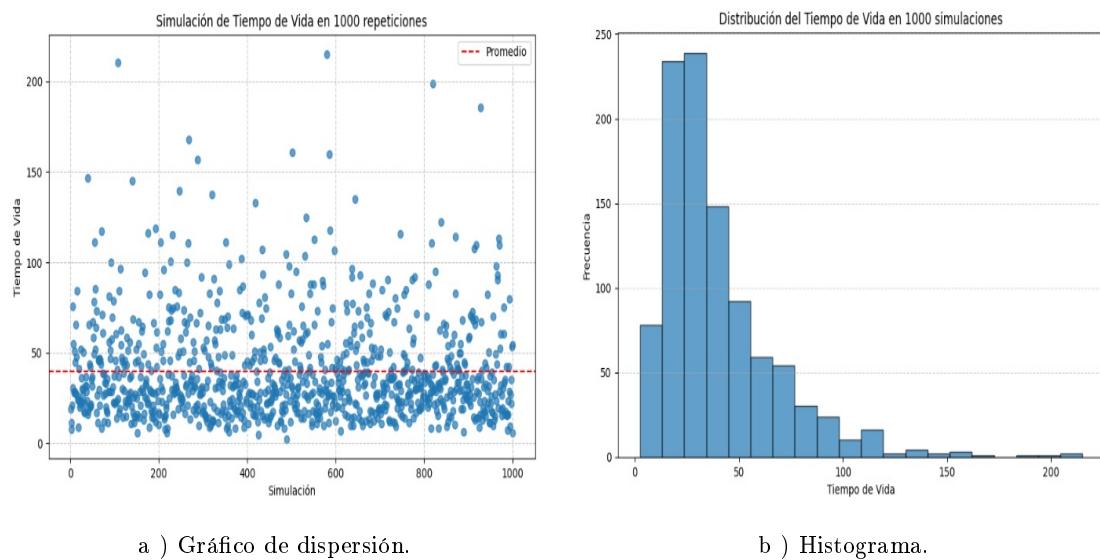


Figura 11.89: Resultados gráficos de varias simulaciones de la etapa 4 de la implementación del MMPP.

6. Superficie de Tiempo Promedio de Vida (E_{\max} vs N_{\max}):

En esta etapa, se realizó un análisis de la superficie de tiempo promedio de vida en función de los parámetros E_{\max} (energía máxima) y N_{\max} (número máximo de nodos). Realizando múltiples simulaciones variando estos parámetros y calculando el tiempo promedio de vida del sistema para cada combinación, como se puede ver a continuación:

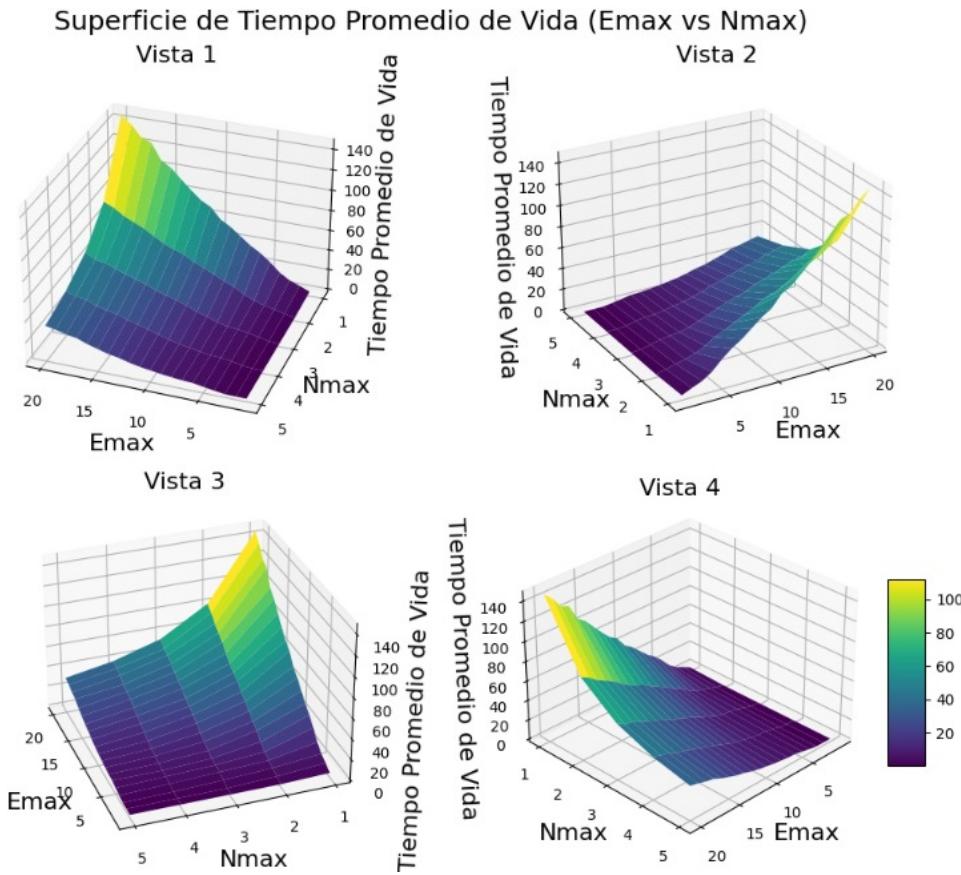


Figura 11.90: Diferentes vistas del gráfico de superficie E_{\max} vs N_{\max} .

El gráfico muestra la superficie tridimensional generada a partir de los valores de E_{\max} , N_{\max} y el tiempo promedio de vida. Se utilizó la librería Matplotlib para visualizar esta superficie, con cuatro subfiguras que muestran la superficie desde diferentes ángulos de vista para proporcionar una comprensión completa de cómo varía el tiempo promedio de vida en función de estos parámetros.

Al visualizar esta superficie, se puede entender mejor cómo varía el tiempo de vida del sistema en respuesta a diferentes niveles de energía máxima y número de nodos máximo. Esto proporciona información valiosa para la optimización y el diseño del sistema en función de estos parámetros.



7. Superficie de Tiempo Promedio de Vida (E_{on} vs E_{sleep}):

Parecido a la etapa anterior, en esta etapa, se analiza la influencia de los parámetros E_{on} (energía consumida cuando los nodos están encendidos) y E_{sleep} (energía consumida cuando los nodos están en modo sleep) con el tiempo promedio de vida del sistema mediante la construcción de otra gráfica de superficie, mostrada a continuación:

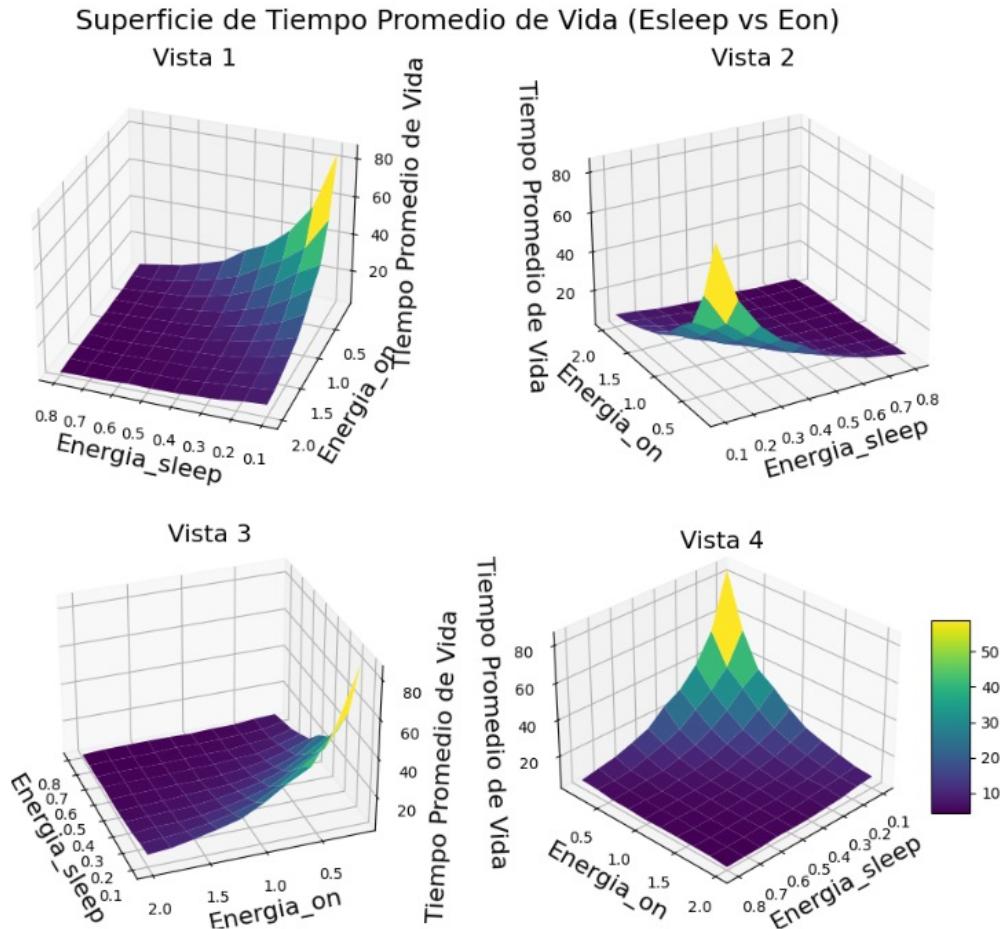


Figura 11.91: Diferentes vistas del gráfico de superficie E_{on} vs E_{sleep} .

Estos resultados ofrecen una visión detallada de cómo cambia el rendimiento del sistema en función de los niveles de consumo de energía durante los estados de reposo y encendido.

11.4. Creación de la página Web

Para esta tarea la estamos dividiendo en varias partes, empezaremos describiendo la arquitectura final para el proyecto junto con todas las configuraciones necesarias para “levantar” el servicio en la nube.

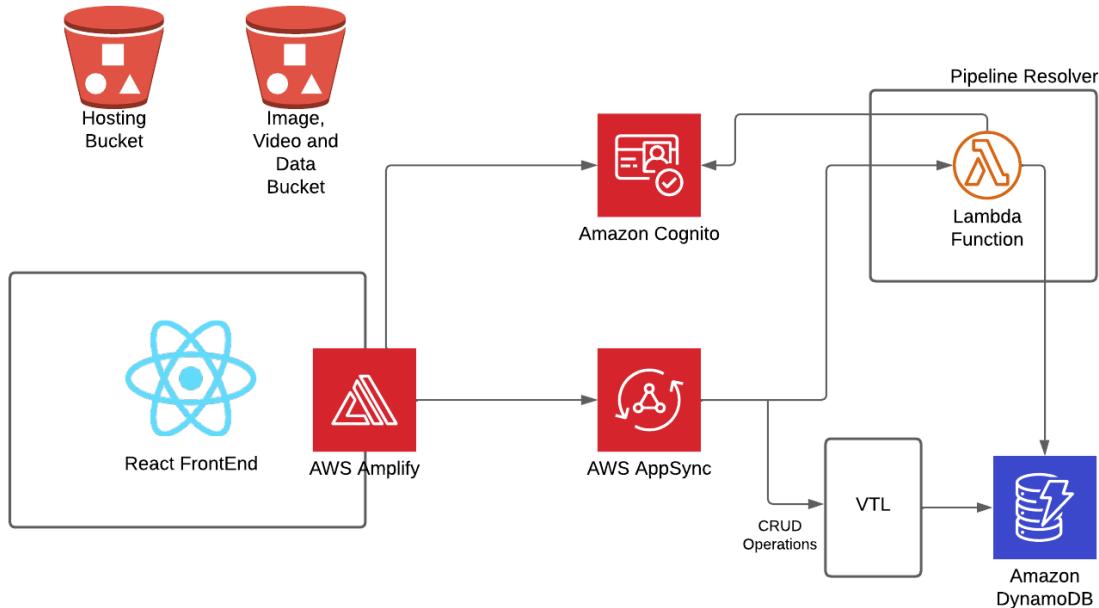


Figura 11.92: Diagrama de arquitectura del subsistema de almacenamiento.

La creación de la Aplicación Web consta de varios módulos que se muestran en el diagrama de la Figura ???. La aplicación web se va a construir con AWS Amplify que nos ayudará a crear el sitio y todo lo requerido para la administración y configuración de la página web, Amplify trabaja con el framework de React para el desarrollo de toda la configuración FrontEnd y BackEnd y toda la información se almacenará en el *Hosting Bucket*, también, se contará con un Bucket para almacenar toda la información multimedia requerida. Debido a la necesidad de contar con seguridad para que no cualquier persona pueda editar y visualizar los datos, se hace uso del servicio de Cognito, el cual dará acceso a toda la información de la aplicación web.



Toda la lógica de negocios del proyecto radica en otros 3 módulos de los cuales consta toda la programación, que son AppSync que funciona con GraphQL para la conexión del backend y que tiene operaciones CRUD (Create, Read, Update and Delete). Una VTL (Velocity Template Language) que son un conjunto de plantillas que otorgan una forma limpia, simple y fácil de incorporar contenido a una página web [?], en otras palabras, toma la solicitud como entrada y genera un documento JSON que contiene las instrucciones para el resolver. VTL puede utilizar plantillas de asignación para instrucciones sencillas, como pasar argumentos de campos GraphQL, o para instrucciones más complejas, como realizar un bucle a través de los argumentos para construir un elemento antes de insertarlo en DynamoDB.

Los *pipeline resolvers* contienen una o varias funciones que se ejecutan en orden secuencial. Cada función incluye una plantilla de solicitud y una plantilla de respuesta. Un pipeline resolver también tiene una plantilla **before** y una plantilla **after** que rodean la secuencia de funciones que contiene la plantilla. La plantilla **before** se ejecuta antes de cualquier función para preparar y transformar los datos de la solicitud inicial, mientras que la plantilla after corresponde al tipo de salida de campo GraphQL. Un resolvedor de canalización puede asignarse a cualquier salida que desee, incluida la entrada para otra función o la plantilla posterior del resolvedor de canalización[?].

Seguridad y Permisos

En la primera parte se crean dos usuarios en donde se le otorgan los permisos de administrador.

Usuarios (2) <small>Información</small>										
Un usuario de IAM es una identidad con credenciales válidas a largo plazo que se utiliza para interactuar con AWS en una cuenta.										
<input type="button" value="C"/> <input type="button" value="Eliminar"/> <input type="button" value="Crear usuario"/>										
<input type="button" value="Buscar"/>										
Nombre de usuario	Ruta	Grupo:	Última actividad	MFA	Antigüedad de	Último inicio de sesión	ID de clave de acceso	Antigüedad de la		
admin-chris	/	0	-	-	-	-	Active - AKIAURENC6U...	6 minutos		
admin-pablo	/	0	-	-	3 minutos	-	-	-		

Figura 11.93: Creación de los usuarios administradores.

Una vez que creados los usuarios, se procede a dirigirse al IDE VisualStudio Code y añadimos seguridad al Bucket, para esto , en la consola del IDE se coloca el comando: *amplify add auth*

Con ese comando, se elegirá la autenticación deseada y se añadirá la autenticación por email.

```
PS C:\Users\Chris\proyecto_PT> amplify add auth
  Using service: Cognito, provided by: awscloudformation

  The current configured provider is Amazon Cognito.

  Do you want to use the default authentication and security configuration? Default configuration
  Warning: you will not be able to edit these selections.
  How do you want users to be able to sign in? Username
  Do you want to configure advanced settings? No, I am done.
  Successfully added resource proyectoPT37912305 locally

  Some next steps:
  "amplify push" will build all your local backend resources and provision it in the cloud
  "amplify publish" will build all your local backend and frontend resources (if you have hosting cat
  (node:10308) NOTE: The AWS SDK for JavaScript (v2) will enter maintenance mode
  on September 8, 2024 and reach end-of-support on September 8, 2025.

  Please migrate your code to use AWS SDK for JavaScript (v3).
  For more information, check blog post at https://a.co/cUPnyil
  (Use `node --trace-warnings ...` to show where the warning was created)
```

Figura 11.94: Configuración por usuario y contraseña.

Creación del S3 Bucket para los archivos multimedia

Se utiliza el siguiente comando para añadir almacenamiento a al servicio: *amplify add storage*.

```
PS C:\Users\Chris\proyecto_PT> amplify add storage
? Please select from one of the below mentioned services: Content (Images, audio, video, etc.)
? Please provide a friendly name for your resource that will be used to label this category in the project: almacenamientoMultimedia
? Please provide bucket name: proyectoPTc9cb5850e864c2eb4ab8042cd12954
? Who should have access: Auth and guest users
? What kind of access do you want for Authenticated users? create/update, read, delete
? What kind of access do you want for Guest users? read
? Do you want to add a Lambda Trigger for your S3 Bucket? No
  Successfully updated auth resource locally.
  Successfully added resource almacenamientoMultimedia locally
```

Figura 11.95: Configuración del servicio de almacenamiento.

Con el comando de añadir almacenamiento, se indica que se pueden contener imágenes, audios, videos y cualquier archivo que se desee almacenar. Además, se configura para que tanto los



miembros autenticados como los visitantes puedan visualizar la información. La diferencia es que los miembros, que serán añadidos a un grupo con reglas específicas, podrán crear, actualizar, leer y eliminar los datos, mientras que los usuarios invitados únicamente podrán ver la información.

Creación de las funciones Lambda

Las funciones Lambda ayudarán a automatizar los procesos de almacenamiento en la base de datos de DynamoDB y a almacenar archivos en el Bucket. Para crear una función Lambda, se utiliza el siguiente comando: *amplify add function*.

```
PS C:\Users\Chris\proyecto_PT> amplify add function
? Select which capability you want to add: Lambda function (serverless function)
? Provide a friendly name for your resource to be used as a label for this category in the project: subirElementos
? Provide the AWS Lambda function name: subirElementos
? Choose the runtime that you want to use: NodeJS
? Choose the function template that you want to use: Hello World
? Do you want to access other resources in this project from your Lambda function? No
? Do you want to invoke this function on a recurring schedule? No
? Do you want to configure Lambda layers for this function? No
? Do you want to edit the local lambda function now? No
Successfully added resource subirElementos locally.
```

Figura 11.96: Creación de funciones Lambda.

Añadiendo GraphQL API

El servicio de GraphQL API permite al FrontEnd comunicarse con el BackEnd. Con el siguiente código: *amplify add api*.

```
PS C:\Users\Chris\proyecto_PT> amplify add api
? Please select from one of the below mentioned services: GraphQL
? Provide API name: proyectoPT
? Choose the default authorization type for the API Amazon Cognito User Pool
Use a Cognito user pool configured as a part of this project.
? Do you want to configure advanced settings for the GraphQL API Yes, I want to make some additional changes.
? Configure additional auth types? Yes
? Choose the additional authorization types you want to configure for the API API key
API key configuration
? Enter a description for the API key: Dar acceso a invitados a archivos.
? After how many days from now the API key should expire (1-365): 100
? Configure conflict detection? No
? Do you have an annotated GraphQL schema? No
? Do you want a guided schema creation? Yes
? What best describes your project: Single object with fields (e.g., "Todo" with ID, name, description)
? Do you want to edit the schema now? Yes
```

Figura 11.97: Creación del GraphQL.



Una vez que se haya creado toda la lógica y se haya completado la arquitectura, los archivos se subirán utilizando el comando: *amplify push*.

```
PS C:\Users\Chris\proyecto_PT> amplify push
- Fetching updates to backend environment: dev from the cloud.(node:4036) NOTE: The AWS SDK for JavaScript (v2) will enter maintenance mode on September 8, 2024 and reach end-of-support on September 8, 2025.

Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check blog post at https://a.co/cUPnyil
(Use `node --trace-warnings ...` to show where the warning was created)
✓ Successfully pulled backend environment dev from the cloud.

Current Environment: dev

| Category | Resource name | Operation | Provider plugin |
| --- | --- | --- | --- |
Auth | proyectopt37912305 | Create | awscloudformation |
Storage | almacenamientoMultimedia | Create | awscloudformation |
Function | subirElementos | Create | awscloudformation |
Api | proyectopt | Create | awscloudformation |

? Are you sure you want to continue? (Y/n) kkk[]
```

Figura 11.98: Archivos que se crearán.

```
Do you want to generate code for your newly created GraphQL API Yes
Choose the code generation language target javascript
Enter the file name pattern of graphql queries, mutations and subscriptions src\graphql\**\*.js
Do you want to generate/update all possible GraphQL operations - queries, mutations and subscriptions Yes
Enter maximum statement depth [increase from default if your schema is deeply nested] 4
```

Figura 11.99: Opciones para creación automatizada de código.

Base de Datos

Los datos que se almacenará en la base de datos son:

- **ID:** ID!
- **Tipo de archivo:** String
- **Fecha de captura del archivo:** String
- **Fecha de subida del archivo:** String
- **Persona que subió el archivo:** String
- **Animal:** String



- **URL del archivo:** String

Página web

La página web tendrá el siguiente diseño:

The wireframe shows a login form with a teal header and footer. The main area has a white background. It features a title 'Inicio de sesión' at the top center. Below it are two input fields: one labeled 'Nombre' with the placeholder 'Nombre' and another labeled 'Contraseña' with the placeholder 'Contraseña'. At the bottom is a blue rectangular button labeled 'Iniciar sesión'.

Figura 11.100: Inicio de sesión.



Figura 11.101: Pestaña de Inicio.

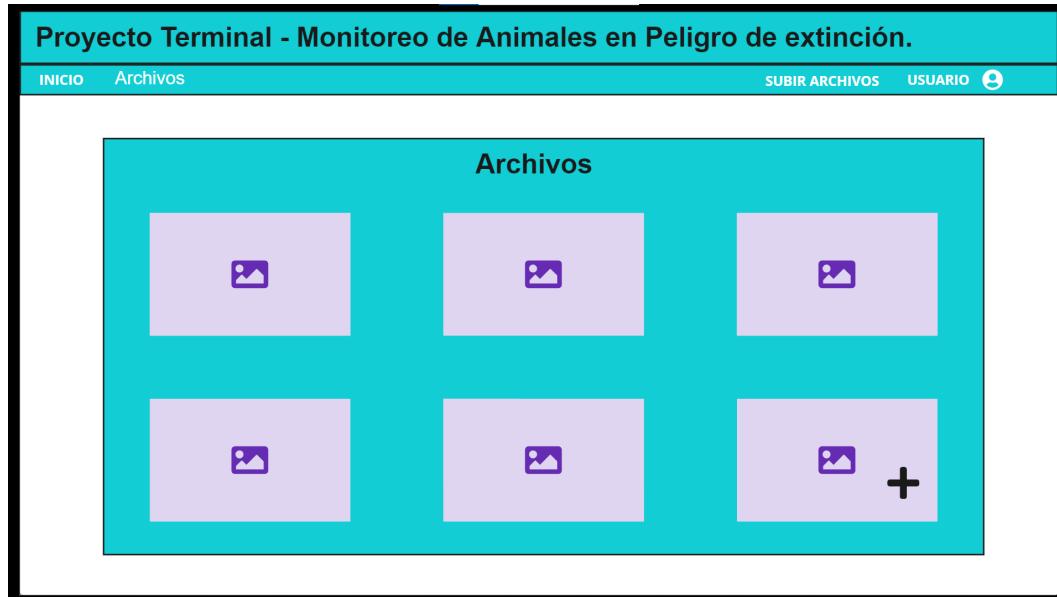


Figura 11.102: Pestaña de Visualización de archivos.



11.5. Integración de la página web.

El prototipo de la aplicación Web se hizo bajo el framework de React, junto TypeScript, NodeJS y varias bibliotecas de diseño como lo son React Bootstrap, Para empezar la integración de todos los componentes de BackEnd, FrontEnd y Base de Datos, hacemos uso de una herramienta que nos ayuda demasiado para el desarrollo web, que se encarga principalmente de hacer las configuraciones de conectividad entre autorización (inicios de sesión por registro en la aplicación), base de datos NoSQL como lo es DynamoDB, el almacenamiento en un Bucket de S3 y gestiona todo el BackEnd para la conectividad entre estos servicios, también nos da una forma de “Hosting” de la aplicación web. Gen 2 es la herramienta que utilizamos a lo largo del proyecto, ya que para comenzar a utilizar el servicio, únicamente debemos clonar la plantilla que nos aparece en la página de inicio para utilizar Amplify Gen 2 [1] que está basado totalmente en TypeScript y una biblioteca de diseños[2] para tener una estética de la página agradable al usuario La página de inicio de Amplify nos ofrece una plantilla con la integración de seguridad (autenticación), conexión con el sitio web, almacenamiento y base de datos; se comienza con una aplicación sencilla de una lista de tareas, la cual tenemos que eliminar y empezar a construir nuestro sitio web, únicamente incluyendo nuestros comandos para la implementación de los servicios, los cuales iremos describiendo en el proceso. Para la Base de Datos, se modificó las tablas que contienen la información, las cuales cuentan con:

- Se le configura la dirección en donde se almacena el archivo subido.
- El nombre de quién lo subió.
- La descripción, en este espacio se recomienda describir todos los datos importantes, sobre la zona en la que se ubicó la cámara, algunas anotaciones y observaciones o algún tipo de información relevante sobre el archivo que se va a subir.
- Por defecto, la base de datos ya cuenta con 2 campos interesantes que aprovechamos y nos son de mucha ayuda, que es la parte de creación (upload) del archivo y de modificación del archivo.



Esta información se envía y recibe con los siguientes comandos respectivamente:

```
client.models.function.create(...); client.models.function.list();
```

La función list nos arroja todos los datos en formato .JSON. Para el almacenamiento en S3, se tiene que seguir una serie de pasos, ya que por defecto, no viene con esta función. Para crear el almacenamiento, en la siguiente dirección de archivos dentro del proyecto **amplify/storage/resource.ts**, vamos a añadir las líneas de código para el almacenamiento.

```
import defineStorage from '@aws-amplify/backend'; export const storage = defineStorage(
  name: 'amplifyTeamDrive');
```

Importamos la biblioteca del backend y nombramos nuestro Bucket. En el directorio **amplify/backend.ts**, añadiremos únicamente la línea de “*Storage*” y ya tendríamos nuestro almacenamiento creado. Cabe recalcar el hecho de que, una vez configurado nuestro FrontEnd junto con los comandos de integración de los servicios ya mencionados, nuestro proyecto se sube en el repositorio de GitHub, Amplify tiene la gran facilidad de que cuando detecta los cambios de con el comando “push”, este automáticamente empieza la implementación de todos los servicios que le administramos a toda nuestra lógica, por lo que no hay que configurar nada más.

Elementos devueltos (4)						
	id (Cadena)	__typename...	content	createdAt	isDone	updatedAt
<input type="checkbox"/>	8966212f-7296-4021...	Todo	Prueba 1	2024-06-1...	false	2024-06-10T03:56:58.020Z
<input type="checkbox"/>	eced3127-336e-4811...	Todo	Prueba 2	2024-06-1...	false	2024-06-10T03:51:25.481Z
<input type="checkbox"/>	349ef4c5-6dc3-494d-...	Todo	Prueba de a...	2024-06-1...	false	2024-06-10T03:56:42.827Z
<input type="checkbox"/>	47a45711-254d-42b...	Todo	Donde esta ...	2024-06-1...	false	2024-06-10T03:58:07.119Z

Figura 11.103: Pruebas de almacenamiento de la base de datos.

<input type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	0224a3e5097427da3f4460b5d6d2b86ad4195c9cb8da7bc13e7187ead1c3ae3.vtl	vtl	6 Jun 2024 10:52:30 AM -06	772.0 B	Estándar
<input type="checkbox"/>	02942b52e6bfbac222f38b518d59907be00b68793c7e0ff87dac75ac45e02.json	json	9 Jun 2024 9:41:08 PM -06	27.2 KB	Estándar
<input type="checkbox"/>	06db846fd14e6fc371f22b12b5545ba8e2dbfedaa85d8c8d586c71c282166657b.vtl	vtl	6 Jun 2024 10:52:30 AM -06	350.0 B	Estándar
<input type="checkbox"/>	08f4d557693d96c1a4efba0f9dc91330e4b19772fd5477c156468843e5d9cb5e.vtl	vtl	6 Jun 2024 10:52:30 AM -06	1.2 KB	Estándar

Figura 11.104: Pruebas de almacenamiento de la base de datos.

11.5.1. Trabajo futuro.

Aunque la aplicación es muy sencilla y funcional, hay mucho trabajo a futuro sobre el rediseño de los módulos que implementamos, así como mayor diseño y uso de las librerías; la parte de autorización también es un tema muy complejo y completo que se le debe tomar mucha importancia, para este proyecto únicamente se toman las medidas de seguridad básicas, que es aplicar un login con usuario y contraseña, sin embargo, se planea modificar todo el inicio de sesión y modificación de información de los usuarios y la personalización de los mismos.

Agregar más módulos y servicios a este proyecto también queda como proyecto a futuro, ya que AWS cuenta con una gran cantidad de servicios que puede ayudar a los expertos en biología a realizar con facilidad sus estudios, recordando que este proyecto se enfrasca en un peldaño para el desarrollo de tecnología en el ámbito de monitoreo animal.

11.6. Integración de los Elementos

Después de haber verificado el funcionamiento de los componentes de manera individual se llevará a cabo su integración. Explorando cómo almacenar los datos de temperatura y GPS en una tarjeta SD para su posterior análisis y cómo utilizar la tecnología LoRa para la transmisión



eficiente de estos datos entre nodos de la red. A través de estos procesos de integración, se espera obtener una red de sensores completamente funcional y lista para su despliegue en el campo.

- **Almacenamiento de datos de temperatura en la tarjeta MicroSD:** Se almacenan los datos recopilados por el sensor de temperatura en la tarjeta MicroSD, mediante la conversión de los valores analógicos obtenidos del termistor a grados Celsius y guardándolos junto con una marca de tiempo en un archivo de texto en la tarjeta MicroSD, además se muestran estos datos en el Monitor Serial. El código realizado comienza inicializando la comunicación serial y configurando un pin para alimentar el termistor directamente, ya que el pin de VCC será utilizado para alimentar el módulo adaptador MicroSD, también se le configura el chip select y se conectan los demás pines como se vio en el Funcionamiento de los Elementos. Después, se inicializa la tarjeta MicroSD y se borra el archivo de temperatura en caso de ya existir. En el bucle principal, se lee el valor analógico del termistor, se convierte a temperatura en grados Celsius y se obtiene la marca de tiempo en formato HH:MM:SS iniciando desde el encendido del Arduino utilizando la función millis() de la librería TimeLib. Esta información se muestra en el Monitor Serial y se guarda en un archivo de texto en la tarjeta MicroSD en el siguiente formato: "temperatura; hora".

```
Output  Serial Monitor X
Message (Enter to send message to 'Arduino Leonardo' on 'COM13')
15:25:43.253 -> Temperatura: 28.84 °C; Hora: 00:01:31
15:25:43.253 -> Temperatura y hora guardadas en SD.
15:25:44.270 -> Temperatura: 28.84 °C; Hora: 00:01:32
15:25:44.270 -> Temperatura y hora guardadas en SD.
15:25:45.273 -> Temperatura: 29.81 °C; Hora: 00:01:33
15:25:45.315 -> Temperatura y hora guardadas en SD.
15:25:46.277 -> Temperatura: 29.81 °C; Hora: 00:01:34
15:25:46.361 -> Temperatura y hora guardadas en SD.
15:25:47.314 -> Temperatura: 29.81 °C; Hora: 00:01:35
```

a) Monitor Serial.

ARCHIVO: Bloc de notas

Archivo Edición Formato Ver Ayuda

Temperatura: 28.84 °C; Hora: 00:01:31
 Temperatura: 28.84 °C; Hora: 00:01:32
 Temperatura: 29.81 °C; Hora: 00:01:33
 Temperatura: 29.81 °C; Hora: 00:01:34
 Temperatura: 29.81 °C; Hora: 00:01:35

b) Archivo de texto.

Figura 11.105: Resultados de guardar la temperatura en la memoria MicroSD.

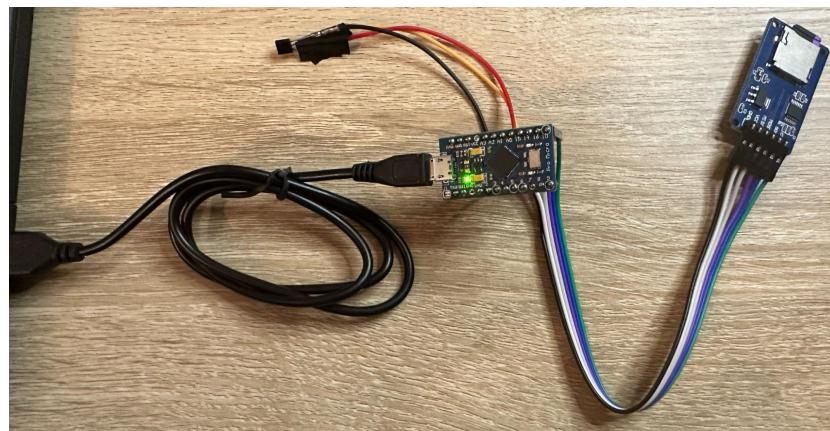


Figura 11.106: Integración del termistor y el módulo adaptador MicroSD.

- **Almacenamiento de datos GPS en la tarjeta MicroSD:** Se registran las coordenadas GPS en un archivo de texto en la tarjeta MicroSD utilizando un Arduino y un módulo GPS. Las coordenadas se registran junto con la hora actual en formato HH:MM:SS ajustada para la Ciudad de México (GMT-6). El código comienza inicializando la comunicación serial con el módulo GPS utilizando una comunicación de software serial a través de los pines RX y TX del Arduino. Luego, se configura la velocidad de comunicación del módulo GPS y se inicializa la tarjeta MicroSD. En el bucle principal, se lee continuamente la información del módulo GPS. Cuando se recibe un mensaje que comienza con "\$ GPRMC"(el mensaje de datos mínimos recomendados de GPS), se procesa para extraer la hora actual, la latitud y la longitud. Estos datos se convierten a formato decimal y se ajusta la hora al huso horario de la Ciudad de México (GMT-6). Luego, se crea una cadena con los datos de coordenadas y hora en el formato "latitud, longitud ; hora". Esta cadena se imprime en el Monitor Serial y se guarda en el archivo de texto en la tarjeta MicroSD.



Serial Monitor X

Message (Enter to send message to 'Arduino Leon')

Datos guardados en la tarjeta SD.
19.454790, -99.095749 ; 23:24:37

Datos guardados en la tarjeta SD.
19.454792, -99.095749 ; 23:24:38

Datos guardados en la tarjeta SD.
19.454792, -99.095749 ; 23:24:39

Datos guardados en la tarjeta SD.
19.454794, -99.095749 ; 23:24:40

GPS1: Bloc de notas

Archivo Edición Formato Ver Ayuda

19.454790, -99.095749 ; 23:24:37

19.454792, -99.095749 ; 23:24:38

19.454792, -99.095749 ; 23:24:39

19.454794, -99.095749 ; 23:24:40

a) Monitor Serial.

b) Archivo de texto.

Figura 11.107: Resultados de guardar la temperatura en la memoria MicroSD.

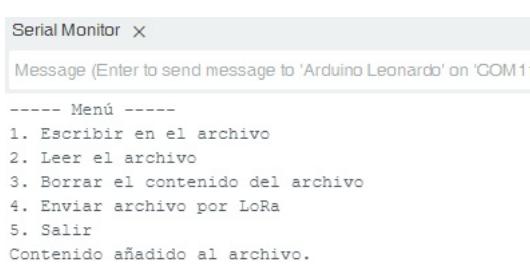


Figura 11.108: Integración del GPS y el módulo adaptador MicroSD.

- **Transmisión de archivos desde la tarjeta MicroSD mediante LoRa:** Este proyecto realiza un intercambio de archivos de texto entre dispositivos usando un Arduino Pro Micro, un módulo adaptador MicroSD y un módulo LoRa E22. El archivo de texto recibido se almacena en una tarjeta MicroSD y el usuario puede interactuar con el



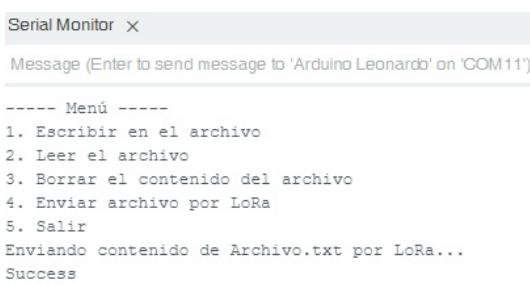
sistema mediante un menú en el monitor serial. El menú permite al usuario seleccionar entre diferentes opciones: escribir en el archivo, leer el archivo, borrar el contenido del archivo y enviar el archivo por LoRa. Dependiendo de la opción seleccionada, se ejecuta la función correspondiente. Durante la ejecución, el programa verifica continuamente si hay datos disponibles en el módulo LoRa. Si hay datos disponibles, estos se reciben y se almacenan en una variable. Los datos recibidos se muestran en el Monitor Serial para que el usuario pueda ver lo que se ha recibido y se guardan en un archivo txt en la tarjeta MicroSD. Si ocurre un error al intentar abrir el archivo, se notifica al usuario. Por otra parte, para enviar el contenido del archivo, primeramente, se abre el archivo, después se guardan los datos en una variable y se envían a través del módulo LoRa. De igual forma, se notifica al usuario sobre el éxito o el fracaso del envío.



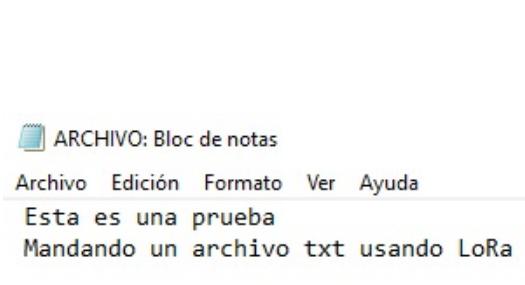
a) Escribir en el archivo.



b) Transmitir archivo.

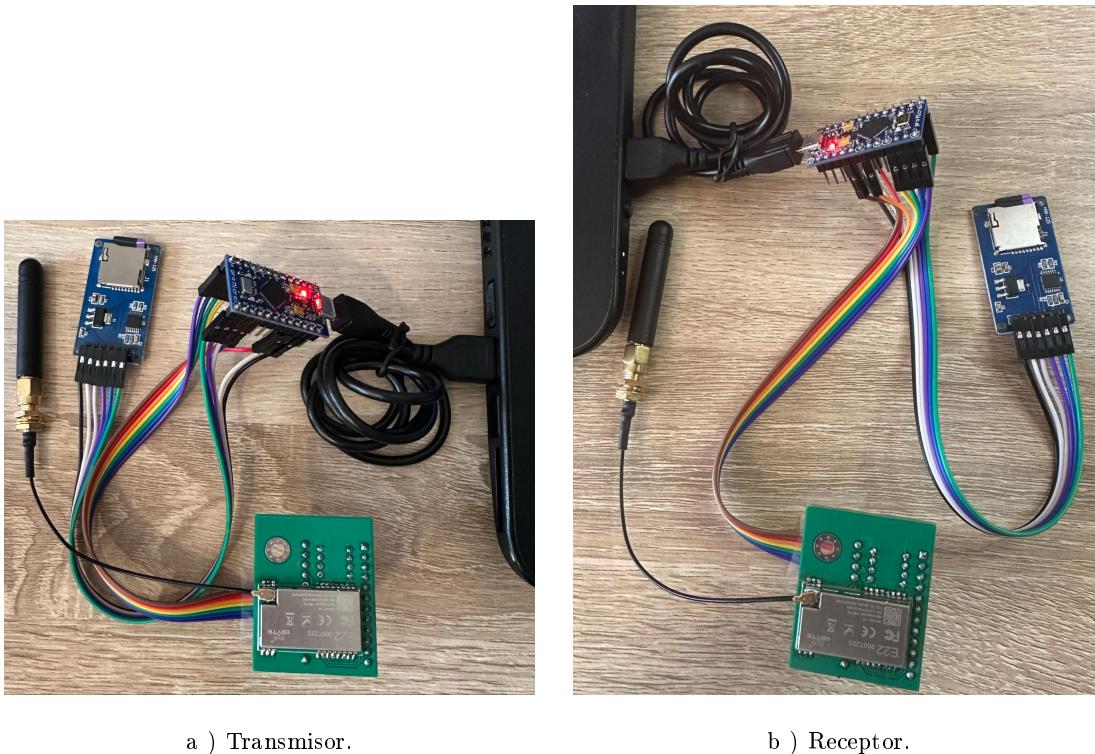


c) Leer archivo.



d) Archivo de texto recibido.

Figura 11.109: Resultados de transmitir el archivo txt de la memoria MicroSD usando LoRa.



a) Transmisor.

b) Receptor.

Figura 11.110: Integración del módulo adaptador MicroSD y el transceptor LoRa.

- **Almacenamiento de datos GPS en la tarjeta MicroSD con transmisión mediante LoRa:** Este proyecto permite registrar coordenadas GPS en una tarjeta MicroSD y transmitir estos datos a través de una transceptores LoRa. Utiliza un Arduino, un módulo GPS, un módulo adaptador MicroSD y un módulo LoRa E22 para realizar estas tareas. El sistema comienza con la inicialización de la tarjeta MicroSD. Si la tarjeta no se inicializa correctamente, se notifica al usuario y el programa se detiene. A continuación, se configura el módulo GPS para comunicarse a través de una conexión serial, ajustando la velocidad de comunicación del módulo GPS NEO-6M. Posteriormente, se inicializa el módulo LoRa E22, preparándolo para la transmisión de datos. El módulo GPS se utiliza para recibir datos de ubicación en formato NMEA, que incluye información sobre la latitud, longitud y hora. Estos datos se procesan para extraer las coordenadas y la hora



actual, convirtiendo las coordenadas a un formato decimal y ajustando la hora al huso horario de la Ciudad de México (GMT-6). Las coordenadas GPS y la hora se guardan en un archivo de texto en la tarjeta MicroSD, actualizándose cada vez que se reciben nuevos datos GPS. El sistema abre el archivo en modo de escritura, guarda los datos y luego cierra el archivo para asegurar que la información se almacene correctamente. Después, se define un intervalo de tiempo en el que se estarán guardando las coordenadas en la tarjeta y concluyendo este tiempo se iniciará la transmisión del archivo. Donde se lee el contenido del archivo y se envía a través del módulo LoRa. La transmisión se realiza en varios pasos: primero se envía un mensaje de inicio, seguido del nombre del archivo y luego cada línea de datos. Finalmente, se envía un mensaje de finalización para indicar que la transmisión ha terminado. Si la transmisión es exitosa, el archivo en la tarjeta MicroSD se borra para preparar el sistema para la próxima serie de datos. Aunque el sistema opera principalmente de forma autónoma, la comunicación con el usuario se realiza a través del Monitor Serial. El usuario puede ver mensajes que indican el estado de la inicialización, la lectura de datos, el almacenamiento y la transmisión. Ilustrando así cómo integrar múltiples tecnologías (GPS, MicroSD y LoRa) para crear un sistema de registro y transmisión de datos eficiente, ideal para aplicaciones que requieren seguimiento de ubicación y comunicación inalámbrica.

Output Serial Monitor X

Message (Enter to send message to 'Arduino Leonardo' on 'COM11')

```
19.4548,99.0957; 19:47:23
Datos guardados en la tarjeta SD.
19.4548,99.0957; 19:47:24
Datos guardados en la tarjeta SD.
19.4548,99.0957; 19:47:25
Datos guardados en la tarjeta SD.
Enviando contenido de gps1.txt por LoRa...
Success
Datos enviados por LoRa.
Archivo borrado correctamente.
19.4548,99.0957; 19:48:01
Datos guardados en la tarjeta SD.
```

(a) Transmisor

Serial Monitor X

Message (Enter to send message to 'Arduino Leonardo' on 'COM13')

```
Esperando señal de inicio...
Recepción de archivo finalizada.
Contenido de gps1.txt:
19.4548,99.0957; 19:47:23
19.4548,99.0957; 19:47:24
19.4548,99.0957; 19:47:25
```

(b) Receptor

Figura 11.111: Resultados de guardar las coordenadas en un archivo txt en la memoria MicroSD y transmitirlo usando LoRa.

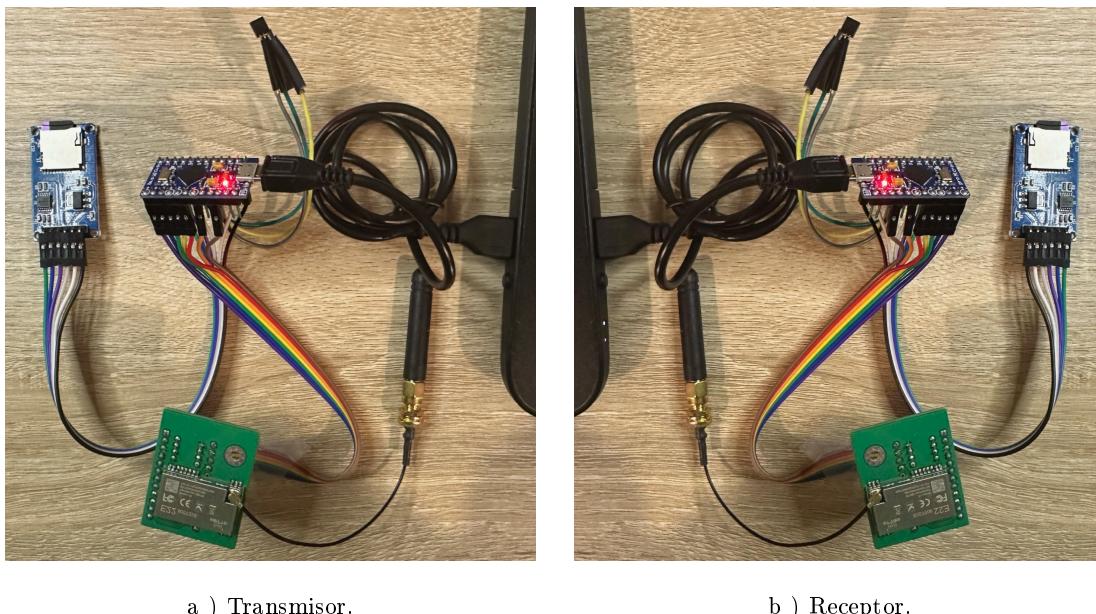


Figura 11.112: Integración del GPS, el módulo adaptador MicroSD y el transceptor LoRa.

- **Recepción y almacenamiento de datos GPS mediante LoRa junto con datos de temperatura en la tarjeta MicroSD:** En este proyecto, se establece el sistema de recepción de datos GPS mediante LoRa, el cual almacena estos datos recibidos en una tarjeta MicroSD con el nombre de archivo recibido, a la par que almacena datos de temperatura. Utilizando un Arduino Pro Micro, un módulo GPS, un termistor, un módulo LoRa E22 y una tarjeta MicroSD. Al iniciar, el sistema configura y alimenta el termistor, el módulo LoRa, y la tarjeta MicroSD. Si la inicialización del dispositivo de almacenamiento falla, se notifica al usuario y se detiene la ejecución para evitar errores posteriores. Una vez que todo está configurado correctamente, el sistema entra en el bucle principal donde se ejecutan dos tareas principales: la lectura y almacenamiento de la temperatura, y la recepción de archivos a través de LoRa. La primera tarea es la lectura de la temperatura. El sistema lee el valor analógico del termistor, convierte este valor a grados Celsius, y obtiene la hora actual utilizando la librería TimeLib. La temperatura



y la hora se imprimen en el Monitor Serial y se almacenan en un archivo en la tarjeta MicroSD. Este proceso se repite cada intervalo de tiempo establecido, proporcionando un registro continuo de la temperatura ambiental. La segunda tarea es la recepción de archivos a través de LoRa. El sistema espera mensajes del módulo LoRa, y cuando recibe un mensaje de inicio ("START"), procede a recibir el nombre del archivo. Una vez recibido el nombre, se crea y abre un archivo con ese nombre en la tarjeta MicroSD y se empiezan a recibir los datos del archivo. Los datos se almacenan línea por línea hasta que se recibe un mensaje de finalización (.END"). Si hay algún error en la recepción de los mensajes o al abrir el archivo, se notifica al transmisor para que no borre el archivo transmitido. Combinando de esta forma, la recepción de datos GPS mediante LoRa con el almacenamiento de datos de temperatura en una tarjeta MicroSD. Utilizando tecnologías de comunicación inalámbrica y almacenamiento digital, se crea una solución robusta para la recopilación y registro de datos ambientales y de ubicación, adecuada para diversas aplicaciones en monitoreo remoto y seguimiento de activos.



```
Serial Monitor X

Message (Enter to send message to 'Arduino Leonardo' on 'COM13')

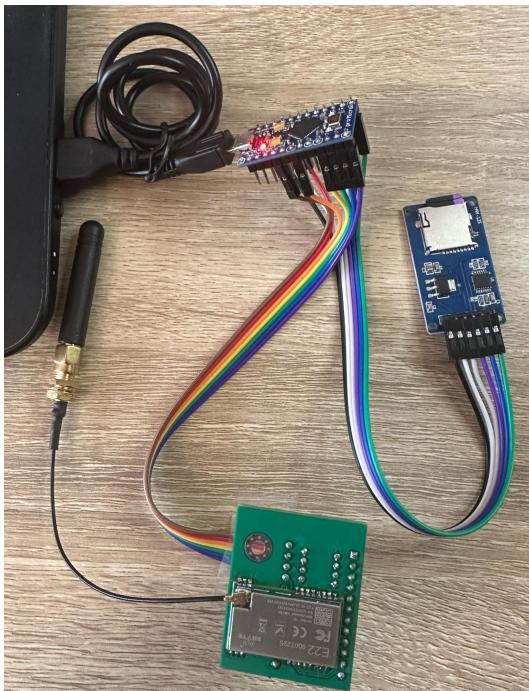
Temperatura: 33.72 °C; Hora: 00:05:52
Temperatura y hora guardadas en SD.
Temperatura: 33.72 °C; Hora: 00:05:54
Temperatura y hora guardadas en SD.
Inicio de recepción de archivo.
Nombre de archivo recibido: gps1.txt
Guardando archivo...
Fin de recepción de archivo.
Temperatura: 32.26 °C; Hora: 00:06:24
Temperatura y hora guardadas en SD.
```

(a) Monitor serial.

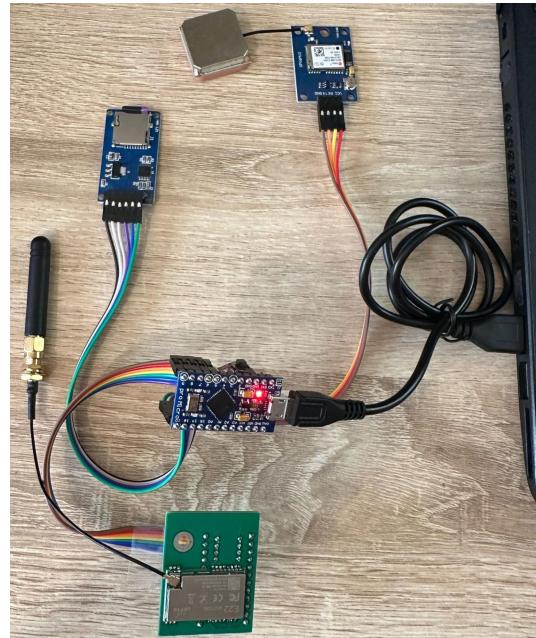
SD > SDHC (G:)			
Nombre	Fecha de modificación	Tipo	Tamaño
GPS1	01/01/2000 01:00 a. m.	Documento de te...	21 KB
TEMP	01/01/2000 01:00 a. m.	Documento de te...	35 KB

(b) Archivos generados.

Figura 11.113: Resultados de recibir las coordenadas usando LoRa, guardarlos en un archivo txt en la memoria MicroSD y recolectar datos de temperatura a la par



a) Transmisor.



b) Receptor.

Figura 11.114: Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa

- **Transmisión de datos GPS y de temperatura de la tarjeta MicroSD mediante LoRa:** En este proyecto, se establece un sistema para la transmisión de datos GPS y de temperatura contenidos en archivos de texto en una tarjeta MicroSD utilizando la tecnología LoRa. El sistema está compuesto por un Arduino Pro Micro, un termistor, un módulo LoRa E22, y una tarjeta MicroSD. La función principal es enviar los datos almacenados en la tarjeta MicroSD a través del módulo LoRa a un receptor remoto.

El programa comienza con la configuración inicial de los componentes. Se configuran los pines del módulo LoRa y de la tarjeta MicroSD. El módulo LoRa es configurado utilizando la biblioteca LoRa_E22, y la tarjeta MicroSD es inicializada utilizando la biblioteca SD. Si la tarjeta no se inicializa correctamente, se notifica al usuario y el



programa se detiene para evitar errores posteriores.

En el bucle principal, el sistema opera en tres modos principales de operación: recepción de archivos GPS por LoRa, medición y almacenamiento de temperatura, y transmisión de archivos GPS y de temperatura. Respecto a la recepción por LoRa, el Arduino monitorea continuamente la disponibilidad de mensajes mediante el módulo LoRa. Cuando se detecta un mensaje de inicio (“START”), se inicia la recepción del archivo correspondiente. Esto implica la espera y recepción del nombre del archivo a través de LoRa, seguido por la recepción secuencial de cada línea de datos. Cada línea de datos recibida se guarda directamente en un archivo con su correspondiente nombre en la tarjeta MicroSD, repitiendo esto hasta recibir un mensaje de finalización (“END”). Notificando al transmisor si ocurrió algún error en la recepción de los datos para anular la eliminación del archivo en el transmisor.

Por otro lado, el Arduino también realiza mediciones periódicas de temperatura a intervalos regulares establecidos. Durante cada ciclo de medición, el termistor conectado se activa para obtener una lectura analógica que representa la temperatura ambiental. Esta lectura se convierte a grados Celsius y se complementa con la hora actual en formato HH:MM. Los datos de temperatura junto con la marca de tiempo se registran de manera sistemática en un archivo específico (temp.txt) en la tarjeta SD. Además de la recepción y almacenamiento local de datos, lo agregado en este sistema es que también está diseñado para la transmisión periódica de sus archivos por LoRa cada cierto periodo de tiempo (mayor al de la medición de temperatura). Durante este proceso, el Arduino verifica la existencia de archivos predefinidos (gps1.txt, gps2.txt, gps3.txt y temp.txt) en la tarjeta MicroSD. Para cada archivo existente, se inicia una secuencia de transmisión por LoRa, comenzando con el envío del nombre del archivo y seguido por el envío de cada línea de datos de manera ordenada. Un mensaje “START” indica el inicio de la transmisión y un mensaje “END” señala su finalización exitosa. Después de completar una transmisión satisfactoria (recibiendo un mensaje de confirmación del receptor), el archivo correspondiente se elimina automáticamente de la tarjeta MicroSD para gestionar eficientemente



el espacio de almacenamiento y garantizar la continuidad en la recolección de datos.

```
Serial Monitor x Output

Message (Enter to send message to 'Arduino Leonardo' on 'COM13')

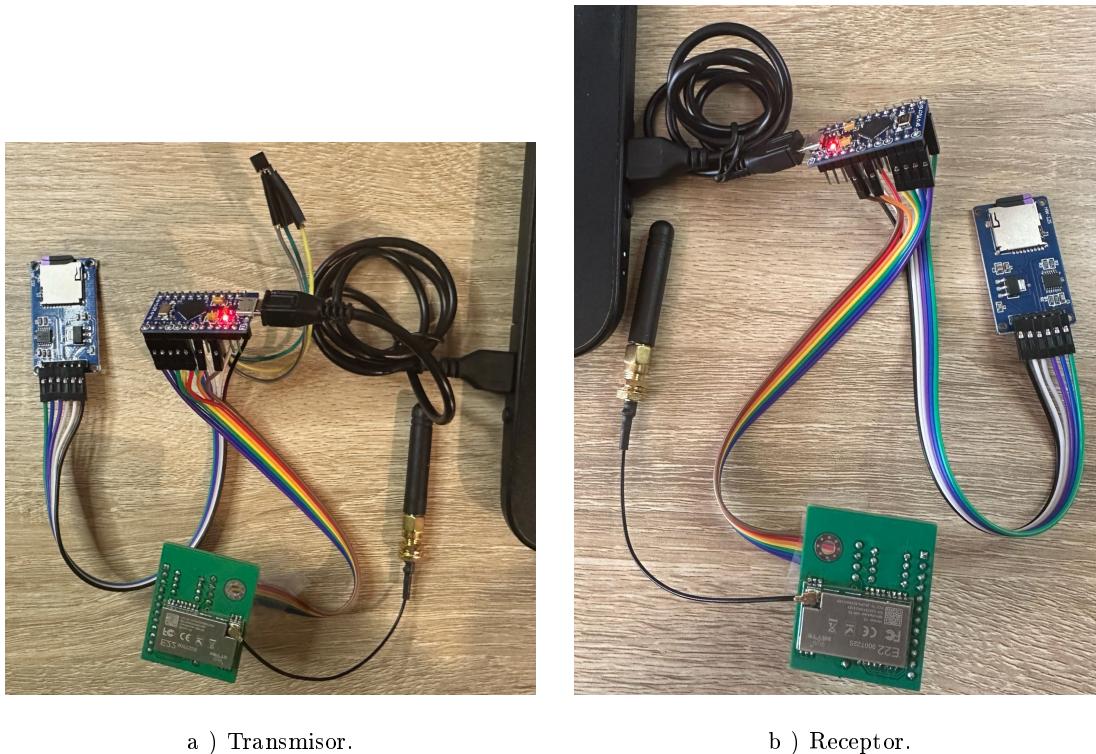
Inicio de recepción de archivo.
Nombre de archivo recibido: gps3.txt
Guardando archivo...
Fin de recepción de archivo.
Temperatura: 32.51 °C; Hora: 00:12:36
Temperatura y hora guardadas en SD.
Temperatura: 32.68 °C; Hora: 00:12:37
Temperatura y hora guardadas en SD.
Enviando contenido de gps1.txt por LoRa...
Success
Datos enviados por LoRa.
Enviando contenido de gps2.txt por LoRa...
Success
Datos enviados por LoRa.
Enviando contenido de gps3.txt por LoRa...
Success
Datos enviados por LoRa.
Enviando contenido de temp.txt por LoRa...
Success
Datos enviados por LoRa.
Temperatura: 33.12 °C; Hora: 00:17:09
Temperatura y hora guardadas en SD.
```

(a) Monitor serial.

SD > SDHC (G:)			
Nombre	Fecha de modificación	Tipo	Tamaño
GPS1	01/01/2000 01:00 a. m.	Documento de te...	106 KB
GPS2	01/01/2000 01:00 a. m.	Documento de te...	105 KB
GPS3	01/01/2000 01:00 a. m.	Documento de te...	104 KB
TEMP	01/01/2000 01:00 a. m.	Documento de te...	75 KB

(b) Archivos generados.

Figura 11.115: Resultados de recibir las coordenadas usando LoRa, recolectar datos de temperatura y transmitir estos archivos txt.



a) Transmisor.

b) Receptor.

Figura 11.116: Integración del GPS, el termistor, el módulo adaptador MicroSD y el transceptor LoRa

Con esta implementación integral de captura, almacenamiento local y transmisión remota de datos utilizando sensores, Arduinos, transceptores LoRa y tarjetas MicroSD, se concluye la configuración de la Red de Sensores Inalámbricos (WSN) propuesta para el monitoreo de especies en peligro de extinción. Este sistema no solo permite el registro en tiempo real de la temperatura ambiental y de coordenadas GPS, sino que también facilita la transmisión remota de datos a largas distancias. Esto garantiza el cumplimiento de las necesidades de monitoreo y conservación animal en entornos remotos y de difícil acceso.

CAPÍTULO 12

Resultados

12.1. INECOL

En el Instituto de Ecología tuvimos la oportunidad de trabajar con el Dr. Salvador Mandujano Rodríguez

12.2. Modelo de propagación



12.2.1. Pruebas en Simulación

Para la parte de análisis de simulación, estamos utilizando el software Mission Planner 1.3.81, en el software tenemos opciones sobre simulaciones de vuelo de un solo dron y de un enjambre de drones. Como se mencionó en los esquemas de vuelo, haremos las 2 pruebas y las analizaremos, para ello cargamos una misión, como se muestra en la figura 12.1:

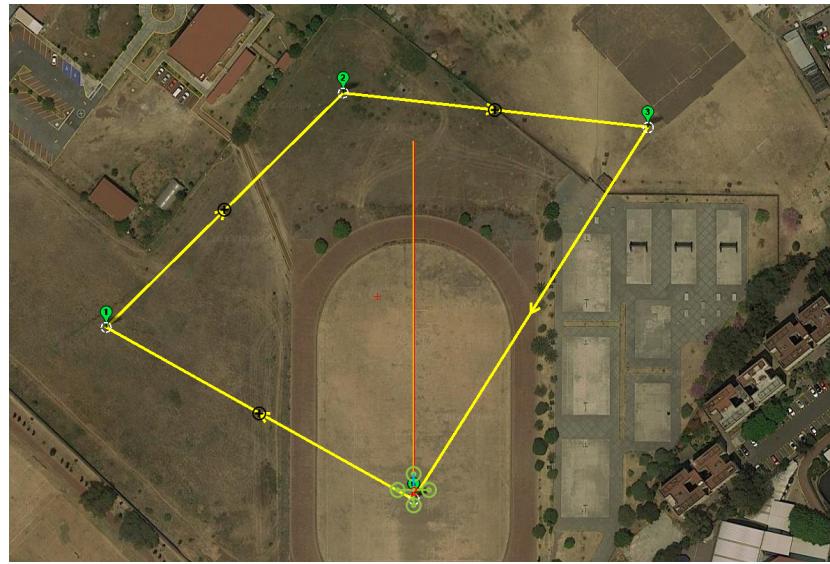


Figura 12.1: Misión de vuelo en el campo de UPIITA-UPIBI



Con las características mostradas en la figura 12.2:

	Command	P1	P2	P3	P4	Lat	Lon	Alt	Frame	Delete		Grad %	Angle	Dist	AZ
▷ 1	WAYPOINT	180	0	0	0	19.5134254	-99.1291505	30	Relative	✓		21.1	11.9	145.2	299
2	WAYPOINT	160	0	0	0	19.5142749	-99.1282386	30	Relative	✓		0.0	0.0	134.4	45
3	WAYPOINT	94	0	0	0	19.514151	-99.1270664	30	Relative	✓		0.0	0.0	123.6	96
4	RETURN_TO_LAUNCH	0	0	0	0	19.5131094	-99.1277799	30	Relative	✓		0	0	0	0
5	LAND	0	0	0	0	19.5128035	-99.1279677	0	Relative	✓		-16.9	-9.6	179.6	212

Figura 12.2: Características de la misión de vuelo.



Para todos tenemos una altura de 30 metros y en cada punto varía el "delay" (simulando que los drones están recibiendo la información captada), posteriormente regresan al unto "Home" y aterrizan en el mismo lugar de despegue.

Otra de las características que tiene es que vuelan a una velocidad de 14.4m/s y no es posible saber en la propia simulación cuanto es el peso de cada Dron, lo que en las pruebas reales puede variar mucho los resultados; cada Dron en la simulación cuenta con una batería de 13 Volts con corriente de 3300mAh, un módulo GPS RTK y una batería cargada al 100 % como se muestra en la figura 12.3:

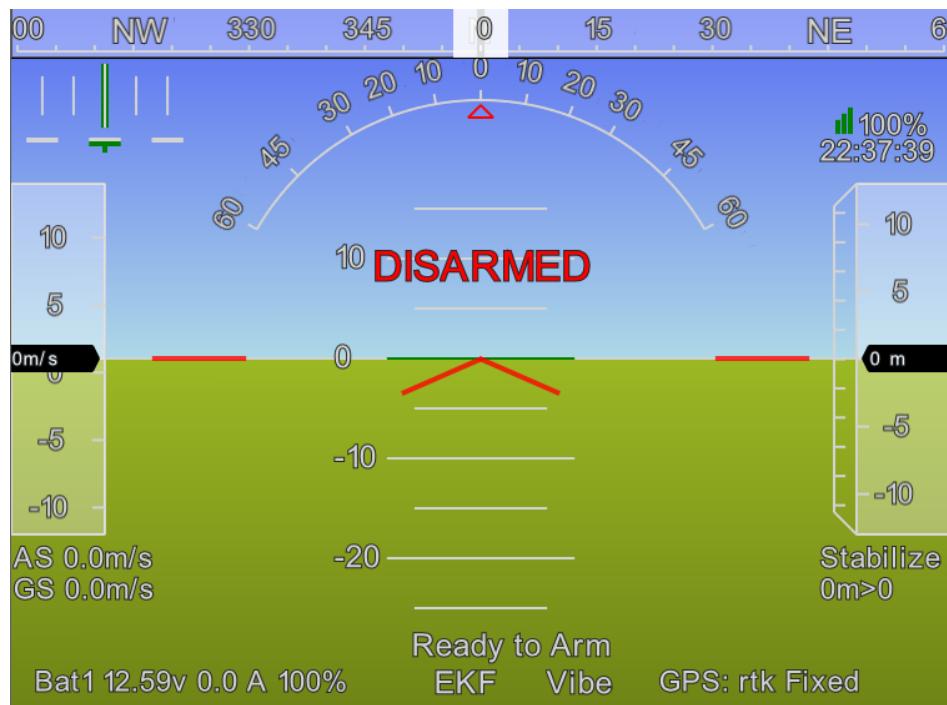


Figura 12.3: Pantalla de Vuelo del Dron en el Software de Mission Planner.



12.2.2. Esquema de vuelo: Enjambre de Drones.

Para esta primer simulación, haremos el esquema donde vuela el enjambre de Drones, volarán los 3 juntos y analizaremos su corriente a lo largo del trayecto.

Para los 3 drones tienen una corriente promedio de 24 A, y en completar la misión de la propuesta tardó un aproximado de 10 minutos.

Para el dron líder tenemos na corriente máxima de 32.2 A, mientras que para el Dron 2 y 3 son de 32.32 y 34.57 A. respectivamente. Sabemos que nos marca como Ampers porque en la figura 12.3 nos muestra las unidades en Ampers.



Figura 12.4: Oscilograma de la corriente del dron líder.



Figura 12.5: Oscilograma de la corriente del Dron 2.



Figura 12.6: Oscilograma de la corriente del Dron 3.



12.2.3. Esquema de Vuelo: Un Solo Dron.

Para esta parte, los drones viajarán a un waypoint individualmente, los parámetros de cada vuelo se describen a continuación con sus respectivas gráficas de corriente de cada vuelo.

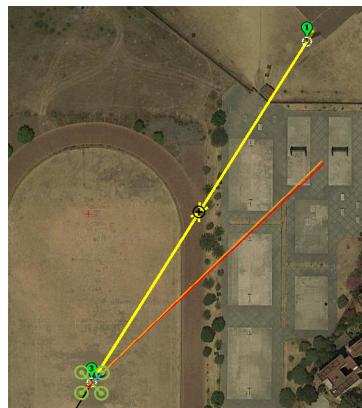


Figura 12.7: Misión de vuelo Dron 1.



Figura 12.8: Misión de vuelo Dron 2.

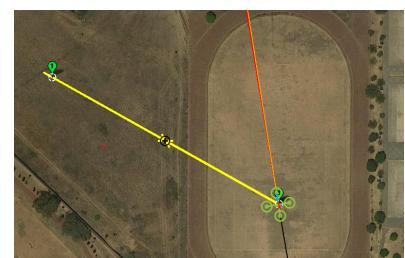


Figura 12.9: Misión de vuelo Dron 3.

Figura 12.10: Misiones de vuelo de los Drones 1, 2 y 3.

Y sus respectivas gráficas de corriente :



Figura 12.11: Oscilograma de misión de vuelo Dron 1.



Figura 12.12: Oscilograma de misión de vuelo Dron 2.

Resultados



Figura 12.13: Oscilograma de misión de vuelo Dron 3.

CAPÍTULO 13

Cronograma

Tabla 13.1: Cronograma de actividades PT1.

Actividad	Semanas																Actividad
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Análisis de los sensores en el mercado	●																Análisis de los sensores en el mercado
Selección de sensores		●	●	●													Selección de sensores
Búsqueda y vinculación con otros proyectos de investigación y/o organizaciones.			●	●	●	●	●										Búsqueda y vinculación con otros proyectos de investigación y/o organizaciones.
Análisis y selección de proveedor de servicios en la nube.				●	●	●											Análisis y selección de proveedor de servicios en la nube.
Analisis de la WSN					●	●	●										Analisis de la WSN
Analisis del animal a estudiar						●	●	●									Analisis del animal a estudiar
Elección del animal a estudiar							●	●									Elección del animal a estudiar
Diseño del dominio en la nube.								●	●	●							Diseño del dominio en la nube.
Creación del dominio en la nube.									●	●	●						Creación del dominio en la nube.
Diseño de la WSN									●	●	●						Diseño de la WSN
Adquisición de los sensores										●	●						Adquisición de los sensores
Diseño de la DB											●	●					Diseño de la DB

CAPÍTULO 14

Escenario de Pruebas

El escenario de pruebas seleccionado para la evaluación de nuestro protocolo es el campo deportivo de la UPIITA-UPIBI del IPN, ubicado en La Laguna Ticomán, Gustavo A. Madero, Ciudad de México. Este espacio es un área verde con acceso exclusivo para estudiantes, profesores y personal del IPN. Para la realización de las pruebas, se instalará una red de sensores inalámbricos (WSN) en una zona del campo. Los sensores recopilarán datos relacionados con variables ambientales como la temperatura, y datos de los animales como las coordenadas GPS y el video de las cámaras trampa. Estos datos permitirán monitorear el entorno y obtener información relevante para la evaluación del sistema.

Para obtener resultados más realistas y representativos, se simulará el comportamiento de una especie animal específica dentro del escenario de pruebas. Se utilizarán modelos de simulación basados en el conocimiento existente sobre la especie, su comportamiento y sus interacciones con el entorno. Estos modelos permitirán generar datos simulados que representen los patrones de movimiento, las interacciones y las preferencias de hábitat de la especie en cuestión.

Posteriormente, se desplegará un enjambre de drones para recolectar los datos de monitoreo



de la WSN. Durante el despliegue, los drones operarán bajo la coordinación del protocolo propuesto, el cual permitirá un mejor control de los drones y asegurará la eficiencia en la recolección de datos.

Además, se llevarán a cabo pruebas para evaluar la eficacia de la recolección de datos durante diferentes momentos del día y bajo diferentes condiciones climáticas, para simular posibles condiciones y evaluar el desempeño del sistema en diferentes escenarios. Durante estas pruebas, se medirán parámetros de desempeño específicos para evaluar el funcionamiento del sistema. Para medir y cuantificar los resultados de cada parámetro de desempeño, se utilizarán métricas específicas. Estas métricas podrían incluir el porcentaje de datos recopilados correctamente, la tasa de errores en la transmisión de datos y el tiempo promedio de respuesta. Las mediciones se llevarán a cabo siguiendo procedimientos establecidos, utilizando herramientas o software especializados y aplicando protocolos de medición adecuados.

Con esta selección, se busca simular un escenario que permita evaluar la eficacia del sistema en situaciones de alta demanda y diferentes condiciones ambientales. De igual manera, se va a utilizar software de simulación para el vuelo de los drones, y se van a desarrollar algunas simulaciones para la red de sensores. Las pruebas se realizarán en cada una de las siguientes tres etapas:

1. Pruebas preliminares de vuelo del subsistema de enjambre.

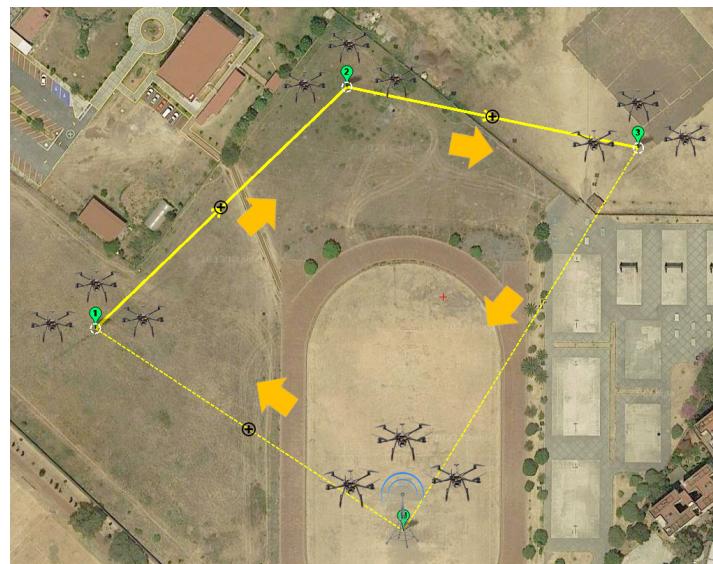


Figura 14.1: Escenario de pruebas finales del subsistema enjambre en el campo UPIITA-UPIBI.

2. Pruebas preliminares de comunicación del subsistema de la WSN.

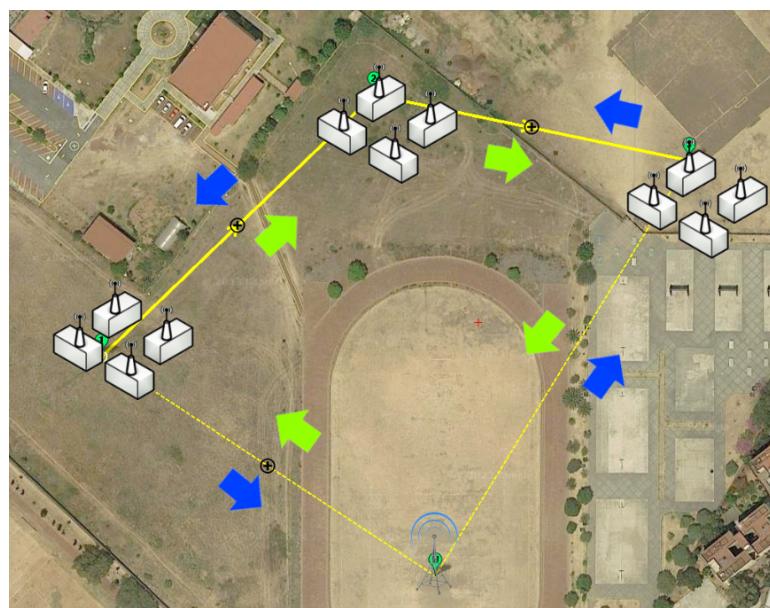


Figura 14.2: Escenario de pruebas finales del subsistema WSN en el campo UPIITA-UPIBI.



3. Pruebas finales del sistema integrado.

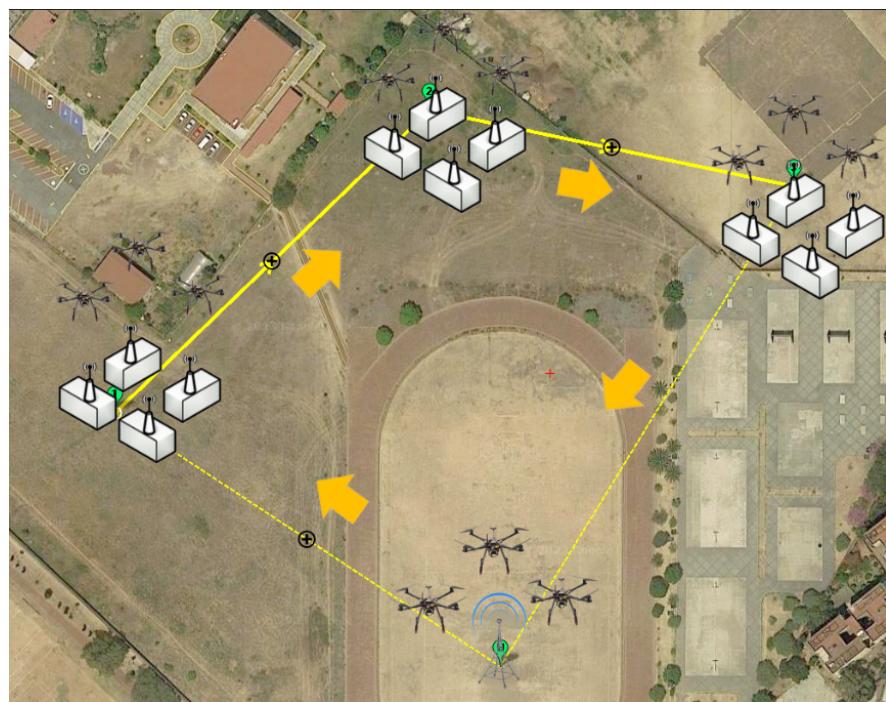


Figura 14.3: Escenario de pruebas finales del sistema integral en el campo UPIITA-UPIBI.