



INSTITUTO TECNOLÓGICO DE BUENOS AIRES - ITBA
ESCUELA DE INGENIERÍA Y TECNOLOGÍA

SISTEMA INTEGRAL DE MONITOREO DE FAUNA SILVESTRE

AUTORES: Rodriguez Turco, Martín Sebastian (Leg. N° 56629)
Mechoulam, Alan (Leg. N° 58438)
Lambertucci, Guido Enrique (Leg. N° 58009)
Lontero Bonaparte, Tomás Guillermo (Leg. N° 58150)

DOCENTES: Carrillo Del Pino, Manuel Esteban
Orchessi, Walter
Gasparini, Ignacio

TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO
ELECTRÓNICO

BUENOS AIRES
SEGUNDO CUATRIMESTRE, 2022



Proyecto final de Ingeniería Electrónica

Sistema Integral de Monitoreo De Fauna Silvestre

Autores: Mechoulam, Alan (58438)
Lambertucci, Guido Enrique (58009)
Rodriguez Turco, Martín Sebastian (56629)
Lonero Bonaparte, Tomás Guillermo (58150)

Tutores: Orchessi, Walter
Carrillo Del Pino, Manuel Esteban
Gasparini, Ignacio

Fecha: 30/11/2022

1. Agradecimientos

A nuestros padres y madres que hicieron posible nuestro pasaje por la institución.
A nuestros hermanos y abuelos que nos acompañaron y apoyaron.
Al CIDEI, a Miguel y Pablo, por brindarnos este increíble proyecto para desarrollar.
A Jorge, Gabriel y Nelson, por estar siempre cerca nuestro, dispuestos a ayudar.
A nuestros amigos, parejas y compañeros de vida.

2. Indice

2.1. Contenidos

1 Agradecimientos	1
2 Indice	2
2.1 Contenidos	2
2.2 Lista de Figuras	5
2.3 Lista de Tablas	6
3 Acrónimos y Definiciones	8
4 Resumen	10
5 Introducción	11
5.1 Antecedentes	11
5.2 Contexto del proyecto	11
6 Objetivos	13
6.1 Finalidad del Proyecto	13
6.2 Planteamiento del Problema a Resolver	13
6.3 Alcance	14
7 Definición de Producto	14
7.1 Requerimientos del Cliente	14
7.1.1 Relevamiento de Datos	14
7.1.2 Requerimientos Finales para Trazabilidad	15
7.2 Diagrama Funcional de Interfaces	16
7.3 Especificaciones de Diseño	16
7.3.1 Especificaciones Funcionales	16
7.3.2 Especificaciones de Interfaz	18
7.3.3 Especificaciones de Dimensionales	18
7.3.4 Especificaciones de Implementación	19
8 Plan de Validación	20
8.1 Diseño de Banco de Pruebas	20
8.2 Especificaciones de Test	21
8.2.1 Tests Básicos	22
8.2.2 Test de Funcionalidad	22
8.2.3 Test de Comunicación	25
8.2.4 Test de Peso, Dimensión y Seguridad	27
8.3 Matriz de Trazabilidad de Validación	28
8.4 Plan de Verificación y Validación	28
9 Análisis de Factibilidad	29
9.1 Factibilidad Tecnológica	29
9.1.1 Esquema Modular	29
9.1.2 Propuesta de Sensores	30
9.1.2.1 Temperatura	30
9.1.2.2 Humedad	30
9.1.2.3 Luminosidad	31
9.1.2.4 Imágenes y video	31
9.1.2.5 Tiempo	32
9.1.2.6 Presencia	32
9.1.3 Propuesta de Almacenamiento	33
9.1.4 Propuesta de Unidad de Procesamiento	33
9.1.5 Propuesta de Batería	34

9.1.6	Propuesta de Paneles solares	34
9.1.7	Regulador de energía	35
9.1.8	Primer Propuesta de Abastecimiento de Energía de UBM	35
9.1.8.1	Planteamiento del Problema	35
9.1.8.2	Carga por Acoplamiento Magnético	36
9.1.8.3	Carga por Radiofrecuencia	36
9.1.8.4	Banda de Frecuencia Adoptada	37
9.1.8.5	Condiciones de Borde	37
9.1.8.6	Interacción entre las Aves y la Radiación Electromagnética	38
9.1.8.7	Esquema de Conexionado	39
9.1.8.8	Amplificador de Potencia	40
9.1.8.9	Integrado de Energy Harvesting	40
9.1.8.10	Comparación entre Antenas	41
9.1.8.11	Selección de Componentes	43
9.1.8.12	Análisis Cuantitativo de la Recarga Inalámbrica	43
9.1.8.13	Conclusiones de la Factibilidad de la Recarga Inalámbrica	47
9.1.9	Propuesta Final de Abastecimiento de Energía de UBM	47
9.1.9.1	Módulos del Sistema de Seguimiento	47
9.1.9.2	Mochila	48
9.1.9.3	Base de Seguimiento	49
9.1.9.4	Base Principal de Seguimiento	50
9.1.9.5	Propuesta de Valor Aumentada	50
9.1.10	Elección de una Solución	50
9.1.10.1	Sensores	50
9.1.10.2	Almacenamiento	51
9.1.10.3	Unidades de procesamiento	51
9.1.10.4	Comunicación	51
9.1.10.5	Batería	52
9.1.10.6	Alimentación	53
9.1.11	DFMEA	54
9.2	Factibilidad de Tiempos	57
9.2.1	Consideraciones	57
9.2.2	Planificación	57
9.2.3	Programación	58
9.3	Factibilidad Económica	59
9.3.1	Modelo de Negocios	59
9.3.2	Investigación y Desarrollo	60
9.3.3	Gastos fijos por unidad	60
9.3.4	Reserva de Contingencia	60
9.3.5	Escenario de Escala	60
9.4	Factibilidad Legal y Responsabilidad Civil	60
10	Ingeniería de Detalle	61
10.1	Hardware	61
10.1.1	Diagrama de Bloques (Hardware)	61
10.1.2	Descripción Detallada de cada Bloque	62
10.1.2.1	Potencia	62
10.1.2.2	Medición	62
10.1.3	Detalles de Selección y Cálculo de los Elementos Circuitales de cada Bloque	64
10.1.3.1	Cálculo Potencias	64
10.1.3.2	Protección para alimentación Raspberry Pi	65
10.1.3.3	Cálculos circuitales	66
10.1.3.4	Cálculo de memoria	67
10.1.4	Plan de Pruebas de cada Módulo	68
10.2	Software	68
10.2.1	Central de Control	68
10.2.2	Configuración Remota	68
10.2.3	Eventos Periodicos	68

10.2.4	Diagrama de Estados y Fluujogramas	69
10.2.5	Descripción de Subrutinas	69
10.2.6	Interfaz Gráfica de Usuario	70
11	Construcción del Prototipo	73
11.1	Prototipo Nido-Sustrato	73
11.2	Prototipo Nido-Electrónica	74
11.2.1	Raspberry Pi	74
11.2.2	Sensores	75
11.3	Prototipo Nido-Potencia	75
12	Validación del Prototipo	76
12.1	Validación pruebas funcionales	76
12.2	Validación Pruebas de Comunicación 1	80
12.3	Validación Pruebas de Comunicación 2	82
12.4	Validación pruebas de seguridad	84
13	Estudios de Confiabilidad	84
13.1	Hardware	84
13.1.1	Raspberry Pi	85
13.1.2	Regulador de Energía	86
13.1.3	Paneles Solares	86
13.1.4	Baterías	86
13.1.5	Conektor Shield	86
13.1.6	RTC	86
13.1.7	Sensor de Presencia	86
13.1.8	Cámara Raspberry Pi	87
13.1.9	Conektor Sensores	87
13.1.10	Sensor Humedad y Temperatura	87
13.1.11	Sensor Luminosidad	87
13.1.12	Tasa de fallas total	87
13.2	Software	88
14	Conclusiones	89
14.1	Excelencias, Objetivos alcanzados	89
14.2	Fallos, Recomendaciones para futuros diseños	89
15	Referencias	90
16	Anexos Técnicos	91
16.1	Esquemáticos	91
16.2	Planos de PCB	100
16.3	Lista de materiales	100
16.3.1	Potencia	100
16.3.2	Sensores	101
16.3.3	Unidad de procesamiento	101
16.3.4	Prototipo para pruebas	101
16.3.5	Miscláneo	101
16.4	Código de software	102
16.4.1	BH1750	102
16.4.2	DHT22	102
16.4.3	getRTMeasurement	102
16.4.4	getTime	103
16.4.5	RTC	104
16.4.6	DatabaseLum	104
16.4.7	DatabaseTyH	105
16.4.8	FileMakerDB	107
16.4.9	makeCsvCli	108
16.4.10	presenceDetector	109

16.4.11 printMeasures	110
16.4.12 Conexión BPS para banco de pruebas Alcance	111
16.4.13 Conexión BPS para banco de pruebas Recepción-Raspberry Pi	112
16.4.14 Conexión BPS para banco de pruebas Transmisión-BPS	113

2.2. Lista de Figuras

5.2.1 Imagen del ave.	12
5.2.2 Morfología nido.	12
7.2.1 Diagrama Funcional de Interfaces.	16
8.4.1 Diagrama de dependencias del plan de validación.	29
9.1.1.1 Diagrama modular del sistema.	29
9.1.8.7.1 Diagrama en bloques cargador.	39
9.1.8.7.2 Módulo utilizado como oscilador HM-TRPW-RS232.	39
9.1.8.7.3 Módulo utilizado como <i>power harvester</i>	40
9.1.8.12.1 Cálculos preliminares de la potencia recibida en las baterías de la mochila del pájaro dada la distancia entre esta y la antena transmisora.	44
9.1.8.12.2 Potencia recibida en las baterías de la mochila del pájaro, dada la distancia entre esta y la antena transmisora, considerando el factor de pérdida por polarización. A los costados de la superficie graficada se observan las curvas de nivel de los distintos ejes del gráficos.	45
9.1.8.12.3 Curvas de nivel de la Figura (9.1.8.12.2) para grados de elevación fijos.	46
9.1.8.12.4 Curvas de nivel de la Figura (9.1.8.12.2) para distancias entre antena receptora y transmisora fijas.	46
9.1.8.12.5 Densidad de potencia radiada a distintas distancias de la antena transmisora dentro del nido.	47
9.1.9.1.1 Elementos ilustrativos de los módulos del sistema de seguimiento.	48
9.1.9.2.1 Beacon EMBC22.	49
9.1.9.3.1 Elementos ilustrativos de los módulos del sistema de seguimiento.	49
9.1.9.3.2 Área total de cobertura y rango de cada módulo individual.	50
9.1.10.5.1 Efectos de la temperatura en relación con la capacidad de la batería.	52
9.1.10.5.2 Ciclos de vida en relación con la profundidad de descarga.	53
9.2.3.1 Diagrama de Gantt del proyecto.	58
9.2.3.2 Simulación de Montecarlo.	58
9.2.3.3 Simulación de Montecarlo para tiempo de ingeniería.	59
9.3.1.1 Modelo de negocio.	59
10.1.1.1 Diagrama en bloques del sistema de <i>hardware</i>	61
10.1.2.2.1 Shield Raspberry Pi.	63
10.1.2.2.2 Placa que contiene a los sensores.	63
10.1.2.2.3 Conexionado cámara.	64
10.1.2.2.4 Conexionado tarjeta SD.	64
10.1.3.2.1 Circuito comparador e interruptores.	65
10.1.3.2.2 Simulación del circuito de protección.	66
10.1.4.1 Esquemático RTC-Tiny.	68
10.2.4.1 Diagrama de estados.	69
10.2.6.1 Flujo de nodos del servidor.	71
10.2.6.2 Página del servidor a la cual accede el usuario.	71
10.2.6.3 Datos de presencia y posición del ave.	72
10.2.6.4 Descarga de datos almacenados.	72
10.2.6.5 Configuración de la interfaz gráfica.	73
10.2.6.6 Comparación entre el uso del filtro de luminosidad.	73
11.1.1 Prototipo del producto montado sobre un nido artificial construido con madera.	74
11.2.1.1 Construcción de la unidad de procesamiento del prototipo.	75
11.2.2.1 Construcción de la placa de sensores y su encapsulado para el prototipo.	75
11.3.1 Comparación entre el uso del filtro de luminosidad.	76
12.1.1 Validación de funcionalidad <i>T-INT-FUN-01</i> y <i>T-INT-FUN-02</i>	77
12.1.2 Validación de seguridad <i>T-INT-FUN-03</i>	77
12.1.3 Validación de funcionalidad <i>T-INT-FUN-04</i> y <i>T-INT-FUN-12</i>	78
12.1.4 Variación de posición del beacon con respecto al nido.	78
12.1.5 Validación de funcionalidad <i>T-INT-FUN-07</i>	79

12.1.6	Validación de funcionalidad <i>T-INT-FUN-08</i>	79
12.1.7	Validación de funcionalidad <i>T-INT-FUN-09</i>	80
12.2.1	Consola del lado del receptor Raspberry Pi.	81
12.2.2	Consola del lado del transmisor <i>BPS</i>	81
12.2.3	Validación <i>T-COM1-02</i>	82
12.3.1	Validación de comunicación <i>T-INT-COM2-01</i>	83
12.3.2	Validación de funcionalidad <i>T-INT-COM2-02</i>	83
12.3.3	Validación de comunicación <i>T-INT-COM2-3</i>	84
12.4.1	Validación de seguridad <i>T-RAM-SEG-01</i>	84
13.1.1	Módulos críticos de funcionamiento.	85
16.1.1	Esquemático de conexionado del <i>shield</i>	91
16.1.2	Esquemático de conexionado para la placa de sensores.	91
16.1.3	Plano de la base del prototipo de nido.	92
16.1.4	Plano de la tapa del prototipo de nido.	93
16.1.5	Plano explotado del prototipo.	94
16.1.6	Plano del encapsulado para la Raspberry Pi con <i>shield</i> , encastre inferior	95
16.1.7	Plano del encapsulado para la Raspberry Pi con <i>shield</i> , encastre superior.	96
16.1.8	Plano del encapsulado para la cámara y sensores, encastre inferior.	97
16.1.9	Plano del encapsulado para la cámara y sensores, encastre superior.	98
16.1.10	Plano del soporte del <i>Power Bank</i>	99
16.2.1	Esquemático del PCB del <i>shield</i>	100
16.2.2	Esquemático del PCB de los sensores.	100

2.3. Lista de Tablas

5.2.1	Medidas del nido.	13
7.1.2.1	Requerimientos de máxima.	15
7.3.1.1	Leyendas para las especificaciones.	16
7.3.1.2	Especificaciones funcionales.	17
7.3.2.1	Especificaciones de interfaz COM1.	18
7.3.2.2	Especificaciones de interfaz COM2.	18
7.3.3.1	Especificaciones dimensionales y de peso.	18
7.3.4.1	Especificaciones de operación.	19
7.3.4.2	Especificaciones de almacenamiento y transporte.	19
7.3.4.3	Especificaciones de confiabilidad.	19
7.3.4.4	Especificaciones de seguridad.	19
7.3.4.5	Especificaciones de costo.	19
8.1.1	Tabla de diseño de los bancos de prueba, mostrando en naranja el DUT.	20
8.2.1	Tabla de plan de validación.	21
8.2.1.1	Tabla de tests básicos.	22
8.2.2.1	Tabla de tests de funcionalidad (Parte 1).	23
8.2.2.2	Tabla de tests de funcionalidad (Parte 2).	24
8.2.2.3	Tabla de tests de funcionalidad (Parte 3).	25
8.2.3.1	Tabla de tests de comunicación.	26
8.2.4.1	Tabla de tests de peso, dimensión y seguridad.	27
8.3.1	Matriz de trazabilidad de validación.	28
9.1.2.1.1	Comparación entre sensores de temperatura.	30
9.1.2.2.1	Comparación de sensores de humedad.	31
9.1.2.3.1	Comparación de sensores de luminosidad.	31
9.1.2.4.1	Comparación entre cámaras.	32
9.1.2.5.1	Comparación entre módulos RTC.	32
9.1.2.6.1	Comparación de módulos para la detección de presencia.	33
9.1.3.1	Comparación entre memorias SD.	33
9.1.4.1	Comparación entre placas Raspberry Pi.	34
9.1.5.1	Comparación entre baterías gel de carga profunda.	34
9.1.6.1	Comparación entre paneles solares.	35
9.1.7.1	Comparativa reguladores de energía.	35
9.1.8.8.1	Comparativa amplificadores de potencia.	40

9.1.8.9.1 Comparación entre cargadores.	41
9.1.8.10.1 Comparación entre antenas transmisoras (Parte 1).	41
9.1.8.10.2 Comparación entre antenas transmisoras (Parte 2).	42
9.1.8.10.3 Comparación entre antenas receptoras (Parte 1).	42
9.1.8.10.4 Comparación entre antenas receptoras (Parte 2).	43
9.1.8.12.1 Especificaciones de los cálculos realizados para el análisis cualitativo de la recarga inalámbrica.	43
9.1.11.1 DFMEA (Parte 1).	54
9.1.11.2 DFMEA (Parte 2).	55
9.1.11.3 DFMEA (Parte 3).	56
9.1.11.4 Criterio de IC.	57
9.1.11.5 Nivel de IC.	57
9.2.2.1 Actividades a realizar en el proyecto en días laborales de 8 horas cada uno.	57
9.3.3.1 Valores de insumos.	60
13.1.1 Ecuaciones para obtener la tasa de fallos de cada módulo acorde a MIL-HDBK-217F.	85
13.1.2.1 Confiabilidad regulador de energía.	86
13.1.5.1 Confiabilidad de la placa <i>shield</i>	86
13.1.6.1 Confiabilidad RTC.	86
13.1.7.1 Confiabilidad sensor de presencia.	86
13.1.9.1 Confiabilidad Conector Placa de Sensores	87
13.1.10.1 Confiabilidad Sensor de Temperatura y Humedad.	87
13.1.11.1 Confiabilidad Sensor de Luminosidad	87
13.1.12.1 Probabilidad de confiabilidad con períodos de 3 meses.	87
13.2.1 Fallas durante el periodo de test del <i>software</i>	88
13.2.2 Valores de la función probabilidad de confiabilidad de software.	88

3. Acrónimos y Definiciones

Acrónimo	Descripción
AMB	Ambiente
AC	Corriente Alterna (<i>Alternate Current</i>)
AGM	Material de Vidrio Absorbente (<i>Absorbent Glass Material</i>)
BLE	Bluetooth Low Energy
BT	Bluetooth
COM	Comunicación
DC	Corriente Continua (<i>Direct Current</i>)
EGM	Electromecánica
EMC	Compatibilidad Electromagnética (<i>Electromagnetic Compatibility</i>)
GUI	Graphical user interface
GPIO	Entradas/salidas de propósito general (<i>General Purpose Input/Output</i>)
GPS	Global Positioning System
HW	Hardware
I2C/I²C	Inter-Integrated Circuit
IC	Circuito Integrado
ISM	Industrial, Scientific, Medical
ITU	International Telecommunication Union
MEC	Mecánico
PCB	Circuito Impreso (<i>Printed Circuit Board</i>)
R-Pi	Raspberry Pi
R/W	Lectura/Escritura (<i>Read/Write</i>)
RF	Radio Frecuencia
RFID	Radio Frequency Identification
RTC	Real Time Clock
RTD	Resistance Temperature Detector
ROE	Razón de Onda Estacionaria
Rx	Receptor
S.I.M.F.S.	Sistema Integral de Monitoreo de Fauna Silvestre
SW	Software
TC	Termocupla
Tx	Transmisor
UBN	Unidad de Batería del Nido
UBM	Unidad de Batería de la Mochila
UP	Unidad de Procesamiento

Término	Definición
RFID	Sistema inalámbrico de identificación y registro de datos compuesto por dos componentes. El <i>lector</i> emite ondas electromagnéticas con el fin de energizar el <i>tag</i> y enviar datos. El <i>tag</i> almacena momentáneamente esta energía para emitir una respuesta al <i>lector</i> .
UBN	La Unidad de Batería del Nido hace referencia a la batería que se encuentra en el nido, la cual es cargada a través de un panel solar y almacena la energía de todo el módulo.
UBM	La Unidad de Batería de la Mochila es la batería que se encuentra en la mochila que dispone el ave.
Schmitt Trigger	Un <i>Schmitt Trigger</i> es un circuito comparador con histéresis. Convierte una señal analógica a una digital, y se le dice <i>trigger</i> debido a que la salida retiene su valor hasta que la entrada cambie lo suficiente como para que sea de un cambio.

4. Resumen

En este proyecto final de ingeniería se detalla el diseño del S.I.M.F.S., una plataforma de adquisición y almacenamiento de datos que reside dentro del nido de un ave durante la etapa de anidamiento. Además, este trabajo incluye desarrollo en investigación de tecnologías de radiación electromagnética para carga inalámbrica. Esta plataforma es utilizada en un estudio de investigación de la especie *Campephilus Magellanicus*, mejor conocido como Carpintero Gigante, en conjunto a un dispositivo ajeno a este proyecto que se coloca sobre el ave.

El S.I.M.F.S. es una red que abarca un amplio margen de tecnologías. Emplea protocolo de comunicación Bluetooth con otros dispositivos; se encarga de la obtención y almacenamiento de datos de periféricos; emplea sistemas de alimentación para áreas remotas; corre un servidor web con acceso restringido donde se alcanza al usuario el acceso a estos datos, entre otras funcionalidades.

Entrando en detalle en lo que respecta a la elección de las tecnologías se puede especificar que:

- Para la alimentación del sistema en ausencia de una red eléctrica cercana, se utilizan paneles solares.
- Para el auxilio de los paneles solares al caer la noche, se utiliza una batería de gel de ciclo profundo.
- Para la recolección de datos, distintos tipos de sensores.
- Para el almacenamiento, una tarjeta SD de grado industrial.
- Para la unidad de procesamiento, una Raspberry Pi 3B.

Además, se incurre en una investigación en el área de transmisión de potencia para analizar la factibilidad tecnológica de la recarga inalámbrica de las baterías de aquel dispositivo cuyo desarrollo es ajeno al proyecto. Se estudia el tipo de acople electromagnético, la banda de frecuencia, se realiza un análisis de distinto tipo de antenas y un estudio de reglamentaciones legales.

A lo largo del trabajo se concluye que la mejor elección para la transmisión inalámbrica de potencia dadas las circunstancias del proyecto es el uso de radiofrecuencia de campo lejano en la banda de frecuencia centrada en 915 MHz con antenas del tipo *patch*. Esto se debe a que para los clientes es de suma importancia la determinación de la factibilidad del desarrollo de una tecnología que cumpla con las características mencionadas. Sin embargo, el estudio de factibilidad dicta que no es posible dentro de las reglamentaciones legales y la eficiencia conseguida obtener la potencia requerida dentro de las restricciones impuestas por la naturaleza del proyecto.

En consecuencia, y para satisfacción de los requerimientos provistos por los solicitantes del proyecto, se propone una alternativa que, si bien no realiza la carga inalámbrica mencionada, garantiza el correcto funcionamiento de los elementos que requieren del uso de dicha carga. Así también aumentando la propuesta de valor provista por la propuesta que incluía el cargador inalámbrico.

Finalmente se detalla sobre la construcción de un prototipo. Con este se aplica el plan de validación, se realiza un estudio económico y legal del proyecto, y se construye continuamente un análisis modal de fallos y efectos.

5. Introducción

5.1 Antecedentes

Cuando se estudian aves, por lo general, los investigadores optan por colocar pequeños dispositivos transmisores sobre el cuerpo de estas. Sin embargo, las soluciones disponibles en el mercado tienen restricciones de energía y peso lo cual resultan incompatibles con las expectativas del grupo INBIOMA.

Actualmente las unidades de recolección de información toman datos sobre la posición, temperatura y el estado vital del espécimen, entre otras. Estos dispositivos comerciales requieren de una antena para la transmisión de datos mediante redes celulares, las cuales no siempre están presentes en las zonas de interés y además generan costos de comunicación. Las antenas que se emplean, cuyo largo es comparable con el largo del ave, no presentan dificultad alguna para aves que duermen y anidan en dormideros o nidos abiertos (al aire libre). Para el caso de las aves que viven en el interior de los árboles, tal como los pájaros carpinteros, el uso de dichos dispositivos es un inconveniente. Estas antenas pueden poner en peligro a las demás aves que habitan dentro del nido y dificultarles la movilidad, haciéndolas más vulnerables ante depredadores.

Por otro lado, los productos existentes que están pensados para especies de menor tamaño, no contemplan la naturaleza territorial y violenta del pájaro carpintero gigante.

También existen productos para aves de mayor tamaño. El problema en estos radica en la incapacidad del sujeto de estudio de transportar el peso de la electrónica asociada a estos productos.

Por último, consideramos las opciones que se pueden conseguir en el mercado no profesional, destinadas para el uso hogareño: pequeños nidos de fácil instalación que poseen sensores variados. Sin embargo, no podría ser utilizado dado que el carpintero no es tomadera de nidos, sino que debe construir el propio.

5.2 Contexto del proyecto

El INBIOMA (Investigaciones en Biodiversidad y Medioambiente radicado en la Universidad Nacional del Comahue) desarrolla un estudio, cuyo resultado busca aprender aspectos de la vida del Carpintero Gigante, ave que sirve de vector de referencia para analizar el estado de otros elementos de la vida silvestre en el área [1]. Es decir, el estudio de los patrones de alimentación y movimiento del ave en cuestión pueden alertar sobre diversos factores que cambian en el ambiente.

En la actualidad no existe en el mercado un dispositivo que permita obtener las variables físicas, tanto del vuelo y comportamiento de las aves, como de su entorno (nido y alrededores). Es por eso que el CIDEI (Centro de Investigación y Desarrollo en Electrónica Industrial del ITBA) busca facilitar este proceso encargando el desarrollo de un dispositivo que satisfaga dichos requerimientos, agregando a su vez aportes que permitan darle un valor agregado al sistema.

El ave de estudio se llama *Campephilus Magellanicus* o “Carpinteros Gigantes”, especie de pájaro carpintero más grande de Sudamérica y de las más grandes del mundo. Los machos llegan a pesar hasta 360 gramos, mientras que las hembras 310 gramos.

Los carpinteros gigantes se encuentran sobre toda la zona andina y la Patagonia. En este caso el estudio se realiza en las cercanías de Bariloche.



Figura 5.2.1: Imagen del ave.

Durante los meses de primavera, estas aves entran en lo que se llama periodo de anidación, donde la hembra pone hasta 4 huevos. A lo largo de este intervalo, el macho y la hembra se turnan para incubar. Los primeros se hacen cargo durante la noche (aproximadamente 6 a 8 horas), mientras que las segundas se ocupan durante el día. Es fundamental saber que esta especie actúa de forma muy agresiva y territorial durante este tiempo, ya que buscan proteger a su descendencia.

Otro aspecto vital es conocer la morfología del nido, lugar donde van los sensores del proyecto. Debido a los hábitos de limpieza del ave, el único lugar factible para la posición de la electrónica resulta ser la bóveda del nido.

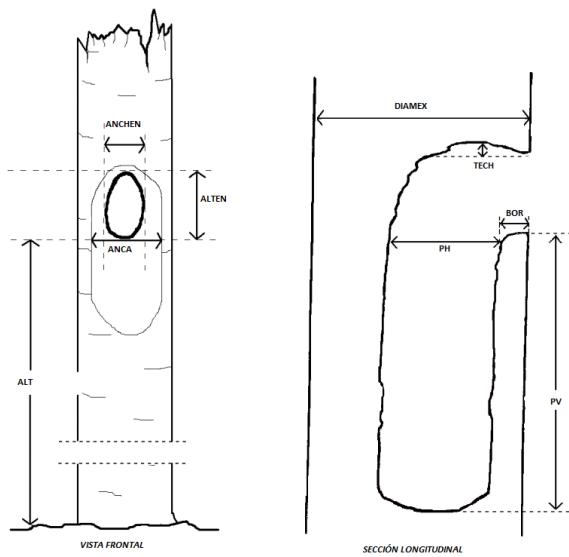


Figura 5.2.2: Morfología nido.

Variable	Media	Rango	Muestras
DIAMEX [cm]	45.56	33.74 ~ 71.94	26
ANCA [cm]	23.41	18.3 ~ 33	26
ALT [m]	8.68	3.3 ~ 17	27
ANCHEN [cm]	8.79	7.9 ~ 9.7	27
ALTEN [cm]	15.40	11.4 ~ 20	27
PV [cm]	32.52	21 ~ 44	27
PH [cm]	26.22	20 ~ 33	27
BOR [cm]	6.41	4 ~ 12	27
TECH [cm]	4.55	0 ~ 10	27

Tabla 5.2.1: Medidas del nido.

Además de esta información, se sabe que el grupo de investigadores se presentan cada dos semanas en la base del árbol, frente al nido, para obtener la información recolectada por el dispositivo.

Finalmente cabe mencionar que este proyecto es complementario de otro ajeno, el cual involucra una mochila que se encuentra sobre el pájaro. Este equipamiento busca obtener datos propios del ave, de su posición y trayectoria. Es esta razón por la cual cobra importancia el comportamiento de la especie durante el anidamiento.

La importancia de este complemento es que la información que recolecte debe ser transmitida y almacenada en el nido, unificando así ambos proyectos al momento de brindarle acceso al usuario.

Otro aspecto que se debe contemplar con respecto a la mochila es que debe proponerse la viabilidad energética de esta. En otras palabras, se deben plantear propuestas que garanticen el funcionamiento del dispositivo a lo largo del proceso de investigación.

6. Objetivos

6.1 Finalidad del Proyecto

En este proyecto se busca desarrollar un equipo electrónico encargado por el CIDEI que permita a un grupo de ornitólogos del CONICET realizar un estudio sobre el ave *Campephilus Magellanicus*. Este equipo adquiere distintas variables del entorno del nido y del ave, sin perturbar la vida normal de este e independizándose de la necesidad de conectarse a una red eléctrica.

El producto brindará al campo de la investigación de la ornitología vital información acerca de uno de los vectores de referencia más importantes del bosque patagónico argentino.

6.2 Planteamiento del Problema a Resolver

El pájaro carpintero habita en zonas urbanas, suburbanas, rurales e intangibles¹. Este factor limita las fuentes de alimentación que se emplean para poder mantener funcionando al sistema. Es por ello que el uso de la red eléctrica no es una opción.

El producto propuesto debe involucrar la adquisición y disposición de distintos parámetros de la vida del ave y de su entorno. Dentro de lo que abarca propiamente el nido, se encuentran como datos elementales la temperatura, luminosidad, humedad y la detección de presencia del ave en el nido. A estas variables se le suma la necesidad de almacenar y transmitir en tiempo real imágenes del interior.

El CIDEI pactó con el grupo de ornitólogos que, una vez cada dos semanas, se acercará una persona a la base del árbol para obtener las mediciones. Esto se repite durante todo el proceso de anidamiento del espécimen en cuestión.

Por otro lado, este proyecto se complementa a su vez con otro dispositivo, el cual se encuentra montado sobre el ave. El elemento mencionado se emplea para poder captar la posición del pájaro durante el día. Como también se debe dar acceso a esos datos, no solo se debe poder almacenar las mediciones captadas en el nido, sino que también se debe incorporar la posibilidad de recibir de manera inalámbrica aquellos datos que obtenga el equipo del ave y permitirle acceso al usuario. Es de vital importancia la necesidad de no perturbar la vida normal del ave.

Un desafío del trabajo se centra en el requisito de lograr mantener energizado ambos sistemas (dispositivo en el nido y dispositivo sobre el ave) sin intervención humana durante todo el periodo de anidamiento del ave, sin la posibilidad de una conexión a la red eléctrica. A pesar de ello, la mayor dificultad se encuentra en el segundo

¹Se definen las zonas intangibles como aquellas zonas a las que no puede acceder el turista, donde se protege la biodiversidad.

dispositivo debido a su tamaño reducido, a que no se encuentra fijo en un lugar y que no puede ser cargado utilizando cables.

Es por ello que un planteamiento del proyecto consiste en determinar la factibilidad de la carga inalámbrica de las baterías del dispositivo del ave. En caso de obtener una respuesta favorable, debe ser llevado a la práctica. Caso contrario, se debe garantizar una solución viable que permita obtener la posición del ave durante el día, garantizando la medición de dicha variable en un entorno mínimo.

6.3 Alcance

El alcance de este proyecto es realizar un prototipo funcional no comercial de un dispositivo capaz de recolectar información para ser utilizado en el ámbito de la investigación. Cuenta con la capacidad de comunicarse por medio de WiFi con la finalidad de habilitar una página web local para uso de las biólogas. En dicha red se puede descargar la información recolectada. Además, cuenta con comunicación *Bluetooth* para recibir los datos recolectados por un proyecto ajeno a S.I.M.F.S., aunque complementario.

Se analiza la factibilidad tecnológica de cargar de manera inalámbrica la UBM colocada sobre el proyecto ajeno. En caso de que el estudio de factibilidad dicte que no sea viable la carga, se debe garantizar la obtención de datos de posición determinadas por el dispositivo montado sobre el ave, cumpliendo aún con los requisitos de peso, dimensiones y transmisión de información de esta.

Dado que el dispositivo está destinado para utilizarse en zonas remotas, se debe conseguir energía mediante medios propios del entorno, como utilizando paneles solares para recolectar energía y baterías para almacenarla.

Se realizan los análisis relevantes para asegurar la viabilidad financiera del proyecto según los requisitos del cliente. La verificación de la calidad del diseño está basada en un único prototipo no comercial, el cual busca cumplir los requerimientos definidos y adquirir las validaciones posibles dentro del marco económico actual y las limitaciones del cliente.

Por ser un proyecto con bajo volumen de producción no se considera el desarrollo de tecnologías de adquisición de datos, almacenamiento o extracción de energía. Tampoco se busca desarrollar un plan de desarrollo económico.

7. Definición de Producto

7.1 Requerimientos del Cliente

El grupo de investigadores del INBIOMA se encuentra en contacto con el CIDEI. Este último encomienda el desarrollo del dispositivo en el cuál se centra este trabajo. De esta forma el centro de investigación del ITBA tiene la libertad de tomar este proyecto y anexarle un interés propio, avocado a la investigación.

De esta forma, no solo se busca desarrollar un proyecto que simplifique el trabajo del usuario de estudiar el ave de interés, sino que también agrega un gran valor en cuanto a lo que se refiere con investigación en electrónica y carga inalámbrica.

En conclusión, se realizó el relevamiento de requerimientos basándose en lo que se obtuvo por parte del CIDEI.

7.1.1 Relevamiento de Datos

La adquisición de datos fue realizada por parte del CIDEI mediante sucesivas reuniones con el equipo de ornitólogas. Estas informaron las necesidades del producto para llevar a cabo su investigación y el CIDEI como cliente de este proyecto relevó estos datos al equipo de desarrollo para fijar los requerimientos del cliente.

Además, se tuvieron en cuenta las diversas normas que rigen los equipos electrónicos vigentes en Argentina como se detalla en la Sección (7.3).

7.1.2 Requerimientos Finales para Trazabilidad

ID	Descripción	Origen
REQ-01	El producto esta colgado de un árbol a una altura de 4 a 14 metros y se instala parcialmente dentro del nido del ave sin comprometer la integridad del árbol.	Cliente
REQ-02	El producto debe mantenerse energizado sin intervención humana.	Cliente
REQ-03	El producto no debe requerir conexión a la red eléctrica para su funcionamiento.	Táctico
REQ-04	El producto debe ser capaz de adquirir los siguientes datos dentro del nido: imágenes, temperatura, humedad, nivel de luz y presencia del ave.	Cliente
REQ-05	El producto debe ser capaz de saber la fecha y hora.	Táctico
REQ-06	Aquellos datos recolectados por dispositivos ajenos al nido deberán ser transmitidos a este por medio del protocolo BLE.	Cliente
REQ-07	El producto debe almacenar los datos adquiridos.	Táctico
REQ-08	Una persona debe recibir los datos almacenados en el nido a la distancia a través de una computadora.	Cliente
REQ-09	El producto no debe llamar la atención de humanos desde el nivel del piso.	Cliente
REQ-10	El producto debe soportar las condiciones meteorológicas del sur Argentino, específicamente los alrededores de Bariloche, Río Negro.	Táctico
REQ-11	El costo del producto debe ser menor o igual a 1000 USD.	Táctico
REQ-12	Analizar la factibilidad tecnológica de la recarga inalámbrica de la batería de la mochila del pájaro.	Cliente
REQ-13	El producto desarmado debe soportar las condiciones de translado impuestas por los caminos rurales hasta llegar a la zona de instalación.	Estado
REQ-14	La vida útil del producto debe ser de por lo menos dos años.	Estado
REQ-15	En caso que no sea factible una recarga inalámbrica de la batería de la mochila, proponer una solución alternativa que permita adquirir los datos de posición del ave a lo largo del estudio.	Cliente

Tabla 7.1.2.1: Requerimientos de máxima.

7.2 Diagrama Funcional de Interfaces

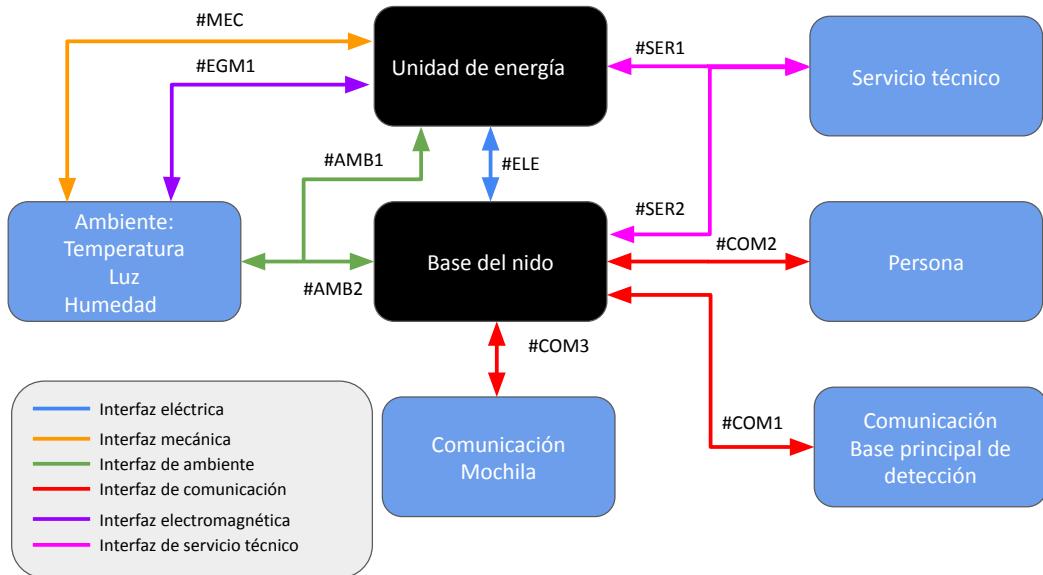


Figura 7.2.1: Diagrama Funcional de Interfaces.

7.3 Especificaciones de Diseño

7.3.1 Especificaciones Funcionales

Leyenda para especificaciones	
Aplicabilidad	Validación
P: Prototipo	I: Inspección Visual
	D: Documentación de Diseño
F: Producto Final	S: Simulación
	T: Test

Tabla 7.3.1.1: Leyendas para las especificaciones.

ID	Descripción	Origen	Aplicabilidad Validación
INT-FUN-01	El dispositivo deberá tener un espacio de almacenamiento de datos de un tamaño entre 32 a 64 GB.	REQ-07	P F - D
INT-FUN-02	El producto debe ser capaz de obtener la temperatura del entorno.	REQ-04	P F - I D T
INT-FUN-03	El producto debe ser capaz de obtener la humedad del entorno.	REQ-04	P F - I D T
INT-FUN-04	El producto debe ser capaz de obtener la intensidad lumínica del interior del nido.	REQ-04	P F - I D T
INT-FUN-05	El producto debe ser capaz de obtener video del interior del nido.	REQ-04	P F - I D T
INT-FUN-06	El producto debe ser capaz de obtener imágenes del interior del nido.	REQ-04	P F - I D T
INT-FUN-07	El producto debe ser capaz de detectar la presencia del ave en el nido.	REQ-04	P F - I D T
INT-FUN-08	El producto debe transmitir de manera inalámbrica los datos almacenados en el nido a un dispositivo según todas las especificaciones de la tabla <i>INT-COM2</i> .	REQ-08	F - D
INT-FUN-09	El producto debe recibir de manera inalámbrica datos almacenados en un dispositivo ajeno al proyecto según todas las especificaciones de la tabla <i>INT-COM1</i> .	REQ-06	F - D
INT-FUN-10	Determinar, dentro de las limitaciones tecnológicas, financieras, legales y morales, además de los requisitos energéticos de la mochila, si es posible la recarga inalámbrica de las baterías de la mochila.	REQ-12	P F - D
INT-FUN-11	En caso que no sea factible una recarga inalámbrica de la batería de la mochila, proponer una solución alternativa que permita adquirir los datos de posición del ave a lo largo del estudio.	REQ-15	P F - D
INT-FUN-12	El sistema debe obtener las variables a medir con una frecuencia de entre 5 y 60 minutos.	REQ-04	P F - T
INT-FUN-13	El sistema debe obtener video del nido en tiempo real transmitido a una computadora ubicada a una distancia que por lo menos abarque entre 3.3 y 17 metros.	REQ-04	P F - T
INT-FUN-14	El sistema utiliza uno o más paneles solares para cargar una batería de gel de ciclo profundo.	REQ-02 REQ-03	P F - D T
INT-FUN-15	El producto debe ser capaz de mantener registro de la hora y fecha correcta luego de ser previamente configurada.	REQ-05	P F - T
INT-FUN-16	Ante la pérdida total de la alimentación de la unidad de energía y el eventual rebastecimiento de esta, el producto debe ser capaz de recuperarse, mantener registro de la hora y fecha correcta luego de ser previamente configurada.	REQ-05	P F - T

Tabla 7.3.1.2: Especificaciones funcionales.

7.3.2 Especificaciones de Interfaz

ID	Descripción	Origen	Aplicabilidad Validación
INT-COM1-01	La transmisión de datos debe efectuarse por medio del protocolo BLE desde el la base principal de seguimiento hacia la base del nido de manera unidireccional.	REQ-06	P F - T
INT-COM1-02	La transmisión de datos se efectúa una vez al día con la información almacenada en la base principal de seguimiento.	REQ-06	F - D
INT-COM1-03	La transmisión de datos entre el nido y la base principal de seguimiento debe tener un alcance de entre por lo menos uno y cinco metros.	REQ-06	P F - T

Tabla 7.3.2.1: Especificaciones de interfaz COM1.

ID	Descripción	Origen	Aplicabilidad Validación
INT-COM2-01	Esta comunicación debe descargar al dispositivo de la persona todos los datos seleccionados almacenados en el nido.	REQ-08	P F - T
INT-COM2-02	La transmisión de datos debe tener un alcance que por lo menos abarque entre 3.3 y 17 m.	REQ-08	P F - T
INT-COM2-03	La transmisión de datos debe ser inicializada por la persona de modo manual.	REQ-08	P F - D
INT-COM2-04	La transmisión de datos debe efectuarse por medio del protocolo WiFi.	REQ-08	P F - D
INT-COM2-05	Los datos descargados por el usuario son descartados de los datos almacenados en el nido una vez el usuario se desconecte de la red WiFi.	REQ-08	F - D T

Tabla 7.3.2.2: Especificaciones de interfaz COM2.

7.3.3 Especificaciones de Dimensionales

ID	Descripción	Origen	Aplicabilidad Validación
IMP-DIM-01	El dispositivo del nido no debe exceder las siguientes dimensiones: Largo <26 cm Ancho <8.79 cm Alto <4.55 cm	REQ-01	F - T
IMP-DIM-02	La unidad de energía no debe exceder las siguientes dimensiones: Largo <1 m Ancho <50 cm Alto <1 m	REQ-01	F - D T
IMP-DIM-03	El equipo dentro del nido no debe exceder los 500 g.	REQ-01	F - D T
IMP-DIM-04	La unidad de energía no debe exceder los 20 kg.	REQ-01	F - D T

Tabla 7.3.3.1: Especificaciones dimensionales y de peso.

7.3.4 Especificaciones de Implementación

ID	Descripción	Origen	Aplicabilidad Validación
IMP-OPE-01	El sistema debe operar normalmente cuando la temperatura ambiente sea $-20^{\circ}\text{C} < T_{\text{AMB}} < 30^{\circ}\text{C}$.	REQ-10	F - D
IMP-OPE-02	Debe operar normalmente cuando la humedad sea: 0 % $< \text{RH} < 100 \%$	REQ-10	F - D
IMP-OPE-03	El dispositivo debe operar normalmente cuando la presión atmosférica sea: 84 kPa $< P_{\text{ATM}} < 90 \text{ kPa}$. Esto equivale a 1500 m de altura para el mínimo de presión, y un máximo de 0 m.	REQ-10	F - D
IMP-DIM-04	La unidad de energía no deberá exceder los 20 kilos.	REQ-01	F - D T

Tabla 7.3.4.1: Especificaciones de operación.

ID	Descripción	Origen	Aplicabilidad Validación
IMP-AYT-01	No se deben sufrir daños cuando, estando desenergizado, la temperatura ambiente sea $-20^{\circ}\text{C} < T_{\text{AMB}} < 40^{\circ}\text{C}$.	REQ-10 REQ-13	F - D
IMP-AYT-02	No se deben sufrir daños cuando, estando desenergizado, la humedad sea 0 % $< \text{RH} < 100 \%$.	REQ-10 REQ-13	F - D
IMP-AYT-03	No se deben sufrir daños cuando, estando desenergizado, la presión atmosférica sea 84 kPa $< P_{\text{ATM}} < 101 \text{ kPa}$.	REQ-10 REQ-13	F - D

Tabla 7.3.4.2: Especificaciones de almacenamiento y transporte.

ID	Descripción	Origen	Aplicabilidad Validación
RAM-CON-01	El producto debe tener una vida útil no menor a tres años.	REQ-14	F - D

Tabla 7.3.4.3: Especificaciones de confiabilidad.

ID	Descripción	Origen	Aplicabilidad Validación
RAM-SEG-01	El dispositivo cuenta con un sistema de autenticación ante el pedido de transmisión de datos definido por INT-COM2.	REQ-08	P F - I D
RAM-SEG-02	El dispositivo no pose luces ni elementos reflectantes para no ser percibido.	REQ-09	F - I

Tabla 7.3.4.4: Especificaciones de seguridad.

ID	Descripción	Origen	Aplicabilidad Validación
IMP-COS-01	El costo del producto debe ser menor o igual a 1000 USD.	REQ-11	F - D

Tabla 7.3.4.5: Especificaciones de costo.

8. Plan de Validación

8.1 Diseño de Banco de Pruebas

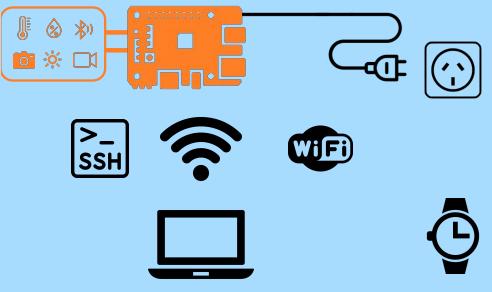
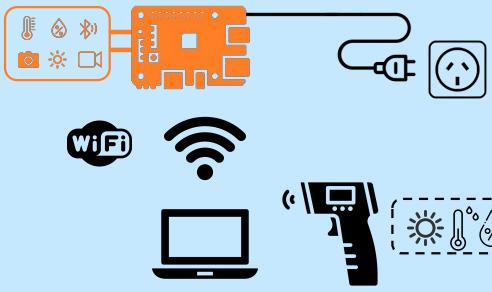
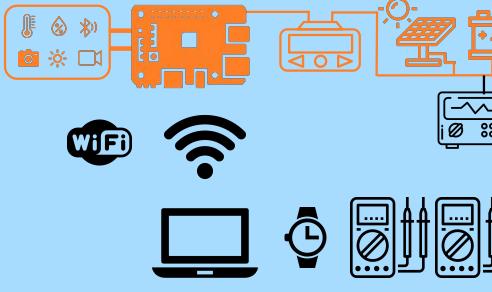
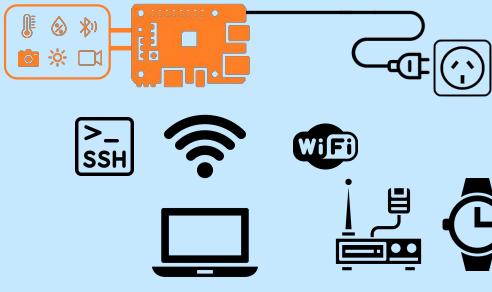
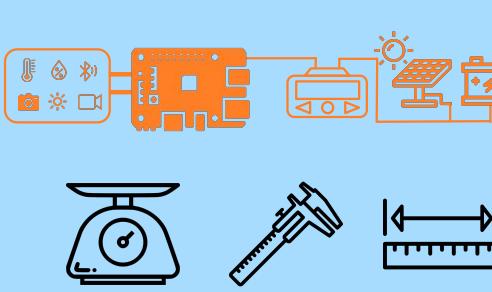
Banco Número	Detalle
BP1	 <ul style="list-style-type: none"> El producto. Fuente de tensión de 5 V y capacidad de proveer 500 mA junto a un cable de alimentación mini-USB. Una computadora capaz de comunicarse con el producto mediante SSH y WiFi. Un reloj.
BP2	 <ul style="list-style-type: none"> El producto. Fuente de tensión de 5 V y capacidad de proveer 500 mA junto a un cable de alimentación mini-USB. Una computadora capaz de comunicarse por WiFi con el producto. Sensor de Humedad calibrado. Sensor de temperatura calibrado.
BP3	 <ul style="list-style-type: none"> El producto. <i>Solar Array Simulator DC Supply</i>. Dos multímetros. Un reloj. Una computadora capaz de comunicarse por WiFi con el producto.
BP4	 <ul style="list-style-type: none"> El producto. Fuente de tensión de 5 V y capacidad de proveer 500 mA junto a un cable de alimentación mini-USB. Una computadora capaz de comunicarse por SSH y WiFi con el producto. Un reloj. Módulo BLE de pruebas.
BP5	 <ul style="list-style-type: none"> El producto. Cinta métrica. Balanza.

Tabla 8.1.1: Tabla de diseño de los bancos de prueba, mostrando en naranja el DUT.

8.2 Especificaciones de Test

ID del test	Aspecto	Dependencia	Banco de pruebas
T-BASIC-01	Validación encendido R-Pi.	-	1
T-BASIC-02	Validación placas manufacturadas.	T-BASIC-01	3
T-BASIC-03	Validación panel solar en correcto estado.	-	3
T-BASIC-04	Validación del regulador de carga en correcto estado.	T-BASIC-03	3
T-INT-FUN-01	Adquisición de datos de temperatura.	T-INT-FUN-09	2
T-INT-FUN-02	Adquisición de datos de humedad.	T-INT-FUN-09	2
T-INT-FUN-03	Adquisición de datos de luminosidad.	T-INT-FUN-09	2
T-INT-FUN-04	Adquisición de video.	T-INT-FUN-09	2
T-INT-FUN-05	Adquisición de imágenes.	T-INT-FUN-09	2
T-INT-FUN-06	Adquisición de datos de presencia.	T-INT-FUN-09	4
T-INT-FUN-07	Periodo activación sensores.	T-INT-FUN-01	1
		T-INT-FUN-02	
		T-INT-FUN-03	
		T-INT-FUN-04	
		T-INT-FUN-05	
T-INT-FUN-08	Validación de fecha y hora cierta.	T-BASIC-02	1
T-INT-FUN-09	Validación de fecha y hora cierta tras pérdida de alimentación.	T-INT-FUN-08	1
T-INT-FUN-10	Recolección de energía en condiciones similares a las de instalación.	T-BASIC-04	3
T-INT-FUN-11	Recuperación ante pérdida de alimentación.	T-INT-FUN-10	3
T-INT-FUN-12	Transmisión de video en tiempo real de dentro del nido.	T-INT-FUN-04	2
		T-INT-FUN-05	
T-INT-COM1-01	Validación transmisión COM1.	T-BASIC-02	4
T-INT-COM1-02	Alcance transmisión COM1.	T-INT-COM1-01	4
T-INT-COM2-01	Validación descarga de datos COM2.	T-BASIC-02	1
T-INT-COM2-02	Alcance transmisión COM2.	T-INT-COM2-01	1
T-INT-COM2-03	Descarte de datos desconexión WiFi COM2.	T-INT-COM2-01	1
T-IMP-DIM-01	Validación dimensiones totales.	T-INT-FUN-11	5
T-IMP-DIM-02	Validación peso total.	T-INT-FUN-11	5
T-RAM-SEG-01	Autorización transmisión nido-persona.	T-INT-COM2-01	1

Tabla 8.2.1: Tabla de plan de validación.

8.2.1 Tests Básicos

ID Aplicabilidad	Procedimiento	Criterio
T-BASIC-01 Prototipo, Final	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Colocar uno de los multímetros en modo continuidad. • Comprobar que no haya pistas cortadas debido al proceso de manufactura. • Comprobar que no haya cortocircuitos entre pistas que no deben estar conectadas debido al proceso de manufactura. 	Todas las pistas presentan correcta continuidad y no hay cortocircuitos entre pistas.
T-BASIC-02 Prototipo, Final	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Posicionar al panel solar de manera tal que a este lo alumbe la luz solar por completo. • Utilizar un multímetro en configuración de medición de tensión en escala 20 V para medir la tensión del panel solar. 	La tensión del panel medida debe ser la tensión a circuito abierto.
T-BASIC-03 Prototipo, Final	<p>Se utiliza el banco de pruebas #1:</p> <ul style="list-style-type: none"> • Proveer alimentación a la R-Pi. 	Se encienden los LEDs rojo y verde de la R-Pi.
T-BASIC-04 Prototipo, Final	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Conectar la batería al regulador de carga DFR0580. • Conectar el simulador de arreglo de paneles solares MPPT al regulador de carga. • Utilizar los multímetros para medir la corriente y tensión de la batería. 	LEDs de carga de batería y entrada de panel solar se encienden. Se registra corriente que ingresa a la batería y tensión de carga.

Tabla 8.2.1.1: Tabla de tests básicos.

8.2.2 Test de Funcionalidad

Precondiciones: Para realizar la prueba T-INT-FUN-11 las anteriores deben haberse realizado satisfactoriamente.

Procedimiento general: Conectar la Raspberry Pi a la alimentación de tensión mediante el cable de alimentación mini USB.

ID Aplicabilidad	Procedimiento	Criterio
T-INT-FUN-01 Prototipo, Final	Se utiliza el banco de pruebas #2: <ul style="list-style-type: none">• Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”.• Abrir un navegador de internet y conectarse a la página “192.168.0.1:1880”.• Prender el sensor de temperatura calibrado y colocarlo próximo al sensor del producto.	La lectura en el sensor calibrado coincide con el provisto por la página con una desviación de como máximo $\pm 0.5^{\circ}C$.
T-INT-FUN-02 Prototipo, Final	Se utiliza el banco de pruebas #2: <ul style="list-style-type: none">• Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”.• Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”.• Prender el sensor de humedad calibrado y colocarlo próximo al sensor del producto.	La lectura en el sensor calibrado coincide con el provisto por la página con una desviación de como máximo $\pm 2\%$.
T-INT-FUN-03 Prototipo, Final	Se utiliza el banco de pruebas #2: <ul style="list-style-type: none">• Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”.• Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”.• Tapar el sensor de luminosidad• Descubrir el sensor• Apuntar al sensor con una linterna prendida	Las lecturas del sensor varían en cada caso de 0 lux (tapado), nivel percibido en el ambiente y un nivel alto cuando es apuntado directamente por la linterna. Valores acorde a la hoja de datos.
T-INT-FUN-04 Prototipo, Final T-INT-FUN-12 Prototipo, Final	Se utiliza el banco de pruebas #2: <ul style="list-style-type: none">• Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”.• Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”.• Tocar el botón de la página web llamado “Ver el nido en vivo”.	Lo visto en la página web coincide con la realidad.
T-INT-FUN-05 Prototipo, Final	Se utiliza el banco de pruebas #2: <ul style="list-style-type: none">• Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”.• Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”.• Abrir una pestaña más y conectarse a la página “192.168.1.101:1880/admin”.• Hacer click en el <i>inject</i> de la zona del <i>flow</i> marcado como “Photo Filter Debug”.• Volver a la primera página abierta e ir al ítem del menú lateral descargas, y seleccionar el archivo con el prefijo Dev.	Lo visto en la imagen descargada coincide con la realidad.
T-INT-FUN-06 Prototipo, Final	Se utiliza el banco de pruebas #4: <ul style="list-style-type: none">• Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”.• Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”• Energizar el módulo BLE previamente programado.• Acercar el dispositivo al nido.	Lo visto en el sector de la página web que indica la presencia, sea “El pájaro no está en casa” o “El pájaro está en el nido” coincide con la realidad.

Tabla 8.2.2.1: Tabla de tests de funcionalidad (Parte 1).

T-INT-FUN-07 Prototipo, Final	<p>Se utiliza el banco de pruebas #1:</p> <ul style="list-style-type: none"> • Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”. • Desde la terminal de la computadora se insertar las siguientes líneas: <pre>ssh pi@[IP Raspberry Pi] cd simfs/Python_Scripts/Clases python3 print_measures.py</pre>	La diferencia horaria entre las mediciones debe corresponder con lo especificado en la pestaña de “Configuración”.
T-INT-FUN-08 Prototipo, Final	<p>Se utiliza el banco de pruebas #1:</p> <ul style="list-style-type: none"> • Utilizar la computadora para conectarse para conectarse a la red WiFi “SIMFS_HOTSPOT”. • Desde la terminal de la computadora se deben insertar las siguientes líneas: <pre>ssh pi@[IP Raspberry Pi] cd simfs/Python_Scripts python3 getTime.py</pre>	La hora impresa en la terminal coincide con la hora correspondiente al lugar de instalación del producto.
T-INT-FUN-09 Prototipo, Final	<p>Se utiliza el banco de pruebas #1:</p> <ul style="list-style-type: none"> • Se debe utilizar la computadora para conectarse a la red WiFi. “SIMFS_HOTSPOT”. • Desde la terminal de la computadora se deben insertar las siguientes líneas: <pre>ssh pi@[IP Raspberry Pi] cd simfs/Python_Scripts/ python3 getTime.py</pre> <ul style="list-style-type: none"> • Anotar el valor que se encuentra en la terminal. • Desde la terminal de la computadora se deben insertar las siguientes líneas: <pre>sudo shutdown -h now</pre> <ul style="list-style-type: none"> • Esperar 1 minuto. • Renergizar la Raspberry Pi. • Desde la terminal de la computadora se deben insertar las siguientes líneas: <pre>ssh pi@[IP Raspberry Pi] cd simfs/Python_Scripts/ python3 getTime.py</pre>	La hora impresa en la terminal coincide con la hora correspondiente al lugar de instalación del producto incluso luego de la pérdida de energía.

Tabla 8.2.2.2: Tabla de tests de funcionalidad (Parte 2).

ID Aplicabilidad	Procedimiento	Criterio
T-INT-FUN-10 Final	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Se debe posicionar el panel solar de manera tal que a este lo alumbe la luz del sol por completo. • Conectar las baterías al módulo DFR0580 en la entrada BAT IN. • Conectar el panel solar al módulo DFR0580 en la entrada SOLAR IN. • Verificar según las instrucciones del módulo DFR0580 el correcto funcionamiento del regulador • Conectar la R-Pi al módulo DFR0580 en la entrada USB1. • Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”. • Corroborar funcionamiento acorde a los tests anteriores. 	El módulo DFR0580 debe reportar correcto funcionamiento y se debe corroborar el correcto funcionamiento del sistema acorde a los tests anteriores.
T-INT-FUN-11 Final	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Se debe conectar el Simulador de Arreglos de Paneles Solares. • Conectarlo al módulo DFR0580. • Cuando aparezca la red WiFi “SIMFS_HOTSPOT” conectarse. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”. • Corroborar funcionamiento acorde a los tests anteriores. • Bajar gradualmente la tensión provista por el simulador hasta el desabastecimiento, por lo que se apagará el sistema. • Esperar 1 minuto. • Subir gradualmente la tensión provista por el simulador hasta una tensión nominal de 21 V. • Cuando aparezca la red WiFi “SIMFS_HOTSPOT” conectarse. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”. • Corroborar funcionamiento acorde a los tests anteriores. 	El funcionamiento al momento de estar energizado no se ve afectado luego de la pérdida de energía.

Tabla 8.2.2.3: Tabla de tests de funcionalidad (Parte 3).

8.2.3 Test de Comunicación

Precondiciones: Se debe tener un conjunto de datos conocidos para enviar por BLE. Una subrutina para transmisión de datos en el módulo BLE de pruebas.

Procedimiento general: Se debe energizar el módulo BLE. Se debe energizar la Raspberry Pi y conectarse a ella utilizando una computadora conectada a la red SIMFS_HOTSPOT. En la terminal de la computadora colocar:

```
ssh pi@[IP Raspberry Pi]
cd simfs/PythonScripts/
```

ID Aplicabilidad	Procedimiento	Criterio
T-INT-COM1-01	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Se debe comenzar la transmisión del módulo BLE. • Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”. • Dirigirse a la sección “Descarga de datos” y tocar el botón “Descargar datos del ave”. 	El set de datos descargado coincide con los transmitidos.
T-INT-COM1-02	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Colocarse a una distancia de 1 metro del producto. • Desde la terminal comenzar el script “BeaconDetectorFinal”. • Se debe comenzar el test de conexión en el módulo BLE. • Alejarse gradualmente una distancia de 15 metros. 	Obtener dos mensajes de detección durante todo el test.
T-INT-COM2-01	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”. • Desde la terminal moverse al directorio: <pre>cd ./simfs_export_data scp pi@./birdData home/desktop/birdDataFolder</pre> • Dirigirse a la sección “Descarga de datos” y tocar el botón “Descargar datos del ave”. 	Si el archivo descargado coincide con el archivo que se encuentra en la carpeta “birdDataFolder”.
T-INT-COM2-02	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Colocarse a una distancia del nido de 3.3 m. • Utilizar la computadora para conectarse a la red WiFi “SIMFS_HOTSPOT”. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”. • Tocar el botón de la página web llamado “Ver el nido en vivo”. • Una vez iniciada la transmisión comenzar a alejarse hasta una distancia de 17 m. 	No se pierde visión de la transmisión de video en ningún momento.
T-INT-COM2-03	<p>Se utiliza el banco de pruebas #3:</p> <ul style="list-style-type: none"> • Luego de realizar T-INT-COM2-01 dirigirse al mismo directorio y verificar que no existe más el archivo. 	El archivo no existe más.

Tabla 8.2.3.1: Tabla de tests de comunicación.

8.2.4 Test de Peso, Dimensión y Seguridad

ID Aplicabilidad	Procedimiento	Criterio
T-IMP-DIM-01	<p>Se utiliza el banco de pruebas #5:</p> <ul style="list-style-type: none"> • Se utiliza la cinta métrica para medir las dimensiones de tanto el dispositivo del nido como de la unidad de energía. 	<p>El dispositivo del nido no debe exceder las siguientes dimensiones:</p> <p>Largo < 26 cm Ancho < 8,79 cm Alto < 4,55 cm</p> <p>La unidad de energía no debe exceder las siguientes dimensiones:</p> <p>Largo < 1 m Ancho < 50 cm Alto < 1 m</p>
T-IMP-DIM-02	<p>Se utiliza el banco de pruebas #5:</p> <ul style="list-style-type: none"> • Se utiliza la balanza para medir el peso tanto del dispositivo del nido como de la unidad de energía. 	<p>El equipo dentro del nido no debe exceder los 500 g. La unidad de energía no deberá exceder los 20 kg.</p>
T-RAM-SEG-01	<p>Se utiliza el banco de pruebas #1:</p> <ul style="list-style-type: none"> • Se debe energizar la Raspberry Pi y conectarse a ella utilizando una computadora en la red “SIMFS_HOTSPOT”. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880”. • Abrir un navegador de internet y conectarse a la página “192.168.1.101:1880/admin”. 	<p>Cuando se conecta a ambos sitios web se pide autenticación de usuario y contraseña.</p>

Tabla 8.2.4.1: Tabla de tests de peso, dimensión y seguridad.

8.3 Matriz de Trazabilidad de Validación

Origen	REQ ID		
	Descripción corta	ESP ID	TEST ID
	REQ 01		
Cliente	El producto esta colgado de un árbol a una altura de 4 a 14 metros y se instala parcialmente dentro del nido del ave sin comprometer la integridad del árbol.	IMP-DIM-01 IMP-DIM-02 IMP-DIM-03 IMP-DIM-04	T-IMP-DIM-01 T-IMP-DIM-02
	REQ 02		
Cliente	El producto debe mantenerse energizado sin intervención humana.	INT-FUN-14 INT-FUN-16	T-INT-FUN-10 T-INT-FUN-11
	REQ 03		
Táctico	El producto no debe requerir conexión a la red eléctrica para su funcionamiento.	INT-FUN-14	T-INT-FUN-10 T-INT-FUN-11
	REQ 04		
Cliente	El producto debe ser capaz de adquirir los siguientes datos dentro del nido: imágenes, temperatura, humedad, nivel de luz y presencia del ave.	INT-FUN-02 INT-FUN-03 INT-FUN-04 INT-FUN-05 INT-FUN-06 INT-FUN-07 INT-FUN-12 INT-FUN-13	T-INT-FUN-01 T-INT-FUN-02 T-INT-FUN-03 T-INT-FUN-04 T-INT-FUN-05 T-INT-FUN-06 T-INT-FUN-07 T-INT-FUN-12
	REQ 05		
Táctico	El producto debe ser capaz de saber la fecha y hora.	INT-FUN-15 INT-FUN-16	T-INT-FUN-08 T-INT-FUN-09
	REQ 06		
Cliente	Aquellos datos recolectados por dispositivos ajenos al nido deberán ser transmitidos a este por medio del protocolo BLE.	INT-COM1-01 INT-COM1-03	T-INT-COM1-01 T-INT-COM1-02
	REQ 08		
Cliente	Una persona debe recibir los datos almacenados en el nido a la distancia a través de una computadora.	INT-COM2-01 INT-COM2-02 INT-COM2-05	T-INT-COM2-01 T-INT-COM2-02 T-INT-COM2-03

Tabla 8.3.1: Matriz de trazabilidad de validación.

8.4 Plan de Verificación y Validación

Las pruebas de verificación y validación se realizan con el propósito de verificar que los conceptos de interés pueden ser llevados adelante de manera satisfactoria. Se emplean también para verificar que un módulo o sistema cumpla con todos los requerimientos, necesidades y especificaciones requeridas.

En la Figura (8.4.1) se presenta dicho plan, mostrando además las dependencias entre las distintas pruebas realizadas, con sus respectivos bancos de prueba. Además, se distinguen las tres etapas de validación, siendo estas las “Pruebas básicas”, las “Pruebas de Prototipo” y las “Pruebas de Producto Final”.

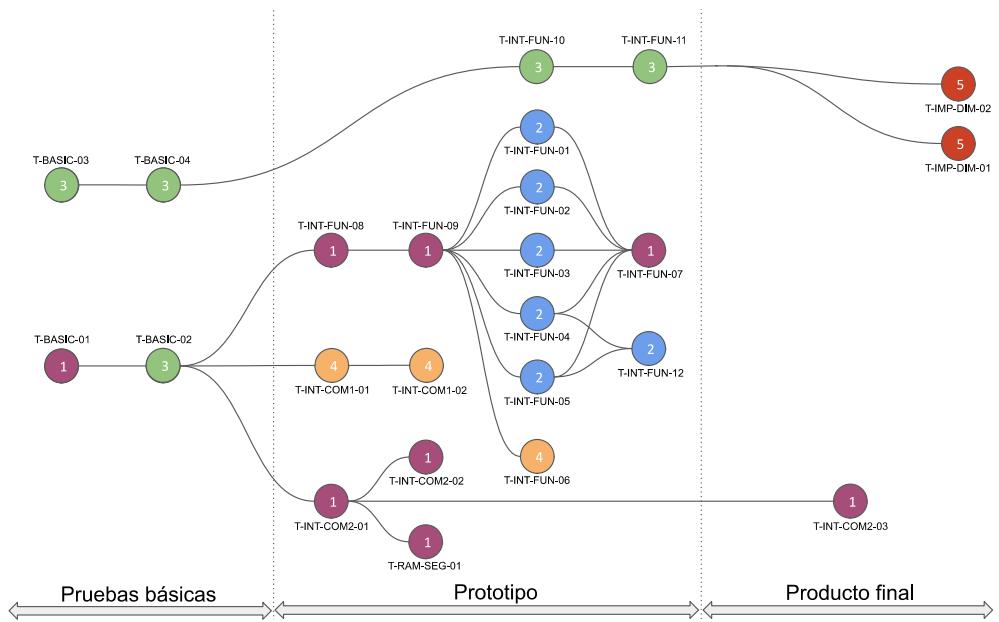


Figura 8.4.1: Diagrama de dependencias del plan de validación.

9. Análisis de Factibilidad

9.1 Factibilidad Tecnológica

9.1.1 Esquema Modular

A continuación, se presentan los distintos módulos que integran el proyecto. Estos fueron agrupados de forma tal que se puedan comunicar de manera ordenada y precisa entre sí.

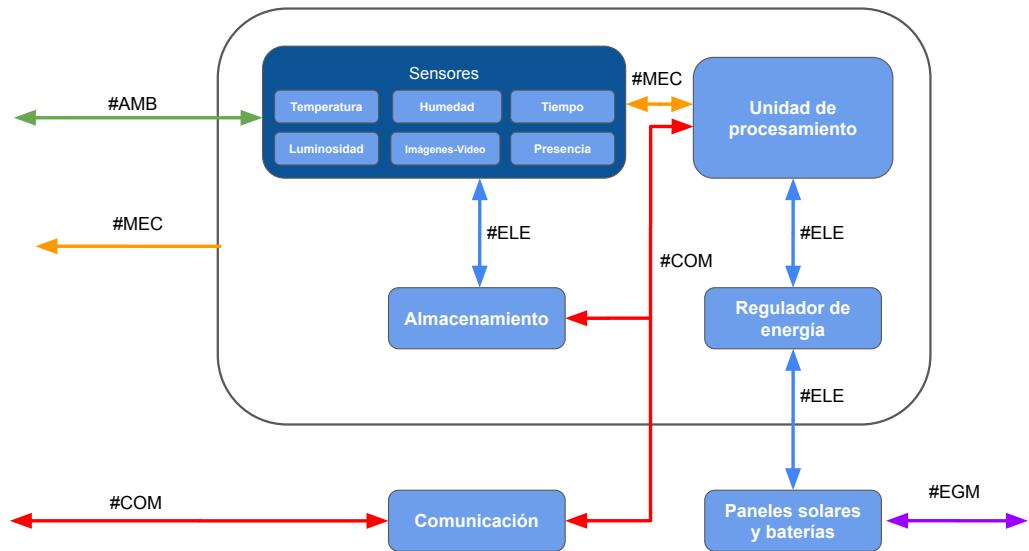


Figura 9.1.1.1: Diagrama modular del sistema.

Además, para un mejor entendimiento, se subdividió el módulo de “Sensores”, ya que de esta forma se interpreta mejor su funcionamiento.

Existen distintas alternativas al momento del desarrollar e implementar los nodos empleados en el trabajo. Estas variedades de implementación se analizan con más profundidad en las siguientes secciones.

9.1.2 Propuesta de Sensores

Para las distintas mediciones se tuvieron en cuenta diversas tecnologías que existen. Se evaluaron parámetros que definen la performance, tales como la linealidad de salida, el costo, el rango de operación, la precisión, el tipo de salida, aplicación, entre otras tantas variables.

9.1.2.1 Temperatura

En el caso de la medición de temperatura, se valoraron diversas tecnologías que existen, siendo por ejemplo la RTD cuyo funcionamiento se basa en el cambio de la resistencia en función de la temperatura bajo al ecuación $R(T) = R_0 + \alpha \cdot \Delta T$. También se consideró la tecnología TC, cuyo funcionamiento se basa en el efecto Seebeck. Finalmente, el uso de un IC, el cual se basa en propiedades de dispositivos semiconductores extrínsecos.

Aspectos comparativos	TC-K	PT-100	Ds18b20	DHT-22
Costo [USD]	4.6	5.2	1.4	4.9
Tipo de salida	Analógico	Analógico	Digital	Digital
Temperatura de operación [°C]	-40 ~ 1200	-50 ~ 200	-10 ~ 85	-40 ~ 80
Interfaz de conexionado	Se debe proporcionar un circuito amplificador	Se debe proporcionar un circuito convertidor de resistencia a tensión	-	-
Precisión [°C]	± 1.5	± 0.1	± 0.5	± 0.5
Estabilidad	Tienden a envejecer	-	-	-
Autocalentamiento	-	Depende de la corriente de medición.	Bajo	Bajo
Imagen				

Tabla 9.1.2.1.1: Comparación entre sensores de temperatura.

9.1.2.2 Humedad

Existen varias maneras de medir la magnitud física de la humedad, dentro de estas la más común se basa en utilizar la dependencia que existe entre la humedad y la capacidad. Es por esto que se utilizan capacitores con un dieléctrico, el cual cambia constante con la humedad. Además, existen sensores que se aprovechan de cómo cambia la resistencia en función de la temperatura, pero estas tecnologías son menos frecuentes.

Aspectos comparativos	DHT-11	AM-2301	DHT-22	AM-1001
Costo [USD]	1.3	7	4.93	5.6
Rango de operación [%RH]	20 ~ 90	0 ~ 100	0 ~ 100	20 ~ 90
Precisión [%RH]	±4	±3	±2	±5
Tipo de salida	Digital	Digital	Digital	Analógica
Imagen				

Tabla 9.1.2.2.1: Comparación de sensores de humedad.

9.1.2.3 Luminosidad

En la medición del nivel de luminosidad se puede optar por diversos caminos. Existen sensores como el BH-1750 y OPT-100 que su funcionamiento se basa en un fotodiodo que conduce cierta corriente a partir de la luz que le impacta. Otros sensores, tales como el TEMT-600, emplean un fototransistor, cuya base se encuentra expuesta. En función de la intensidad lumínica en dicha zona, circulará cierta corriente por el colector. Finalmente existen fotoresistores, los cuales, tal como su nombre indica, cambian la resistencia en función del nivel de luz.

Aspectos comparativos	BH-1750	TEMT-6000	OPT-101	GL55-LM393
Costo [USD]	1.54	2.27	2.26	1.26
Temperatura de operación [°C]	-40 ~ 85	-40 ~ 85	0 ~ 70	-30 ~ 70
Potencia disipada [mW]	-	100	-	75
Tipo de salida	I2C	Analógica (Corriente)	Analógica (Tensión)	Analógica Digital
Aplicación	-	Necesita un amplificador de corriente	-	-
Tensión de alimentación [V]	2.4~3.6	< 6	2.7 ~ 36.0	3.3 ~ 5.0
Rango de medición [nm]	450 ~ 650	400 ~ 900	450 ~ 1000	450 ~ 750
Imagen				

Tabla 9.1.2.3.1: Comparación de sensores de luminosidad.

9.1.2.4 Imágenes y video

Para la obtención de captura imágenes como de video, teniendo en cuenta la tecnología utilizada para la unidad de procesamiento, se encontraron diversos módulos de cámara que se pueden usar:

Aspectos comparativos	RPi-CMOD-V1	RPi-CMOD-V2	RPi-HQC	RPi-ZEROc
Costo [USD]	25	25	50	20
Tamaño [mm]	25 x 24 x 9	25 x 24 x 9	38 x 38 x 18.4	8.6 x 8.6 x 5.2
Resolución de la cámara [MP]	5	8	12	5
Integración Linux	V4L2 driver	V4L2 driver	V4L2 driver	V4L2 driver
Capacidad nocturna	No	Filtro	No	No
Peso [g]	3	3.4	12	1.1
Sensor	OmniVision OV5647	Sony IMX219	Sony IMX219	OV5647
Temperatura de operación [°C]	-25~80	-25~80	-25~80	-25~80
Imagen				

Tabla 9.1.2.4.1: Comparación entre cámaras.

9.1.2.5 Tiempo

Es fundamental contar con un método para conocer la fecha y hora de manera confiable. Tanto como para saber en qué momento fueron tomadas las mediciones. Teniendo esto en cuenta se proponen los siguientes módulos RTC:

Aspectos comparativos	Ds3231	Ds1302	Ds1307
Costo [USD]	2.33	0.81	1.37
Temperatura de Operación	-40 ~85	-40 ~85	-40 ~85
Rango de tensiones	2.3 ~5.5	2 ~5.5	4.5~5.5
Protocolo de Comunicación	I2C	SPI	I2C
Precisión	3.5ppm	-	-
Imagen			

Tabla 9.1.2.5.1: Comparación entre módulos RTC.

9.1.2.6 Presencia

Se desea conocer cuándo está presente el ave dentro del nido. Para esto, se decide utilizar BLE por su baja utilización de recursos energéticos. Una opción es utilizar la R-Pi como receptora. Otras opciones que se exploran involucran utilizar un transceptor² *Bluetooth*.

²Un transceptor es aquel cuyo se encuentra compuesto con un transmisor y un receptor, compartiendo circuito y/o encapsulado.

Aspectos comparativos	RPI 3B	NRF24L01	WIM2480	ESP32-S3-WROOM-1-N4
Costo [USD]	-	3	10.77	3.25
Compatibilidad Bluetooth	Bluetooth 4.1	Bluetooth 4.1	Bluetooth 5.2	Bluetooth 5.0
BLE	Sí	Sí	Sí	Sí
Sensibilidad Rx [dBm]	-	-85	-92	-103.5
Imagen				

Tabla 9.1.2.6.1: Comparación de módulos para la detección de presencia.

9.1.3 Propuesta de Almacenamiento

Para almacenar información, se considera la utilización memorias SD debido a su compacto tamaño comparado a la densidad de información que puede contener. Existe una gran variedad de memorias SD, permitiendo priorizar diversos aspectos a la hora de optar por una opción. La velocidad de lectura, la de escritura y el almacenamiento son algunos de estos aspectos, aunque en este proyecto también es importante considerar el rango de temperatura de operación.

Aspectos comparativos	SDCG3	SDCE	SDSDQAF3-XI
Costo [USD]	20	52	22
Temperatura de operación [°C]	-25 ~ 85	-25 ~ 85	-40 ~ 85
Almacenamiento [GB]	64 ~ 512	64 ~ 256	8 ~ 128
Velocidad R/W [MB/s]	170 / 90	285 / 165	50 / 80
Alimentación [V]	3.3	3.3	2.7 ~ 3.6
Imagen			

Tabla 9.1.3.1: Comparación entre memorias SD.

9.1.4 Propuesta de Unidad de Procesamiento

La UP representa el cerebro del proyecto. Este se ocupa de procesar la información de los sensores, almacenarla en la SD e iniciar los procesos de comunicación. En otras palabras, el integrado se ocupa de conectar los distintos módulos entre sí y garantizar su adecuado funcionamiento.

Aspectos comparativos	RPi 4	RPi Zero W	RPi 3B
Costo [USD]	235	70	130
Temperatura de operación [°C]	0 ~ 50	-20 ~ 85	-20 ~ 85
Memoria	1 [GB] ~ 8 [GB]	512 [MB]	1 [GB] ~ 8 [GB]
Conexiones	Wireless LAN, Bluetooth 5.0, Ethernet, USB	Wireless LAN, Bluetooth 4.1 (BLE), Micro USB, mini HDMI	Wireless LAN, Bluetooth 5.0 (BLE), Ethernet, USB, antena externa
Sonido y video	Micro HDMI, MIPI DSI y CSI	Mini HDMI, HDMI, CSI, PAL/NTSC pads	HDMI, MIPI DSI y CSI, SDIO
Soporte SD	Almacenamiento y carga de SO	Micro SD	Entrada SD para tarjeta o eMMC externo
Dimensiones [mm]	85.6 x 56.5	65 x 30	40 x 55
Alimentación	5 V (3 A)	5 (1.2 A)	5 V (1.4 A)
Imagen			

Tabla 9.1.4.1: Comparación entre placas Raspberry Pi.

9.1.5 Propuesta de Batería

Para la elección de las baterías, se evaluaron unidades de tecnología de gel de carga profunda.

Aspectos comparativos	Kijo Serie JDG	Ultracell UL24-12	Fenk JS12-20	Probattery BVLA-12280E-CPB
Costo [USD]	102 ~ X	86	22 ~ 29	82
Temperatura de operación [°C]	-20 ~ 50	0 ~ 40	-20 ~ 50	0 ~ 40
Tensión nominal [V]	12	12	12	12
Capacidad nominal [Ah]	33 ~ 250	24	13.2 ~ 20.0	28
Dimensiones (máximas) [mm]	520 x 268 x 220	166.5 x 175 x 125	181 x 77 x 167	166 x 175 x 125
Peso [kg]	10 ~ 61	7.2	5.45	11.467
Porcentaje de autodescarga (mensual a 25 °C)	3 %	- %	3 %	- %
Imagen				

Tabla 9.1.5.1: Comparación entre baterías gel de carga profunda.

9.1.6 Propuesta de Paneles solares

Para poder abastecer a todos los módulos anteriormente mencionados, es necesario la existencia de un módulo que brinde dicha energía. Dadas la ubicación remota donde se encontrará el producto final, se opta por emplear un panel solar, capaz de obtener energía del entorno y no de la red eléctrica.

Lo que principalmente determinará la elección de este componente es el consumo de las demás partes.

Aspectos comparativos	DSP-20P	DSP-30M	LN-50P	ESPMC210	LNSE-260P
Costo [USD]	17	25	30	78	72.5
Temperatura de operación [°C]	-45 ~ 85	-45 ~ 85	-45 ~ 85	-40 ~ 85	-45 ~ 85
Potencia máxima [W]	20 ± 3 %	30 ± 3 %	50 ± 3 %	210	260
Tensión a potencia máxima [V]	17.6	18.0	18.0	18.85	30.4
Corriente a potencia máxima [A]	1.14	1.67	2.78	11.15	8.55
Tensión a circuito abierto máxima [V]	22.0	21.5	22.3	23.2	37.4
Corriente a corto circuito máxima [A]	1.39	1.86	3.01	11.8	9.11

Tabla 9.1.6.1: Comparación entre paneles solares.

9.1.7 Regulador de energía

En cuanto a la carga de la batería principal a partir de los paneles solares se tuvieron en cuenta los siguientes integrados

Aspecto comparativo	DFR0580	P1698	CN3306
Costo [USD]	18	26	11
Temperatura de operación	-40°C ~ 85°C	-35°C ~ 60°C	-40°C ~ 85°C
Tensión de operación [V]	6.6 ~ 30	12 / 24	4.5 ~ 32
Tecnología de batería	Lead-Acid	Lead-Acid	Multi-Chemistry
Máxima corriente salida [A]	4	2/10	1.5
MPPT	Si	No	Si
Imagen			

Tabla 9.1.7.1: Comparativa reguladores de energía.

9.1.8 Primer Propuesta de Abastecimiento de Energía de UBM

La primera solución planteada para el abastecimiento de energía de la UBM se basa en la transferencia de energía al dispositivo del ave por medio de radiación electromagnética. A continuación, se detalla el análisis de factibilidad tecnológica realizado y las conclusiones tomadas de esta propuesta.

9.1.8.1 Planteamiento del Problema

La morfología del nido plantea la necesidad de recargar la UBM a distancias de entre 32 y 64 cm. Esto es dictado por las mínimas y máximas dimensiones medidas del nido entre la bóveda, donde se puede colocar electrónica, y el fondo, donde duerme el ave [1].

En el mejor de los casos, el pájaro carpintero permanece por ocho horas en el nido mientras duerme, y luego cuida a las crías turnándose con la hembra durante el día. Por lo tanto, este permanecería un total de aproximadamente doce horas en el nido.

En el peor de los casos, el ave únicamente permanece seis horas dentro del nido para dormir. La UBM tiene un consumo diario de

$$2.5 \text{ V} \cdot 1 \text{ mA} \cdot 24 \text{ hs} = 60 \text{ mWh} \quad (1)$$

por lo que, tomando ambos casos, se obtiene un requerimiento de transmisión de potencia de

$$\begin{cases} \frac{60 \text{ mWh}}{11 \text{ hs}} = 5.5 \text{ mW} & \text{Mejor caso} \\ \frac{60 \text{ mWh}}{8.5 \text{ hs}} = 7 \text{ mW} & \text{Peor caso} \end{cases} \quad (2)$$

9.1.8.2 Carga por Acoplamiento Magnético

La transmisión inalámbrica de potencia por acoplamiento magnético se basa en generar un campo magnético al hacer circular corriente por un arreglo de bobinas gracias a la Ley de Ampere.

$$\nabla \times \mathbf{B} = \mu_0 \left(\mathbf{J} + \epsilon_0 \frac{\partial \mathbf{B}}{\partial t} \right) \quad (3)$$

Este campo magnético es captado por una o más bobinas receptoras las cuales generan a causa de este una fuerza electromotriz según la Ley de Faraday.

$$\nabla \times \mathbf{E} = - \frac{\partial \mathbf{B}}{\partial t} \quad (4)$$

De esta manera, se genera un sistema que actúa como transformador, utilizando como reluctancia al aire que separa ambas bobinas, con un factor de acople k definido como

$$k = \frac{M}{\sqrt{L_1 L_2}} \quad (5)$$

donde M es el coeficiente de mutua-inductancia entre las bobinas y L_1 y L_2 las auto-inductancias de las bobinas.

La eficiencia en la transmisión de energía de este método es alta [2], pero depende en gran medida por el factor de acople entre ambas bobinas. Este factor de acople disminuye considerablemente con la distancia y la desalineación entre bobinas [3] [4].

Como las distancias a las que la transmisión se debe efectuar son de entre 32 y 64 cm y no se puede garantizar alineación entre bobinas al estar sujeto al comportamiento impredecible del ave, se descarta este método como solución.

9.1.8.3 Carga por Radiofrecuencia

Si se parte de la ecuación del campo magnético en el eje azimutal de un dipolo de Hertz, se tiene que

$$E_\theta = -\frac{\eta}{4\pi} I \cdot \Delta L \cdot k^2 \cdot \sin\theta \cdot e^{-jkr} \left[\frac{1}{jkr} + \left(\frac{1}{jkr} \right)^2 + \left(\frac{1}{jkr} \right)^3 \right] \quad (6)$$

donde los últimos tres términos se denominan, en orden de aparición, término de campo lejano, campo cercano radiativo, y campo cercano reactivo.

Las fronteras entre estos campos no están estrictamente fijadas, ya que varían con el tipo y tamaño de antena. Para el caso de antenas eléctricamente cortas, es decir, más cortas que media longitud de onda, se adopta el siguiente criterio

$$\begin{cases} 0 < d < \frac{\lambda}{2\pi} & \text{Campo cercano reactivo o inductivo} \\ \frac{\lambda}{2\pi} \approx d \approx \lambda & \text{Campo cercano radiativo o de Fresnel} \\ \lambda \approx d \approx 2\lambda & \text{Zona de transición} \\ 2\lambda \approx d < \infty & \text{Campo lejano o de Fraunhofer} \end{cases} \quad (7)$$

La zona de campo cercano puede dividirse entre la zona reactiva-inductiva y la zona radiativa o de Fresnel.

En la zona reactiva la relación entre los campos eléctricos y magnéticos no es predecible. Además, como no solo hay ondas electromagnéticas siendo irradiadas en esta zona, sino que también hay una cierta cantidad de energía siendo almacenada en la cercanía de la antena, la verdadera densidad de potencia se torna difícil de encontrar.

En el caso de la zona radiativa o de Fresnel, toda la energía es radiada. Sin embargo, la relación entre el campo eléctrico y magnético sigue siendo impredecible.

A una distancia entre una y dos longitudes de onda, los efectos de campo cercano comienzan a cesar, mientras que los efectos de campo lejano comienzan a aparecer. Por lo tanto, es en esta zona que ambos efectos están presentes y tienen importancia [5]. Los dispositivos RFID suelen operar en esta zona [6].

Por otro lado, el campo lejano es el utilizado para realizar todo tipo de telecomunicaciones hoy en día. En esta zona, el campo eléctrico y campo magnético son ortogonales y la razón entre ambos es la impedancia del medio. Además, el vector de Poynting, definido como $\vec{S} = \vec{E} \times \vec{H}$, provee una medida de la energía electromagnética radiada [7].

Para analizar la potencia recibida en la antena receptora, la cual estará montada en la mochila, se realiza el balance de potencias del circuito electromagnético, por lo que partiendo de la ecuación de transmisión de Friis, se tiene que

$$\frac{P_r}{P_t} = \left(\frac{A_r A_r}{d^2 \lambda^2} \right) \quad (8)$$

reescribiendo esta fórmula para utilizar las ganancias de las antenas en vez de las áreas efectivas, e incluyendo otras pérdidas del circuito electromagnético, se arriba a

$$P_r[dBm] = P_t[dBm] + G_t[dB] + G_r[dB] - L_{bf}[dB] - L_{cab}[dB] - L_{roe}[dB] - L_r[dB] \quad (9)$$

donde P_r es la potencia recibida en la antena receptora, P_t la potencia emitida por la antena transmisora, G_t la ganancia de la antena transmisora, G_r la ganancia de la antena receptora, L_{bf} las pérdidas por espacio libre, L_{cab} las pérdidas en los cables de ambas antenas, L_{roe} las pérdidas por retorno en ambas antenas, y L_r las pérdidas por desacople entre las líneas de transmisión y las antenas.

Para el caso de la pérdida por espacio libre, esta se puede calcular como

$$L_{bf} = 32.5dB + 20\log_{10}f[MHz] + 20\log_{10}R[km] \quad (10)$$

mientras que el resto de los datos se puede obtener por medio de las hojas de datos o ensayos de las antenas, exceptuando la potencia a ser calculada y las pérdidas por desacople, las cuales dependen constructivamente del diseño de las líneas de transmisión que conectan con las antenas.

9.1.8.4 Banda de Frecuencia Adoptada

Se puede observar que las pérdidas de espacio libre aumentan mediante crece la frecuencia de la onda electromagnética emitida. Esto plantea una situación de compromiso. Si la frecuencia es muy alta, las pérdidas por espacio libre serán muy grandes. Mientras que, si es muy baja, la longitud de onda será muy grande, por lo que se estaría trabajando en el campo reactivo. Esto no es deseado debido a la imposibilidad de determinar con precisión la densidad de potencia, y la gran pérdida de potencia con la distancia.

Se decidió utilizar la banda de 915 MHz por las siguientes razones:

- La zona de transición ocurre entre 32.8 y 64.6 cm que concuerda con las distancias mínimas y máximas entre emisor y receptor. Si bien no estaremos trabajando en campo lejano, esto no acarrea problemas, ya que hay basta cantidad de antecedentes del uso de los efectos de esta zona en dispositivos comerciales [6] [8].
- Esta frecuencia pertenece a la banda ISM, la cual está reservada para propósitos industriales, científicos o médicos, excluyendo las aplicaciones de telecomunicaciones [9], atribuidas a la Región 2 definida por la ITU como América [10].

9.1.8.5 Condiciones de Borde

Para realizar las comparaciones entre antenas transmisoras, se tuvieron en cuenta los siguientes criterios:

- **Dimensiones:** La antena transmisora deberá ser colocada en la bóveda del nido, dado que ese es el único lugar donde se puede colocar electrónica sin que esta sea perturbada por las aves y viceversa. El volumen de la bóveda se puede aproximar al de un cilindro macizo chato de diámetro entre 7.9 y 9.7 cm y aproximadamente 5 cm de altura, por lo que las dimensiones de la antena emisora estarán acotadas por estos valores.
- **Directividad:** Se quiere que la potencia enviada a la antena se transforme en radiación electromagnética que llegue a la mochila del ave, por lo que radiación que no sea dirigida directamente hacia el fondo del nido será potencia desperdiciada. Es por esto que se quiere una alta directividad en la antena emisora. También

hay un límite máximo en la directividad de la antena. Sin embargo, esta limitación no se alcanzará, dado que la tecnología a utilizar será de antenas del tipo planas, más cortas eléctricamente que media longitud de onda por cuestiones de limitaciones en las dimensiones.

- **Potencia Máxima:** Como las pérdidas en el circuito electromagnético son grandes, una muy baja parte de la potencia enviada a la antena transmisora formará parte de la potencia entregada a las baterías de la mochila, por lo que, para recibir la potencia necesaria, se debe transmitir en el orden de los watts. Es por esto que la potencia máxima es una especificación relevante al momento de decidir entre soluciones.

Mientras que, para el caso de las antenas receptoras, se tuvieron en cuenta los siguientes criterios:

- **Eficiencia:** Esta será la especificación más importante y es la que determinará la factibilidad de la solución. Se buscará la mayor eficiencia posible para lograr transmitir a las baterías la potencia recibida por el campo electromagnético.
- **Lóbulo isotrópico:** Como se desconoce cuál será la posición del ave dentro del nido, se requiere que el lóbulo de radiación de la antena receptora sea lo más isotrópico posible, garantizando una recepción de potencia uniforme sin importar la posición del ave. Se buscará una ganancia menor a 2 dBi .
- **Peso:** El ave no puede cargar con más de un cierto porcentaje de su propio peso, por lo que minimizar esta especificación es crucial.
- **Dimensiones:** Es necesario no perturbar al ave con la mochila. Esto requiere que la antena receptora posea las mínimas dimensiones posibles. Sin embargo, como la banda a utilizar será la de 915 MHz y un cuarto de onda en esta frecuencia es alrededor de 8 cm , existe una relación de compromiso entre las dimensiones de la antena receptora y la eficiencia de esta.

Finalmente, una restricción a tener en cuenta para ambas antenas será el costo.

9.1.8.6 Interacción entre las Aves y la Radiación Electromagnética

Se recopiló información sobre experimentos donde se ha evaluado las consecuencias de exponer distintos tipos de aves frente a ondas electromagnéticas.

Esta radiación produce dos tipos de efectos sobre los tejidos biológicos, clasificándose en térmicos y no térmicos. Los primeros se manifiestan como un aumento de la temperatura del sistema que recibe la emisión. Dependiendo de la duración y la intensidad de la exposición, se pueden observar respuestas físicas o psicológicas. Por otro lado, los no térmicos se manifiestan como cambios en el metabolismo celular. Se observan cambios en los comportamientos de los individuos expuestos cuando se comprometen estructuras del sistema nervioso. La principal diferencia entre estos dos procesos se centra en el tiempo de exposición.

Se ha corroborado que gallinas expuestas a ondas en un rango de densidad de potencia de 200 a 500 W/m^2 han reaccionado evitando o escapando de este campo al poco tiempo de exposición [11].

En un segundo estudio realizado sobre las especies *Cyanocitta Cristata* y *Haemorhous Mexicanus* determina un nivel máximo de densidad de potencia aplicada sobre estas aves, siendo este de 230 W/m^2 . La primera especie tiene un tamaño que oscila entre los 22 y los 30 centímetros, y un peso entre 70 y 100 gramos. Para la segunda especie, su largo varía entre 12.5 y 15 centímetros, mientras que en cuanto a peso varía entre 16 y 27 gramos. Esto se remarca porque las aves expuestas a la misma densidad mencionada pueden afrontar mayores riesgos si poseen tamaños superiores, siendo este el caso del *Campephilus Magellanicus* [12].

En otro estudio similar al primero, se radió a gallinas, palomas y gaviotas con una densidad de 100 a 300 W/m^2 , las cuales se encontraban encerradas dentro de cajas. Se dispuso un arreglo que permitiera posicionar la antena tanto por encima como por debajo del especie de estudio. Además, se repitió la experiencia primero cubriendo la cabeza del ave y luego su cuerpo.

Se observó para las gallinas que, al colocar la antena por encima de ellas, a los pocos segundos los animales comenzaron a realizar movimientos de extensión de todas sus extremidades. Se atribuyó dicha reacción a la penetración directa sobre la médula espinal, ya que al repetir el experimento radiando desde abajo del ave se notaron sobresaltos de los especímenes pero sin tantos signos de disturbios musculares. No se notaron cambios al cubrir la cabeza y el cuerpo.

Para las palomas y gaviotas se obtuvieron resultados similares pero menos dramáticos. Es importante remarcar que el tamaño de estas dos últimas es menor al de las gallinas, siendo consistente con los resultados obtenidos en el segundo informe. Todas las aves de estudio mostraron que sus alas no retornaron a su posición normal de retracción por un período mayor o igual a una hora [13].

9.1.8.7 Esquema de Conexionado

El bloque del cargador de la UBM consiste en un receptor y un transmisor de potencia. La transmisión inalámbrica consta, por el lado del transmisor, de un oscilador HM-TRPW-RS232 de 915 MHz el cual está comandado por la R-Pi y se comunica mediante UART; como así también de un amplificador de potencia alimentado por una etapa DC-DC XI6009 de 12 V a 15 V.

Por el lado del receptor, se encuentra el integrado P1110B, el cual almacena energía temporalmente en un capacitor para realizar posteriormente la carga de la UBM.

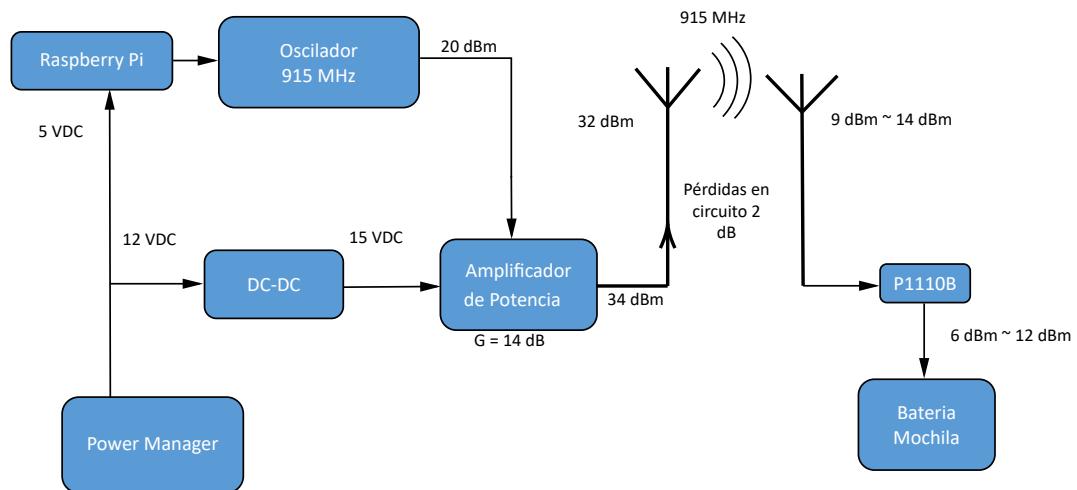


Figura 9.1.8.7.1: Diagrama en bloques cargador.

Para la trasmisión de la señal de 915 MHz se utiliza un cable del tipo RG-213. Este se caracteriza por poseer bajas pérdidas de $23.054 \text{ dB}/100\text{m}$.



Figura 9.1.8.7.2: Módulo utilizado como oscilador HM-TRPW-RS232.

Debido a la falta de stock mundial de componentes electrónicos a causa de la pandemia, se utilizó un módulo transciever FSK que opera en la banda de 915MHz. Este módulo HM-TRPW-RS232 puede generar la señal carrier de hasta 20 dBm y posee interfaz RS-232 para la comunicación UART.

- Potencia máxima de 20 dBm.
- Alimentación 100 mA@20 dBm.
- Corriente suspendido 1 μA .
- Velocidad de comunicación 1.2 kbps – 115.2 kbps.
- Dimensiones 44.1 × 30 × 1.2 mm.

Para el amplificador de potencia de RF, como las unidades a producir serán muy pocas, no se justifican las horas necesarias para diseñar el circuito y la placa impresa de un amplificador de RF. Por esta razón, se utilizará uno comercial.



Figura 9.1.8.7.3: Módulo utilizado como *power harvester*.

Para la recepción de la radiofrecuencia se utiliza el *power harvester* P1110B. Este integrado toma la señal de radiofrecuencia y realiza con esta la carga de un capacitor. Cuando el capacitor llega a un valor de tensión máximo, habilita la salida mediante una etapa DC-DC hasta que el capacitor baje su tensión por debajo de un mínimo para volver a cargarse hasta el máximo.

- Eficiencia del 60 %.
- Adaptado internamente a 50Ω .
- Operación por encima de -5 dBm .
- Opción de carga para Li-ion y baterías alcalinas.

9.1.8.8 Amplificador de Potencia

Para el amplificador de potencia se tuvieron en cuenta distintos módulos armados.

Aspectos comparativos	NWDZ RF PA V2.0	Unbranded Broadband RF Amplifier	DH-RF V2017	JP-2Y560895 RF Amplifier
Costo [USD]	18.59	9.82	23	69.31
Alimentación [V]	12-15	12	15	24-28
Frecuencia [MHz]	2-700	1-930	1-1000	915 +- 25
Potencia salida máxima [dBm]	34.8	29	35	42
Potencia entrada máxima [dBm]	10	0	0	23
Ganancia de potencia [dB]	35	29	32	Ajustable
Dimensiones [cm]	7x3	5x5	3.7x5.6	4x8
Imagen				

Tabla 9.1.8.8.1: Comparativa amplificadores de potencia.

9.1.8.9 Integrado de Energy Harvesting

Se investigaron diversos métodos de transmisión de energía inalámbrica a través de campos electromagnéticos. Dentro de estos se en cuenta el campo cercano y el de radiofrecuencia. El primero se caracteriza por ser

puramente imaginario y por poseer una caída proporcional al cuadrado de la distancia, mientras que el segundo (radiación) cae linealmente.

Debido a que la aplicación es de una distancia que cae en el rango del campo lejano, se optó por la radiofrecuencia. Es así como surgieron los integrados IC-P2110 y IC-P1110 de PowerCast, los cuales permiten la recolección de energía de radio frecuencia almacenando está en capacitores o con la opción de directamente cargar una batería (P1110). Se analiza a modo comparativo ambos IC en la Tabla (9.1.8.9.1).

Aspectos comparativos	IC-P1110	IC-P2110
Costo [USD]	48.33	32
Funcionalidad principal	Recolección y almacenamiento de energía para uso variado	Recolección de energía para carga de baterías/Capacitores
Frecuencia de trabajo [MHz]	910 ~ 928	910 ~ 928
Eficiencia del PH para RFin = 11 dBm	60 %	45 %
Corriente de salida para RFin = 11 dBm	3 mA	-
Tiempo de carga inicial del capacitor [s]	-	< 5
Posee placa de evaluación	Sí	Sí
Impedancia de entrada [Ω]	50	50
Temperatura de operación [°C]	-40 ~ 85	-40 ~ 85

Tabla 9.1.8.9.1: Comparación entre cargadores.

9.1.8.10 Comparación entre Antenas

Aspectos comparativos	APAE915R2540ABDB1-T	W3215	ISPC.91A.09.0092E
Costo [USD]	3.66	12.47	20.91
Dimensiones [mm]	25 x 25 x 4	40 x 40 x 6	47 x 47 x 6.5
Frecuencia Central [MHz]	915	915	915
Impedancia [Ω]	50	50	50
Polarización	RHCP	Lineal vertical	RHCP
Ganancia [dBi]	1.5	4.5	5 (30 x 30 ground plane)
ROE	1.5	1.23	1.28
Imagen			
Lóbulo			

Tabla 9.1.8.10.1: Comparación entre antenas transmisoras (Parte 1).

Aspectos comparativos	APAES915R80C16-T	ARRKP7059-S915B
Costo [USD]	34.28	50.73
Dimensiones [mm]	80 x 80 x 6	70 x 70 x 5.9
Frecuencia Central [MHz]	915	915
Impedancia [Ω]	50	50
Polarización	RHCP	RHCP
Ganancia [dBi]	2 (120 x 120 ground plane)	2.8 (70 x 70 ground plane)
ROE	1.3	≤ 2
Imagen		
Lóbulo		

Tabla 9.1.8.10.2: Comparación entre antenas transmisoras (Parte 2).

Aspectos comparativos	ANT-915-CPA	FPX290.07.0100A	NN01-105
Costo [USD]	3.9	15.39	3.53
Dimensiones [mm]	25 x 25 x 4	70 x 45 x 0.1	18 x 7.3 x 0.8
Frecuencia Central [MHz]	915	915	915
Impedancia [Ω]	50	50	50
Polarización	RHCP	Lineal	Lineal
Ganancia [dBi]	1.5	0.5	1.7
Eficiencia [%]	38	43	85
ROE	≤ 1.2	1.5	1.4
Peso [gr]	13.2	1.5	0.2
Imagen			
Lóbulo			"Omnidireccional"

Tabla 9.1.8.10.3: Comparación entre antenas receptoras (Parte 1).

Aspectos comparativos	ANT1204F005R0915A	1513156-1	ANT-915-USP410
Costo [USD]	1.52	2.8	1.45
Dimensiones [mm]	12.20 x 4 x 1.6	38.1 x 6.6 x 1.57	13.2 x 9.1 x 2.9
Frecuencia Central [MHz]	915	915	915
Impedancia [Ω]	50	50	50
Polarización	Lineal	Lineal	Lineal
Ganancia [dBi]	1.59	1	0
Eficiencia [%]	-	88	27
ROE	-	1.85	1.5
Peso [gr]	-	< 0.9	0.6
Imagen			
Lóbulo			

Tabla 9.1.8.10.4: Comparación entre antenas receptoras (Parte 2).

9.1.8.11 Selección de Componentes

Para el integrado de *energy harvesting* se optó por el P1110 debido a su elevada eficiencia en la zona donde se transmitirá potencia.

Por un lado, para el caso de las antenas transmisoras, se descartan APAE915R2540ABDB1-T, APAES915R80C16-T y ARRKP7059-S915B por su baja ganancia, además del alto costo de las últimas dos. También se descarta W3215 por ser de polarización lineal. Por lo que resta la antena **ISPC.91A.09.0092E**.

Por otro lado, para las antenas receptoras, se descarta ANT-915-USP410 y ANT-915-CPA por su baja eficiencia. Se quita FXP290.07.0100A por ser demasiado grande. Se descarta ANT1204F005R0915A por especificaciones poco claras. Entre las restantes se escoge **NN01-105** por su tamaño a pesar de ser de polarización lineal, lo cual se deberá tener en cuenta en los cálculos, además de tener una potencia máxima de 5 W.

9.1.8.12 Análisis Cuantitativo de la Recarga Inalámbrica

Utilizando las antenas ISPC.91A.09.0092E como transmisora y NN01-105 como receptora, así como el *energy harvester* elegido, y siguiendo el esquema de conexión planteado anteriormente, se calcula la potencia recibida en las baterías de la mochila del ave según las Ecuaciones (9.1.8.3) y (9.1.8.3). Se tiene en cuenta que la potencia en la antena transmisora es la máxima que esta admite de 5 W.

Además, debido a que la antena transmisora es de polarización circular derecha y la antena receptora es de polarización lineal, se debe tener en cuenta que la eficiencia equivalente de la antena receptora será la mitad. Estas especificaciones se resumen en la Tabla (9.1.8.12.1).

Especificación	Valor
Potencia transmitida	5 W
Ganancia Tx	5 dBi
Ganancia Rx	1.7 dBi
Eficiencia Tx	0.73
Eficiencia Rx	$0.82 \cdot 0.5 = 0.41$
Eficiencia Harvester	0.6

Tabla 9.1.8.12.1: Especificaciones de los cálculos realizados para el análisis cualitativo de la recarga inalámbrica.

Se puede observar en la Figura (9.1.8.12.1) que a priori la potencia a la salida del *energy harvester* es superior a la del peor caso para todas las distancias comprendidas entre los 32 y 64 cm.

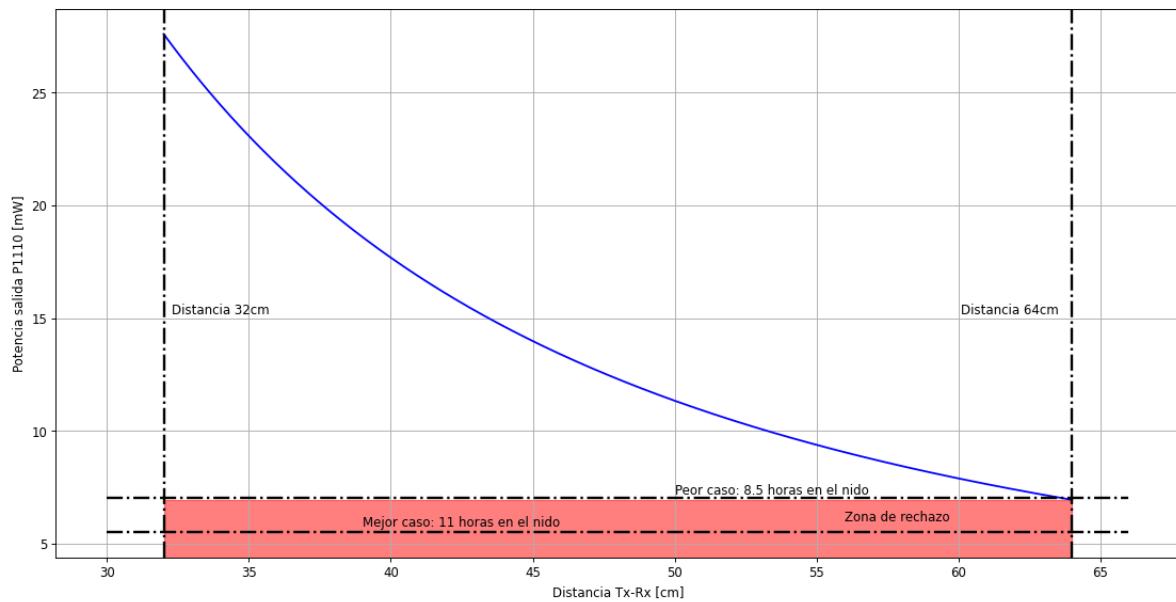


Figura 9.1.8.12.1: Cálculos preliminares de la potencia recibida en las baterías de la mochila del pájaro dada la distancia entre esta y la antena transmisora.

Sin embargo, estos cálculos tienen en cuenta que ambas antenas se encuentran perfectamente alineadas siempre. Esto en la práctica no es así. Como la mochila del ave va sobre su lomo, la antena receptora no siempre está completamente horizontal. Es por esto que se debe multiplicar a la potencia recibida por el factor de pérdida por polarización o *PLF* por sus siglas en inglés. Este factor se puede calcular como:

$$PLF = \cos^2(\phi) \quad (11)$$

En la Figura (9.1.8.12.2) se puede observar la potencia recibida en las baterías de la mochila, teniendo en cuenta el factor de pérdida por polarización para todos los ángulos de elevación ϕ posibles, siendo estos entre -90 y 90 grados.

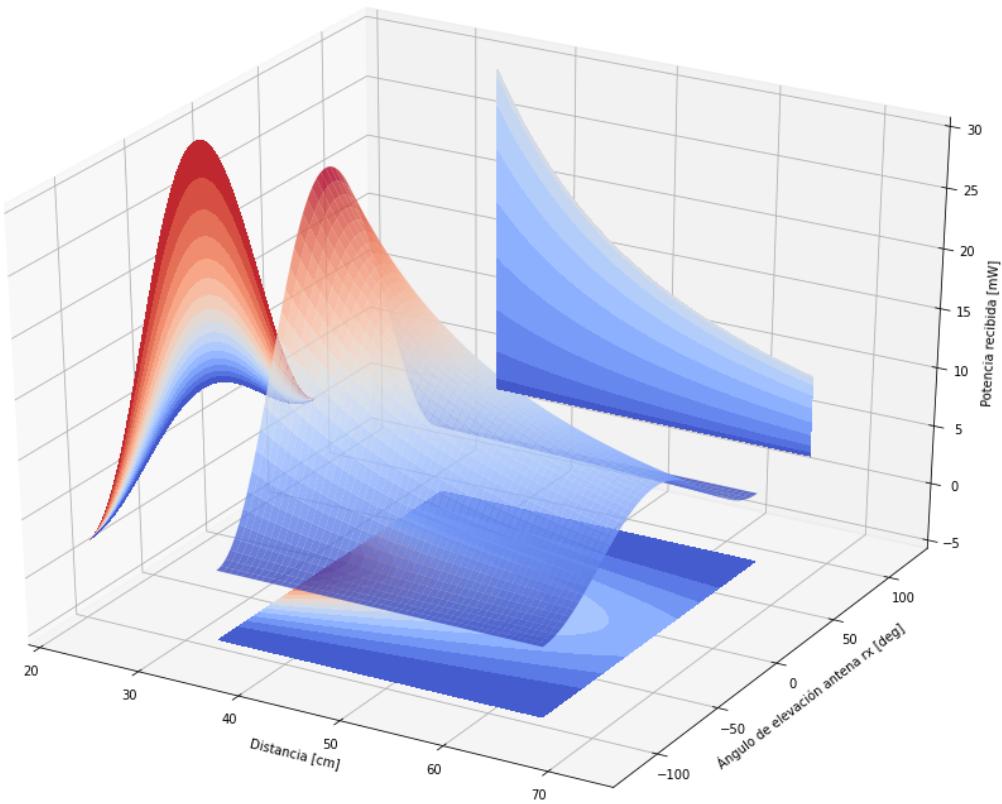


Figura 9.1.8.12.2: Potencia recibida en las baterías de la mochila del pájaro, dada la distancia entre esta y la antena transmisora, considerando el factor de pérdida por polarización. A los costados de la superficie graficada se observan las curvas de nivel de los distintos ejes del gráficos.

Se observa en las curvas de nivel que la potencia recibida decae rápidamente cuando el ángulo de elevación ϕ se aleja de su valor nulo. La Figura (9.1.8.12.3) muestra algunas curvas de nivel para ciertos grados de elevación. Cabe notar que cuando el ángulo es nulo, la potencia recibida cumple con el requisito de peor caso para todas las distancias entre 32 y 64 cm. Sin embargo, en cuanto el valor de ϕ se vuelve distinto de cero, rápidamente decrece la distancia máxima en la que los requisitos de potencia recibida se cumplen.

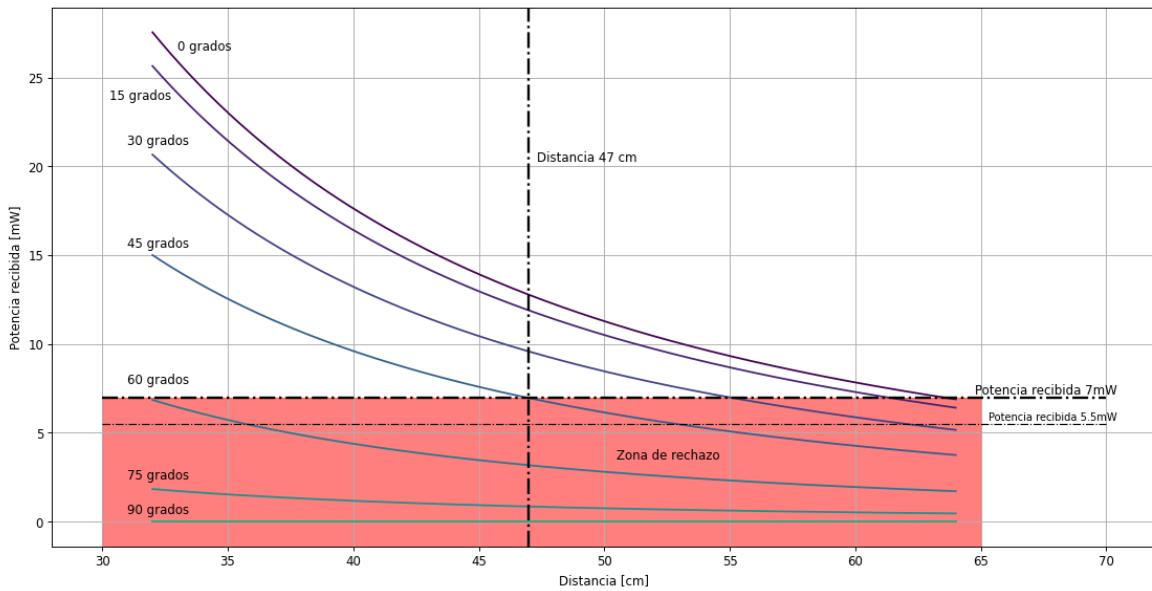


Figura 9.1.8.12.3: Curvas de nivel de la Figura (9.1.8.12.2) para grados de elevación fijos.

Si se toma la curva de 45 grados como referencia, se observa que el rango de distancias en las que se cumple el requisito del peor caso es de 32 a 44.8 cm. Esto se puede notar también en la Figura (9.1.8.12.4), donde aquellas curvas de nivel para distancias fijas que se encuentran fuera de la zona de rechazo, son aquellas en las que la distancia es menor a 44.8 cm.

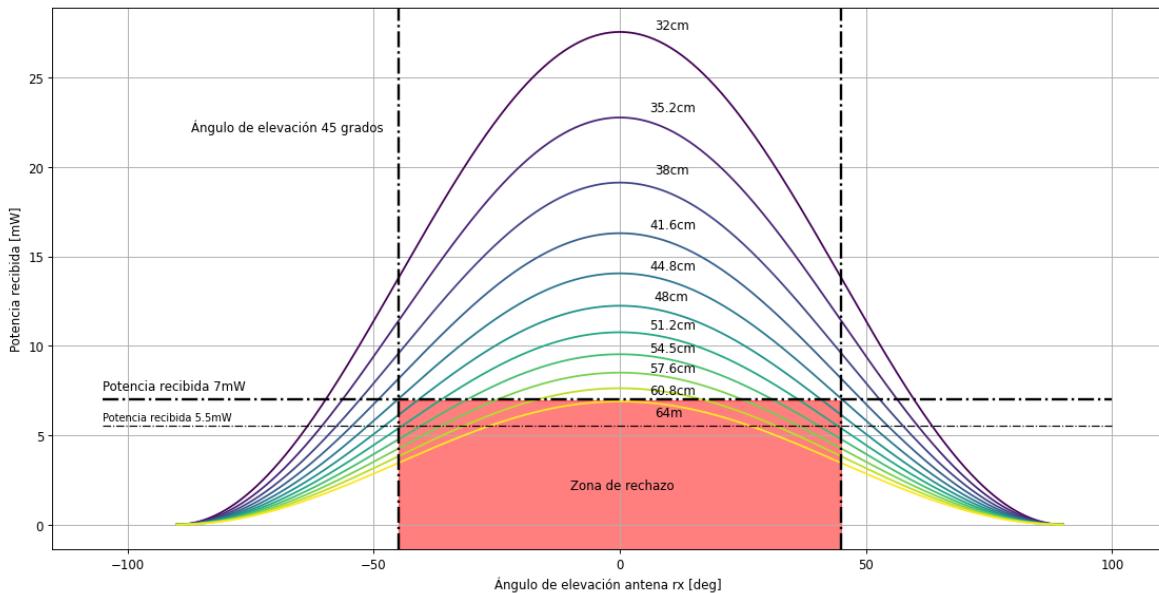


Figura 9.1.8.12.4: Curvas de nivel de la Figura (9.1.8.12.2) para distancias entre antena receptora y transmisora fijas.

Si se tiene en cuenta el requisito de potencia recibida en el mejor de los casos, la carga inalámbrica es factible dadas las condiciones impuestas para distancias entre 32 y 51.2 cm. Teniendo en cuenta el requisito del peor de los casos este rango se reduce a distancias entre 32 y 44.8 cm.

Otro factor importante que se debe tener en cuenta en el análisis cuantitativo es el de la densidad de potencia radiada por el sistema y la necesidad de no perturbar al ave en el nido. Por lo presentado en la Tabla (5.2.1), la distancia que existe entre la bóveda y la entrada es de hasta 10 cm, mientras que el largo de la entrada es de entre 11.4 y 20 cm. A esto se le suma el tamaño del ave, el cual es de hasta 47 cm. Esto significa que el ave al entrar o salir del nido se encuentra por lo menos alrededor de los 10 cm de distancia de la antena transmisora.

Tomando las especificaciones de la Tabla (9.1.8.12.1) y teniendo en cuenta que la densidad de potencia P_D se puede aproximar para el tipo de antena utilizada como

$$P_D = \frac{P_{tx}G_{tx}}{4\pi d^2} \quad (12)$$

donde P_{tx} es la potencia emitida en la antena transmisora, G_{tx} es la ganancia de esta y d la distancia a la que se quiere evaluar la densidad de potencia, se obtiene el gráfico de la Figura (9.1.8.12.5).

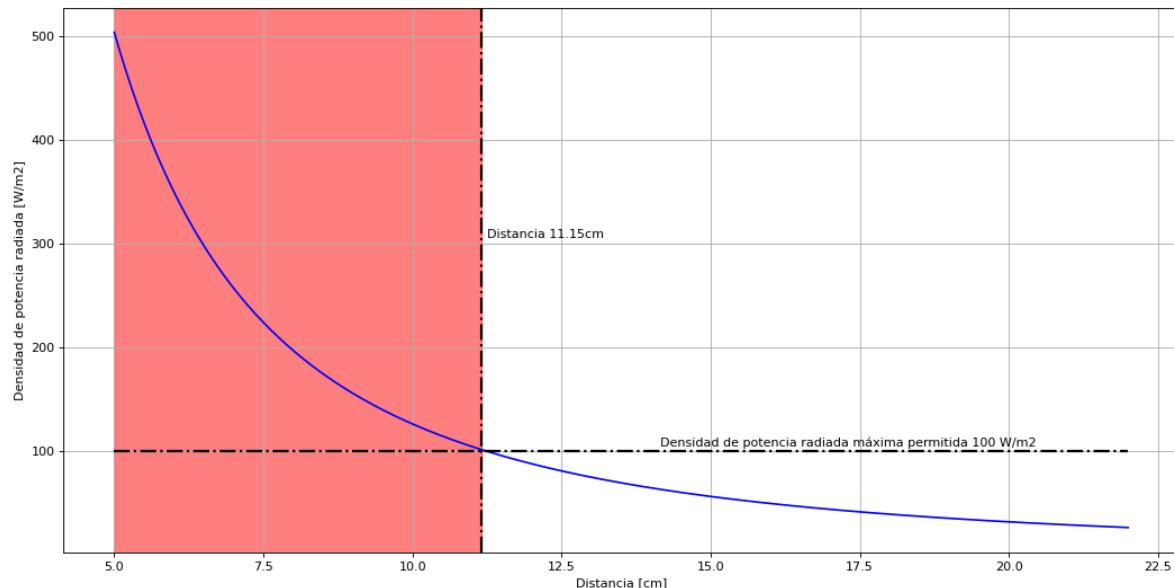


Figura 9.1.8.12.5: Densidad de potencia radiada a distintas distancias de la antena transmisora dentro del nido.

Se puede observar que la densidad de potencia radiada a una distancia menor a 11.15 cm es mayor que la de 100 W/m^2 utilizada en el estudio de palomas y gaviotas [13], animales más chicos que el carpintero gigante. Por esta razón, no se puede garantizar que el ave no sea perturbada al entrar y salir del nido debido a la radiación electromagnética de la recarga inalámbrica.

9.1.8.13 Conclusiones de la Factibilidad de la Recarga Inalámbrica

Tanto por el rango de distancias en las que los requisitos de potencia recibida en el peor de los casos se cumplen, como la restricción de no perturbar al ave a causa de la radiación electromagnética, se concluye inviable la carga inalámbrica para la batería de la mochila del ave dadas las condiciones planteadas y se investigará una solución alternativa.

9.1.9 Propuesta Final de Abastecimiento de Energía de UBM

Ante la imposibilidad de realizar la carga inalámbrica se tuvo una reunión con el cliente para discutir el proceder. El cliente indicó que el dato principal que se requiere del ave es la posición. Para la obtención de datos de la trayectoria del ave se propuso el esquema que se explica a continuación.

9.1.9.1 Módulos del Sistema de Seguimiento

La propuesta se basa en cuatro partes: la mochila en el ave, las bases de seguimiento, la base principal de seguimiento y la comunicación de esta con la base del nido.



(a) Base de seguimiento.



(b) Base principal de seguimiento.



(c) Base Nido.



(d) Mochila.

Figura 9.1.9.1.1: Elementos ilustrativos de los módulos del sistema de seguimiento.

9.1.9.2 Mochila

Para la propuesta de la mochila, se llegó a la conclusión de que utilizar GPS es muy costoso en materia de energía, peso y dimensiones. Es por esto que tomó un camino alternativo, recomendando la tecnología de *beacon Bluetooth*.

Se toma como ejemplo el modelo de *beacon* EMBC22, cuyas dimensiones son útiles para la aplicación deseada, dado que miden 30 mm de diámetro y 10 mm de altura. Junto a la batería se llega a un peso de 10 gramos, cumpliendo así con los requerimientos de dimensiones y de peso estipulados.

La característica más notable del *beacon* es que permite mandar paquetes personalizables por *Bluetooth 5.0* a bases receptoras para informar su presencia. A partir estas se puede obtener la ubicación. Además, este dispositivo soporta un amplio rango de temperaturas y cuenta con un grado de protección IP-64.

Para garantizar el funcionamiento durante todo el período del estudio se calcula una vida útil de la batería³. Con un rango de detección de hasta 200 metros y un microcontrolador con acelerómetro y *firmware* personalizable, se obtiene un valor de 4 años.

³Para calcular la vida útil de la batería se utiliza la calculadora de vida útil del proveedor del *beacon*.



Figura 9.1.9.2.1: Beacon EMBC22.

9.1.9.3 Base de Seguimiento

Las bases de seguimiento consisten en una red de módulos receptores esparcidos por el bosque a una distancia de aproximadamente 90 metros entre sí. Además, se hicieron las siguientes recomendaciones:

- Comunicación *Bluetooth* 5.0 para el *beacon* acorde al estándar, con la finalidad de poder triangular su posición.
- Uso de red Lora o *Bluetooth* 5.0 para la comunicación entre bases de seguimiento de ser necesario y con la base principal de seguimiento para reporte de datos.
- Uso de paneles solares y baterías para lograr independencia de la red eléctrica.



(a) Ubicación de bases de seguimiento.



(b) Triangulación de posición.

Figura 9.1.9.3.1: Elementos ilustrativos de los módulos del sistema de seguimiento.

La zona que se debe cubrir por estas bases corresponde a un círculo de radio 500 metros. El rango aproximado de cada base con la que consistentemente pueden comunicarse sin pérdida de paquetes es de aproximadamente 95 metros. Por lo que utilizando estos datos se puede obtener la zona de cobertura utilizando un total de 43 receptores.

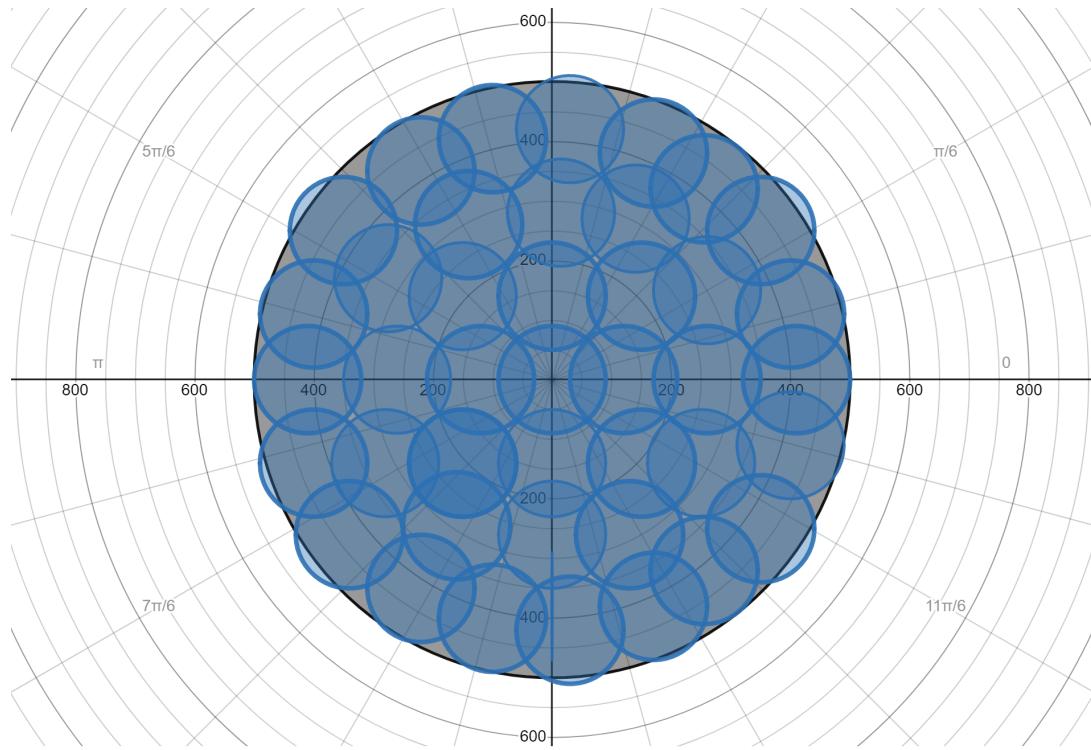


Figura 9.1.9.3.2: Área total de cobertura y rango de cada módulo individual.

9.1.9.4 Base Principal de Seguimiento

La base principal de seguimiento junta toda la información de las demás centrales. Se considera la existencia de repetidores, los cuales son empleados para la señal del módulo principal. Esto permite extender la distancia en el caso de que la red receptora se encuentre fuera de alcance de la principal.

Dicha base también debe contar con independencia de la red eléctrica, y debe poder comunicarse con la del nido mediante *Bluetooth*. El propósito de esta comunicación es el enviar un set de datos del ave una vez por día.

9.1.9.5 Propuesta de Valor Aumentada

Se destaca que, al realizar el diseño sugerido, aumenta con creces la modularidad del producto final. Con la red de receptores desplegada, es posible escalar el sistema para más de un animal a la vez. Esto es posible simplemente con utilizar un *beacon* en las especies que se deseen seguir.

Debido al nivel de configuración que estos poseen, se puede distinguir a los distintos dispositivos entre sí, creando así un bosque inteligente para el seguimiento de toda su fauna.

9.1.10 Elección de una Solución

9.1.10.1 Sensores

Para el sensor de temperatura la primera opción a descartar es aquella que no cumple con el rango de temperaturas a medir, por lo que el DS18B20 queda descartado a pesar de su bajo costo. Luego, de las opciones que quedan, todas son de un costo similar. Sin embargo, hay que tener en cuenta que para la termocupla se debe proporcionar una manera de medir la temperatura de referencia, la cual puede ser tanto una RTD como un IC, aumentando el costo de la termocupla. Tanto la TC como la RTD necesitan un circuito convertidor para poder medir directamente el valor de la temperatura con un micro controlador, mientras que los IC ofrecen directamente una salida digital.

Otra desventaja de la TC es que tiende a envejecer rápidamente. Si bien el dispositivo no se usará más de tres meses seguidos, este podrá ser reutilizado, dándole mayor peso al factor del envejecimiento. El autocalentamiento también es contra productivo en la medición de temperatura debido a que puede alterar la misma si no es tenido en cuenta. Las TC no cuentan con este inconveniente debido a su principio de funcionamiento, mientras las otras opciones si lo tienen. Con la RTD este efecto depende directamente de la corriente que se suministra para la medición, y con los IC esto es un aspecto que es considerado por los diseñadores de estos.

Por estas razones los candidatos terminan siendo la DHT-22 y la PT-100. Un punto favorable para la DHT-22 es que no necesita un circuito extra. Adicionalmente, esta unidad cuenta con una medición de humedad, lo que brinda la posibilidad de usarlo también para dicha variable o como un complemento de otro sensor.

En la elección de componente para la medición de humedad, como primer criterio, se busca que pueda medir el rango entero de humedad relativa y que cuente con una precisión considerable. Dadas estas consideraciones, se descarta la DHT-11 y AM-1001. Es así que de los dos restantes, se opta por el DHT-22 debido a que por un menor costo se obtienen mejores prestaciones. Teniendo en cuenta esto se utilizará tanto para la medición de temperatura como humedad el DHT-22.

En cuanto a la luminosidad, principalmente se deberá asegurar el funcionamiento en el rango de temperatura en el cual operará el dispositivo. Por esto, el OPT-101 queda descartado. Además, se tiene en cuenta la potencia utilizada, el rango de medición de los sensores y el tipo de alimentación.

La comunicación puede ser analógica en corriente para el TEMT-6000, pero este necesitará un amplificador de corriente o un convertidor, para llevar a esta corriente a un nivel medible.

Existen también otros sensores que tienen una salida analógica de tensión como el GL55-LM393, con un rango entre 0 y V_{dd} . Este también provee una salida digital que funciona con un *Schmitt Trigger*⁴.

El BJ-1750 cuenta con una salida digital con el protocolo de comunicación I2C. Teniendo en cuenta esto se opta por utilizar este último sensor.

Luego para el módulo RTC se tomó uno que tenga I2C, y que cuyo rango de alimentación se encuentre entre 3.3 y 5 V, por lo que se opta por el DS3231.

Para la detección de presencia se descarta el uso de la R-Pi 3B, dado que esta debe estar por fuera del nido al no haber suficiente espacio en la bóveda. De la misma forma, el módulo NRF24L01 por no ser compatible con *Bluetooth* 5.0. Finalmente se decide utilizar el ESP32-S3-WROOM-1 por el menor costo y mayor sensibilidad de recepción de la antena.

Finalmente, para la cámara que obtendrá las imágenes, se tuvo en cuenta fundamentalmente la relación precio-resolución de la cámara, al igual que la integración con Linux y el factor de contar con una API para el lenguaje C. Por lo que la R-Pi-CMOD-V2 fue la cámara seleccionada.

9.1.10.2 Almacenamiento

El factor principal para seleccionar la memoria SD a utilizar es el rango de temperaturas de operación. Es por este factor que se elige la SDSDQAF3-XI, ya que esta se encuentra en un rango seguro (mayor a las demás).

Se recolectará un volumen de datos del que no se tiene una gran certeza, ya que una parte será lo que provenga de la mochila. Se estima en función de los datos del nido y del periodo de activación de los sensores, que una memoria de 16 GB es suficiente incluso si aumenta el volumen de datos.

9.1.10.3 Unidades de procesamiento

Para este módulo se opta por la Raspberry Pi 3B, ya que posee *Bluetooth* BLE, es el dispositivo más económico, más pequeño y cuenta con soporte para micro SD.

En cuanto las temperaturas de funcionamiento, este dispositivo se encuentra dentro del rango necesario. Además, los módulos Raspberry Pi trabajan entre 20 °C y 30 °C por encima de la temperatura ambiente debido a su autocalentamiento. También es sabido que los integrados R-Pi pueden llegar a soportar temperaturas extremas, tales como las que se dan en la Antártida [14].

Además, este dispositivo seleccionado se caracteriza por ser compatible con la cámara seleccionada sin la necesidad de utilizar adaptadores.

9.1.10.4 Comunicación

En cuanto a la comunicación se utilizará BLE para la comunicación con la base principal de seguimiento y la detección del ave en el nido, y WiFi para la comunicación con un tercero. Ambos de estos se encuentran disponibles para su uso en el módulo Raspberry Pi 3B.

⁴Un *Schmitt Trigger* es un circuito comparador con histéresis. Convierte una señal analógica a una digital, y se le dice *trigger* debido a que la salida retiene su valor hasta que la entrada cambie lo suficiente como para que se dé un cambio.

9.1.10.5 Batería

La batería a emplear queda determinada en función del consumo del sistema y de la energía que se deba almacenar en caso de emergencias.

Los consumos de potencia de las partes del sistema se detallan en la Sección (10.1.3).

Para alimentar tanto a los sensores como a la R-Pi en un día se requieren 151.2 kJ . Con una batería de 21 Ah de capacidad se consigue la especificación mencionada, por lo que se opta por la batería UL24-12 de Ultracell ya que la batería de Probattery es mucho más pesada.

Cabe notar que los 24 Ah de la batería seleccionada son nominales, por lo que se debe calcular la capacidad real (la cual será ligeramente menor debido a condiciones del entorno y calidad de esta). Para obtener la capacidad real de la batería se utiliza la información proporcionada en la hoja de datos. En la Figura (9.1.10.5.1) se puede observar como a una temperatura de trabajo de 10°C y una razón de descarga de 0.05 CA (descarga total en 20 h) la capacidad efectiva restante de la batería es del 90% . Por otro lado, en la Figura (9.1.10.5.2) se observa como teniendo en cuenta una vida útil de 540 ciclos (es decir, 2 años de uso ininterrumpido) la capacidad efectiva restante sigue siendo del 100% .

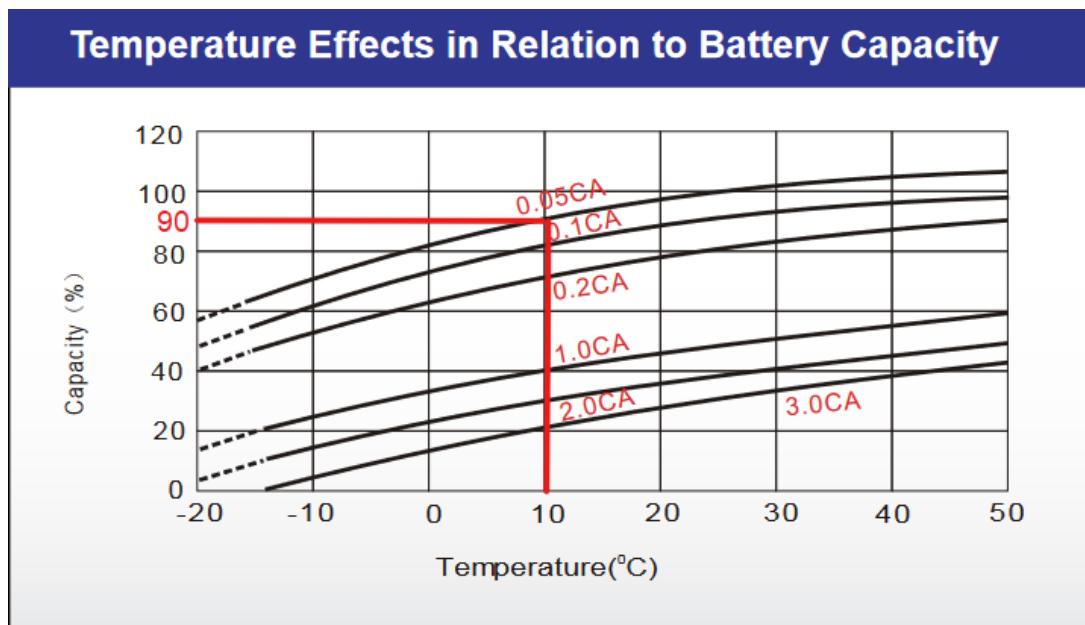


Figura 9.1.10.5.1: Efectos de la temperatura en relación con la capacidad de la batería.

Cycle Life in Relation to Depth of Discharge

Testing condition
 Discharging: current 0.17C (FV 1.7V/cell);
 Charging: current 0.25C max, voltage 2.45V/cell;
 Charging volume: 125% of discharged capacity.

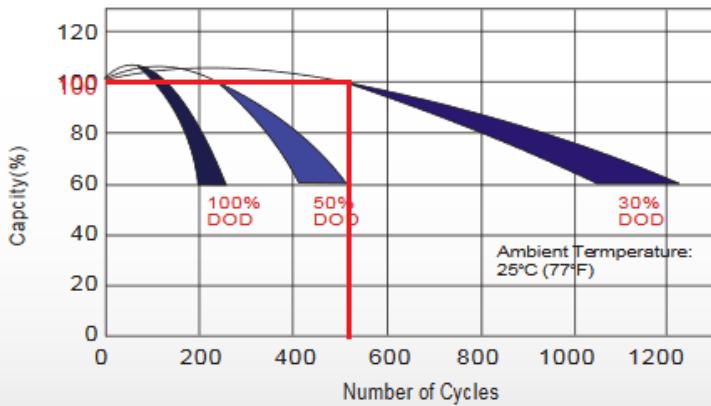


Figura 9.1.10.5.2: Ciclos de vida en relación con la profundidad de descarga.

Por ende, la capacidad efectiva, calculada como:

$$C_{ef} = C_{nom} \cdot 0.9 \cdot 1 = 21.6 \text{ Ah} \quad (13)$$

por lo que la batería elegida es apta para el proyecto.

9.1.10.6 Alimentación

Para poder abastecer a la batería seleccionada, con un panel de 23.1 W o más se puede proveer la potencia requerida. Por esta razón, se decide utilizar el panel solar DSP-30M de 30 W.

En cuanto a la etapa de carga, para el cargador MPPT de la batería principal se optó por la DFR0580 debido al precio, la documentación clara y la tecnología de la batería a usar.

9.1.11 DFMEA

ANÁLISIS DE RIESGOS												
Nº	Efectos de falla	Modo de la falla	Causas de la falla	NS	Aceptabilidad PO	DE	IC	Acción de reducción	NS	Aceptabilidad PO	DE	IC
1	No se pueden realizar mediciones	Los sensores dejan de funcionar	Los sensores son dañados por el ave	4	4	2	32	Ocultar los sensores en la bóveda	4	2	2	16
2	No se pueden realizar mediciones	Los sensores dejan de funcionar	El conexionado es dañado por el ave	4	4	2	32	Hacer más robusto el cableado	4	2	2	16
3	Falta de energía solar	Falla en los paneles solares	Los paneles se encuentran dañados	5	3	2	30	Colocar protección para los paneles	5	2	2	20
4	Falta de energía solar	La electrónica no funciona correctamente	Fue colocado en un lugar con obstrucciones	5	3	3	45	Instalar el panel solar sobre un tronco, donde no haya ramas u objetos que puedan obstruir	5	1	3	15
5	Falta de energía en la UBN	No hay alimentación	Cuatro días con un nivel de luz menor al necesario para la carga de baterías	5	3	3	45	Seleccionar una batería más grande de la necesaria para almacenar un excedente	5	1	3	15
6	Falta de energía en la UBN	La batería no funciona	Se inundó el contenedor de la batería	5	3	4	60	Se utiliza una carcasa para la batería con protección IP-23 que asegure protección contra agua	5	1	4	20

Tabla 9.1.11.1: DFMEA (Parte 1).

ANÁLISIS DE RIESGOS														
	Fecha de elaboración:	25/05/21	Fecha de revisión:	28/11/22										
Nº	Efectos de falla	Modo de la falla	Causas de la falla	NS	Aceptabilidad PO	DE	IC	Acción de reducción			Aceptabilidad NS	PO	DE	IC
7	La electrónica deja de funcionar	La UP deja de funcionar	La UP se encuentra a una temperatura baja	5	2	2	20	Colocar la UP en un encapsulado			5	1	2	10
8	Falla de almacenamiento	No se pueden guardar más datos	La temperatura de operación es menor al mínimo aceptable	5	4	3	60	Cambiar la memoria por una de nivel industrial			5	1	3	15
9	Falla de almacenamiento	La memoria sufre una pérdida de información	La memoria es defectuosa	5	2	3	30	Se verifica la funcionalidad de la misma antes de integrarla al proyecto			5	1	3	15
10	Interrupción en la transmisión nido - base principal de seguimiento	Se pierde la comunicación con la base principal de seguimiento	La base principal de seguimiento es colocada fuera del rango de la comunicación	5	3	2	30	Se verifica el alcance de la transmisión de la base principal de seguimiento luego de su colocación			5	1	2	10
11	Interrupción en la transmisión nido - persona	Se pierde la comunicación con la persona	El dispositivo receptor no se encuentra en el rango de transmisión	5	3	2	30	Informar la existencia del error en el dispositivo receptor			3	3	2	18
12	Interrupción en la transmisión nido - persona	La transmisión de datos se ve interrumpida	La persona que recibe la información está posicionada demasiado lejos del transmisor							La información es borrada únicamente cuando se recibe un mensaje de OK de la persona en la base del árbol a través del WIFI				

Tabla 9.1.11.2: DFMEA (Parte 2).

ANÁLISIS DE RIESGOS													
	Fecha de elaboración:	25/05/21	Fecha de revisión:	28/11/22	NS	PO	DE	IC	Aceptabilidad	NS	PO	DE	IC
13	Efectos de falla	Modo de la falla	Causas de la falla	NS PO DE IC	Acción de reducción		Se realiza la detección del ave utilizando un módulo BLF en la placa de sensores dentro del nido		5	1	3	15	
14	No se detecta el ave nunca en el nido	Falla la detección del ave en el nido	La R-Pi no consigue detectar la señal del beacon del ave fuera del nido	5 4 3 60	Reiniciar Node Red cada cierto tiempo así se refresca el servidor		4	1	5	20			
15	Falla en el servidor	No se obtiene respuesta de Node Red	Sobrecarga del servidor	4 3 5 60	Identificar si ya se solicitó el servicio y bloquear el ingreso de solicitudes		5	1	1	5			
16	Falla en el servidor	No se obtiene respuesta de Node Red	Reiteradas solicitudes del usuario de un mismo servicio	5 5 1 25	Limitar el tiempo de transmisión a un periodo de 5 minutos y brindar la posibilidad de reiniciar el video		5	1	3	15			
17	Inconsistencia del intervalo de tiempo entre mediciones	No se puede visualizar el nido en tiempo real	Sobre carga debido a una transmisión constante	5 5 3 75	Se sobrecarga temporalmente o se reinicia el servidor		4 2 32	Se almacena en memoria la fecha y hora de la última medición		4	1	2	8
18	El controlador deja de funcionar	Corrupción de la tarjeta SD de la Raspberry Pi	Se pierde seguimiento del último dato tomado	5 4 3 60	Se agrega un circuito de protección ante desabastecimiento de energía		5	1	3	15			

Tabla 9.1.11.3: DFMEA (Parte 3).

Severidad	Probabilidad	Detectabilidad	
Insignificante	Remota	Completa	1
Poco significante	Poco remota	Mayor	2
Moderado	Media	Moderada	3
Grave	Alta	Pequeña	4
Muy grave	Muy alta	Mínima	5

Nivel de IC	
Aceptable	$IC \leq 27$
Bajar hasta razonablemente práctico	$27 < IC < 48$
No aceptable	$48 \leq IC$

Tabla 9.1.11.5: Nivel de IC.

Tabla 9.1.11.4: Criterio de IC.

9.2 Factibilidad de Tiempos

9.2.1 Consideraciones

Para el desarrollo de las siguientes secciones, se tiene en cuenta un equipo de trabajo de 4 personas, con días laborales de 8 horas. Además, se consideran 15 días de vacaciones en la primera quincena de enero.

9.2.2 Planificación

Se procede a presentar un cuadro con las tareas a realizar. En la siguiente tabla se observa el tiempo más probable, el optimista y el pesimista, para tener un análisis más real de la duración.

Nº	Nombre de tarea	Duración Optimista	Duración Media	Duración Pesimista	Predecesora
1	Detectar Necesidad	1	3	4	-
2	Definir el alcance	2	3	4	1
3	Antecedentes y Contexto	1	2	3	1
4	Reuniones con cliente	0.25	0.25	0.25	1
5	Definir objetivos de diseño	1	2	5	2, 3
6	Definir requerimientos	4	5	8	4, 5
7	Definir Especificaciones	3	5	8	5
8	Planes de validación	3	5	8	11
9	DFMEA 1º reunión	0.25	0.25	0.25	8
10	Investigación antenas y radiopropagación	44	45	54	6, 7
11	Análisis de factibilidad Tecnológica	40	45	47	6, 7
12	Análisis de presupuesto y costos	4	5	9	6, 7
13	Análisis de factibilidad económica	3	5	8	12
14	DFMEA 2º reunión	0.25	0.25	0.25	9, 11, 12
15	Cálculo y selección de HW	12	15	20	11, 12, 13
16	Diagrama de HW y plan de implementación	14	15	18	15
17	DFMEA 3º reunión	0.25	0.25	0.25	15
18	Diagrama de SW y plan de implementación	11	15	19	15
19	Integración a nivel módulos	23	25	27	16, 18
20	Integración general	18	20	21	19
21	Integración Prototipo	9	12	13	20
22	Vacaciones	15	15	15	21
23	Integración a Prototipo	9	12	13	22
24	Validación de prototipo	19	20	21	23
25	Estudio de confiabilidad	6	10	13	24

Tabla 9.2.2.1: Actividades a realizar en el proyecto en días laborales de 8 horas cada uno.

9.2.3 Programación

Se realizó un diagrama de Gantt acorde a la Tabla (9.2.2.1). En este se marcó el camino crítico en rojo.

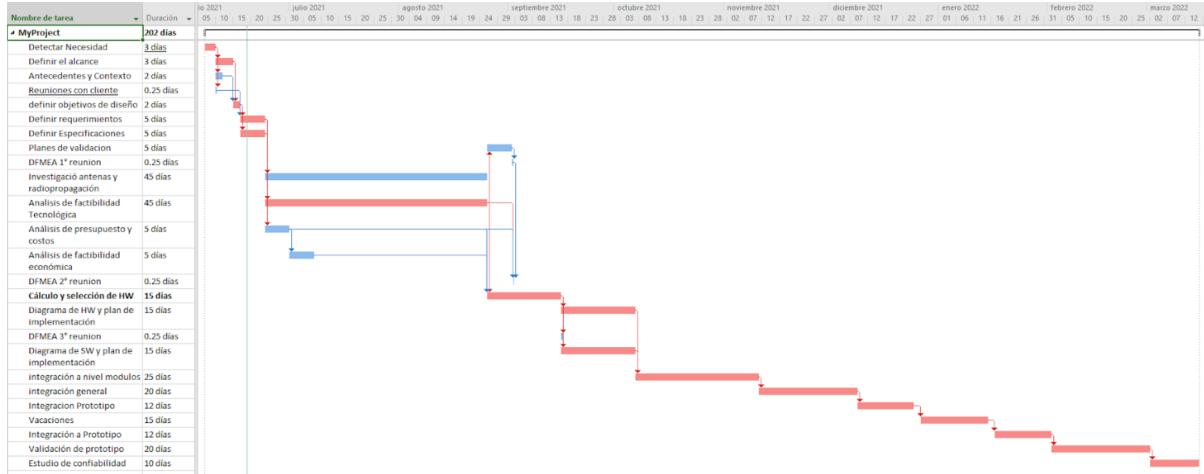


Figura 9.2.3.1: Diagrama de Gantt del proyecto.

Luego se realizó una simulación de Montecarlo utilizando la distribución β para las variables aleatorias, obteniendo como resultado el análisis plasmado en la Figura (9.2.3.2). Se tiene en cuenta que por teorema central del límite la suma de las variables aleatorias β convergen a una normal.

En este gráfico se ve la probabilidad de terminar el proyecto en un intervalo de entre 1533 a 1957 horas, con una probabilidad del 95 %. Esta distribución es el lapso entre que se comienza el proyecto y se termina, a través del camino crítico.

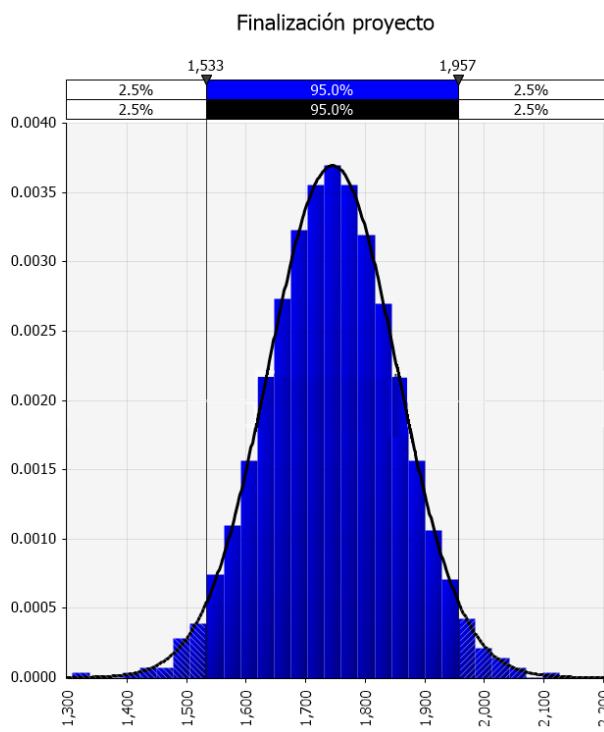


Figura 9.2.3.2: Simulación de Montecarlo.

A continuación, se muestran la cantidad de horas de ingeniería total. Para obtener los resultados de la Figura (9.2.3.3), se tuvieron cuenta la paralelización de actividades y la disponibilidad de 4 trabajadores.

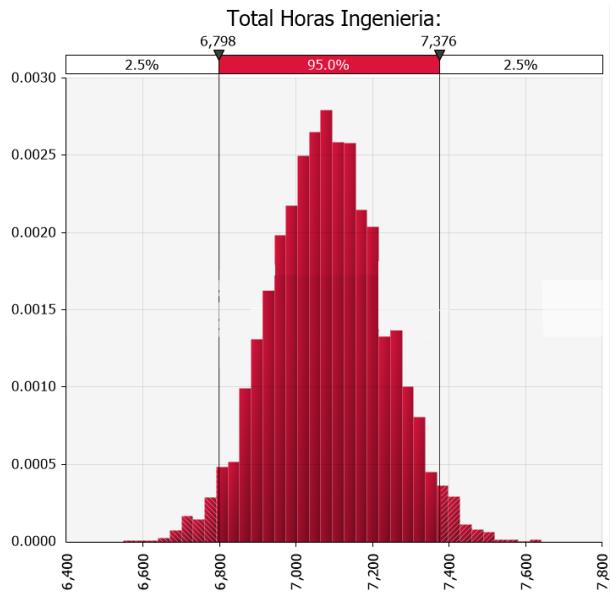


Figura 9.2.3.3: Simulación de Montecarlo para tiempo de ingeniería.

Se puede observar que el tiempo total de horas de ingeniería corresponde a un rango entre aproximadamente 6800 a 7400 horas, con una media de aproximadamente 7100 horas.

9.3 Factibilidad Económica

Para poder estudiar la viabilidad financiera de todo emprendimiento es necesario hacer un balance cuidadoso de los diferentes costos en los que se va a incurrir. Se debe analizar cuáles son las fuentes de ingresos que hacen del modelo de negocio planteado un negocio sostenible en el tiempo.

En el caso de este diseño en particular, se enfrenta el desarrollo de una única unidad. Sin embargo, no se descarta la posibilidad de realizar más unidades en el futuro por fuera del marco del proyecto. Es por esto que se realiza un modelo de negocios, basado en el planteo de una referencia futura.

9.3.1 Modelo de Negocios

Este diseño se trata de un proyecto único, con posibilidad de realizar hasta quince unidades adicionales posteriores a su finalización. Para dar una visión global se planteó el siguiente modelo de *Canvas*:



Figura 9.3.1.1: Modelo de negocio.

9.3.2 Investigación y Desarrollo

Este proyecto contiene una gran componente de diseño. Según lo establecido en la programación del proyecto, el tiempo invertido de diseño estimado es de 5472 horas. Una vez completada la etapa de diseño el equipo se aboca a la implementación del primer prototipo, es decir a la integración del *hardware* a utilizar y el desarrollo del *software* que controla al nido, permitiendo la interacción con el mismo.

Realizando la estimación sobre un sueldo de 30 USD/hora y multiplicándolo por la cantidad de horas de ingeniería esperada por persona se obtiene un total de:

$$\text{Costo mano de obra} = 7100 \text{ hora} \cdot 30 \frac{\text{USD}}{\text{hora}} = 213.000,00 \text{ USD}$$

9.3.3 Gastos fijos por unidad

Los gastos principales considerados son 257.34 USD , considerando el valor de los insumos de *hardware* y de montaje:

Ítem	Precio [USD]
Sensor humedad-temperatura	4.9
Sensor luminosidad	1.54
ESP32-S3-WROOM-1-N4	3.25
Módulo RTC	2.33
Cámara	20
SDI 32GBy	32
R-Pi 3B	25.5
Batería	86
Panel solar	25
MPPT	18
Encapsulado	16
Módulo Bluetooth 5.0	3.82
Montaje	31
Extras	15
Total	257.34

Tabla 9.3.3.1: Valores de insumos.

9.3.4 Reserva de Contingencia

Dado que este proyecto cuenta con un alto grado de investigación es necesario contar con un cierto margen de seguridad en caso de que ocurra un cambio de planes necesario para alcanzar el objetivo del proyecto. Es por eso que se decidió incluir en el análisis un adicional del 5 % sobre el costo total del proyecto. Esta suma es devuelta al cliente en caso de no necesitarla. De esta forma, la reserva de contingencia se valúa en 11.000 USD .

9.3.5 Escenario de Escala

Se contempla que, en el caso de que la producción sea de quince unidades, sea posible conseguir los insumos necesarios para el ensamblado de los dispositivos a precio de venta al por mayor.

9.4 Factibilidad Legal y Responsabilidad Civil

En cuanto a la emisión de ondas electromagnéticas, las normas y regulaciones que se deben considerar son:

- Resolución 202/95 (Estándar Nacional de Seguridad para la Exposición a radiofrecuencias comprendidas entre 100 KHz y 300 GHz).
- Resolución 530/2000 (Estándar Nacional de Seguridad de aplicación obligatoria a todos los sistemas de telecomunicaciones que irradian en determinadas frecuencias).
- Resolución 1994/2015 (Regulación SAR).
- *Guidelines* de la FCC en exposición máxima recomendada. Una densidad de potencia de $580 \mu\text{W/cm}^2$ @ 850 MHz .

- Ministerio de Salud y Acción Social admite como máxima una densidad de potencia de $450 \mu W/cm^2$ @ $850 MHz$.
- *The Worldwide Approval Status for 900 MHz and 2.4 GHz Spread Spectrum Radio Products.* Establece un límite de $4 W$ para la potencia de la antena transmisora.

En lo que a la preservación del medio ambiente respecta, la Ley 26.331 estipula los siguientes aspectos de interés:

- Se debe considerar un Plan de Manejo Sostenible de Bosques Nativos.
- El Plan debe cumplir las condiciones mínimas de persistencia, producción sostenida y mantenimiento de los servicios ambientales de los bosques.

A esto se le suma lo estipulado en la Ley 13.273. En esta se encuentra que se constituye como contravenciones forestales destruir, remover o suprimir señales o indicadores colocados por la autoridad forestal, entre otras cosas.

10. Ingeniería de Detalle

10.1 Hardware

10.1.1 Diagrama de Bloques (Hardware)

A continuación, se muestra el diagrama en bloques de *hardware* y los delimitadores de las distintas secciones, siendo estas: potencia y sensado.

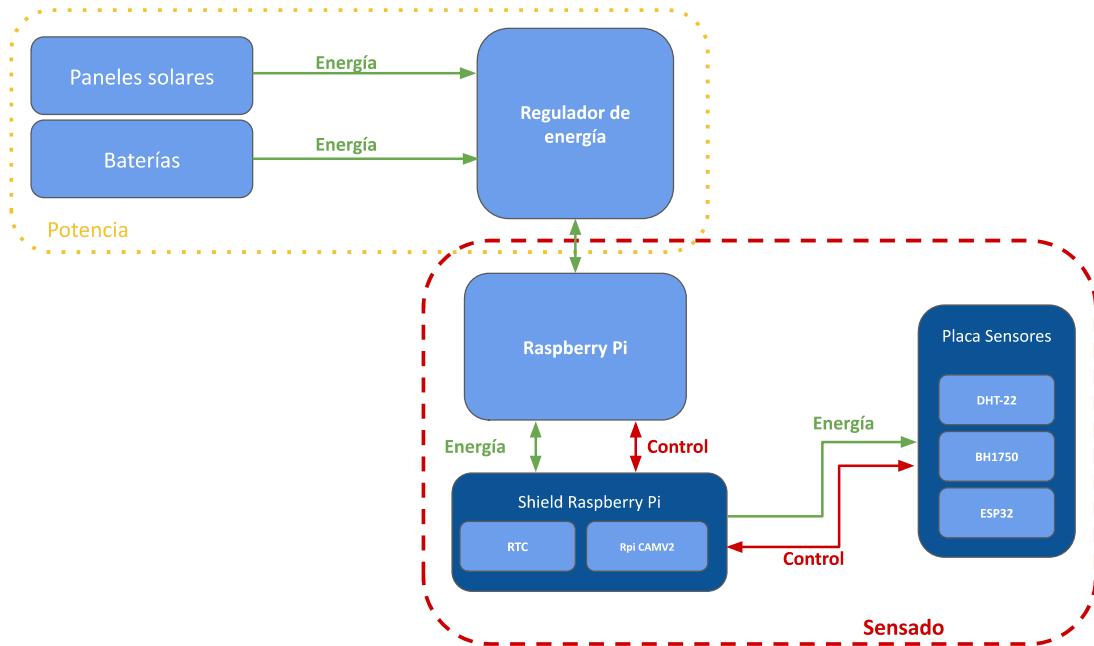


Figura 10.1.1.1: Diagrama en bloques del sistema de *hardware*.

La etapa de potencia es el conjunto de elementos necesarios para proveer de energía a toda la electrónica del proyecto. Es decir, a aquellos componentes y elementos que se encuentran en el nido.

Por otro lado, la etapa de sensado es aquella que se encarga de la medición de las variables físicas y de su correcto almacenamiento, teniendo en cuenta que esto implica un conocimiento preciso de la hora.

10.1.2 Descripción Detallada de cada Bloque

10.1.2.1 Potencia

Para la unidad de potencia se utilizan paneles solares y una batería de gel de ciclo profundo conectados a la placa DFR0580, la cual se ocupa de cargar la batería. De esta se obtienen cuatro salidas de tensión:

- Dos salidas de 5 V y 2.5 A [USB].
- Una salida de 5 V y 5 A.
- Una salida de 12 V y 8 A.

De estas salidas solo es necesaria aquella que posee 5 V. En cuanto a la corriente, si bien es indistinto, se emplea la de 5 A para evitar problemas de falta de corriente.

10.1.2.2 Medición

Para el conexionado de los sensores se diseñó una placa de soporte para la Raspberry Pi, la cual se denomina *shield*. En esta se encuentra:

- El módulo RTC.
- El acondicionamiento para la conexión de la cámara.
- El conexionado para el abastecimiento de energía para los sensores.
- Un conector para los sensores, que abarca al de humedad, temperatura, luminosidad y presencia.
- Un sistema de protección ante desabastecimiento de energía.
- Interruptores de prendido y apagado seguro.

La placa de sensores utiliza alimentación de 3.3 V, la referencia de tierra, dos pines GPIO y el bus I^2C de la R-Pi. Este último se emplea para administrar la comunicación con el sensor de luminosidad y el módulo RTC, ya que ambos emplean el respectivo protocolo de comunicación del bus, en modo *multi-slave*. Por otro lado, un pin GPIO corresponde a la comunicación serial del sensor de temperatura y humedad DHT-22; y el otro se utiliza para la comunicación con el detector de presencia. Además, la placa de sensores posee cinco pines denominados "PROG" los cuales se utilizan para programar al ESP32-WROOM-1-N4.

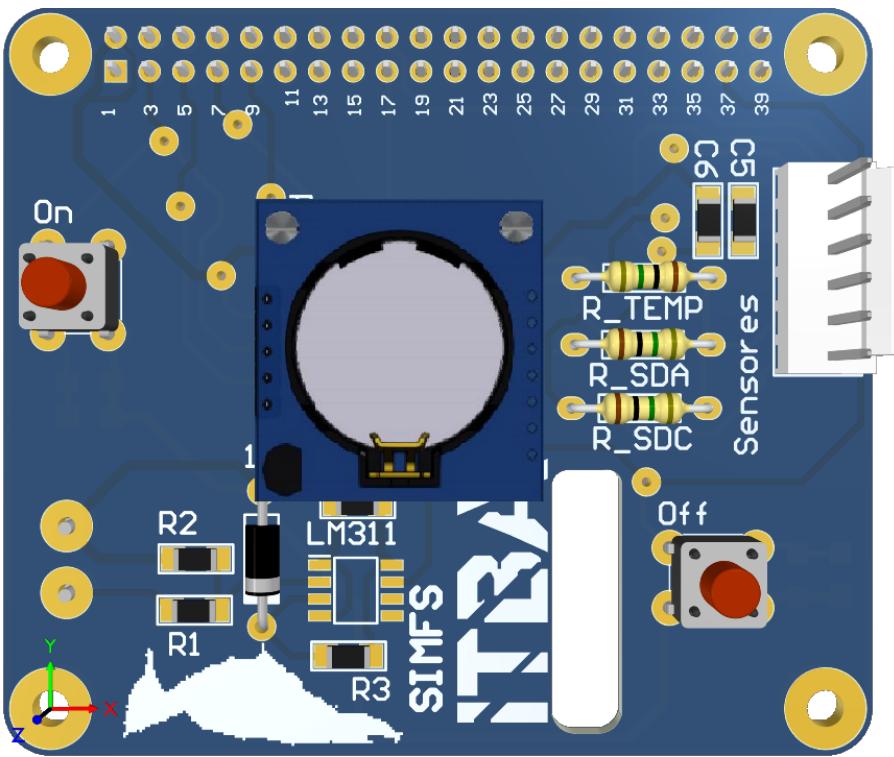
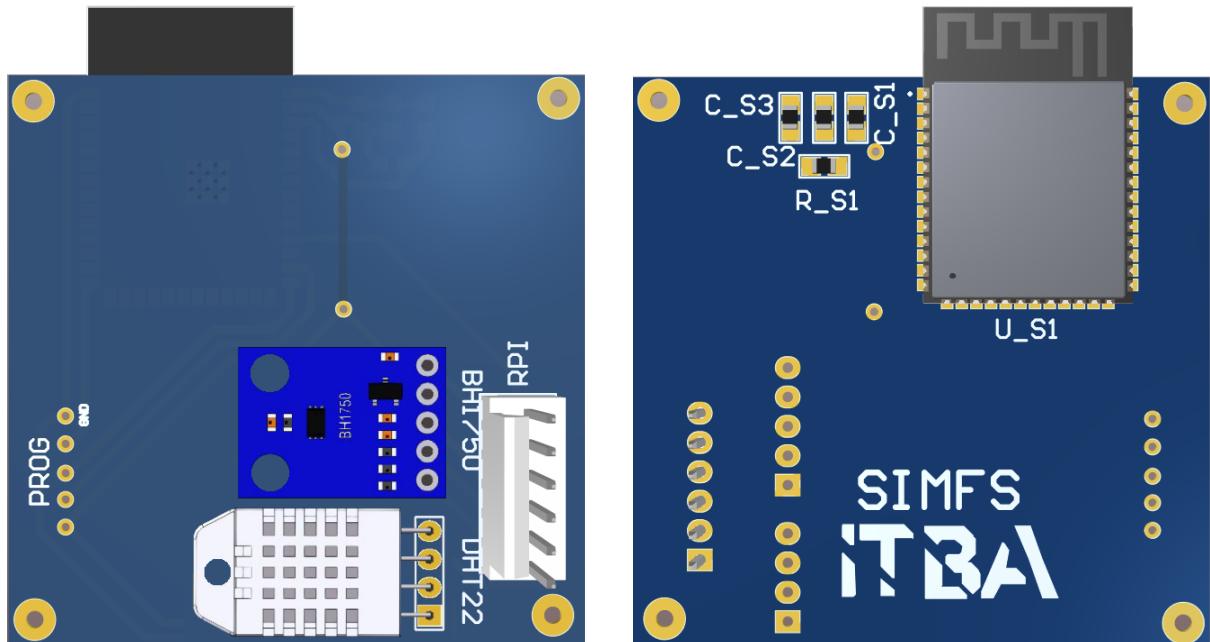


Figura 10.1.2.2.1: Shield Raspberry Pi.



(a) Vista top de la placa de sensores.

(b) Vista bottom de la placa de sensores.

Figura 10.1.2.2.2: Placa que contiene a los sensores.

Por fuera del *shield*, la Raspberry Pi cuenta tanto con un conector destinado al uso de una cámara como un conector mini USB por el cual se energiza la unidad. Ambos conectores se muestran en la Figura (10.1.2.2.3).

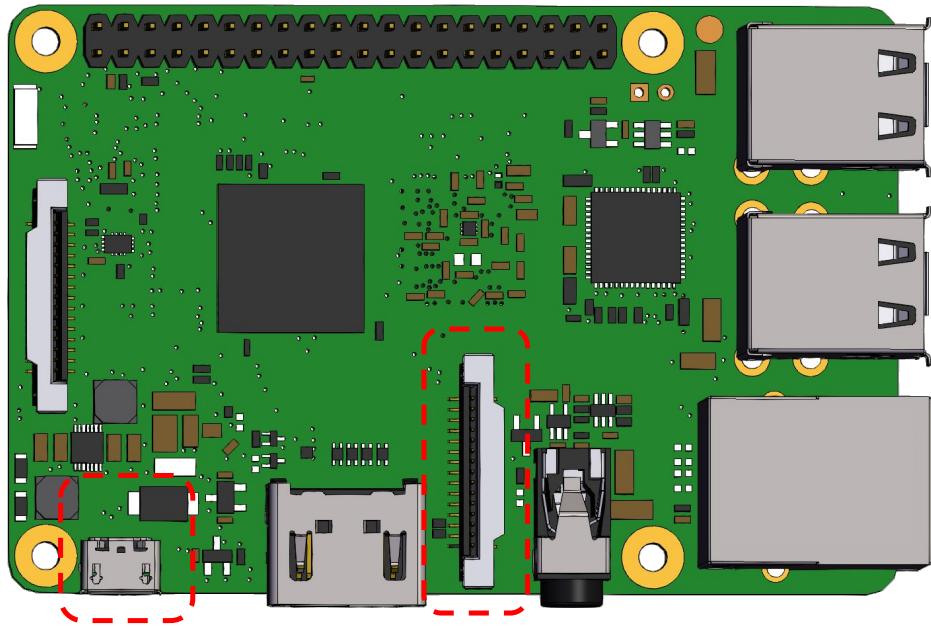


Figura 10.1.2.2.3: Conexionado cámara.

Además, cuenta con una ranura para una tarjeta SD. Esta es usada no solo para la imagen de sistema operativo, sino que también para el almacenamiento de datos.

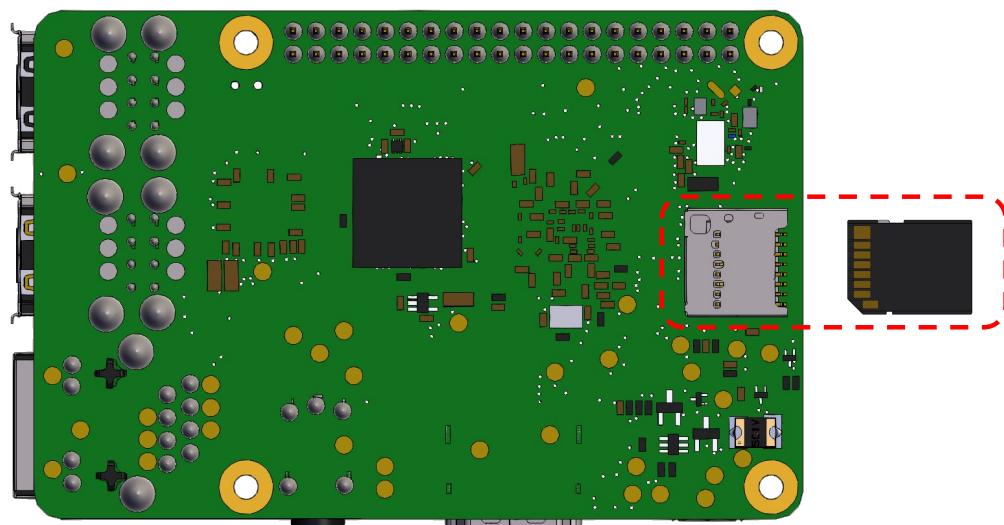


Figura 10.1.2.2.4: Conexionado tarjeta SD.

10.1.3 Detalles de Selección y Cálculo de los Elementos Circuitales de cada Bloque

10.1.3.1 Cálculo Potencias

Se asumen las siguientes variables para los cálculos de potencias, obtenidas de estadísticas propias de la zona donde se realiza el estudio [17].

1. Duración noche = 12.8 horas.
2. Duración día = 11.2 horas.

3. Horas de sol efectivas = 8 horas.

Para el caso de la R-Pi, se tiene que su consumo mínimo normal es de $P_{rpi_{min}} = 1.2 W$. Sin embargo, para reducir este consumo, se desactivan los puertos de Ethernet y HDMI ya que no se utiliza un entorno gráfico. Esto permite reducir aún más el consumo mínimo. Se estima que la Raspberry Pi consume como máximo alrededor de $P_{rpi_{est}} = 1.75 W$ en su funcionamiento normal [15]. Luego, se tiene que el consumo energético por día es de

$$E_{sist} = P_{rpi_{est}} \cdot 24 \text{ hs} = 151.2 \text{ kJ} \quad (14)$$

Teniendo en cuenta que la batería tiene una tensión de $V_{bat} = 12 V$ debe tener una capacidad de almacenamiento equivalente a $T_{reserva} = 4 \text{ días}$ sin recarga [17] y utilizando un coeficiente de seguridad de $\gamma_{bat} = 1.5$, se obtiene

$$\text{Capacidad}_{bat} = \frac{E_{sist} \cdot T_{reserva} \cdot 1000}{V_{bat} \cdot 3600} \cdot \gamma_{bat} = 21 \text{ Ah} \quad (15)$$

Por otro lado, para los cálculos del panel solar, se tienen en cuenta que se quiere que en un día de sol promedio se logre abastecer al sistema de su consumo energético normal diario. También, se requiere recargar un $\rho = 0.2$ de la capacidad total de la batería. Además, se considera un coeficiente de seguridad de $\gamma_{panel} = 2$. Es así que se llega a

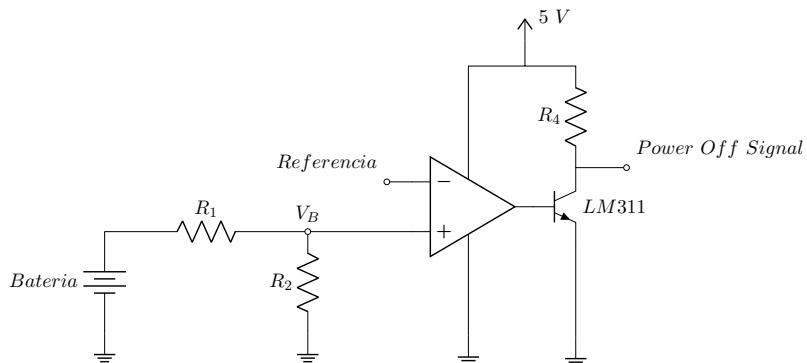
$$Pot_{panel} = \left(E_{sist} + \frac{\text{Capacidad}_{bat} \cdot V_{bat} \cdot 3600 \cdot \rho}{1000} \right) \cdot \gamma_{panel} \cdot \frac{1000}{60 \cdot 60 \cdot 8 \text{ hs}} = 23.1 \text{ W} \quad (16)$$

10.1.3.2 Protección para alimentación Raspberry Pi

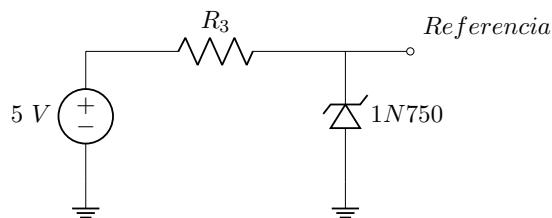
Ante la eventual pérdida de energía, en condiciones de funcionamiento normal, es sumamente probable que la memoria SD de la Raspberry Pi sea corrompida, dejándola inutilizable. Esto está contemplado en *T-INT-16*, por lo que se diseña un circuito de detección de baja batería, que apaga de manera segura la UP.

Para esto se agrega una bornera, la cual se conecta directamente la batería. De esta manera, se logra medir el nivel de tensión de alimentación, mediante un divisor resistivo.

Comparando este nivel con una tensión de referencia, cuando la batería alcanza un valor inferior al 25 % de su capacidad, se cambia el estado de la salida y se apaga la Raspberry Pi.



(a) Comparador de protección.



(b) Sección de referencia.

Figura 10.1.3.2.1: Circuito comparador e interruptores.

Se realizó la simulación del circuito propuesto, obteniendo los resultados esperados.

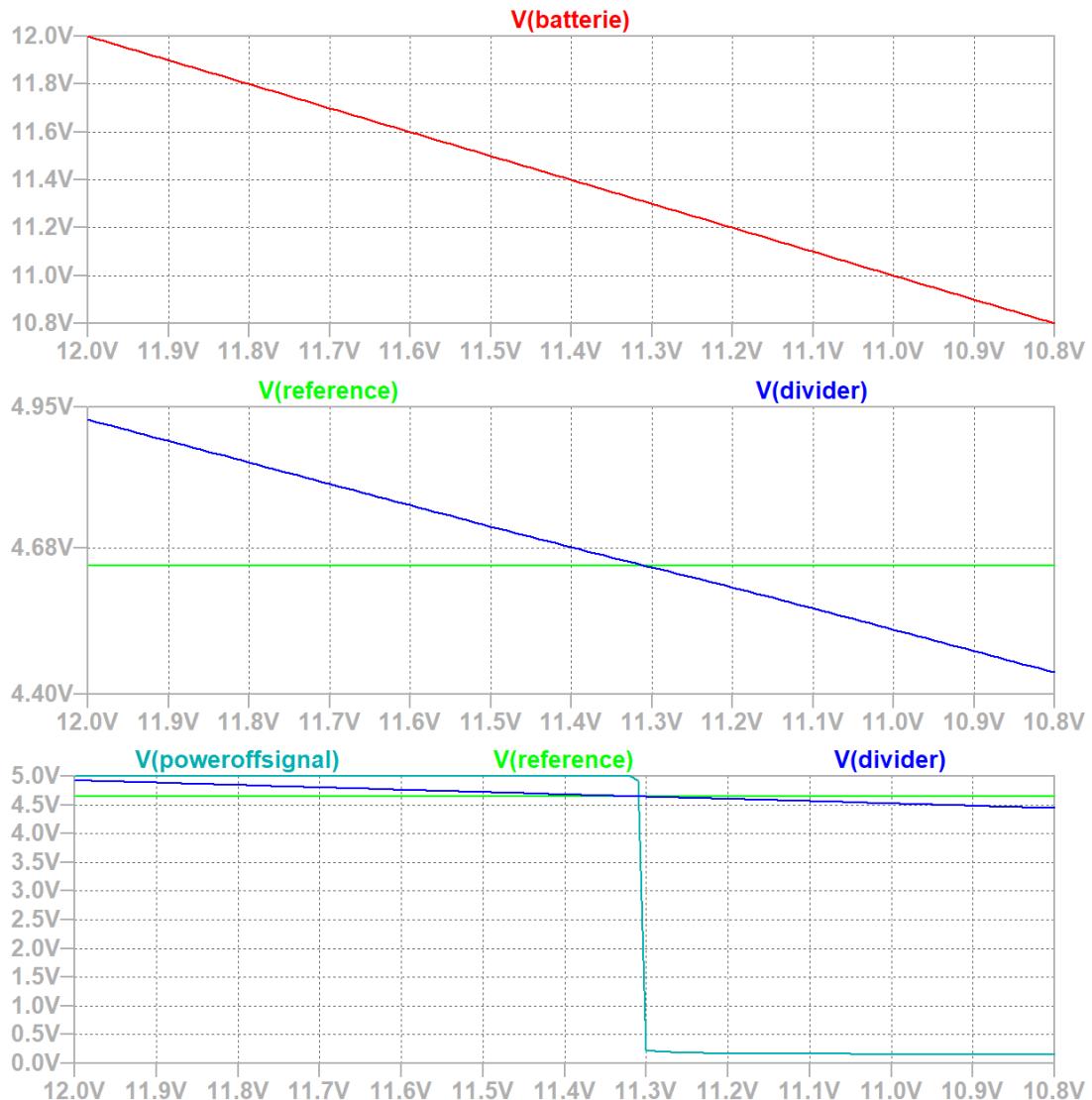


Figura 10.1.3.2.2: Simulación del circuito de protección.

Se observa como al caer la tensión de la batería, el circuito comparador lo detecta y activa la señal de apagado. Adicionalmente se agregan dos interruptores de apagado y prendido de modo seguro. Estos están destinados para los operarios de instalación y desinstalación, en caso de ser necesarios.

Finalmente, se configura la Raspberry Pi para que, con el cambio de tensión del comparador, se efectúen los cambios deseados.

```
sudo nano /boot/config.txt
dtoverlay=pio-shutdown, gpio_pin=26, active_low=1, gpio_pull=up
```

10.1.3.3 Cálculos circuitales

Los valores de las resistencias y capacitores utilizados como acondicionamiento de las señales del ESP32-WROOM-S3-1-N4 fueron extraídos de las recomendaciones de las hojas técnicas del dispositivo.

Tanto para el Bus de I^2C como para el DHT22, se calcularon las resistencias de *pull-up*⁵ de la siguiente manera:

$$R_p = \frac{V_{dd}}{I_{R_p}} = \frac{3.3\text{ V}}{1\text{ mA}} = 3.3\text{ k}\Omega \quad (17)$$

Para el circuito de protección, se tiene en cuenta que el nivel de tensión de la batería gradualmente disminuye con el uso, mientras que el valor del voltaje regulado de la Raspberry Pi no. Esto ocurre siempre y cuando se encuentre en el rango de funcionamiento habitual de la batería.

Es por esto que se decide utilizar esta última tensión para realizar una referencia utilizando una resistencia y un diodo zener de 4.6 V como el 1N750. Esta medida es comparada con la de la batería.

La comparación no es directa, ya que el nivel de tensión de la batería es mayor a los 5 V provistos por la Raspberry Pi. Es así que se emplea un divisor resistivo para bajar el valor. Para ello se tuvo en cuenta que la tensión a la que se baja debe ser mayor a la de referencia para el rango de capacidad de la batería de 25 % a 100 % y menor caso contrario.

$$R_1 + R_2 = R_{tot} \quad (18)$$

$$I_{leak} = \frac{V_{bat}}{R_{tot}} = 70\mu A \quad (19)$$

$$R_1 = \frac{V_{LowBat} \cdot R_2}{V_{trigger}} - R_2 \quad (20)$$

Con estas ecuaciones y teniendo en cuenta que $V_{LowBat} = 11.2\text{ V}$ y $V_{trigger} = V_{ref} = 4.6$, se definen los valores de las resistencias.

$$R_1 = 100\text{ K}\Omega \quad R_2 = 68\text{ K}\Omega \quad (21)$$

Por último, el valor del resistor utilizado para la referencia de tensión se calcula de la siguiente manera:

$$R_3 = \frac{V_{cc} - V_{zener}}{I_{zener}} = \frac{5\text{ V} - 4.6\text{ V}}{8.5\text{ mA}} = 47\text{ }\Omega \quad (22)$$

10.1.3.4 Cálculo de memoria

Se definen los siguientes valores para cada medición:

- Medición temperatura y humedad: 16 Bytes (fecha y hora) + 4 Bytes (humedad) + 4 Bytes (temperatura) = 24 Bytes.
- Medición luminosidad: 16 Bytes (fecha y hora) + 4 Bytes (luminosidad) = 20 Bytes.
- Medición cámara: cada imagen pesa 1.2 MBytes [16] (considerando una calidad alta).
- Información recibida por BT: aproximadamente 2 KByte.
- Sistema operativo: 2 GBytes.
- Programas y bibliotecas: 2.8 GBytes.

De esta forma se tiene el tamaño de las mediciones durante un período de 14 días:

$$\begin{aligned} 12 \text{ medición/hora} \cdot 24 \text{ hora/día} \cdot 14 \text{ día} \cdot (24 \text{ Bytes/medición} \\ + 16 \text{ Bytes/medición} + 1.2 \text{ MBytes}) + 2 \text{ KByte/día} \cdot 14 \text{ día} = 4.84 \text{ GBy} \end{aligned}$$

Si a esto se le suma el espacio dedicado al sistema operativo, programas y bibliotecas, se obtiene que el tamaño total de memoria necesaria es de aproximadamente 9 GBy.

Además, de lo mencionado en las cuentas, se tiene en cuenta el espacio utilizado por el *overhead* de la base de datos, así también un espacio adicional por si no se visita al nido en el tiempo pactado. Es así que con una memoria de 32 GBy basta para el proyecto.

⁵En los circuitos digitales, un resistor *pull-up* se utiliza para garantizar que la señal siempre se encuentre en un estado conocido.

10.1.4 Plan de Pruebas de cada Módulo

El plan de pruebas corresponde a las descritas en [8.1]. Para el uso correcto de los módulos se tuvieron en cuenta los siguientes aspectos:

- Definir un estado predeterminado del Bus I^2C mediante *pull-ups*.
- El uso de capacitores de desacople
- Para el módulo RTC, el integrado DS1307 requiere 5 V para funcionar correctamente, por lo que para que sea compatible con la lógica de 3.3 V de la Raspberry Pi, se realiza un cambio en el esquemático al remover dos resistores del módulo.

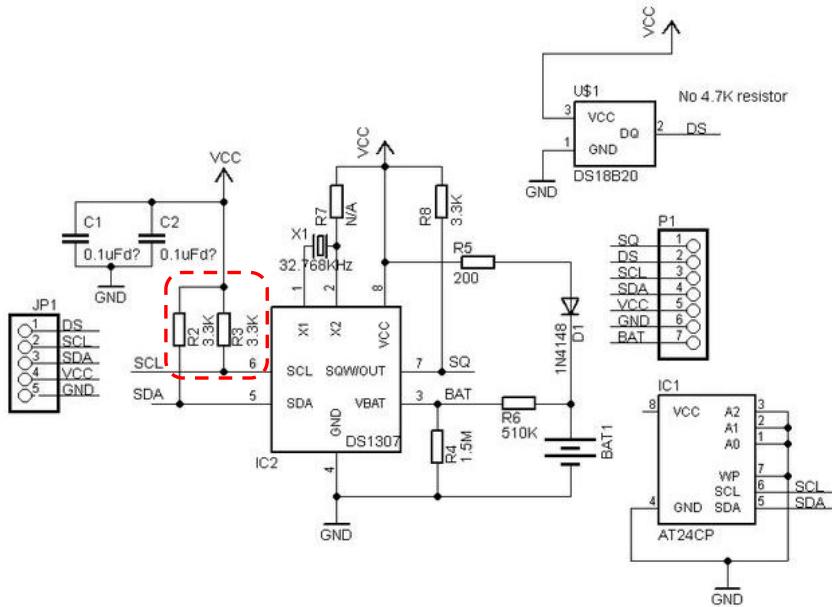


Figura 10.1.4.1: Esquemático RTC-Tiny.

Con este cambio se quita la referencia a 5 V del bus en el módulo. Por lo que la lógica queda en el rango de 3.3 V a 5 V. Cabe destacar que utilizar los niveles lógicos de 3.3 V son compatibles con el integrado *DS1307*

10.2 Software

10.2.1 Central de Control

El centro de control, compuesto por una Raspberry Pi, utiliza una distribución de *Linux Server*. Esto se debe a que no es de interés contar con una interfaz gráfica y es importante reservar espacio para almacenar los datos obtenidos desde los sensores y la cámara. Prescindir de la interfaz gráfica también permite ahorrar potencia dado que se apaga el puerto HDMI.

La lógica de control está desarrollada utilizando el lenguaje de programación *Python*. Su rico ecosistema de paquetes permite utilizar los diversos sensores y manejar las bases de datos.

10.2.2 Configuración Remota

La configuración de la Raspberry Pi se realiza de forma remota mediante SSH (*secure shell*). Dado que el proyecto apunta a ser escalable y tener más de un nido operativo, se buscó una forma de configurar la R-Pi manera automática y replicable. Para ello se utiliza una herramienta de automatización llamada Ansible. Esta nos permite guardar sobre archivos de texto las configuraciones que necesita la Raspberry Pi.

10.2.3 Eventos Periodicos

Durante la duración del proyecto la base de control deberá ejecutar una serie de acciones que se repetirán en el tiempo a intervalos regulares. Para poder orquestar estas actividades se hace uso de la utilidad nativa de Linux, los cronjobs. Utilizar esta actividad es ventajosa por sobre el método clásico de generar un único script que

se encargue de todo. De esta forma evitamos un punto de falla que pondría en riesgo la ejecución de las demás acciones.

Por lo tanto, según los intervalos pre-establecidos se ejecutarán diferentes tareas. A continuación se detallan las acciones que se ejecutarán de forma periódica

- Encendido de la antena BLE durante el día.
- Reinicio periódico cada ocho horas.
- Apagado y encendido del server.

10.2.4 Diagrama de Estados y Flujogramas

En este pasaje se habla sobre el diagrama de estados del sistema. El objetivo de este consiste en representar el funcionamiento de la lógica que se encuentra en *Node Red*.

Cabe aclarar que, por cuestiones de practicidad, en la Imagen (10.2.4.1) se simbolizan con nodos rojos la interacción que el usuario tiene con la interfaz gráfica. Estos nodos representan los datos que se ingresan en el servidor por parte del cliente. Es decir, una comunicación unidireccional de la persona hacia el nido.

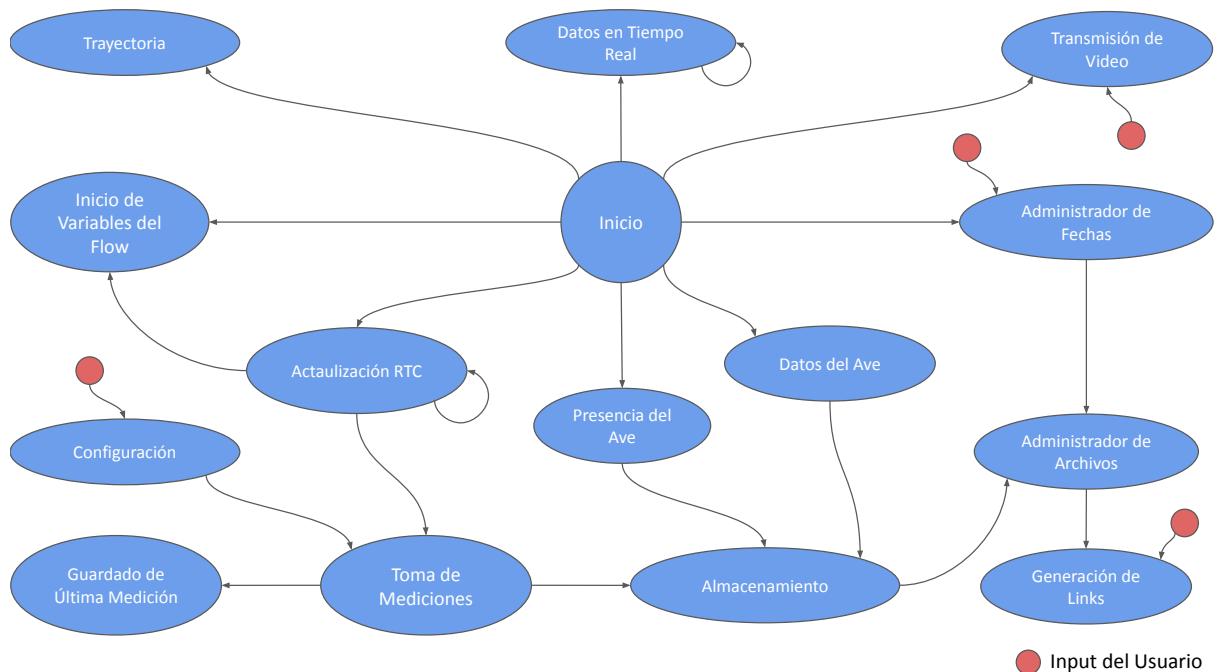


Figura 10.2.4.1: Diagrama de estados.

10.2.5 Descripción de Subrutinas

El nodo *Inicio* es el principal. Si bien es el más simple, es el más importante porque inicia todo el sistema y su configuración. Este estado se ocupa de mandar distintos tipos de señales de arranque. Estas varían dependiendo del tipo de inicio que se busca determinar, distinguiendo así tres versiones distintas:

- *Init*: esta señal se envía cuando se inicia el servidor o cuando se realiza un cambio en los nodos mediante el usuario de administrador.
- *F5*: se envía esta indicación cuando el usuario actualiza la página a la que tiene acceso.
- *Init And F5*: esta es una combinación de las dos anteriores. El objetivo de esta es simplificar el esquema del sistema al momento de conectar los distintos estados y manejar su comunicación.

El siguiente estado en cuanto a importancia es el de *Actualización RTC*. Este es un nodo que se ocupa de leer el periférico de RTC y actualizar el valor dentro de *Node Red* de manera periódica.

Con una de sus salidas se ocupa de inicializar las demás variables que se emplean en el servidor llamando a *Inicio de Variables del Flow*. Es importante aclarar que las variables del flow son del tipo globales, es decir, cada

nodo puede leerlas y modificarlas. Por cuestión de orden se asignó un estado propio para asignar valores por defecto y garantizar un correcto funcionamiento desde el arranque.

Con la salida faltante se comunica con el estado de *Toma de Mediciones*. Como su nombre lo indica, se ocupa de verificar si ha transcurrido el intervalo de tiempo requerido con respecto a la última medición realizada. En caso afirmativo, se realiza la medición y luego informa a *Guardado de Última Medición*, el cual actualiza la fecha y hora de este dato. Si bien *Inicio de Variables del Flow* configura un intervalo por defecto para cada variable, *Configuración* brinda la posibilidad de que el usuario cambie esos intervalos, manteniéndose en un margen de entre 5 minutos a 1 hora.

Con otra de las salidas del estado de *Toma de Mediciones* se indica al nodo de *Almacenamiento* de guardar el dato determinado. La razón por la cual *Almacenamiento* tiene una salida es porque debe informar a *Administración de Archivos* que hay datos nuevos. Este a su vez se ocupa de verificar cuáles de todos los datos que posee deben ser mostrados al usuario.

Esa decisión se encuentra regida en función de intervalos de tiempos, los cuales se determinan por una fecha inicial y final. Por defecto, *Variables del Flow* asigna como día inicial y final la fecha en la cual se haya inicializado el servidor. Sumado a eso, *Administrador de Fechas* se ocupa de brindarle al usuario la posibilidad de cambiar dicho período y de informar al *Administración de Archivos* de dicho cambio.

El estado *Generación de Links* se encarga de cargar los archivos adecuados para ser descargados y brindar al usuario acceso a estos.

Datos del Ave obtiene los datos que se encuentran en la mochila del ave, para luego facilitárselos a *Almacenamiento*. De una forma similar, *Presencia del Ave* se ocupa de detectar si el ave se encuentra en el entorno del nido, mostrarlo en pantalla.

Su conexión con el estado de *Almacenamiento* se da a que se guarda esta base de datos en la memoria de la R-Pi para aliviar el peso que debe soportar el servidor. Es por ello que estos a pesar de que el nodo que se ocupa de almacenar datos no brinda estos a *Administrador de Archivos* ya que no es de interés que el usuario tenga acceso a la descarga de estos.

El nodo de *Transmisión de Video* muestra en pantalla una transmisión en tiempo real del nido. Este se activa cuando el usuario lo solicite. Si bien es posible reiniciarlo y finalizarlo a voluntad, para evitar un uso excedido de dicha cualidad, y así evitar también una sobrecarga, el video se termina luego de 5 minutos de transmisión.

Datos en Tiempo Real toma una medida de luminosidad, temperatura y humedad de manera periódica. Para evitar un uso excesivo de memoria, estos datos se superponen. El objetivo de este es mostrar al usuario en tiempo real las condiciones por dentro del nido.

Por último, el nodo de *Trayectoria* se ocupa de mostrar la posición del ave en el entorno de medición de la red de seguimiento.

10.2.6 Interfaz Gráfica de Usuario

Para el desarrollo de la GUI, se valió del uso de *Node Red, software* que permite el desarrollo de servicios online. *Node Red*, mediante un sistema de nodos y sumado a la programación en *JavaScript*, permite generar una página web en una red local, lo cual cumple la función de interfaz con el usuario.

El sistema de edición y desarrollo de la GUI puede ser accedido a través de un navegador, de la misma manera que la interfaz del usuario. Para evitar que el usuario o alguna persona no deseada acceda a esta sección, se emplea un link privado, el cual es confidencial y no es publicado. También se vale de un sistema de autenticación que emplea la interfaz gráfica, creando un usuario de administrador.

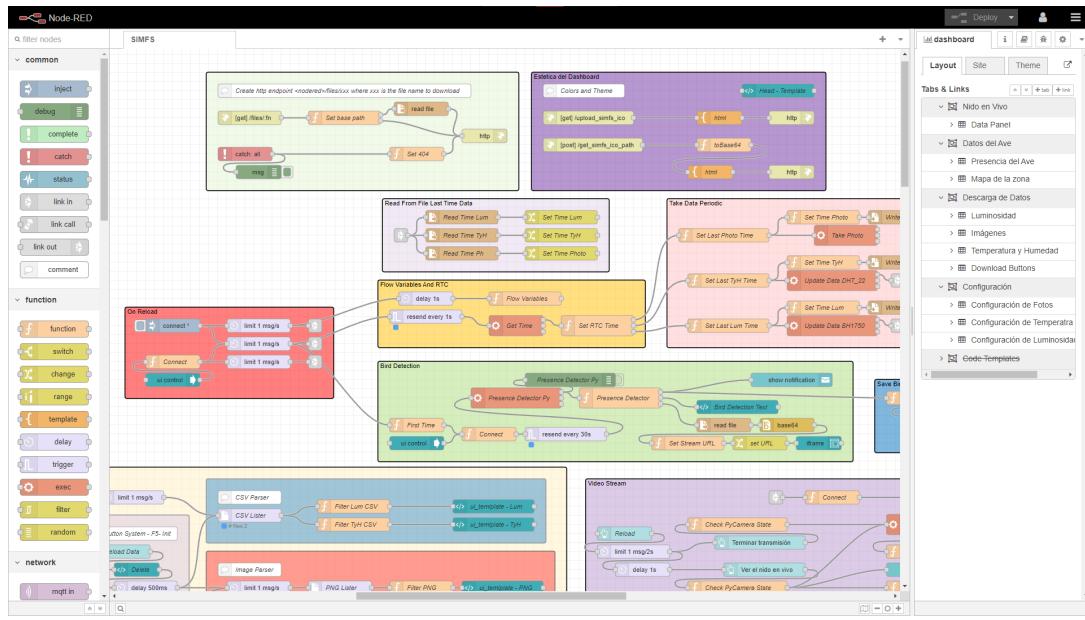
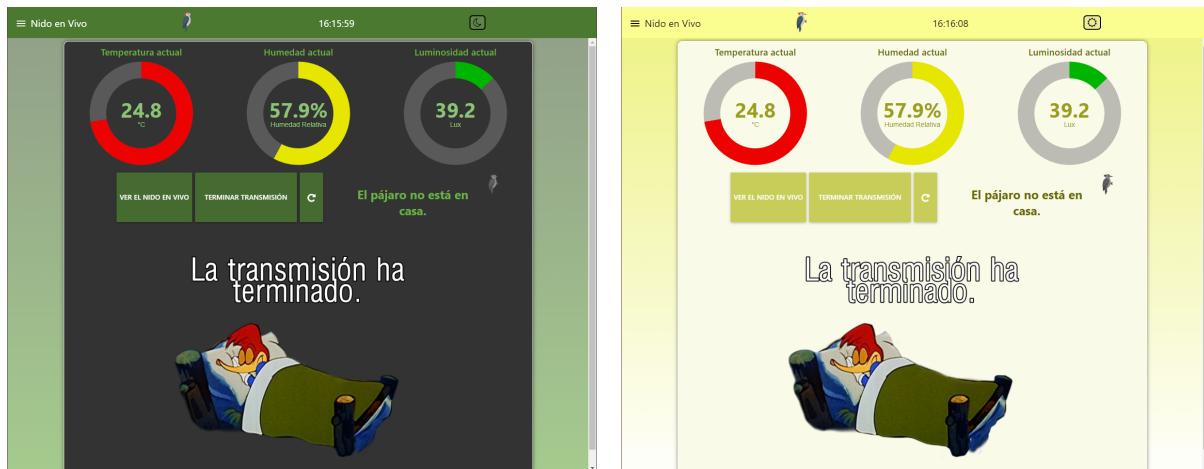


Figura 10.2.6.1: Flujo de nodos del servidor.

Dado que este servidor se encuentra corriendo en la R-Pi, al conectarse a la red de esta, se puede acceder a la página mencionada, donde se brindan varias funcionalidades adicionales a las descargas básicas de los datos requeridos.

La interfaz gráfica a la cual el usuario tiene acceso posee distintas pestañas, donde cada una brinda un parámetro diferente. En la primera pestaña es posible ver las mediciones realizadas de luminosidad, temperatura y humedad en tiempo real. Además, como el sistema detecta la presencia del ave, se muestra si este se encuentra dentro del nido o no. Por último, se brinda al usuario la posibilidad de ver la una transmisión en tiempo real del nido por dentro.



(a) Interfaz con usuario versión oscura.

(b) Interfaz con usuario versión clara.

Figura 10.2.6.2: Página del servidor a la cual accede el usuario.

En la segunda pestaña, es posible ver datos propios del ave, referidos a su posición. El primero muestra en un gráfico si el ave se ha encontrado dentro o fuera del nido durante un lapso de 24 horas, siendo posible seleccionar el día del cual se quiere ver los datos. Las zonas verdes denotan su estadía en el nido, mientras que las rajas su ausencia. Se muestra también cuanto tiempo duró cada intervalo. El segundo gráfico muestra el recorrido que ha realizado el ave en el entorno del nido en tiempo real.

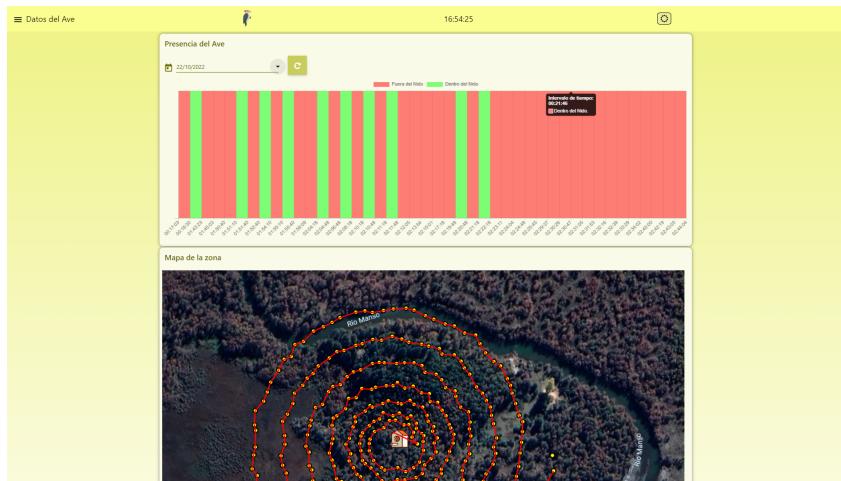


Figura 10.2.6.3: Datos de presencia y posición del ave.

La tercera pestaña es la más importante en cuanto a los requisitos del proyecto. En esta se presentan links de descarga para los datos obtenidos del nido y aquellos provistos por la unidad del pájaro. El usuario puede seleccionar el intervalo de tiempo en el cual desea obtener los datos.

Es posible descargar archivos del formato “csv” para los datos de luminosidad, temperatura y humedad, “png” si se desea descargar imágenes de forma individual o “zip” para el todo el conjunto de fotos mostradas.



Figura 10.2.6.4: Descarga de datos almacenados.

La cuarta y última pestaña le brinda al usuario cierto margen de configuración de la toma de datos. Estas opciones le permiten modificar el intervalo de cada medición, es decir cada cuanto tiempo se desea tomar una medición de las variables de interés o cada cuanto se desea tomar una foto.

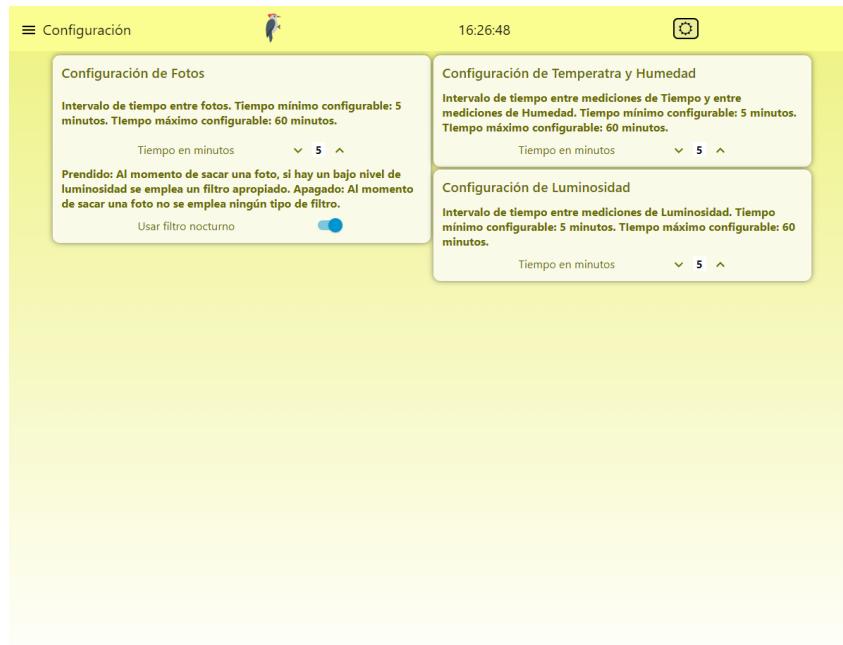


Figura 10.2.6.5: Configuración de la interfaz gráfica.

También es posible determinar si se desea o no emplear un filtro en la cámara al momento tomar fotos, considerando el nivel luminosidad. En caso de que dicha variable posea un valor bajo se emplea el filtro mejorando así la luminosidad de la foto.



(a) Foto tomada sin filtro de luminosidad.



(b) Foto tomada con filtro de luminosidad.

Figura 10.2.6.6: Comparación entre el uso del filtro de luminosidad.

Cabe destacar que de la misma forma que se considera un usuario de administrador, la GUI cuenta con una cuenta con contraseña para brindarle acceso al usuario y así evitar el uso de cualquier persona que consiga conexión a la red.

11. Construcción del Prototipo

11.1 Prototipo Nido-Sustrato

Para la construcción del prototipo del nido se tuvo en cuenta que el sustrato debe ser madera. Para las medidas se tuvieron en cuenta las dimensiones promedio de un nido de carpintero [1].

Por otro lado, para alojar y proteger la electrónica se contempló el desarrollo de encapsulados. Primero se trabajó con la Raspberry Pi y sus complementos. Se adaptó un diseño existente agrandando su tamaño para que

puedan introducirse las placas requeridas, se extendieron y fabricaron ranuras para el conexionado y se agregó un soporte que permita atornillar el conjunto. Para los sensores, se diseñó otro contenedor el cual contempla ranuras para que la cámara y los demás componentes estén en contacto con el exterior y ranuras para el conexionado con la R-Pi.

Es así que el diseño del prototipo de nido se puede observar en los planos especificados en las Figuras (16.1.3), (16.1.4) y (16.1.5), mientras que los encapsulados se encuentran en las Figuras (16.1.6), (16.1.7), (16.1.8) y (16.1.9).

Finalmente, en la Figura (11.1.1) se puede observar el prototipo del producto montado sobre un nido artificial construido con madera. En dicha imagen se puede observar un elemento extra cerca del encapsulado de la Raspberry Pi. Este complemento puede ser observado con mayor detalle en la Figura (16.1.10). La justificación del elemento se explica más adelante.

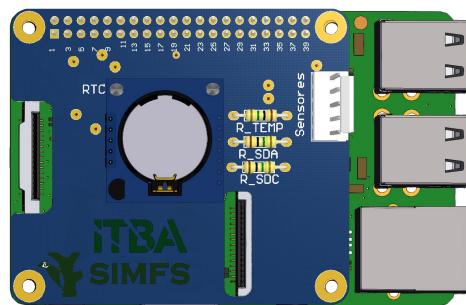


Figura 11.1.1: Prototipo del producto montado sobre un nido artificial construido con madera.

11.2 Prototipo Nido-Electrónica

11.2.1 Raspberry Pi

Para el prototipo se utiliza una Raspberry Pi 3B, sobre la cual se desarrollan los *drivers* de comunicación con los diversos sensores. Además se confeccionó un *shield* (encapsulado), el cuál realiza el acondicionamiento de las señales e interfaz para la placa de sensores, como se observa en la Figura (11.2.1.1a).



(a) Esquemático de la Raspberry Pi y su *shield*.



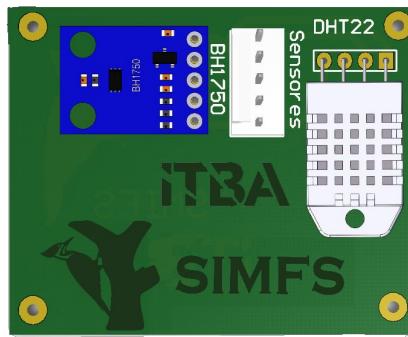
(b) Raspberry Pi y su shield implementado.

(c) Encapsulado de la Raspberry Pi y su *shield*.

Figura 11.2.1.1: Construcción de la unidad de procesamiento del prototipo.

11.2.2 Sensores

En cuanto a los sensores se diseña una placa que se ubica en la bóveda del nido, la cual reúne todos los sensores necesarios, y ofrece una interfaz con la placa *shield*.



(a) Esquemático de la placa de sensores.



(b) Placa de sensores implementada.

(c) Encapsulado de la placa de sensores.

Figura 11.2.2.1: Construcción de la placa de sensores y su encapsulado para el prototipo.

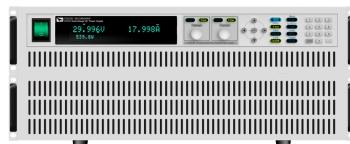
11.3 Prototipo Nido-Potencia

Para el prototipo del sistema de potencia del nido, se cuenta con la batería **TAIYO TYD12-33** de plomo-ácido de ciclo profundo AGM de 33 Ah y la fuente de tensión **ITECH IT6536C** con capacidad de simular la curva de

tensión-corriente de un panel solar.



(a) Batería utilizada en la construcción del prototipo.



(b) Fuente de tensión con capacidad de simular la curva tensión-corriente de un panel solar utilizado en el prototipo.

Figura 11.3.1: Comparación entre el uso del filtro de luminosidad.

Se utiliza el *Solar Power Manager DFR0580* que se encarga de extraer eficientemente la energía generada por el panel solar mediante el algoritmo de MPPT, cargar la batería y proveer las salidas de alimentación acondicionadas para la R-Pi.

Alternativamente, para disminuir el tamaño del prototipo, se reemplaza la fuente de alimentación del sistema. Este cambio se realiza únicamente en aquellos procesos en los cuales no influya utilizar otro sistema como batería, como son los bancos de prueba o el uso del prototipo en ambientes de desarrollo que no abarcan el sistema de abastecimiento, entre otros.

Se opta por utilizar un *Power Bank* (batería externa) que cumpla con los requisitos de tensión y corriente necesarios. El dispositivo elegido para esta tarea es el cargador *CP-V3* de la empresa *Sony*.

Las razones por las cuales se decidió utilizar dicha tecnología como alternativa son porque posee alta portabilidad, un tamaño reducido, es independiente de la red eléctrica, brinda una señal de alimentación estable y es fácilmente recargable.

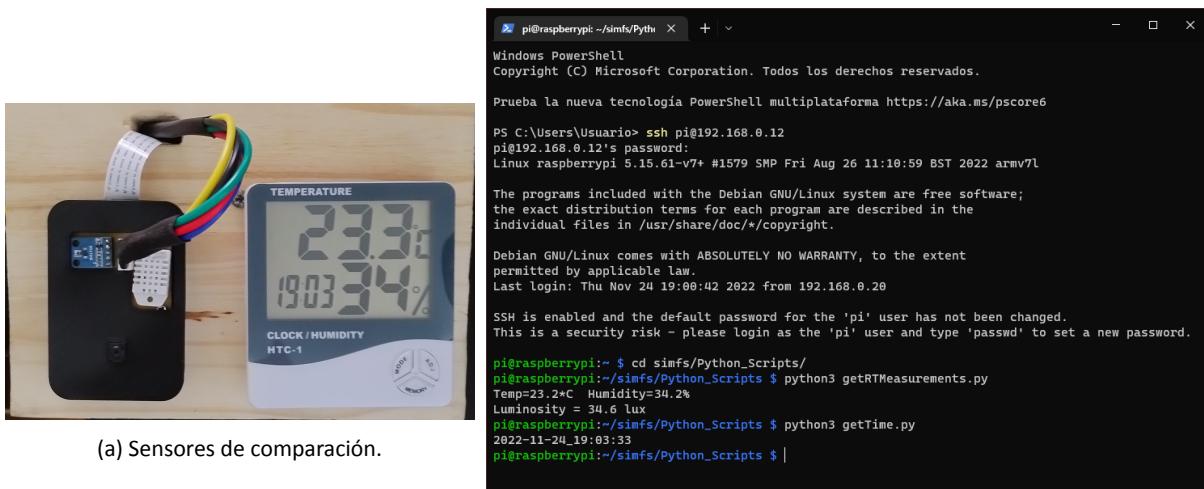
Como se mencionó en la Sección (11.1), este elemento se adosa al nido artificial mediante el soporte presentado en la Figura (16.1.10).

12. Validación del Prototipo

Para la validación del prototipo se emplean los mismos bancos de prueba mencionados en la Sección (8.2), ya que no existe una diferencia sustancial que implique la necesidad de alterar dichos ensayos.

12.1 Validación pruebas funcionales

Se empieza por las pruebas de *T-INT-FUN-01* y *T-INT-FUN-02*. Ambas pruebas se realizan a la par, no solo porque comparten el mismo banco de pruebas, sino porque se emplea el mismo sensor calibrado para la corroboración. Este sensor es capaz de medir tanto temperatura como humedad.



(a) Sensores de comparación.

(b) Medición obtenida.

Figura 12.1.1: Validación de funcionalidad *T-INT-FUN-01* y *T-INT-FUN-02*.

De esta forma se verifica que los sensores se encuentran calibrados con un margen de error menor al máximo permitido (desviación máxima de $\pm 0.5^\circ C$ y $\pm 2\%$ respectivamente).

Para las validaciones del sensor de luminosidad se desarrollaron las pruebas acorde a *T-INT-FUN-03*, obteniendo los resultados esperados para todos los casos, como se observa en la Figura (12.1.2).



(a) Sensor cubierto.

(b) Sensor descubierto.

(c) Sensor descubierto y apuntado con una linterna.

Figura 12.1.2: Validación de seguridad *T-INT-FUN-03*.

Disponiendo del banco de pruebas #3, se corroboró mediante observación los tests de funcionalidad *T-INT-FUN-04* y *T-INT-FUN-12*. Se garantizó que la imagen mostrada al usuario sea la adecuada y que se vea de manera fidedigna, es decir que la transmisión sea de calidad y fluida.



(a) Vista de la transmisión en vivo del nido.



(b) Imagen del exterior del nido en el momento de la transmisión.

Figura 12.1.3: Validación de funcionalidad T-INT-FUN-04 y T-INT-FUN-12.

El test de T-INT-FUN-05 se concluyó de manera satisfactoria. Resultados similares a los obtenidos se encuentran presentes en la Figura (10.2.6.6), donde se comparan fotos similares con y sin filtro. Por otro lado, se realiza la prueba T-INT-FUN-06, donde se varió la posición del módulo BLE, observando los cambios de su presencia.



(a) "El pájaro está en el nido".



(b) "El pájaro no está en casa".



(c) Beacon fuera del nido.



(d) Beacon en el nido.

Figura 12.1.4: Variación de posición del beacon con respecto al nido.

Las validaciones a continuación están relacionadas con aspectos funcionales de los sensores. Se comienza con el periodo de muestreo de estos como indica T-INT-FUN-07. Se puede corroborar que las mediciones son cada 5 minutos.

```

pi@raspberrypi:~/simfs/Python_Scripts/Clases $ ssh pi@192.168.0.12
pi@192.168.0.12's password:
Linux raspberrypi 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 18 17:42:29 2022 from 192.168.0.20

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~/simfs/Python_Scripts/Clases $ python3 print_measures.py
[('2022-11-18_16:54:28', 22.780000762939495, 0.9, 980001525878986), ('2022-11-18_16:59:26', 22.780000762939495, 0.9, 980001525878986), ('2022-11-18_17:04:31', 22.799999237860547, 0.9, 70000076293945), ('2022-11-18_17:09:31', 22.809999618539273, 0.5, 599998474121094), ('2022-11-18_17:14:31', 23.1000003814659727, 0.5, 599998474121094), ('2022-11-18_17:19:34', 23.309999618539273, 0.5, 20000076293945), ('2022-11-18_17:39:38', 24.5, 60.799999237860547, 0.6, 79999923786055)], ('2022-11-18_17:39:44', 24.5, 60.799999237860547, 0.6, 79999923786055)]
pi@raspberrypi:~/simfs/Python_Scripts/Clases $

```

Figura 12.1.5: Validación de funcionalidad *T-INT-FUN-07*.

Para *T-INT-FUN-08* se observa que el tiempo obtenido en consola como el de la computadora, presentado en la parte inferior derecha de la imagen, coinciden. Es por esto que se considera que la prueba concluyó de manera exitosa.

```

PS C:\Users\Guido L> ssh pi@192.168.0.12
pi@192.168.0.12's password:
Linux raspberrypi 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST
2022 armv7l

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in t
he
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the exten
t
permitted by applicable law.
Last login: Thu Nov 17 17:15:06 2022 from 192.168.0.6

SSH is enabled and the default password for the 'pi' user has no
t been changed.
This is a security risk - please login as the 'pi' user and type
'passwd' to set a new password.

pi@raspberrypi:~ $ cd simfs/Python_Scripts/
pi@raspberrypi:~/simfs/Python_Scripts $ python3 getTime.py
2022-11-17_17:17:41
pi@raspberrypi:~/simfs/Python_Scripts $ |

^  ENG  5:17 PM
11/17/2022

```

Figura 12.1.6: Validación de funcionalidad *T-INT-FUN-08*.

La siguiente prueba corresponde a *T-INT-FUN-09*. En esta se desenergiza la Raspberry Pi, y luego de esperar el tiempo apropiado, se la energiza nuevamente corroborando que la hora correspondiente al módulo RTC sea correcta.

```
pi@raspberrypi:~ $ cd simfs/Python_Scripts/
pi@raspberrypi:~/simfs/Python_Scripts $ python3 getTime.py
2022-11-17_18:01:13
pi@raspberrypi:~/simfs/Python_Scripts $ sudo shutdown -h now
Connection to 192.168.0.12 closed by remote host.
Connection to 192.168.0.12 closed.
PS C:\Users\Guido L> ssh pi@192.168.0.12
pi@192.168.0.12's password:
Linux raspberrypi 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST
2022 armv7l

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in t
he
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the exten
t
permitted by applicable law.
Last login: Thu Nov 17 18:01:28 2022 from 192.168.0.6

SSH is enabled and the default password for the 'pi' user has no
t been changed.
This is a security risk - please login as the 'pi' user and type
'passwd' to set a new password.

pi@raspberrypi:~ $ cd simfs/Python_Scripts/
pi@raspberrypi:~/simfs/Python_Scripts $ python3 getTime.py
2022-11-17_18:03:09
pi@raspberrypi:~/simfs/Python_Scripts $ |
```

Figura 12.1.7: Validación de funcionalidad *T-INT-FUN-09*.

12.2 Validación Pruebas de Comunicación 1

T-INT-COM1-01 se efectúa disponiendo del banco de pruebas número #3. Se emplea un dispositivo de comunicación mediante protocolo BLE. Se observa en las diversas consolas que no solo se efectúa la transmisión de datos, sino que el archivo no sufre ninguna pérdida de paquetes ya que ambos hash coinciden.

```

~/Desktop/pi@raspberrypi:simfs/BLECommunication
$ python receiveBPSData.py
Connecting to: 8C:AA:B5:87:6F:12
Connected
---get service(UUID:0x1825)-characteristic(UUID:0x2A68)---
Service <wid=0x1825 handleStart=40 handleEnd=65535>
<class 'bluepy.btle.Characteristic'>
<bluepy.btle.Characteristic object at 0xb5518150>
Characteristic <Navigation>
{Date, longitud, latitud
 2022-09-01 07:00, -70.99546440522245, -44.98539606888342
 2022-09-01 08:00, -70.98940077464881, -44.96978485265774
 2022-09-01 09:00, -70.97283729921473, -44.957373488811655
 2022-09-01 10:00, -70.95658068749987, -44.96493015263342
 2022-09-01 11:00, -70.98331579257334, -44.971547817520715
 2022-09-01 12:00, -70.9970284517916, -44.96869887746417
 2022-09-01 13:00, -70.97057429048496, -44.97260842382192
 2022-09-01 14:00, -70.96614905334705, -44.98701675325146
 2022-09-01 15:00, -70.96491593058316, -44.97651013046015
 2022-09-01 16:00, -70.96340409254134, -44.97401276549228
 2022-09-01 17:00, -70.97935037411875, -44.99423472922937
 2022-09-01 18:00, -70.98917608906055, -44.9651622651613
 2022-09-01 19:00, -70.99626670660012, -44.97224164530085, ...
Transmission Ended.
Hash of received file: FB9774E1965600BC4E706D87816688A1B0C2B0EAE591E9C72504EF4D6C8A7B29

```

Figura 12.2.1: Consola del lado del receptor Raspberry Pi.

```

// start advertising
BLE.advertise();

Serial.println("BLE Object transfer Peripheral");
}

void loop() {
  // listen for Bluetooth® Low Energy peripherals to connect:
  BLEDevice central = BLE.central();

  // if a central is connected to peripheral:
  if (central) {
    Serial.print("Connected to central: ");
    Serial.println("Device connected");
    Serial.println(central.address());

    // while the central is still connected to peripheral:
    while (central.connected()) {
      // Post bird data
      Serial.println("Publishing Service-Characteristic");
      Serial.print("Hash of file: ");
      Serial.println(hash);
      navigationCharacteristic.writeValue(BirdData);
    }
    // when the central disconnects, print it out:
    Serial.print(F("Disconnected from central: "));
    Serial.println(central.address());
  }
}

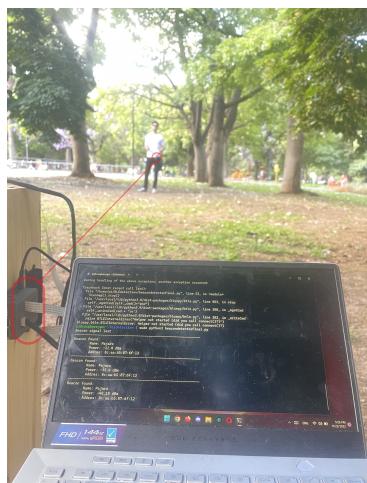
```

Figura 12.2.2: Consola del lado del transmisor BPS

Se realizó la prueba acorde a *T-INT-COM1-02*, donde se obtuvieron mediciones a distintas distancias, variando entre 1 y 30 metros. Se obtuvieron resultados satisfactorios, por lo que la prueba es superada con especificaciones incluso superiores a las definidas.



(a) Distancia de 1 metro.



(b) Distancia de 15 metros.

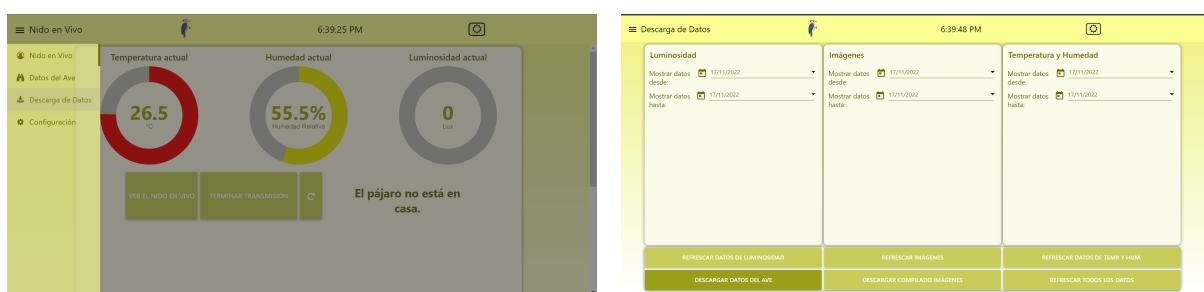


(c) Distancia de 30 metros.

Figura 12.2.3: Validación T-COM1-02.

12.3 Validación Pruebas de Comunicación 2

Para las pruebas de Comunicación 2 se requiere la computadora de un tercero (biólogo en campo). La primera prueba corresponde a T-INT-COM2-01 la cual consta en analizar a la capacidad de acceder a los datos del ave guardados en el nido.



(a) Pestaña de descargas.

(b) Botón de descarga.

The screenshot shows two windows side-by-side. On the left is the Node-RED Dashboard with a sidebar titled 'Descarga de Datos' containing 'Temperatura y Humedad' and 'Luminosidad'. Below it are buttons for 'REFRESCAR DATOS DE LUMINOSIDAD', 'DESCARGAR LOS DATOS DEL AIRE', and 'REFRESCAR TODOS LOS DATOS'. On the right is a terminal window titled 'pi@raspberrypi: ~/simfs/simf' showing the contents of 'BirdData.csv'. The terminal output is a list of data rows:

	Date	longitud	latitud
1	9/1/2022 7:00	-70.9955	-44.9854
2	9/1/2022 8:00	-70.9894	-44.9698
3	9/1/2022 9:00	-70.9728	-44.9574
4	9/1/2022 10:00	-70.9566	-44.9649
5	9/1/2022 11:00	-70.9833	-44.9715
6	9/1/2022 12:00	-70.997	-44.9687
7	9/1/2022 13:00	-70.9706	-44.9726
8	9/1/2022 14:00	-70.9561	-44.987
9	9/1/2022 15:00	-70.9649	-44.9765

(c) Consola y csv descargado.

Figura 12.3.1: Validación de comunicación *T-INT-COM2-01*.

Se observa que los datos descargados que se ven en la izquierda se corresponden con los datos que se encuentran en el nido como se ve a la derecha. Por lo que no hubo una pérdida de datos. Adicionalmente se utilizó una función de *hash* para comparar los contenidos de ambos archivos y fehacientemente comprobar que son el mismo archivo.

Por su parte, el desarrollo de las pruebas de *T-INT-COM2-02* se efectuó sin mayores inconvenientes, logrando la conexión continua variando la distancia en rango de 4 a 14 metros.

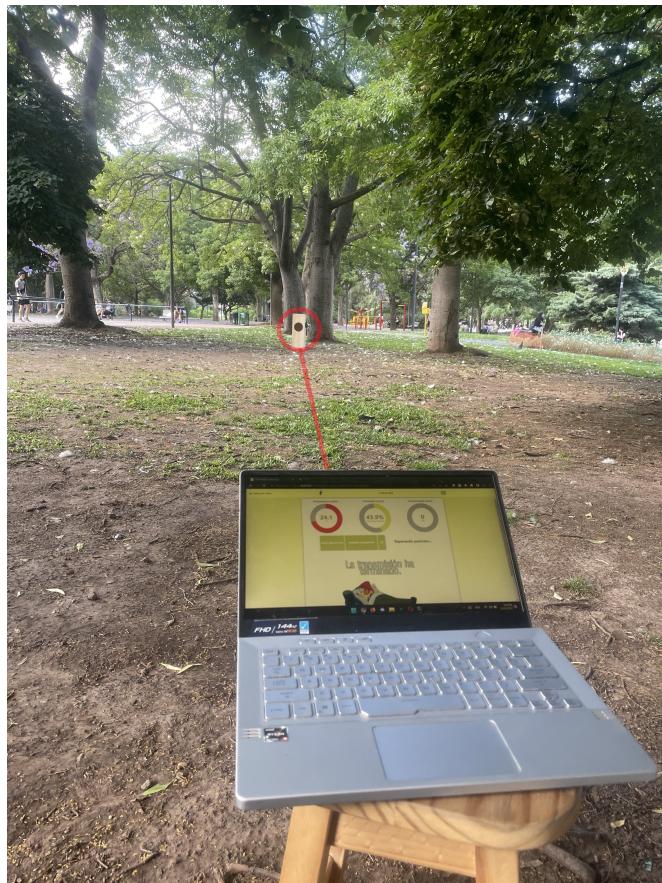


Figura 12.3.2: Validación de funcionalidad *T-INT-COM2-02*.

Finalmente con lo que respecta a los bancos de prueba de comunicación, se procede a realizar la validación

correspondiente a *T-INT-COM2-03*. En esta prueba se de verificar la existencia del archivo *BirdData* luego descargarlo y al acceder nuevamente a la Raspberry Pi comprobar que este fue borrado exitosamente.

```
pi@raspberrypi:~/simfs/simfs_export_data $ ls
BirdData.csv
Chart.js
Datos_Fotos.zip
Imagen_2022-10-31_16-59-39.png
Imagen_2022-10-31_17-29-40.png
Imagen_2022-10-31_17-37-01.png
Imagen_2022-10-31_17-42-01.png
Imagen_2022-10-31_17-47-01.png
Imagen_2022-10-31_17-52-01.png
Imagen_2022-10-31_17-57-01.png
Imagen_2022-10-31_18-02-01.png
Imagen_2022-10-31_18-07-02.png
Imagen_2022-10-31_18-12-02.png
Imagen_2022-10-31_18-17-02.png
Imagen_2022-10-31_18-22-02.png
Imagen_2022-10-31_18-27-02.png
Imagen_2022-10-31_18-32-02.png
Imagen_2022-10-31_18-37-02.png
Imagen_2022-10-31_18-42-03.png
Imagen_2022-10-31_18-47-03.png
Imagen_2022-10-31_18-52-03.png
Imagen_2022-10-31_18-57-03.png
Imagen_2022-10-31_19-02-04.png
Imagen_2022-10-31_19-07-04.png
Imagen_2022-10-31_19-12-04.png
Imagen_2022-10-31_19-17-04.png
Imagen_2022-10-31_19-22-04.png
Imagen_2022-10-31_19-27-04.png
Imagen_2022-11-01_16-54-18.png
Imagen_2022-11-01_16-59-18.png
Imagen_2022-11-01_17-04-18.png

pi@raspberrypi:~/simfs/simfs_export_data $ ls
Chart.js
Datos_Fotos.zip
Imagen_2022-10-31_16-59-39.png
Imagen_2022-10-31_17-29-40.png
Imagen_2022-10-31_17-37-01.png
Imagen_2022-10-31_17-42-01.png
Imagen_2022-10-31_17-47-01.png
Imagen_2022-10-31_17-52-01.png
Imagen_2022-10-31_17-57-01.png
Imagen_2022-10-31_18-02-01.png
Imagen_2022-10-31_18-07-02.png
Imagen_2022-10-31_18-12-02.png
Imagen_2022-10-31_18-17-02.png
Imagen_2022-10-31_18-22-02.png
Imagen_2022-10-31_18-27-02.png
Imagen_2022-10-31_18-32-02.png
Imagen_2022-10-31_18-37-02.png
Imagen_2022-10-31_18-42-03.png
Imagen_2022-10-31_18-47-03.png
Imagen_2022-10-31_18-52-03.png
Imagen_2022-10-31_18-57-03.png
Imagen_2022-10-31_19-02-04.png
Imagen_2022-10-31_19-07-04.png
Imagen_2022-10-31_19-12-04.png
Imagen_2022-10-31_19-17-04.png
Imagen_2022-10-31_19-22-04.png
Imagen_2022-10-31_19-27-04.png
Imagen_2022-11-01_16-54-18.png
Imagen_2022-11-01_16-59-18.png
Imagen_2022-11-01_17-04-18.png
```

Figura 12.3.3: Validación de comunicación *T-INT-COM2-3*.

12.4 Validación pruebas de seguridad

Se dispuso del banco de pruebas #1 para analizar la aplicabilidad de *T-RAM-SEG-01*. El acceso tanto a la interfaz gráfica del usuario como la del administrador se encuentran restringidas para cualquiera que no posea las credenciales adecuadas.

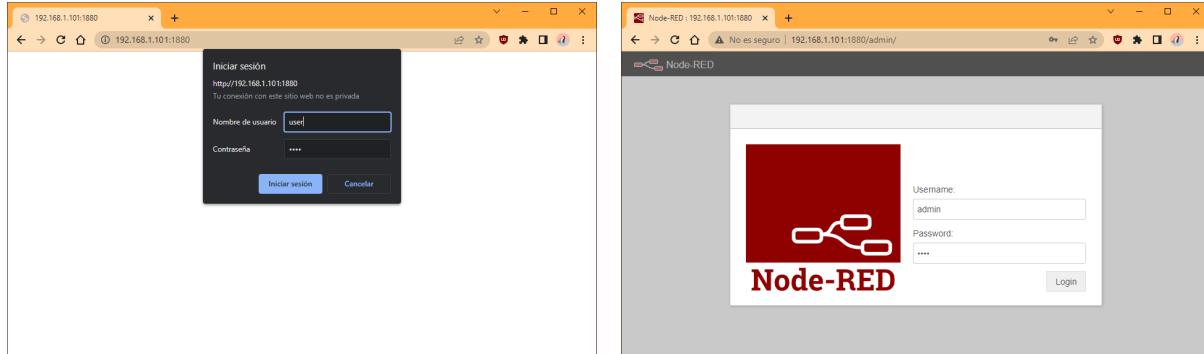


Figura 12.4.1: Validación de seguridad *T-RAM-SEG-01*.

Se obtuvo un resultado mejor del esperado ya que el hecho de emplear una red WiFi para conectar la unidad del nido con el usuario también requiere del uso de una contraseña. Esto agrega una capa de seguridad extra al sistema.

13. Estudios de Confiabilidad

En esta sección se analiza la confiabilidad de los elementos de *hardware* y *software*, exponiendo en cada caso los criterios utilizados. A su vez se separa la parte de *hardware* observando cada parte de manera individual para luego obtener un resultado en conjunto.

13.1 Hardware

El estudio se lleva a cabo según la norma MIL-HDBK-217F, la cual estima la cantidad de fallas en unidad de tiempo para diferentes tipos de componentes electrónicos. Este estándar tiene en cuenta factores como las con-

diciones de uso, temperatura.

El primer paso consiste en realizar un gráfico de todos los módulos considerados críticos que garantizan las funcionalidades fundamentales del producto. Esto se presenta en la Figura (13.1.1).

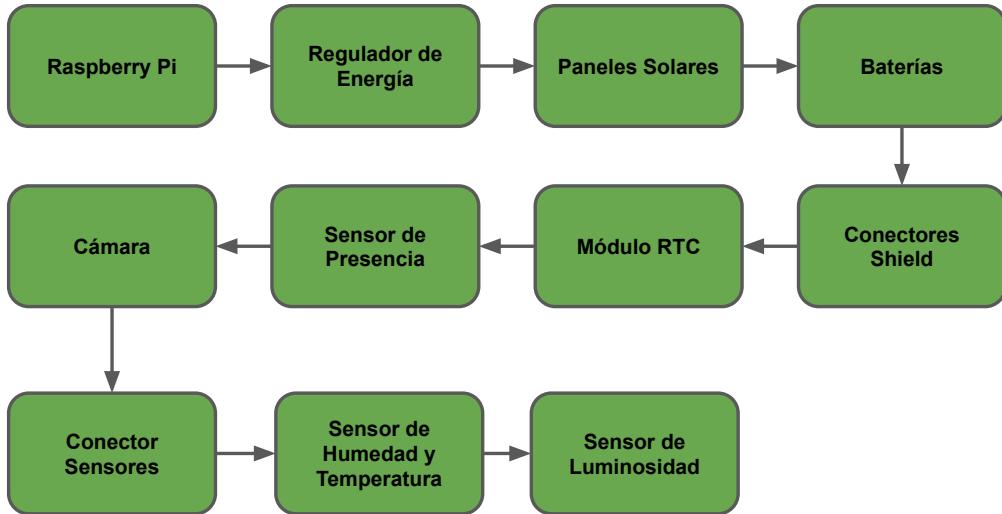


Figura 13.1.1: Módulos críticos de funcionamiento.

Se calcula una estimación de la confiabilidad de cada módulo utilizando una estrategia “Top-Down”. Se encuentra la tasa de falla de cada submódulo mediante un proceso multiplicativo a la tasa de fallos base provista por las distintas cargas a las que son sometidos los componentes. No se cuenta con redundancias, por lo que el valor de confiabilidad del módulo estará dado por la confiabilidad en serie de cada uno de sus componentes. Para todos los cálculos se considera una temperatura de 16°C, la cual es la temperatura media de la zona d estudio en el periodo del mismo.

Tipo de componente	Cálculo de tasa de fallos [failures / 10 ⁶ .hour]
Resistencia	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_P \cdot \pi_S \cdot \pi_Q \cdot \pi_E$
Capacitor	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_C \cdot \pi_V \cdot \pi_{SR} \cdot \pi_Q \cdot \pi_E$
IC DHT-22	$\lambda_p = (C_1 \cdot \pi_T + C_2 \cdot \pi_E) \cdot \pi_Q \cdot \pi_L$
IC BH1750	$\lambda_p = \lambda_{BD} \cdot \pi_{MFG} \cdot \pi_T \cdot \pi_{CD} + \lambda_{BP} \cdot \pi_E \cdot \pi_Q + \lambda_{EOS}$
Shield	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_K \cdot \pi_Q \cdot \pi_E$
Conexionado Sensores	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_K \cdot \pi_Q \cdot \pi_E$
IC DS1307	$\lambda_p = \lambda_{BD} \cdot \pi_{MFG} \cdot \pi_T \cdot \pi_{CD} + \lambda_{BP} \cdot \pi_E \cdot \pi_Q + \lambda_{EOS}$
Diodo	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_S \cdot \pi_C \cdot \pi_Q \cdot \pi_E$
Regulador de Tensión	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_S \cdot \pi_C \cdot \pi_Q \cdot \pi_E$
Transistor Mosfet	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_A \cdot \pi_R \cdot \pi_S \cdot \pi_Q \cdot \pi_E$
VHSIC	$\lambda_p = \lambda_{BD} \cdot \pi_{MFG} \cdot \pi_T \cdot \pi_{CD} + \lambda_{BP} \cdot \pi_E \cdot \pi_Q + \lambda_{EOS}$
Batería	Documentación
Panel Solar	Documentación
Cámara Raspberry Pi	Documentación
Inductor	$\lambda_p = \lambda_b \cdot \pi_T \cdot \pi_Q \cdot \pi_E$
ESP8266	$\lambda_p = (C_1 \cdot \pi_T + C_2 \cdot \pi_E) \cdot \pi_Q \cdot \pi_L$
Raspberry Pi	Documentación

Tabla 13.1.1: Ecuaciones para obtener la tasa de fallos de cada módulo acorde a MIL-HDBK-217F.

13.1.1 Raspberry Pi

Para el módulo Raspberry Pi se utiliza la documentación de esta para obtener el valor de la tasa de fallos.

$$\lambda_p = 3 \quad (23)$$

13.1.2 Regulador de Energía

Componente	Cantidad	λ_p	λ_{Tot}
Capacitor	9	0.00356	0.03204
Resistor	36	0.00267	0.09612
Diodo	7	0.0007	0.0049
Inductor	2	0.06138	0.12276
Transistores MOSFET	13	0.1776	2.3088
IC Step-Down	1	1.106	1.106
IC MPPT	1	1.017	1.017
IC Protocolo USB	1	1.004	1.004
Total			5.69162

Tabla 13.1.2.1: Confiabilidad regulador de energía.

13.1.3 Paneles Solares

Para determinar a tasa de fallas de los paneles solares se utiliza la documentación de estos.

$$\lambda_p = 0.057 \quad (24)$$

13.1.4 Baterías

Para determinar a tasa de fallas de las baterías se utiliza la documentación.

$$\lambda_p = 0.005 \quad (25)$$

13.1.5 Conector Shield

Componente	Cantidad	λ_p	λ_{Tot}
Conectores PCB	1	0.00182	0.00182
Conector Molex	1	0.00182	0.00182
Total			0.00364

Tabla 13.1.5.1: Confiabilidad de la placa *shield*.

13.1.6 RTC

Componente	Cantidad	λ_p	λ_{Tot}
Capacitor	3	0.00356	0.01068
Resistor	5	0.00267	0.01335
Diodo	1	0.0007	0.0007
DS1307	1	1.106	1.106
Total			1.13073

Tabla 13.1.6.1: Confiabilidad RTC.

13.1.7 Sensor de Presencia

Componente	Cantidad	λ_p	λ_{Tot}
ESP32 WROOM S3	1	1.08	1.08
Total			1.08

Tabla 13.1.7.1: Confiabilidad sensor de presencia.

13.1.8 Cámara Raspberry Pi

Para determinar la tasa de fallas del módulo de cámara se recurre a la hoja de datos.

$$\lambda_p = 3.16 \quad (26)$$

13.1.9 Conector Sensores

Componente	Cantidad	λ_p	λ_{Tot}
Conector Molex	1	0.00182	0.00182
Total	0.00182		

Tabla 13.1.9.1: Confiabilidad Conector Placa de Sensores

13.1.10 Sensor Humedad y Temperatura

Componente	Cantidad	λ_p	λ_{Tot}
Capacitor	2	0.00356	0.00712
Resistor	1	0.00267	0.00267
IC-DHT22	1	3.036	3.036
Total	3.04579		

Tabla 13.1.10.1: Confiabilidad Sensor de Temperatura y Humedad.

13.1.11 Sensor Luminosidad

Componente	Cantidad	λ_p	λ_{Tot}
Capacitor	4	0.00356	0.01424
Resistor	3	0.00267	0.00801
Regulador de tensión	1	0.00059	0.00059
IC-BH1750	1	1.106	1.106
Total	1.12884		

Tabla 13.1.11.1: Confiabilidad Sensor de Luminosidad

13.1.12 Tasa de fallas total

Utilizando los valores finales obtenidos en las anteriores tablas se calcula el λ_{Total} de la siguiente manera:

$$\lambda_{Total} = \sum_{Modulos} \lambda_{Totn} = 18.8174 \frac{\text{fallas}}{10^6 \text{ horas}} \quad (27)$$

Luego asumiendo que λ es constante y la función de confiabilidad definida como:

$$R(t) = e^{-\frac{\lambda_{Total}}{10^6} \cdot t} \quad (28)$$

Con esto definido se obtiene el tiempo medio a la falla para todo el sistema:

$$MTTF = \int_0^{\infty} R(t) dt \approx 53142.19 \text{ horas} \approx 6.066 \text{ años} \quad (29)$$

Además, se calcula valores para $R(T)$ con un paso de 3 meses (Duración base del proyecto) hasta 3 años.

Tiempo	3 Meses	6 Meses	9 Meses	1 Año	1 Año y 3 Meses	1 Año y 6 Meses
$R(t)$	0.96	0.922	0.885	0.850	0.816	0.786
Tiempo	1 Año y 9 Meses	2 Años	2 Años y 3 Meses	2 Años y 6 Meses	2 Años y 9 Meses	3 Años
$R(t)$	0.752	0.722	0.70	0.677	0.647	0.622

Tabla 13.1.12.1: Probabilidad de confiabilidad con períodos de 3 meses.

Teniendo en cuenta que el tiempo de vida del producto está estimado en 3 años y que la duración del estudio es de 3 meses. Que el tiempo medio a la falla sea de 6 años, resulta satisfactorio. Finalmente cabe mencionar que luego de cada uso se recomienda una revisión técnica para asegurar un correcto funcionamiento de todos los módulos así aumentando la confiabilidad del sistema.

13.2 Software

Para el estudio de confiabilidad de software se utilizó un modelo de estimación de tipo exponencial. Este modelo asume que todas las fallas que se presentan son independientes entre sí. Más específicamente se utilizará el modelo de Shooman, el cual normaliza las fallas por la cantidad de líneas de código I_T sigue:

$$R(t) = e^{-\lambda t} \quad (30)$$

donde se toma

$$\lambda = -k [\epsilon_T - \epsilon(\zeta)] t \quad (31)$$

Además, se define ϵ_T como la tasa de errores totales, $\epsilon(\zeta)$ la tasa de errores corregidos hasta un tiempo de corrección ζ y k una constante del modelo de estimación.

Semanas	E_t	E_c	E_r	H [hrs]	λ_i	$\frac{\lambda_i}{\lambda_{i-1}}$	\hat{E}_{tot}	\hat{k}
1	20	17	3	168	0.11905	-	-	-
2	37	33	4	336	0.05060	0.42500	11.82609	32.32474
3	50	43	7	504	0.02579	0.50980	10.40000	37.93184
4	61	52	9	672	0.01637	0.63462	15.63158	12.34174
5	74	65	9	840	0.01548	0.94545	225.33333	0.35769
6	89	80	9	1008	0.01488	0.96154	375.00000	0.20329
7	105	96	9	1176	0.01361	0.91429	170.66667	0.42079
8	115	105	10	1344	0.00744	0.54688	10.86207	43.15476

Tabla 13.2.1: Fallas durante el periodo de test del software.

La Tabla (13.2.1) presenta la cantidad acumulada de errores E_t , la cantidad acumulada de corregidos E_c , la cantidad acumulada resultante de los residuales E_r , las horas de test acumuladas H [hrs], la tasa de fallos λ y los parámetros invariables del modelo \hat{E}_{tot} y \hat{k} a lo largo del periodo de test. Estos son calculados mediante la formula:

$$\hat{E}_{tot} = \frac{\frac{\lambda_i}{\lambda_{i-1}} E_c(\tau_{i-1}) - E_c(\tau_i)}{\frac{\lambda_i}{\lambda_{i-1}} - 1} \quad (32)$$

y

$$\hat{k} = \lambda_i \frac{I_T}{\hat{E}_{tot} - E_c(\tau_i)} \quad (33)$$

Finalmente, se adoptan los valores $\hat{E}_{tot} = 1$, $I_T = 5000$ y $\hat{k} = 43.15476$. Aproximando según

$$\lambda = k \frac{E_r}{I_T} \quad (34)$$

se obtiene $\lambda = 0.00863$ que, tomando el inverso de este valor se obtiene el *TMEF*, que resulta ser de 116 hs. La tasa media entre fallos es pequeña, y se debe a dos errores a la hora de realizar el estudio de confiabilidad. Si bien este se realizó en una etapa tardía del desarrollo como postula la hipótesis del modelo, uno de los errores corregidos en la última etapa del *testeo* redujo la frecuencia de aparición de errores subsiguientes significativamente. Además, la mayoría de los errores que surgieron eran de carácter visual y no impedían el correcto uso del programa. En la Tabla (13.2.2) se observan los valores de la función $R(t)$.

Tiempo	8 Horas	24 Horas	50 Horas	100 Horas	200 Horas
$R(t)$	0.933	0.813	0.65	0.422	0.178

Tabla 13.2.2: Valores de la función probabilidad de confiabilidad de software.

Otra cosa que cabe notar es que varios de los errores surgidos en la etapa de desarrollo eran de carácter acumulativo y tenían mayor probabilidad de aparición cuanto más tiempo se encuentre activo el software. Es por esto que se realiza cada ocho horas un reinicio del programa disminuyendo significativamente la tasa de aparición de errores.

14. Conclusiones

14.1 Excelencias, Objetivos alcanzados

Comparando los inicios del proyecto con respecto a las instancias finales, se presentan sensaciones ambiguas. No todos los objetivos alcanzados arrojaron los resultados deseados. No obstante, las metas fueron completadas y cubiertas con éxito.

Se logró obtener un alto nivel de confiabilidad en el producto superando con creces el periodo de tres meses por lo que un mismo dispositivo puede ser utilizado en varias investigaciones al cabo de los años. Además, el uso de los recursos naturales como abastecimiento energético permite que el producto sea utilizado hasta en las zonas más inhóspitas.

Gracias al uso de recursos como Ansible, que permite replicar de manera automática configuraciones en la unidad de procesamiento, este producto se vuelve fácil de replicar sin conocimientos técnicos avanzados tornándolo del tipo *plug and play*.

Se aplicaron distintas áreas de conocimiento, algunas de estas desarrolladas a lo largo de la carrera, mientras que otras no formaron parte de dicho proceso. Los desafíos que se presentaron muchas veces tuvieron una dificultad mayor a la esperada. A pesar de ello, estas adversidades no fueron impedimento para enfrentarlas y seguir adelante.

Pero la reflexión más importante se centra en la investigación acerca de la carga inalámbrica de la mochila. Como se mencionó, se esperaba que, de una forma u otra, sea posible lograr dicha transferencia de energía. Este desenlace no quita mérito al trabajo ni a su desarrollo. Lo que se destaca no es el resultado sino lo que proveyó la investigación. Se aprendió acerca del uso de antenas, transmisión de potencia por radiofrecuencia, *energy harvesting*, manejo de datos (*databases*) y a desarrollarse en el entorno de investigación, sobre todo a trabajar en el límite del desarrollo tecnológico del campo.

14.2 Fallos, Recomendaciones para futuros diseños

La principal adversidad del desarrollo del proyecto es la más notable por varias razones. Se puso mucha confianza en que iba a ser factible realizar la carga inalámbrica de la UBM. En retrospectiva, se debería de haber comenzado el proyecto ya con un plan B planteado. Además, en caso de futuros trabajos, se considera que se debe empezar mucho antes con el proceso de investigación.

En este proyecto se hizo evidente la falta de experiencia y conocimientos previos del manejo de antenas y radiopropagación del equipo.

Por último, se proponen recomendaciones para futuros diseños además de la ya mencionada. Estas propuestas surgen a partir de lo experimentado a lo largo del proceso. Una mejora inmediata que se recomienda es la de no utilizar módulos prefabricados. En su lugar emplear circuitos que integren toda electrónica en una sola placa. Otra recomendación es la de usar un *Flex-PCB* para la tecnología situada en la bóveda del nido, debido al poco espacio presente en esta zona.

15. Referencias

- [1] V. Ojeda, M. L. Chazarreta, C. M. Pozzi. *El Carpintero Gigante: Especie Clave Del Bosque Andino Patagónico*. Difundiendo Saberes, Vol. 8, 2011.
- [2] Large.stanford.edu. n.d. Wireless Power Efficiency. [online] Disponible en: <http://large.stanford.edu/courses/2016/ph240/surakitbovorn1/> [Accedido 26 Jun 2021].
- [3] Palmberg, E., Lundmark, S., Alatalo, M., Thiringer, T. and Karlsson, R., 2012. Wireless Charging - some key elements. [online] Publications.lib.chalmers.se. Disponible en: <https://publications.lib.chalmers.se/records/fulltext/175567/175567.pdf> [Accedido 26 Jun 2021].
- [4] Abatti, P., Pichorim, S. and de Miranda, C., 2015. Maximum Power Transfer versus Efficiency in Mid-Range Wireless Power Transfer Systems. *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, 14(1), pp.97-109.
- [5] Occupational Safety and Health Administration, Cincinnati Technical Center (May 20, 1990). "Electromagnetic Radiation and How It Affects Your Instruments. Near field vs. Far field" Disponible en: <https://www.osha.gov/radiofrequency-and-microwave-radiation/electromagnetic-field-memo> [Accedido 25 Jun 2021].
- [6] V. Nikitin, P., Rao, K. and Lazar, S., n.d. An Overview of Near Field UHF RFID.
- [7] Vistnes, A., 2018. Physics of Oscillations and Waves. Oslo, Norway: Springer, Chapter 9.
- [8] Humavox. n.d. Our Technology – Humavox. [online] Available at: <http://www.humavox.com/our-technology/> [Accedido 27 June 2021].
- [9] Life.itu.int. n.d. Terms and definitions. [online] Disponible en: <https://life.itu.int/radioclub/rr/art1.pdf> [Accedido 26 Jun 2021].
- [10] Sma.gov.jm. n.d. ITU Radio Regulations, CHAPTER II – Frequencies, ARTICLE 5 Frequency allocations, Section IV – Table of Frequency Allocations. [online] Disponible en: https://www.sma.gov.jm/sites/default/files/publication_files/ITU-R_Radio_Regulations_2012_%202015_%20Article_5_Table%20of%20Frequencies.pdf [Accedido 26 Jun 2021].
- [11] Tanner, J.A., Romero-Sierra, C. and Davie, S.J. (1967) Non-thermal effects of microwave radiation on birds, Nature News. Nature Publishing Group. [online] Disponible en: <https://www.nature.com/articles/2161139a0> [Accedido 21 Nov 2022].
- [12] Wasserman, F.E. et al. Thermoregulatory Behavior of Birds in Response to Continuous Wave 2.45-GHz Microwave Radiation, JSTOR. The University of Chicago Press. Disponible en: https://www.jstor.org/stable/30161222#metadata_info_tab_contents [Accedido 21 Nov 2022].
- [13] Tanner, J.A. Effect of microwave radiation on birds, Nature News. Nature Publishing Group. [online] Disponible en: <https://www.nature.com/articles/210636a0> [Accedido 21 Nov 2022].
- [14] L. Upton, 2014. Penguin Lifelines. [Blog] Raspberry Pi Blog, Disponible en: <https://www.raspberrypi.org/blog/penguin-lifelines/> [Accedido 24 May 2021].
- [15] Raspberry Projects [Blog] www.projects-raspberry.com [online]. Disponible en: <https://projects-raspberry.com/how-much-power-does-raspberry-pi-3b-use-power-measurements/> [Accedido 25 May 2021].
- [16] "Raspberry Pi Documentation - Camera", Raspberrypi.org, 2021. [online]. Disponible en: <https://www.raspberrypi.org/documentation/accessories/camera.html>. [Accedido 05 Oct 2021].
- [17] "El tiempo en San Carlos de Bariloche en la primavera, temperatura promedio (Argentina) - Weather Spark", Es.weatherspark.com, 2021. [online]. Disponible en: <https://es.weatherspark.com/s/25786/0/Tiempo-promedio-en-la-primavera-en-San-Carlos-de-Bariloche-Argentina>. [Accedido 05 Oct 2021].

16. Anexos Técnicos

16.1 Esquemáticos

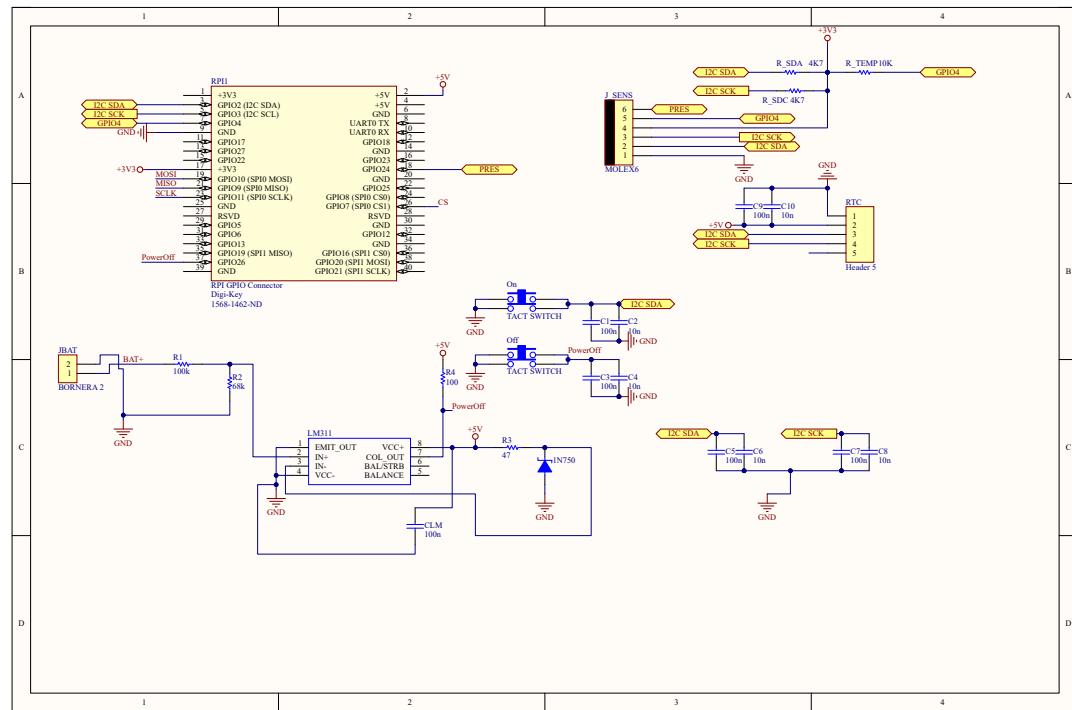


Figura 16.1.1: Esquemático de conexionado del *shield*.

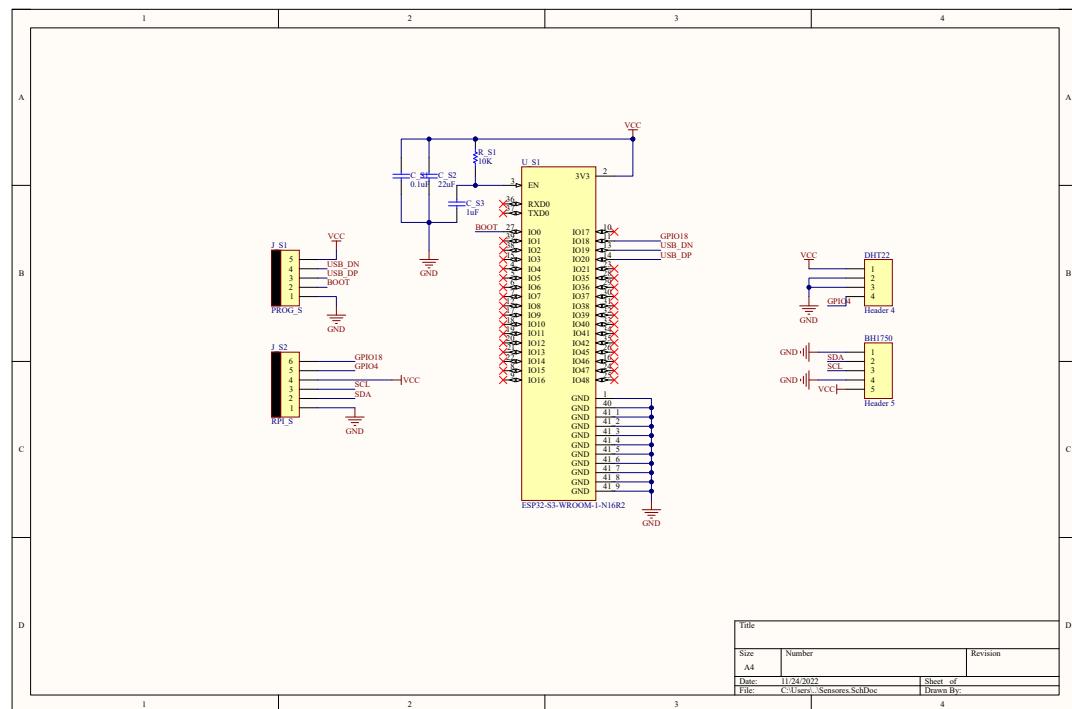


Figura 16.1.2: Esquemático de conexionado para la placa de sensores.

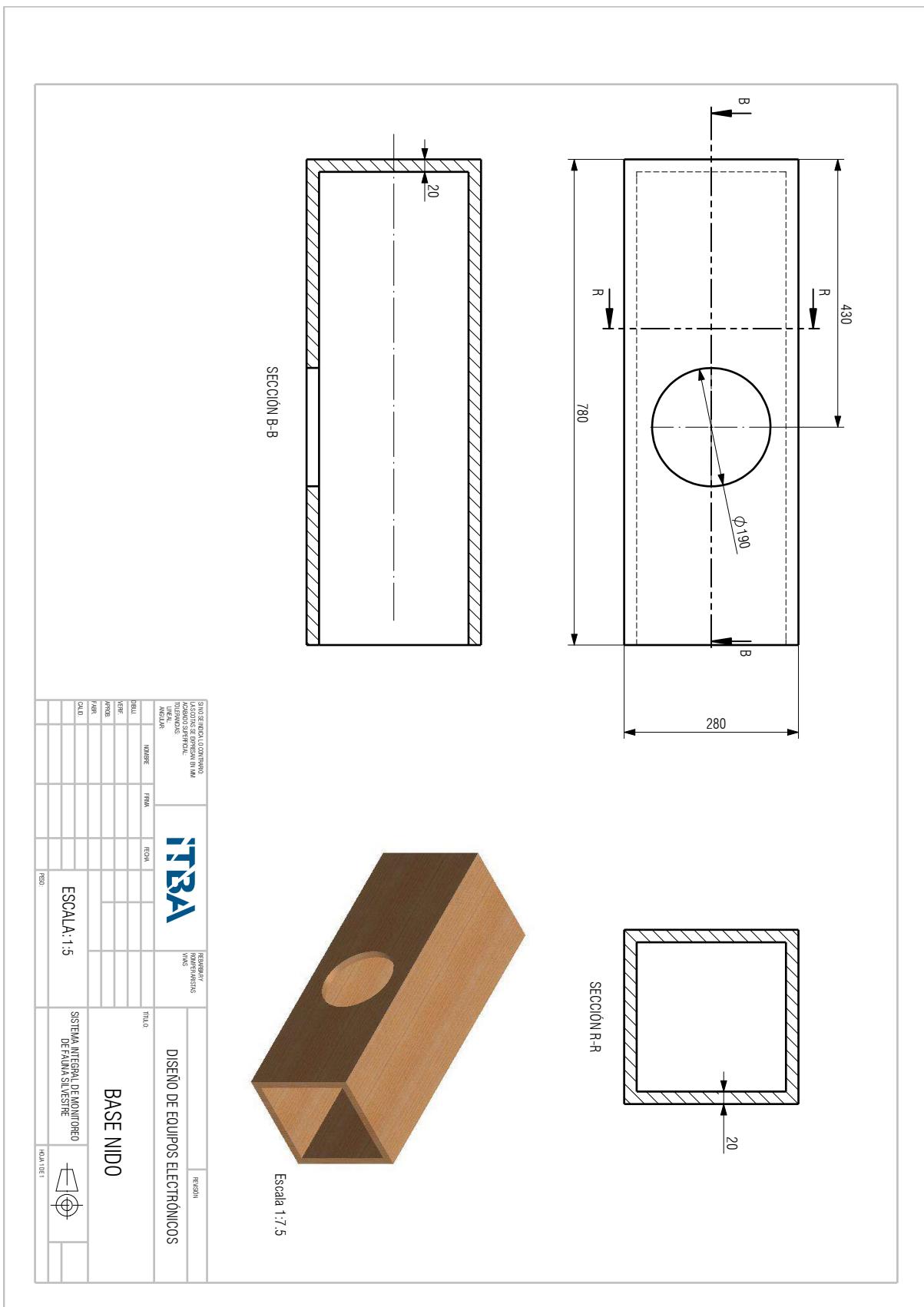


Figura 16.1.3: Plano de la base del prototipo de nido.

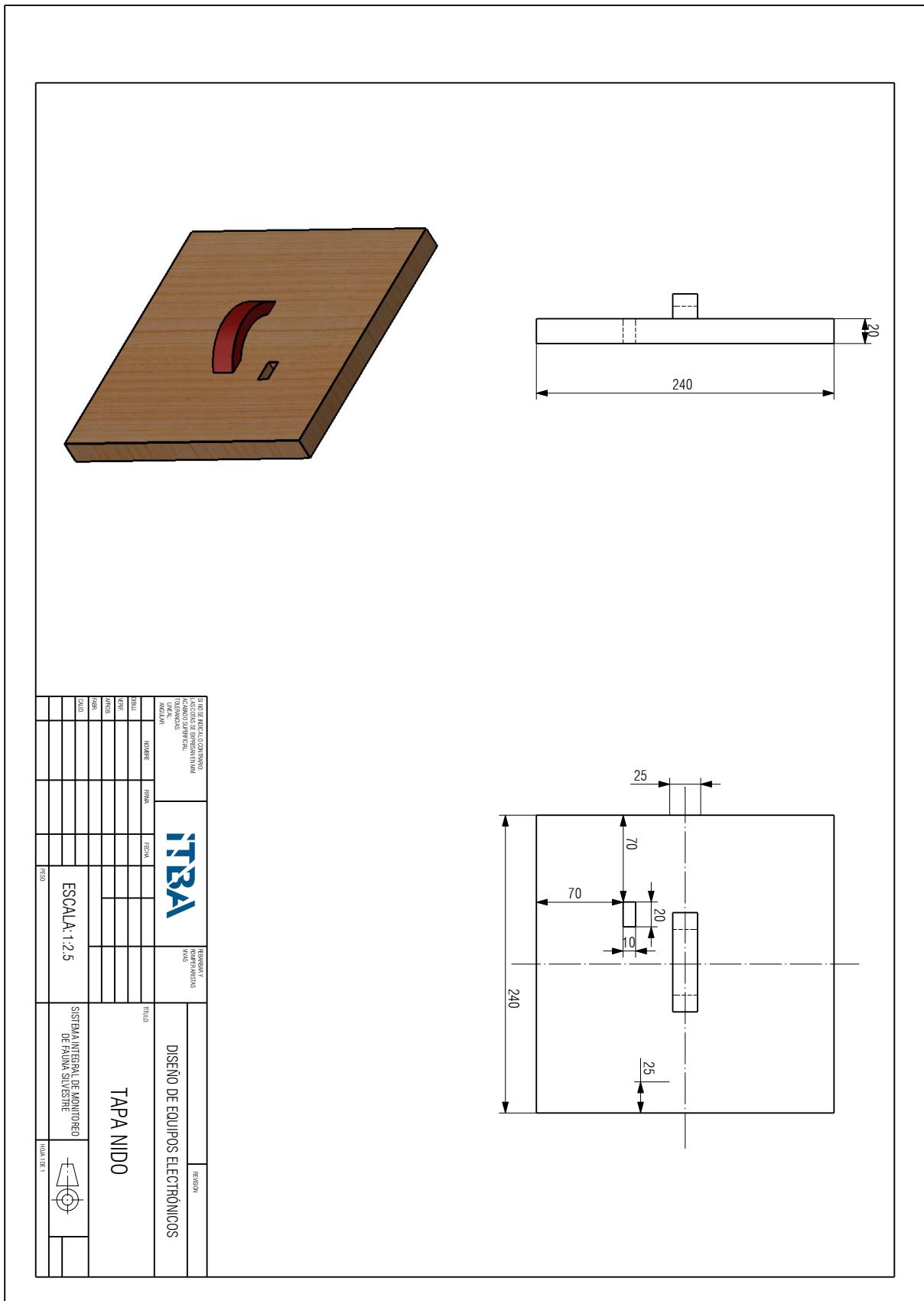


Figura 16.1.4: Plano de la tapa del prototipo de nido.

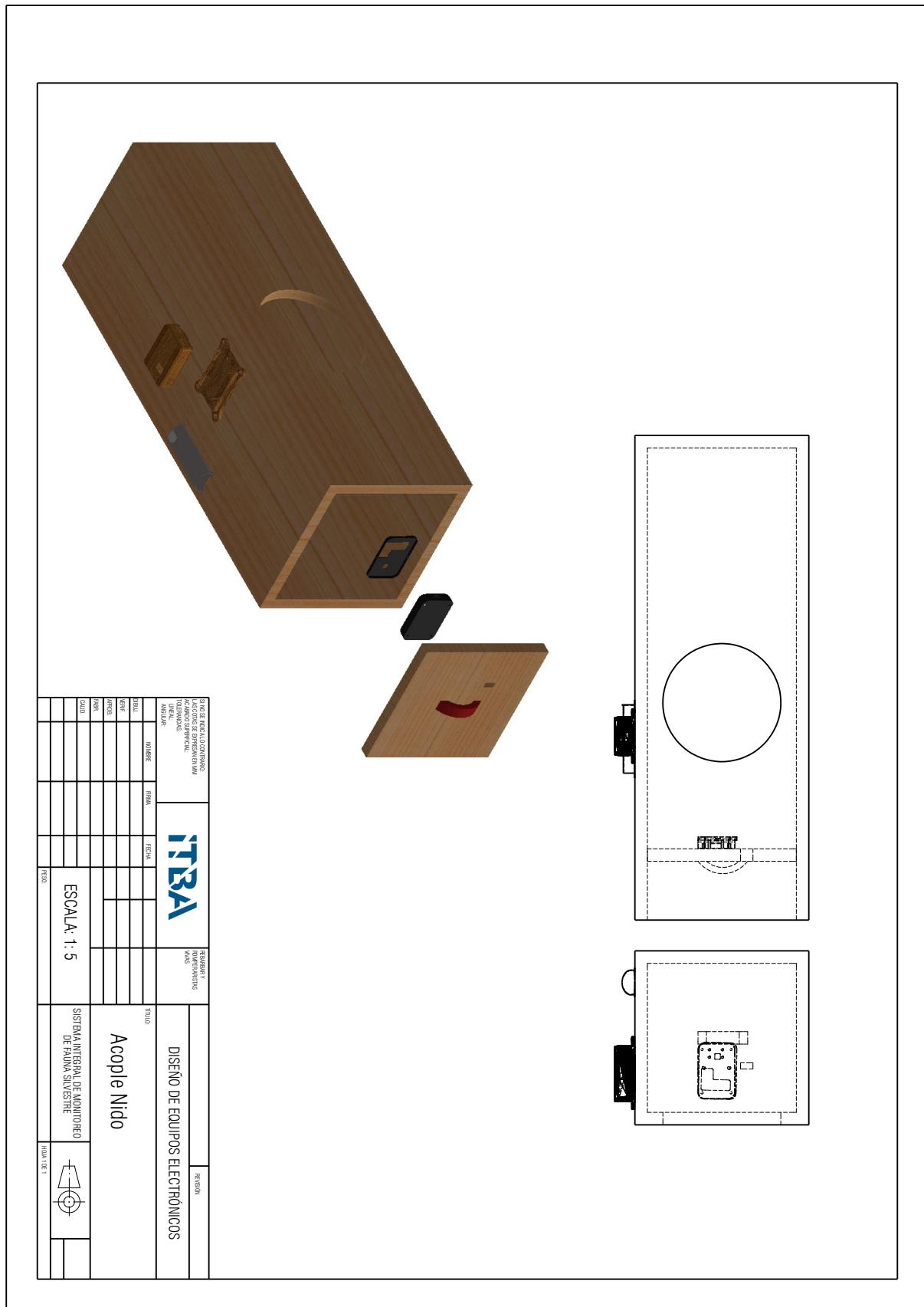


Figura 16.1.5: Plano explotado del prototipo.

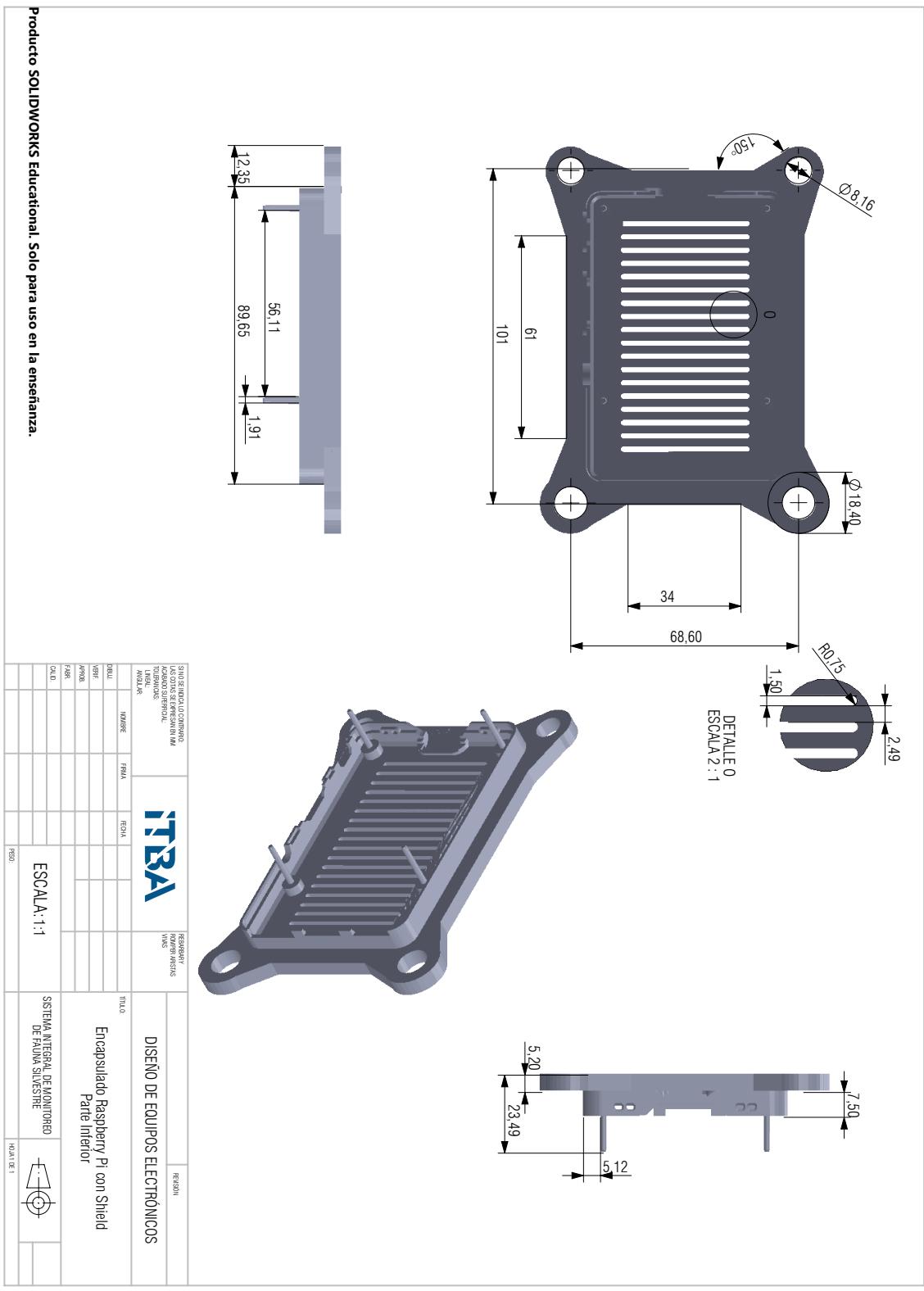


Figura 16.1.6: Plano del encapsulado para la Raspberry Pi con shield, encastre inferior.

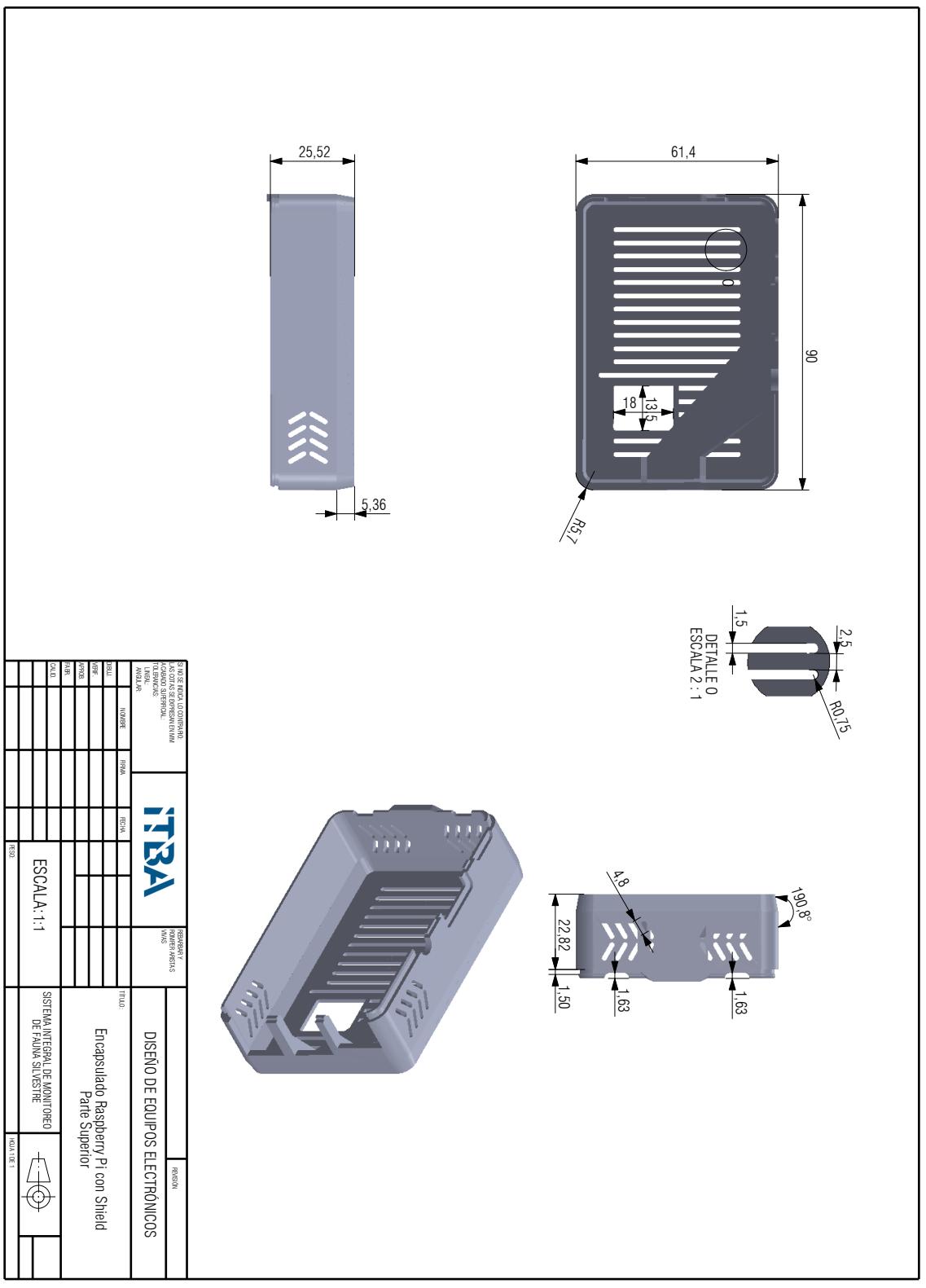


Figura 16.1.7: Plano del encapsulado para la Raspberry Pi con shield, encastre superior.

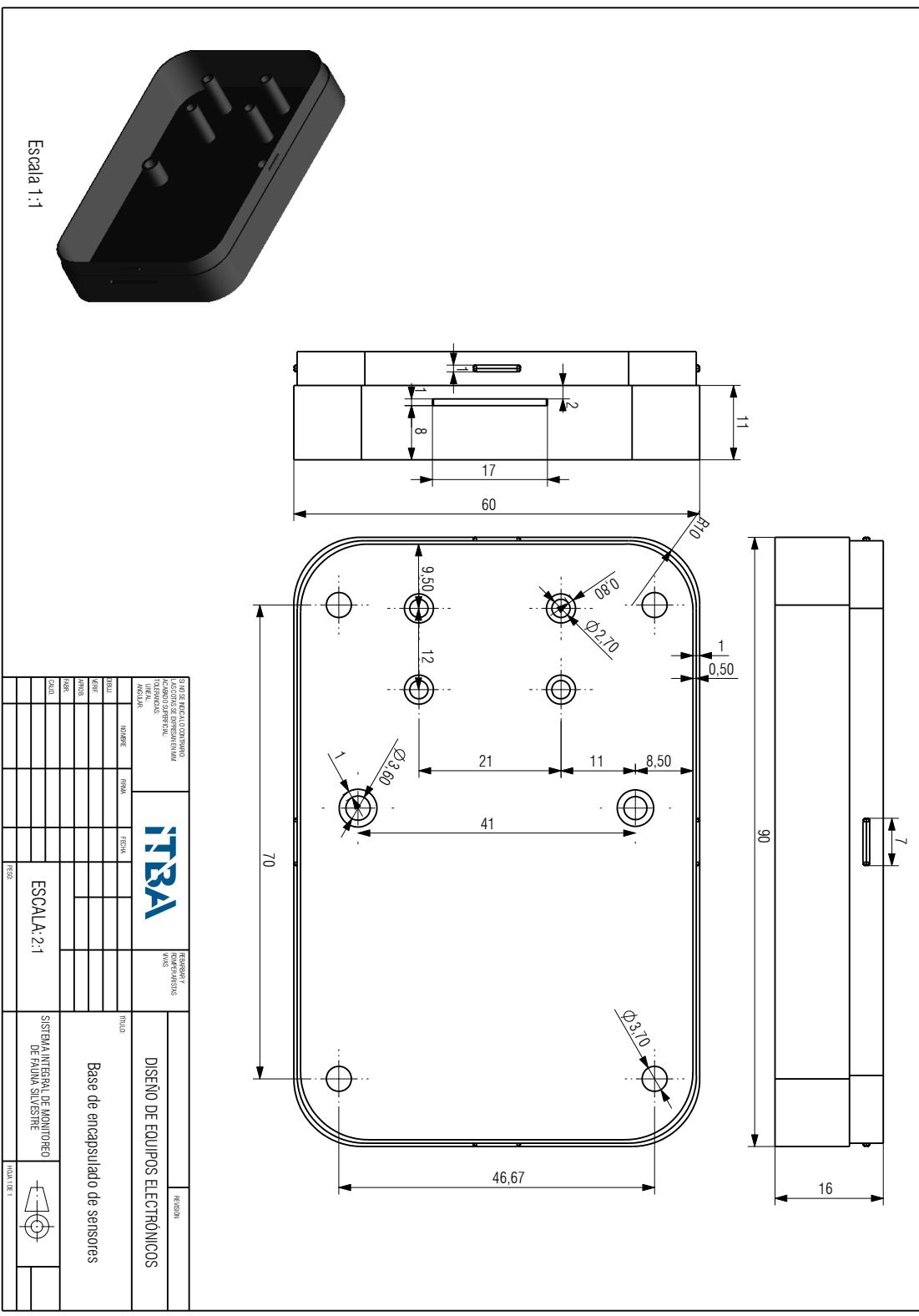


Figura 16.1.8: Plano del encapsulado para la cámara y sensores, encastre inferior.

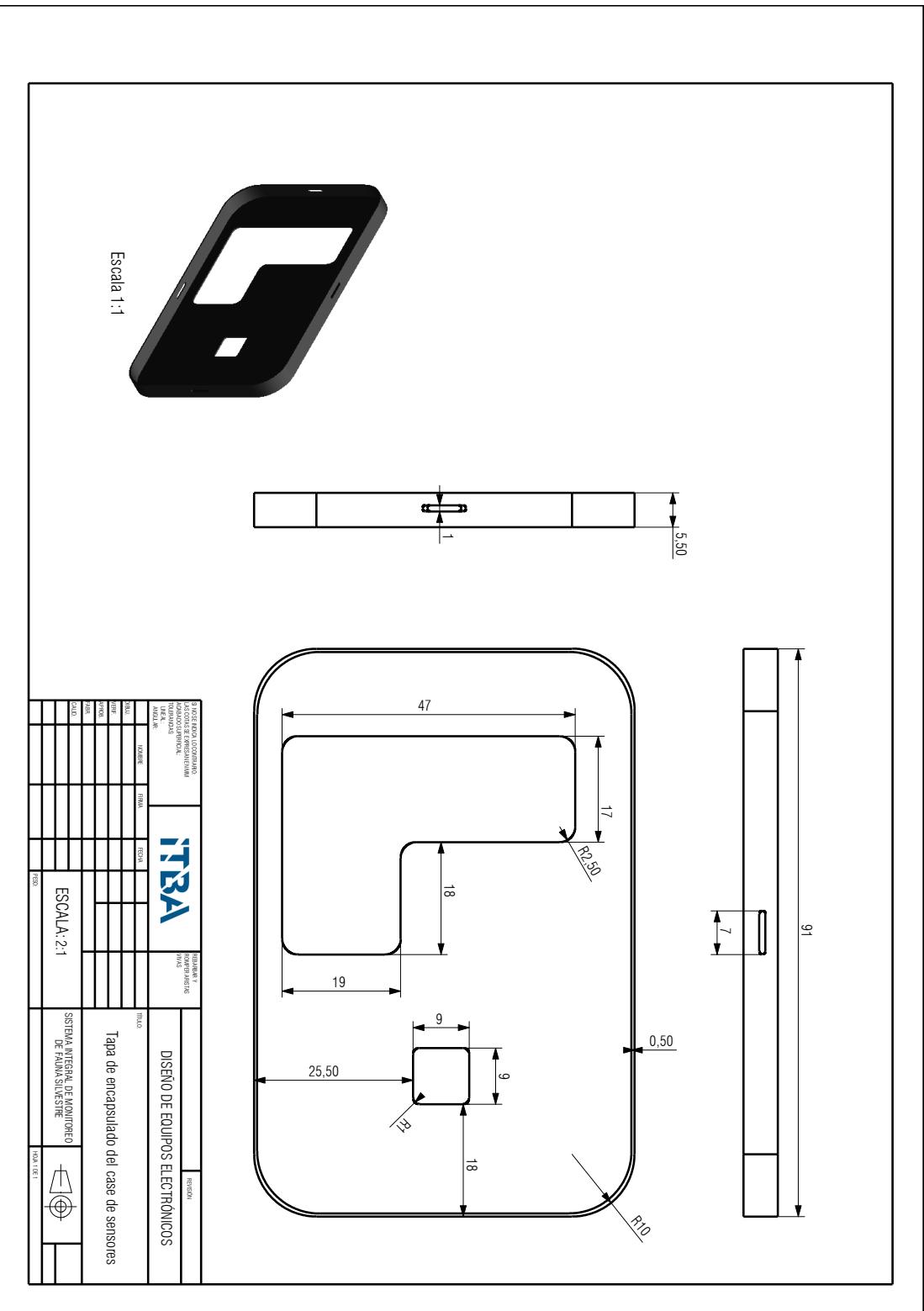


Figura 16.1.9: Plano del encapsulado para la cámara y sensores, encastre superior.

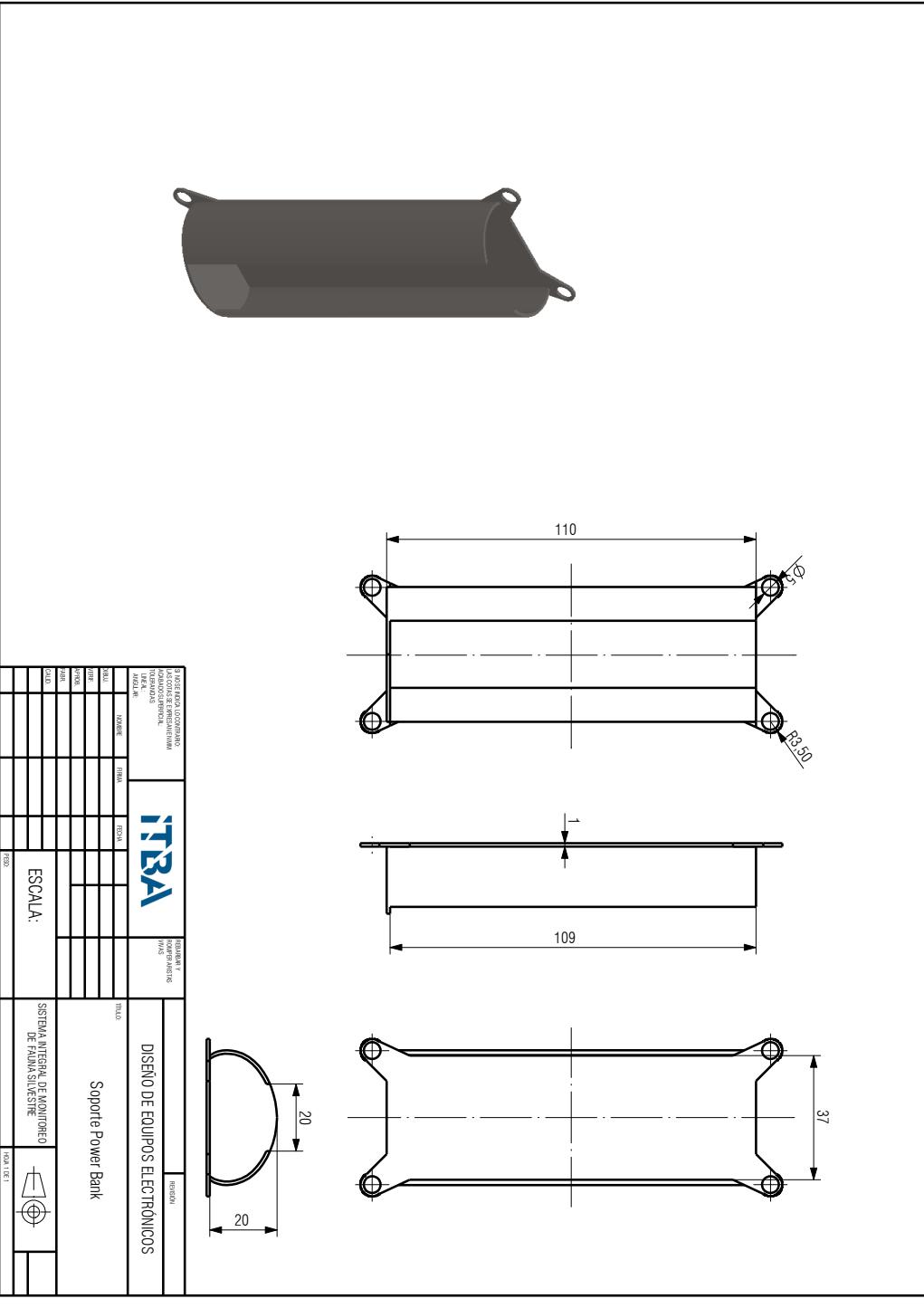


Figura 16.1.10: Plano del soporte del *Power Bank*.

16.2 Planos de PCB

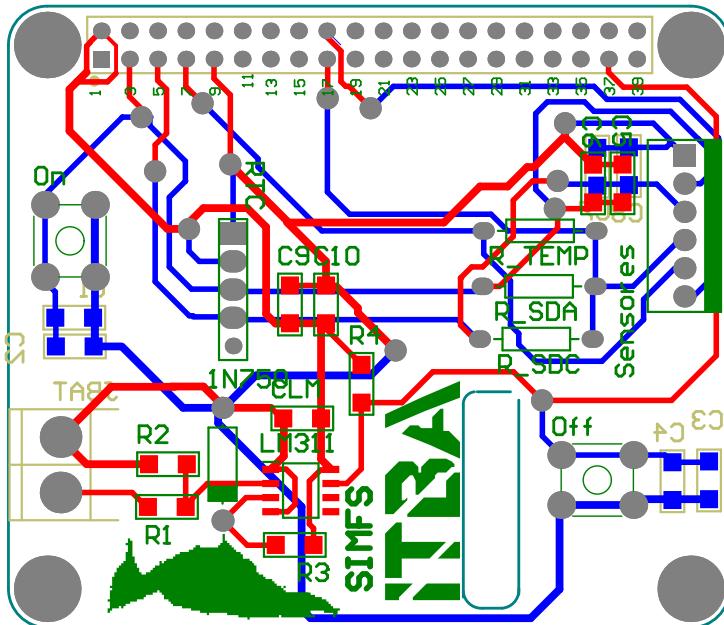


Figura 16.2.1: Esquemático del PCB del *shield*.

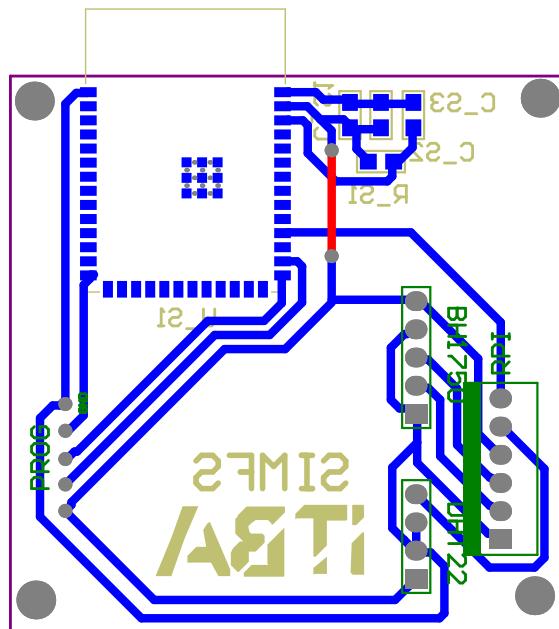


Figura 16.2.2: Esquemático del PCB de los sensores.

16.3 Lista de materiales

Para la lista de materiales se separó en seis grupos, los cuales se enumeran a continuación.

16.3.1 Potencia

- Panel Solar LN-50P.
- Regulador MPPT DFR0580.
- Batería de gel de ciclo profundo UL24-12 de Ultracell.

16.3.2 Sensores

- DHT-22.
- BH-1750.
- ESP32-WROOM-S3-1-N4
- Placa 5x5 epoxi.
- Placa 10x10 epoxi.
- 40 pines hembra.
- RPi-CMOD-V2.
- DS-3231.
- Cinco resistencias de $3.3\text{ k}\Omega$.
- Una resistencia de $10\text{ k}\Omega$
- Una resistencia de $100\text{ k}\Omega$
- Una resistencia de $68\text{ k}\Omega$
- Una resistencia de $47\text{ }\Omega$
- Cinco capacitores de 100 pF .
- Un capacitor de $0.1\text{ }\mu\text{F}$.
- Seis capacitores de 100 nF .
- Cinco capacitores de 10 nF .
- Un capacitor de $1\text{ }\mu\text{F}$.
- Un capacitor de $22\text{ }\mu\text{F}$.

16.3.3 Unidad de procesamiento

- RPi-3B
- SDSDQAF3-XI 32GBy

16.3.4 Prototipo para pruebas

- Caja de madera correspondiente a los planos (16.1.3), (16.1.4) y (16.1.5).

16.3.5 Misceláneo

- 5 m de cable rojo de 2 mm.
- 5 m de cable negro de 2 mm.
- 3 m de cable amarillo de 2 mm.
- 3 m de cable verde de 2 mm.
- Pines macho.
- Termocontraíble de 2 mm.
- 3 m de estaño.

16.4 Código de software

16.4.1 BH1750

```
1 import board
2 import adafruit_bh1750
3 import Clases.Database_Lum
4
5 from RTC import getTime
6
7 import time
8
9 i2c = board.I2C()
10 sensor = adafruit_bh1750.BH1750(i2c)
11
12 db_Lum = Clases.Database_Lum.Database_Lum(using_file_db=True,
13                                              foldername='db')
14 db_Lum.insert_lum_meas(sensor.lux, getTime())
```

16.4.2 DHT22

```
1 import Adafruit_DHT
2 import Clases.Database_TyH
3 from RTC import getTime
4
5 DHT_SENSOR = Adafruit_DHT.DHT22
6 DHT_PIN = 4
7 TEMPERATURE_OUTLIER_RANGE = 50
8 HUMIDITY_OUTLIER = 101
9
10 waiting_measurement = True
11 while waiting_measurement:
12     humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
13
14     if humidity is not None and temperature is not None:
15         waiting_measurement = False
16     else:
17         print("Failed to retrieve data from humidity sensor")
18
19 db = os.getenv('SIMFS_DATA_DB_PATH', 'db')
20
21 print("Temp={0:0.1f}*C  Humidity={1:0.1f}%" .format(temperature, humidity))
22
23 tempOutlier = temperature < TEMPERATURE_OUTLIER_RANGE
24 humOutlier = humidity < HUMIDITY_OUTLIER
25
26 if( tempOutlier and humOutlier ):
27     db_TyH = Clases.Database_TyH.Database_TyH(using_file_db=True,
28                                              foldername='db')
29     db_TyH.insert_tyh_meas(temperature, humidity, getTime())
```

16.4.3 getRTMeasurement

```
1 import Adafruit_DHT
2 import board
3 import adafruit_bh1750
```

```

4 import Clases.Database_Lum
5 import Clases.Database_TyH
6 import json
7 import os
8
9 DHT_SENSOR = Adafruit_DHT.DHT22(4)
10 DHT_PIN = 4
11 TEMPERATURE_OUTLIER_RANGE = 50
12 HUMIDITY_OUTLIER = 101
13
14 i2c = board.I2C()
15 sensor = adafruit_bh1750.BH1750(i2c)
16
17 waiting_measurement=True
18
19 while waiting_measurement:
20     humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
21
22     if humidity is not None and temperature is not None:
23         waiting_measurement = False
24     else:
25         print("Failed to retrieve data from humidity sensor")
26         waiting_measurement = False
27         humidity = 999
28         temperature = 999
29
30 lum = round(sensor.lux,1)
31 temp = round(temperature,1)
32 hum = round(humidity,1)
33 print("Luminosity = {0:0.1f} lux".format(lum))
34
35 tempOutlier = temperature > TEMPERATURE_OUTLIER_RANGE
36 humOutlier = humidity > HUMIDITY_OUTLIER
37
38 if( tempOutlier and humOutlier ):
39     temp = 999
40     hum = 999
41
42 print("Temp={0:0.1f}*C  Humidity={1:0.1f}%".format(temperature, humidity))
43 print("lum={0:0.1f} lux".format(lum))
44
45 data = {
46     "lum": lum,
47     "temp": temp,
48     "hum": hum
49 }
50
51 json_data = json.dumps(data)
52 with open("../LastData/last_data.json", "w") as outfile:
53     outfile.write(json_data)

```

16.4.4 getTime

```

1 from RTC import getTime
2
3 print(getTime().strip())

```

16.4.5 RTC

```
1 import board
2 import adafruit_ds1307
3 import time
4
5 def getTime():
6     i2c = board.I2C()
7     rtc = adafruit_ds1307.DS1307(i2c)
8     t = rtc.datetime
9
10    day = t.tm_mday
11    if day < 10:
12        day = '0' + str(day)
13
14    mon = t.tm_mon
15    if mon < 10:
16        mon = '0' + str(mon)
17
18    year = t.tm_year
19
20    hour = t.tm_hour
21    if hour < 10:
22        hour = '0' + str(hour)
23
24    min = t.tm_min
25    if min < 10:
26        min = '0' + str(min)
27
28    sec = t.tm_sec
29    if sec < 10:
30        sec = '0' + str(sec)
31
32    return f"{year}-{mon}-{day}_{hour}:{min}:{sec}"
33
34 def setTime(year,month,day,hour,minute):
35     i2c = board.I2C()
36     rtc = adafruit_ds1307.DS1307(i2c)
37     rtc.datetime = time.struct_time((year,month,day,hour,minute,0,0,9,-1))
```

16.4.6 DatabaseLum

```
1 import sqlite3
2 import os
3
4 class Database_Lum:
5     """Luminosity database class for sqlite3"""
6
7     def __init__(self, db_dir: str, using_file_db=True) -> None:
8         if using_file_db is True:
9             self.foldername = db_dir
10            self.usingfolder = True
11            if(os.path.isdir(db_dir) == 0):
12                os.makedirs(db_dir)
13            self.conn = sqlite3.connect(db_dir + '\Lum.db')
14        else:
15            self.usingfolder = False
```

```

16     self.conn = sqlite3.connect(':memory:')
17
18     self.c = self.conn.cursor()
19     self.c.execute(
20         ''' SELECT count(name) FROM sqlite_master WHERE type='table'
21             AND name='measurements' '''
22     )
23     if self.c.fetchone()[0] == 0:
24         {
25             self.c.execute("""CREATE TABLE measurements (
26                 date_meas TEXT, lum_meas REAL)""")
27         }
28
29     def insert_lum_meas(self, lum_, date) -> None:
30         if(self.usingfolder):
31             with self.conn:
32                 self.c.execute("INSERT INTO measurements VALUES
33                     (:date_, :luminosity_meas)",
34                     {'luminosity_meas': lum_, 'date_': date})
35
36     def get_meas_in_range(self, start_time: str, end_time: str):
37         if (self.usingfolder):
38             self.c.execute("SELECT * FROM measurements WHERE date_meas
39                 BETWEEN (:start_) AND (:end_)",
40                 {'start_': start_time, 'end_': end_time})
41             return self.c.fetchall()
42
43     def remove_meas_in_range(self, start_time: str, end_time: str):
44         with self.conn:
45             self.c.execute("DELETE FROM measurements WHERE date_meas
46                 BETWEEN (:start_) AND (:end_)", {
47                     'start_': start_time, 'end_': end_time})
48
49     def get_all_meas(self):
50         if (self.usingfolder):
51             self.c.execute("SELECT * FROM measurements "
52                     )
53             return self.c.fetchall()
54
55     def __del__(self):
56         self.conn.close()

```

16.4.7 DatabaseTyH

```

1 import sqlite3
2 import os
3
4 class Database_TyH:
5     """Temperature and Humidity database class for sqlite3"""
6
7     def __init__(self, db_dir: str, using_file_db=True) -> None:
8         if using_file_db is True:
9             self.foldername = db_dir
10            self.usingfolder = True
11            if(os.path.isdir(db_dir) == 0):
12                os.makedirs(db_dir)
13            self.conn_temp_n_hum = sqlite3.connect(
14                db_dir + 'Temp_and_hum.db')
15        else:

```

```

16     self.usingfolder = False
17     self.conn_temp_n_hum = sqlite3.connect(':memory:')
18
19     self.ctyh = self.conn_temp_n_hum.cursor()
20     self.ctyh.execute(
21         ''' SELECT count(name) FROM sqlite_master WHERE type='table'
22             AND name='measurements' ''')
23     if self.ctyh.fetchone()[0] == 0:
24         {
25             self.ctyh.execute("""CREATE TABLE measurements (
26                 date_meas TEXT, temperature_meas REAL,
27                 humidity_meas REAL )""")
28         }
29
30     def insert_tyh_meas(self, temp_, hum_, date) -> None:
31         if(self.usingfolder):
32
33             with self.conn_temp_n_hum:
34                 self.ctyh.execute("INSERT INTO measurements VALUES
35                     (:date_, :temperature_meas, :humidity_meas)", {
36                         'temperature_meas': temp_, 'humidity_meas': hum_,
37                         'date_': date})
38
39     def get_meas_in_range(self, start_time, end_time):
40         if (self.usingfolder):
41             self.ctyh.execute("SELECT * FROM measurements WHERE
42                 date_meas BETWEEN (:start_) AND (:end_)", {
43                     'start_': start_time, 'end_': end_time})
44             return self.ctyh.fetchall()
45
46     def get_all_meas(self):
47         if (self.usingfolder):
48             self.ctyh.execute("SELECT * FROM measurements")
49             return self.ctyh.fetchall()
50
51     def remove_meas_in_range(self, start_time: str, end_time: str):
52         with self.conn_temp_n_hum:
53             self.ctyh.execute("DELETE FROM measurements WHERE
54                 date_meas BETWEEN (:start_) AND (:end_)", {
55                     'start_': start_time, 'end_': end_time})
56
57     def __del__(self):
58         self.conn_temp_n_hum.close()

```

16.4.8 FileMakerDB

```
1 import Database_Lum
2 import Database_TyH
3 import sys
4 import csv
5
6 def write_TyH(start_time: str, end_time: str):
7     db_TyH = Database_TyH.Database_TyH(using_file_db=True, db_dir='../db')
8
9     data = db_TyH.get_meas_in_range(start_time, end_time)
10    csv_name = start_time[0:10] + '_' + end_time[0:10]
11    db_export_path = "/home/pi/simfs_export_data/CSV/TemperaturaYHumedad_"
12    filename = db_export_path + csv_name + '.csv'
13    with open(filename, 'w') as f:
14        writer = csv.writer(f, lineterminator='\n')
15        writer.writerow(['Time',
16                         'Temperature', 'Humidity'])
17        for tup in data:
18            writer.writerow(tup)
19
20 def write_Lum(start_time: str, end_time: str):
21     db_Lum = Database_Lum.Database_Lum(using_file_db=True, db_dir='../db')
22
23     data = db_Lum.get_meas_in_range(start_time, end_time)
24     csv_name = '../CSV/Luminosidad_' + start_time[0:10]
25             + '_' + end_time[0:10]
26     db_export_path = "/home/pi/simfs_export_data/CSV/TemperaturaYHumedad_"
27     filename = db_export_path + csv_name + '.csv'
28     with open(filename, 'w') as f:
29         writer = csv.writer(f, lineterminator='\n')
30         writer.writerow(['Time',
31                         'Lux'])
32         for tup in data:
33             writer.writerow(tup)
34
35 if __name__ == "__main__":
36     argv = sys.argv[1:]
37     if(len(argv) != 3):
38         print('Wrong amount of parameters!')
39     else:
40         if(argv[0].lower() == '-l'):
41             if((len(argv[1]) != 16) or (len(argv[2]) != 16)):
42                 print('Bad date formating!')
43             else:
44                 write_Lum(argv[1], argv[2])
45                 print('Writing CSV')
46         elif(argv[0].lower() == '-tyh'):
47             if((len(argv[1]) != 16) or (len(argv[2]) != 16)):
48                 print('Bad date formating!')
49             else:
50                 print('Writing CSV')
51                 write_TyH(argv[1], argv[2])
52         else:
53             print('Wrong key word!')
```

16.4.9 makeCsvCli

```
1 import argparse
2 from enum import Enum
3 from File_maker_DB import write_Lum, write_TyH
4
5 class MeasDevices(Enum):
6     TEMP_AND_HUM_DEV = 1
7     LUX_DEV = 2
8
9 if __name__ == '__main__':
10    parser = argparse.ArgumentParser(description='Generate csv data')
11    parser.add_argument('--start_time', dest='start_time',
12                        help='csv start_time', type=str)
13    parser.add_argument('--end_time', dest='end_time',
14                        help='csv end_time', type=str)
15
16    parser.add_argument('--meas_type', type=int,
17                        help="""Select information source
18                                1: Temperature and Humidity
19                                2: Luminosity""")
20
21 args = parser.parse_args()
22 start_time = args.start_time
23 end_time = args.end_time
24 print("Generating CSV")
25 print(f"from: {start_time}")
26 print(f"to: {end_time}")
27 print(f"for device {MeasDevices.TEMP_AND_HUM_DEV}")
28 try:
29     if args.meas_type == MeasDevices.TEMP_AND_HUM_DEV.value:
30         print("Doing temp")
31         write_TyH(start_time, end_time)
32     elif args.meas_type == MeasDevices.LUX_DEV.value:
33         print("Doing light")
34         print("start_time", start_time)
35         print("end_time", end_time)
36         write_Lum(start_time, end_time)
37 except Exception as e:
38     print("Generating CSV failed")
39     print(e)
```

16.4.10 presenceDetector

```
1 from bluepy.btle import Scanner
2 import numpy as np
3 import os
4 RETRIES = 3
5
6 def resetBL():
7     os.system("sudo hciconfig hci0 down && sudo hciconfig hci0 up")
8
9 def IsInNest(mac, timeout, minDistance = -50):
10    avg = 0
11    scan_time = 0.1
12    iterations = int(timeout/0.1)
13    for i in range(iterations):
14        try:
15            ble_list = Scanner().scan(0.1)
16            for dev in ble_list:
17                if mac.lower() == dev.addr:
18                    avg = avg + 0.2*(dev.rssi - avg)
19                    .format(dev.rssi, dev.addr, dev['scanData'][8]))
20        except:
21            continue
22    if 0.0 > avg >= minDistance:
23        return True, avg
24    else:
25        return False, avg
26
27 if __name__ == "__main__":
28    devices = []
29    mac = "8C:AA:B5:87:6F:12"
30    isBirdInNest = False
31    secondsToScan = 5
32    for attempt in range(RETRIES):
33        try:
34            isBirdInNest, value = IsInNest(mac, secondsToScan)
35            if value == 0:
36                resetBL()
37                continue
38            else:
39                break
40        except BTLEManagementError:
41            resetBL()
42        except KeyboardInterrupt:
43            Scanner().stop()
44            raise SystemExit
45        except:
46            raise Exception("Error occurred")
47
48    print(isBirdInNest)
```

16.4.11 printMeasures

```
1 import Database_TyH
2 import Database_Lum
3
4 db_TyH = Database_TyH.Database_TyH(using_file = True, foldername = '../db')
5 data = db_TyH.get_all_meas()
6 print("=====Temperature and Humidity measures=====")
7 print(data)
8 db_Lum = Database_Lum.Database_Lum(using_file = True, foldername = '../db')
9 data = db_Lum.get_all_meas()
10 print("=====Luminosity measures=====")
11 print(data)
12 db_Pres = Database_Pres.Database_Pres(using_file = True,
13                                     foldername = '../db')
14 data = db_Pres.get_all_meas()
15 print("=====Presence measures=====")
16 print(data)
```

16.4.12 Conexión BPS para banco de pruebas Alcance

```
1 from bluepy.btle import Scanner
2 import numpy as np
3 import time
4 # 8C:AA:B5:87:6F:12
5
6 class Beacon(object):
7
8     def __init__(self, data, address):
9         self._uuid = data[0]
10        self._major = data[1]
11        self._minor = data[2]
12        self._power = data[3]
13        self._rssi = data[4]
14        self._address = address
15
16    def __str__(self):
17        ret = "Beacon: address:{ADDR} uuid:{UUID} major:{MAJOR}" \
18              " minor:{MINOR} txpower:{POWER} rssi:{RSSI}" \
19              .format(ADDR=self._address, UUID=self._uuid, MAJOR=self._major,
20                      MINOR=self._minor, POWER=self._power, RSSI=self._rssi)
21        return ret
22
23
24 if __name__ == "__main__":
25     devices = []
26     COUNT = 20
27     avg = 0
28
29     while True:
30         try:
31             ble_list = Scanner().scan(5)
32             beacon_mentioned = False
33             for dev in ble_list:
34                 try:
35                     if "8C:AA:B5:87:6F:12".lower() == dev.addr:
36                         avg = avg + 0.3*(dev.rssi - avg)
37                         print("-----")
38                         print(f"Beacon found:\n\tName: {dev.getScanData()[1][2]}\n\tPower: {np.round(avg, 3)} dBm\n\tAddress: {dev.addr}")
39                         beacon_mentioned = True
40                 except:
41                     continue
42                 if (beacon_mentioned == False):
43                     print('Beacon signal lost')
44             except KeyboardInterrupt:
45                 print("Raising SystemExit")
46                 Scanner().stop()
47                 raise SystemExit
48             except:
49                 raise Exception("Error occurred")
```

16.4.13 Conexión BPS para banco de pruebas Recepción-Raspberry Pi

```
1 from bluepy import btle
2 import hashlib
3 MAC = "8C:AA:B5:87:6F:12"
4 SERVICE_UUID = "1825" #Object Transfer service 0x1825
5 CHARACTERISTIC_UUID = "2A68" #Navigation 0x2A68
6 while True:
7     try:
8         print("Connecting to: "+MAC)
9         dev = btle.Peripheral(MAC)
10        print("Connected")
11        for svc in dev.services:
12            print(str(svc))
13        print("\n ----get service-characteristic----")
14        service = dev.getServiceByUUID(btle.UUID(SERVICE_UUID))
15        print(service)
16        characteristic = service.getCharacteristics()
17        for char in characteristic:
18            if(char.uuid == CHARACTERISTIC_UUID):
19                print(char.read())
20                print("Hash of received file: ",hashlib.sha256(char.read()))
21    except KeyboardInterrupt:
22        print("Raising SystemExit")
23        raise SystemExit
24    except:
25        raise Exception("Error occurred")
26 print("Transmission Ended.")
```

16.4.14 Conexión BPS para banco de pruebas Transmisión-BPS

```
1 #include <ArduinoBLE.h>
2
3 BLEService OTService("1825"); // Object Transfer service 0x1825
4 char* BirdData ="Date,longitude,latitude\n 2022-09-01 07:00,-70.99,-44.98..";
5 char* hash="FB9774E1965600BC4E706D87816688A1B0C2B0EAE591E9C72504EF4D6C8A7B29";
6 // BLE Navigation 0x2A68 Characteristic
7 // custom 128-bit UUID, read and writable by central
8 BLEStringCharacteristic navigationCharacteristic("2A68",BLERead|BLEWrite,15);
9
10 void setup() {
11   Serial.begin(9600);
12   while (!Serial);
13   // begin initialization
14   if (!BLE.begin()) {
15     Serial.println("starting Bluetooth® Low Energy failed!");
16     while (1);
17   }
18   // set advertised local name and service UUID:
19   BLE.setLocalName("BPS-BLE");
20   BLE.setAdvertisedService(OTService);
21   // add the characteristic to the service
22   OTService.addCharacteristic(navigationCharacteristic);
23   // add service
24   BLE.addService(OTService);
25   // set the initial value for the characteristic:
26   navigationCharacteristic.writeValue("");
27   // start advertising
28   BLE.advertise();
29   Serial.println("BLE Object transfer Peripheral");
30 }
31 void loop() {
32   // listen for Bluetooth® Low Energy peripherals to connect:
33   BLEDevice central = BLE.central();
34   // if a central is connected to peripheral:
35   if (central) {
36     Serial.print("Connected to central: ");
37     Serial.println("Device connected");
38     Serial.println(central.address());
39     // while the central is still connected to peripheral:
40     while (central.connected()) { // Post bird data
41       Serial.println("Publishing Service-Characteristic");
42       Serial.print("Hash of file: ");
43       Serial.println(hash);
44       navigationCharacteristic.writeValue(BirdData);
45     } // when the central disconnects, print it out:
46     Serial.print(F("Disconnected from central: "));
47     Serial.println(central.address());
48   }
49 }
```