

Universidad Tecnológica Nacional

Proyecto Final

Diseño e implementación de cámara trampa
con sistema de posicionamiento y
catalogación de imágenes con visión artificial

Autor:

- *Andrian Franco*

Director:

- *Esp. Ing. Cabral Alejandro*

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero Electrónico*

en la

Facultad Regional Paraná

abril de 2023

Declaración de autoría:

Yo declaro que el Proyecto Final "Diseño e implementación de cámara trampa con sistema de posicionamiento y de catalogación de imágenes con visión artificial" y el trabajo realizado son propio. Declaro/declaramos:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pregrado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

-
-
-

Fecha:

Agradecimientos:

A mi docente director, Ing. Alejandro Ariel Cabral, por sus consejos y buena predisposición ante todas mis consultas.

Al Dr. Ing. Julian Sabattini, por brindarme la posibilidad de realizar este proyecto, el cual financio en su totalidad.

A la Universidad Tecnológica Nacional, Regional Paraná, por brindarme los conocimientos y el espacio para formarme como profesional y a su cuerpo docente, por su calidez humana y su preparación.

A mi Familia, que me brindo el espacio y la comodidad para dedicarme a tiempo completo a mis estudios.

A mi Novia Camila y su familia, por brindarme la calidez y el soporte cuando lo requerí.

A mis compañeros de estudio, en especial a mi amigo Lautaro, por compartir esta etapa juntos y por su ayuda desinteresada en los momentos en los que más lo necesite.

Andrian Franco

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

**Diseño e implementación de cámara trampa
con sistema de posicionamiento y
catalogación de imágenes con visión artificial**

Andrian Franco

Abstract:

This project consists of developing a trap camera with an extended detection and vision field, using an array of 4 PIR motion sensors and a positioning system integrated with a stepper motor.

The equipment can detect motion, taking images and videos in both daytime and nighttime scenes, and can be remotely accessed for configuration and control through the developed graphical interface.

In addition, a Python script was coded that uses pre-trained artificial vision models VGG-16 and VGG-19, for the cataloging of animal images according to the following species: capybara, deer, wild cat, sheep, puma, and cow.

The system integrates a Raspberry Pi 3A+ computer for executing the graphical interface, detecting motion, positioning the equipment, and controlling the camera, programmed in Python using the PyQt5 toolkits for interface design, and an Arduino Nano for sensing and controlling power, lighting, and temperature.

The camera module used is the Raspberry Pi Noir Camera V2.1, and the nighttime lighting consists of a series array of eight 3W infrared LEDs at 850nm.

Finally, a trap camera capable of detecting motion in 4 detection regions, arranged at 360° was obtained. With ranges of 15 meters for motion detection and 10 meters for nighttime lighting.

On the other hand, the generated artificial vision models achieved a precision of 94.25% (VGG-19) and 90.80% (VGG-16) in daytime scenes.

Keywords:

Arduino nano, PIR sensors, Raspberry Pi, Solar battery, Stepper Motor.

Resumen:

Este proyecto consiste en la elaboración de una cámara trampa con un campo de detección y de visión ampliados, haciendo uso de un arreglo de 4 sensores de movimiento PIR y un sistema de posicionamiento integrado por un motor paso a paso, respectivamente.

El equipo es capaz de detectar movimiento, tomar imágenes y videos en escenas diurnas y nocturnas y puede ser accedido de manera remota para su configuración y control a través de la interfaz gráfica desarrollada.

También se codificó un script de Python que hace uso de los modelos preentrenados de visión artificial VGG-16 y VGG-19, para la catalogación de imágenes de animales según las siguientes especies: carpincho, ciervo, gato montés, oveja, puma y vaca.

El sistema integra una computadora Raspberry Pi 3A+ para la ejecución de la interfaz gráfica, detección del movimiento, posicionado del equipo y control de la cámara, programada en Python utilizando el toolkit PyQT5 para el diseño de la Interfaz y un Arduino Nano para el sensado y control de la alimentación, la iluminación y la temperatura.

El módulo de cámara utilizado es el Noir Camera V2.1 de Raspberry Pi y la iluminación nocturna consiste en un arreglo serie de ocho leds infrarrojos de 3W a 850nm.

Se obtuvo finalmente una cámara trampa capaz de detectar movimiento en 4 regiones de detección, dispuestas en 360°. Con alcances de 15 metros para la detección de movimiento y 10 metros para la iluminación nocturna.

Por otra parte, los modelos de visión artificial generados obtuvieron una precisión de 94,25% (VGG-19) y 90,80% (VGG-16) en escenas diurnas.

Palabras Clave:

Arduino nano, Batería solar, Motor de Pasos, Raspberry Pi, Sensores PIR.

Índice:

Capítulo 1: Introducción.....	1
Capítulo 2: Antecedentes.....	3
Capítulo 3: Marco teórico.....	5
3.1 Sistema de activación automático.....	5
3.2 Clasificación de imágenes utilizando visión artificial	7
Capítulo 4: Requerimientos	11
4.1 Requerimientos funcionales.....	12
4.1.1 RF01: Alimentar sistema con batería de 12V.....	13
4.1.2 RF02: Supervisar tensiones del sistema.....	13
4.1.3 RF03: Medir intensidad de luz	13
4.1.4 RF04: Iluminar ambiente en condiciones de iluminación escasa ..	14
4.1.5 RF05: Conectar por bluetooth para Seteo inicial.....	14
4.1.6 RF06: Conectar a internet para acceso remoto	14
4.1.7 RF07: Configurar mediante interfaz gráfica	14
4.1.8 RF08: Detectar movimiento	15
4.1.9 RF09: Posicionar Cámara en región de detección.....	15
4.1.10 RF10: Capturar imágenes y grabar videos.....	15
4.1.11 RF11: Almacenar archivos de imágenes y videos en dispositivo de almacenamiento externo	15
4.1.12 RF12: Almacenar archivos de imágenes y videos en carpetas específicas.....	15
4.1.13 RF13: Nombrar archivos de imágenes y videos con nombres específicos	16
4.1.14 RF14: Reconocer animales mediante script de visión artificial....	16
4.1.15 RF15: Estar contenido en un Gabinete Estanco	17
4.1.16 RF16: Refrigerar sistema.....	17
4.2 Requerimientos no funcionales.....	17
4.2.1 RF01: Capacidad de almacenamiento de dispositivo externo.....	18
4.2.2 RF02: Capturar imagen con tiempo de obturación de 1 ms	18
Capítulo 5: Desarrollo	19
5.1 Arquitectura	19
5.1.1 Actores externos.....	20
5.1.1.1 Actor externo: Usuario	20
5.1.1.2 Actor externo: Animal.....	20
5.1.2 Interfaces.....	20
5.1.2.1 Interfaces entre etapas	20
5.1.2.2 Interfaces con actores externos	21
5.2 Diseño	21
5.2.1 Diseño de Hardware	21
5.2.1.1 Selección de hardware para el acceso y control del sistema. .	21
5.2.1.2 Selección de hardware para el sensado de la iluminación.....	25
5.2.1.3 Selección de hardware para el sensado de temperatura	26
5.2.1.4 Selección de hardware para el sensado del movimiento	28

5.2.1.5 Selección de hardware para la ventilación.....	29
5.2.1.6 Selección de hardware para el posicionamiento.....	29
5.2.1.7 Selección de hardware para la iluminación nocturna.....	31
5.2.1.8 Selección de hardware para la captura de imágenes y videos	32
5.2.1.9 Módulos conversores de tensión.....	35
5.2.1.10 Selección de hardware para el entrenamiento del modelo....	35
5.2.1.3 Selección de hardware para el sensado de voltaje.....	37
5.2.2 Diseño de Software.....	38
5.2.2.1 Diseño de Software de Cámara Trampa.....	38
5.2.2.2 Diseño de GUI (Interfaz Gráfica de Usuario).....	39
5.2.2.3 Diseño de Scripts para la Catalogación de imágenes.....	39
5.2.3 Definición de módulos de software de Cámara Trampa.....	41
5.2.3.1 Módulo Control.....	43
5.2.3.2 Módulo Ventana Principal de Interfaz.....	47
5.2.3.3 Módulo Inicio Script.....	49
5.2.3.4 Módulo Detección, Captura y actualización de datos.....	51
5.2.3.5 Módulo Cierre de Interfaz.....	53
5.2.4 Definición de scripts para la Catalogación de imágenes.....	55
5.3 Implementación.....	57
5.3.1 Implementación de Hardware.....	57
5.3.1.1 Sensado y Control de voltaje de batería.....	58
5.3.1.2 Sensado de luz y Control de iluminación nocturna.....	59
5.3.1.3 Sensado de Temperatura y Control de ventilación.....	61
5.3.1.4 Circuito de Iluminación nocturna.....	62
5.3.1.5 Sensado de Movimiento.....	64
5.3.1.6 Control de motor.....	67
5.3.1.7 Circuito final.....	69
5.3.1.8 Placa PCB.....	73
5.3.2 Implementación de Software.....	76
5.3.2.1 Implementación de software de Cámara Trampa.....	76
5.3.2.2 Implementación de scripts para la Catalogación de imágenes	79
5.3.3 Consumo: Elección de batería y panel solar.....	85
5.3.4 Implementación de Gabinete.....	88
5.4 Interfaz Humano-Maquina.....	90
5.4.1 Uso de Cámara Trampa.....	90
5.4.2 Uso de scripts de Vision Artificial.....	91
5.4.2.1 Agregado de imágenes a set de datos para reentrenamiento.	92
5.5 Validación y verificación.....	94
5.5.1 Casos de prueba.....	96
5.5.1.1 Caso de prueba N° 01: Alimentación del sistema.....	96
5.5.1.2 Caso de prueba N° 02: Iluminación nocturna.....	96
5.5.1.3 Caso de prueba N° 03: Conexión mediante Bluetooth.....	96
5.5.1.4 Caso de prueba N° 04: Conexión mediante WiFi.....	96
5.5.1.5 Caso de prueba N° 05: Configuración mediante GUI.....	96

5.5.1.6 Caso de prueba N° 06: Detección de movimiento y reposicionamiento del sistema.....	97
5.5.1.7 Caso de prueba N° 07: Almacenado de archivos.....	97
5.5.1.8 Caso de prueba N° 08: Catalogado de imágenes.....	97
5.5.1.9 Caso de prueba N° 09: Gabinete Estanco.....	97
5.5.1.10 Caso de prueba N° 10: Refrigeración del sistema.....	98
5.5.2 Resultados de los casos de prueba.....	102
5.5.3 Pruebas de campo.....	107
5.5.3.1 Prueba en área despejada de radio 15 metros.....	108
5.5.3.2 Prueba de iluminación nocturna.....	112
Capítulo 6: Resultados.....	113
Capítulo 7: Análisis de Costos.....	116
Capítulo 8: Discusión y Conclusión.....	117
8.1 Trabajos Futuros.....	118
Capítulo 9: Literatura Citada.....	119
Capítulo 10: Anexos.....	123
Anexo 1: Cuadro comparativo de versiones de Raspberry Pi.....	123
Anexo 2: Diagrama de conexionado completo.....	124
Anexo 3: Instalación y configuración de programas en Raspberry.....	125
Anexo 4: Instalación y configuración de programas en computadora.....	131

Lista de Figuras:

Lista de Tablas:	13
Figura 1. Formato de neurona artificial [17].	8
Figura 2. Comparativa de arquitecturas de VGG-16 y VGG-19 [23].	10
Figura 3. Partes de la arquitectura a implementar.	19
Figura 4. Computadora de placa única Raspberry Pi 3 A+ [29].	24
Figura 5. Placa de desarrollo Arduino Nano [31].....	25
Figura 6. Fotorresistencia.	26
Figura 7. Termistor NTC [33].	27
Figura 8. Sensor PIR Optex Saver.....	28
Figura 9. Ventilador de 5V de la marca Pengda Technology.	29
Figura 10. Motor Paso a Paso Nema 17 modelo 17HD34008-22B.	30
Figura 11. (a) Shield y (b) Controlador de motor paso a paso A4988	31
Figura 12. Sensibilidad relativa del sensor de la cámara vs longitud de onda de iluminación infrarroja [35].....	32
Figura 13. Led High Power 3W infrarrojo de 850nm [36].	32
Figura 14. Raspberry Pi Camera Module 2 NoIR [37].....	33
Figura 15. Filtro infrarrojo de corte [38].....	34
Figura 16. Módulo conversor DC-DC Step Down.....	35
Figura 17. Notebook HP Victus 16.1' GTX 1650.	36
Figura 18. Diagrama en bloques final del sistema.	36
Figura 19. Resistencia de 1/2W y 5% de tolerancia.....	37
Figura 20. Lenguaje de programación Python [40].	38
Figura 21. Diagrama de estados de Cámara Trampa.	43
Figura 22. Diagrama de flujo de Módulo Control.....	44
Figura 23. Curva de histéresis para control de tensión.	45
Figura 24. Curva de histeresis para control de iluminación nocturna.	46
Figura 25. Curva de histéresis para control de Temperatura.	47
Figura 26. Diagrama de flujo de Módulo Ventana Principal de Interfaz.....	48
Figura 27. Diagrama de flujo de Módulo Inicio Script.....	50
Figura 28. Diagrama de flujo de Módulo detección, captura y actualización.	53
Figura 29. Diagrama de flujo de Módulo Cierre de Interfaz.....	54
Figura 30. Diagrama de flujo de Script de entrenamiento.	56
Figura 31. Diagrama de flujo de Script de Predicción y Catalogación.....	57
Figura 32. Sensado de tensión mediante divisor resistivo.	58
Figura 33. Sensado de intensidad de luz con circuito divisor resistivo.	59
Figura 34. Circuito de control de filtro de corte infrarrojo.....	60
Figura 35. Sensado de temperatura mediante divisor resistivo con NTC.....	62
Figura 36. Circuito de iluminación nocturna.	63
Figura 37. (a) Circuito de Alarma desactivada y (b) activada sensores PIR.	64
Figura 38. (a) Vista Superior y (b) Vista de Lado de cobertura [50].	65
Figura 39. Ángulo de detección de sensor PIR.....	66
Figura 40. Ubicación de sensores, regiones de detección y cobertura.	67
Figura 41. Circuito de control de motor paso a paso.....	68

Figura 42. Conexión de switch de fin de carrera.	68
Figura 43. Esquema circuital simplificado de Cámara Trampa.	69
Figura 44. Bloque de alimentación de circuito final.	69
Figura 45. Bloque de conversor Step Down de Arduino y módulos relé.....	70
Figura 46. Bloque Arduino Nano – Sensado y control.	71
Figura 47. Bloque de conversor Step Down 2 para alimentar Raspberry.....	71
Figura 48. Bloque Raspberry – Sensado y control.....	72
Figura 49. Conexión de puertos GPIO de Raspberry.....	72
Figura 50. Diagrama esquemático de conexiones de cámara trampa.	73
Figura 51. (a) Vista superior y (b) inferior de diseño Placa PCB.	74
Figura 52. (a) Vista superior y (b) inferior de Placa PCB.....	75
Figura 53. Interfaz Gráfica de Cámara Trampa.....	76
Figura 54. Sección de ayuda de Interfaz Gráfica.	77
Figura 55. GUI de Cámara Trampa con visión nocturna desactivada.	78
Figura 56. GUI de Cámara Trampa con visión nocturna activada.....	78
Figura 57. Navegador de archivos en directorio de capturas y grabaciones.	79
Figura 58. Imágenes utilizadas en el entrenamiento del modelo.....	79
Figura 59. Razas de las distintas especies a detectar.	80
Figura 60. Métricas de desempeño para (a) VGG-16 y (b) VGG-19	81
Figura 61. Matriz de confusión para (a) VGG-16 y (b) VGG-19	82
Figura 62. Consumo de la cámara trampa en condición más desfavorable.	86
Figura 63. División del circuito en 2 niveles para simple acceso al almacenamiento.	88
Figura 64. Gabinete de camara trampa con conectores estancos, portaleds y porta cámara.	88
Figura 65. Gabinete de sensores PIR.....	89
Figura 66. Redes creadas para el ingreso remoto.	90
Figura 67. Apertura automática de interfaz Gráfica.....	90
Figura 68. Interfaz Web de Jupyter Notebook.....	92
Figura 69. descripción de recorte de imagen [51].	93
Figura 70. Tester o multímetro digital.	95
Figura 71. Fuente de tensión continua regulable.	95
Figura 72. Configuración de sistema establecida.....	107
Figura 73. Sistema de cámara trampa, montado en poste.....	108
Figura 74. Detección y captura a 10 metros, demarcada con cinta blanca.	109
Figura 75. Detección y captura a 15 metros.	110
Figura 76. Detección y captura a 7 metros, caminando rapido.	110
Figura 77. Seguimiento del objetivo hacia la derecha.....	111
Figura 78. Seguimiento del objetivo hacia la izquierda.	111
Figura 79. Imagen diurna: 8MP, ISO 100, velocidad de obturación 500. ...	112
Figura 80. Imagen nocturna: 8MP, ISO 800, velocidad de obturación 300000.	112
Figura 81. Modelos y versiones de Raspberry Pi [53].....	123
Figura 82. Diagrama circuital de sistema completo.....	124
Figura 83. Captura de interfaz de software BalenaEtcher.....	125

Figura 84. Icono de VNC Viewer.....	127
Figura 85. Nueva red para acceso por servidor de VNC Viewer.	127
Figura 86. Icono de bluetooth Blueman.	129
Figura 87. Creación de nueva red para acceso por bluetooth.....	130
Figura 88. Anaconda Navigator: Interfaz Gráfica de Anaconda.	131
Figura 89. Detección de placa de video NVIDIA GeForce GTX 1650.	133

Lista de Tablas:

Tabla 1. Requerimientos funcionales.....	13
Tabla 2. Requerimientos no funcionales.....	18
Tabla 3. Resistencia de Termistor NTC según temperatura medida.....	61
Tabla 4. Tensión de salida de divisor resistivo según temperatura medida.	62
Tabla 5. Errores e imágenes desconocidas para % de certeza > 0,9995.....	83
Tabla 6. Errores e imágenes desconocidas para % de certeza > 0,999.	83
Tabla 7. Errores e imágenes desconocidas para % de certeza > 0,995.	84
Tabla 8. Errores e imágenes desconocidas para % de certeza > 0,99.	84
Tabla 9. Errores e imágenes desconocidas para % de certeza > 0,95.	84
Tabla 10. Errores e imágenes desconocidas para % de certeza > 0,9.	84
Tabla 11. Matriz de trazabilidad de requerimientos funcionales del sistema.	94
Tabla 12. Herramientas utilizadas para cada caso de prueba.....	95
Tabla 13. Escenario, precondiciones y casos/sub-casos de cada prueba.	102
Tabla 14. Resultados obtenidos de la ejecución de cada caso de prueba.	105
Tabla 15. Fallas de los casos de prueba y sus soluciones.....	106
Tabla 16. Costo final del proyecto.....	116

Lista de Abreviaciones y Símbolos

ms	Milisegundos
PIR	Passive Infrared (Sensor infrarrojo pasivo)
AIR	Active Infrared (Sensor infrarrojo activo)
IA	Artificial Intelligence (Inteligencia Artificial)
ADC	Analog to Digital Converter (Conversor analógico-digital)
GUI	Graphical User Interface (Interfaz Gráfica de Usuario)
SBC	Single Board Computer (Computadora de placa única)
PIC	Programmable Integrated Circuit (Circuito integrado programable)
LDR	Light Dependent Resistor (Resistencia dependiente de la luz)
RTD	Resistance Temperature Detector (Detector resistivo de temperatura)
NTC	Negative Temperature Coefficient (Termistor de coeficiente negativo)
IDE	Integrated Development Environment (Entorno de Desarrollo Integrado)
CPU	Central Processing Unit (Unidad central de procesos)
GPU	Graphical Processing Unit (Unidad central de procesamiento gráfico)
PCB	Printed Circuit Board (Placa de Circuito Impreso)
ANN	Artificial Neural Network (Red Neuronal Artificial)
CNN	Convolutional Neural Network (Red Neuronal Convolutiva)
CUDA	Compute Unified Device Architecture (Arquitectura Unificada de Dispositivos de Computo)
CUDNN	NVIDIA CUDA Deep Neural Network (Red Neuronal Profunda CUDA de NVIDIA)
RTC	Real Time Clock (Reloj en Tiempo Real)
<i>IPYNB</i>	<i>Interactive Python Notebook (cuaderno interactivo de python)</i>

Dedicado a:

Mi abuela Lucy, por su lucha incesante por mi bienestar, por su amor incondicional y por brindarme las oportunidades y herramientas necesarias para proyectar mi vida.

Capítulo 1: Introducción

Una cámara trampa es un dispositivo formado por una cámara que cuenta con un sistema de disparo automático el cual la activa al detectar movimiento, y en función de la configuración previa del usuario captura fotografías y/o videos [1].

En sus inicios, las cámaras trampa utilizaban sistemas de disparo automático mecánicos [1] [2] [3] [4], pero a partir de 1980, los mismos fueron reemplazados por sensores infrarrojos activos [3]. Estos nuevos sensores presentan como desventaja, la necesidad de conocer la ruta que sigue el animal para activar la cámara [1].

En la actualidad, la gran mayoría de cámaras trampa integra un sistema de disparo automático pasivo [5], denominado *Sensor infrarrojo pasivo* (PIR acrónimo del inglés, *Passive Infrared*). Su funcionamiento consiste en captar una diferencia de temperatura en los elementos que lo componen, al percibir la radiación infrarroja emitida por los cuerpos presentes en el área de detección, generando una corriente eléctrica [6].

La utilización de estos dispositivos para estudios de monitoreo sobre la distribución y abundancia de fauna silvestre, son ampliamente utilizados en la academia científica como en áreas protegidas que realizan tareas de conservación [7]. Estas cámaras permiten capturar registros de especies que son sensibles a la presencia de seres humanos, facilitando el conocimiento de la diversidad de fauna del lugar. Por lo tanto, son un complemento esencial para los inventarios manuales que se realizan. Las aplicaciones de uso son diversas y dependen de los objetivos de estudio, de manera simplificada, se puede decir que consisten en conocer la presencia y ocurrencia de una cierta especie en un sector específico, realizar estudios comportamentales de actividad en distintos periodos de tiempo, inferir en estudios relacionados a la conservación de una especie, entre otras aplicaciones [1] [3].

Sin embargo, las cámaras trampa tradicionales cuentan con una desventaja ya que a partir de un punto fijo de colocación, el ángulo de detección es estrecho, y en consecuencia implica aumentar el número de cámaras por punto de muestreo. Una alternativa es aumentar el ángulo de detección de la cámara trampa [7] [8], es decir aquel ángulo en que los sensores son capaces de detectar movimiento en varias direcciones. Algunos autores sostienen que estaría asociado al ángulo de la lente de la cámara, siendo también otra limitante en las cámaras comerciales [1]. Por este motivo, para obtener mayores ángulos de detección, se utilizan cámaras con lentes gran angulares, pero estas generan excesiva distorsión en la imagen [9], dificultando la identificación posterior de las especies por parte del operario.

Otro aspecto a tener en cuenta durante el proceso de monitoreo utilizando cámaras trampa tradicionales, es la cantidad de fotografías que son capturadas durante el período de muestreo, las cuales deben ser procesadas manualmente por un operario entrenado en la identificación de las especies. Esto es una tarea compleja y trabajosa, pudiendo estar semanas o meses dependiendo del período de estudio [3]. Si bien existen softwares para aminorar la labor, solo organizan las imágenes por sitio, estación y revisiones en fechas específicas [1], pero no son capaces de identificar la especie. Para evitar prolongadas jornadas de trabajo, una alternativa es utilizar un método de clasificación de imágenes denominado Visión Artificial que consiste en entrenar previamente al ordenador, y luego con ese aprendizaje, ordenar automáticamente las imágenes [10]. La elaboración de estos modelos implica contar con un número suficientemente grande de fotografías de cada especie a identificar, para aumentar la cantidad de rasgos posibles de cada especie.

Por tal motivo, el objetivo de este proyecto final consistió en diseñar y elaborar un dispositivo que permita capturar imágenes y/o videos con un campo de detección aumentado a 360°, y acoplado al mismo, un sistema de identificación de imágenes basado en especie animal.

Capítulo 2: Antecedentes

La observación es el método más utilizado para estudiar el comportamiento animal en su hábitat natural [2] cuando las especies pueden coexistir en el mismo lugar y al mismo tiempo con la presencia del ser humano. La cámara trampa es un dispositivo interesante para el estudio de la fauna silvestre [3] porque permite ampliar las listas de especies cuando las mismas son temerarias a la presencia humana. También hay animales que poseen hábitos nocturnos dificultando su documentación cuando los muestreos son por observación directa [3] [4].

La primera captura de una cámara trampa fue realizada en 1878, por E. J. Muybridge quien estaba interesado en la locomoción de los caballos. Para documentarlo, utilizó múltiples cámaras, las cuales fueron activadas por cuerdas que chocaba el caballo al galopar [2] [3]. Luego, en 1890, se fotografió, por primera vez de manera remota, a la fauna silvestre. El método fue implementado por George Shiras III, quien unió cámaras con linternas utilizando cables para ser activadas al animal tirar de ellas [1] [4].

En la década de los 20', fue realizado el primer inventario de mamíferos en Barro Colorado, por F.M. Chapman [2] y fue fotografiado el primer tigre salvaje, por F.W. Champion, dejando en evidencia que cada tigre tenía un patrón particular de rayas por las que podía ser identificado [2] [3].

Los primeros *sensores infrarrojos activos* (AIR acrónimo del inglés, *Active Infrared*) fueron utilizados en 1980. Los mismos consistían en un emisor y un receptor, estos enviaban y recibían pulsos, respectivamente. La interrupción de los pulsos enviados era indicativa de obstrucción activando la cámara trampa [7].

En 1984, A. Seydack realizó en Sudáfrica el primer estudio de mamíferos, utilizando un sistema de activación automático mecánico con placas de presión logrando más de 500 imágenes [7].

En la década de los 90', se acuñó el término "cámara trampa" a este tipo de dispositivos [7] y su venta se vio magnificada para realizar inventarios faunísticos entre las acciones de conservación de una región [1] [2].

En la década de los 2000, las cámaras trampa digitales reemplazaron a las de película permitiendo el uso de flash infrarrojo, dispositivos de almacenamiento de tamaño reducido y sensores PIR [7].

En la actualidad, algunas cámaras trampa poseen conectividad WiFi o celular permitiendo el envío de datos a distancia y la configuración remota del sistema. La tendencia de las cámaras trampa es hacia la miniaturización [1] [3] debido al vandalismo que sufren constantemente, y también porque ciertas especies animales aún pueden detectarlas a pesar del camuflaje utilizado.

Capítulo 3: Marco teórico

3.1 Sistema de activación automático

Las cámaras trampa se pueden clasificar según su sistema de activación automático en sistema activo y sistema pasivo [2].

En el primer tipo, el sensor del sistema de activación está compuesto por 2 partes, un emisor y un receptor, el sistema emite energía de forma constante a través del emisor y la recibe por el receptor. En el caso de que la recepción de energía se vea afectada por una obstrucción, el sistema emitirá un aviso. Los sistemas de detección de movimiento activos se clasifican por la utilización de microondas, ultrasonidos o luz [3].

Los sensores de movimiento por microondas permiten detectar distintas magnitudes asociadas con el movimiento, tales como la velocidad y la dirección, entre otras. Su principio de funcionamiento se basa en el efecto Doppler por lo que al emitir una onda electromagnética con una frecuencia establecida, si la onda que retorna posee una frecuencia distinta, indica que se encontró un objeto en movimiento. El área de detección de estos dispositivos es de 360° sin puntos ciegos [11].

Se considera ultrasonido a cualquier onda acústica de frecuencia superior a 20 kHz. Los sensores activos de movimiento por ultrasonido también utilizan el efecto Doppler para la detección de movimiento y tienen una alta sensibilidad [12]. Aun así, aunque el ultrasonido es imperceptible por el oído humano, genera irritación en animales incitando a que abandonen el lugar donde está dispuesto dicho sensor [13].

Los sensores de movimiento por luz pueden emitir haces continuos o pulsados de luz, donde los de haces continuos obtienen tiempos de respuesta rápidos pero su distancia se ve acotada y son interferidos por la luz externa, mientras

que los de haces pulsados pueden obtener distancias de censados superiores y un menor efecto de interferencia de luz externa [14].

Existen distintos sensores de acuerdo con la disposición del emisor y receptor, los mismos pueden estar en el mismo equipo o en equipos separados para realizar distintas configuraciones. Los sensores AIR utilizados en cámaras trampa son del tipo barrera, lo que indica que el emisor y el receptor se encuentran enfrentados, donde el objeto a detectar debe interferir el recorrido del haz, además, los mismos pueden utilizar tecnología LED o LASER [14].

A diferencia de los sensores activos, los sensores pasivos no emiten energía, por lo que se requiere solamente de un elemento receptor para la detección de movimiento. El sensor pasivo más utilizado es el PIR, este contiene un sensor piroeléctrico con múltiples elementos que generan una señal eléctrica al detectar una diferencia de temperatura entre sus elementos. Dicha temperatura es representada por las distintas radiaciones infrarrojas presentes en la zona de detección, emitidas por los distintos cuerpos [6].

Los animales de sangre caliente, como por ejemplo los mamíferos, regulan su temperatura corporal por un proceso denominado homeostasis, es decir, son capaces de mantener una temperatura aproximadamente constante, independientemente de la temperatura ambiente. En consecuencia en la mayoría de los casos se traduce en una temperatura diferente a la del entorno, aumentando la probabilidad de detección de su movimiento [7].

Se considera entonces que la mejor opción para la detección de movimiento animal es el sensor PIR, ya que es capaz de detectar la radiación de los cuerpos y su movimiento. Este no emite ningún tipo de energía evitando modificar o alterar el comportamiento de los animales y no requiere una ruta fija por la ausencia de un dispositivo emisor. Además, a diferencia del sensor de movimiento por microondas, posee un campo de visión más acotado permitiendo obtener distintas señales de activación en distintos sectores circulares.

3.2 Clasificación de imágenes utilizando visión artificial

Realizar la clasificación de imágenes de forma manual es una de las tareas más tediosas del relevamiento con fototrampeo. La implementación de un sistema automático capaz de separar las fotografías por especies utilizando visión artificial, puede ser una gran utilidad para optimizar el proceso, reduciendo el tiempo de procesamiento [7].

La Visión Artificial forma parte de la *Inteligencia Artificial* (AI acrónimo del inglés, *Artificial Intelligence*). La misma tiene por objetivo comprender el contenido de las imágenes digitales [15]. Para cumplir dicho objetivo utiliza Redes Neuronales Artificiales (ANN acrónimo del inglés, *Artificial Neural Network*), que adoptan su funcionamiento de las neuronas de los animales [16].

Una neurona artificial integra:

- Múltiples entradas, con valores determinados entre 0 y 1.
- Múltiples pesos asignados a cada entrada, los cuales pueden ser negativos o positivos, indicando si la entrada estimulará o suprimirá la acción de la neurona. Además, a mayor magnitud, se obtiene un mayor efecto.
- Una función de activación que limitará su salida, determinando su aporte a la red neuronal.

La neurona realizará una suma de sus entradas ponderadas, luego se aplicará la función de activación determinando el valor de salida de dicha neurona.

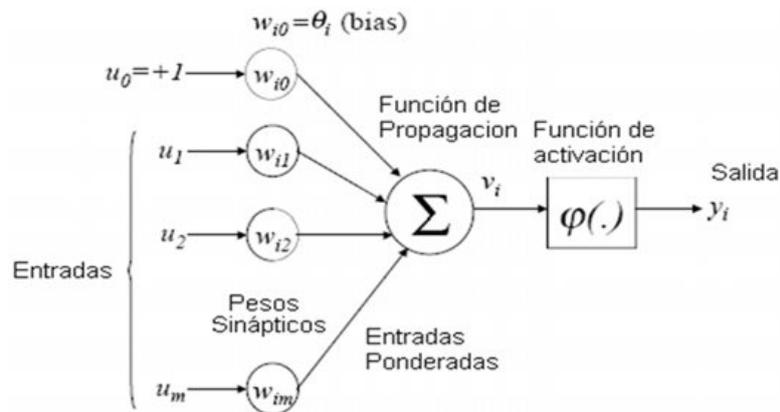


Figura 1. Formato de neurona artificial [17].

Las redes neuronales consisten en múltiples neuronas interconectadas que pueden estar dispuestas en una capa simple o en múltiples capas.

La operación de una red neuronal se divide en dos etapas: entrenamiento y ejecución. Durante el entrenamiento, la red neuronal comienza con pesos aleatorios que se van ajustando en cada iteración con el fin de lograr una mayor precisión. En la segunda operación, los pesos ya han sido fijados, y se utiliza el modelo para procesar la información de entrada [18].

El proceso de visión artificial consiste en: adquirir la información de la imagen, procesarla, detectar bordes, segmentarla, extraer sus características, realizar un reconocimiento y localización e interpretarla [19].

A diferencia de las técnicas de detección y segmentación utilizadas en visión artificial, que consisten en reconocer múltiples objetos en una imagen, la clasificación busca determinar a qué objeto representa la totalidad de la imagen [20].

Para la clasificación de imágenes por visión artificial, se utilizan redes neuronales convolucionales que reducen la información a procesar, utilizando menos recursos computacionales. Un conjunto de datos etiquetados y de redes neuronales conforman un modelo [21].

Las *Redes Neuronales Convolucionales* (CNN, *Convolutional Neural Network*) consisten en capas específicas de ANN ordenadas por jerarquía, para obtener un resultado efectivo en el procesamiento de la información. Las primeras capas detectan los patrones más sencillos y a medida que avanza la información por la CNN, las capas se complejizan para detectar distintos patrones.

En el procesamiento de una imagen se requiere una cantidad de neuronas determinada por el alto y el ancho en pixeles y el modelo de color, en caso de ser escala de grises, el número de neuronas solo se determina por el alto y el ancho mientras que si se usa el modelo RGB se debe multiplicar el número de neuronas por los 3 canales del modelo: rojo, verde y azul.

Una vez ingresada la información de la imagen, la CNN realiza un proceso de convolución o filtrado, entre la matriz generada por los datos de la imagen y la matriz del filtro obteniendo una matriz de salida. Luego se realiza un submuestreo donde se reduce el tamaño de la imagen, conservando las características más destacadas y evitando que la matriz de salida crezca exponencialmente, ya que es requerido el uso de una amplia cantidad de imágenes.

Por último, se aplica la función de activación. Para la clasificación de imágenes se utiliza la función Softmax por su carácter multicapa, lo que permite el uso de más de una clase. Esta función de activación entrega como resultado un valor entre 0 y 1 para cada una de las clases entrenadas, donde la sumatoria de todos los resultados es igual a 1 [18]. Existen modelos que poseen un entrenamiento previo, haciendo uso de millones de imágenes, para mejorar los porcentajes de predicción, lo que permite reducir el tiempo de ejecución [21]. Estos modelos son ampliamente utilizados debido a que el diseño de una red neuronal no sigue patrones estrictos de número de capas y orden, complejizando el proceso de creación de un modelo efectivo.

Los modelos preentrenados VGG-16 y VGG-19 presentan suficiente evidencia para determinar que brindan muy buenos resultados para la clasificación de imágenes en comparación con otros modelos [22]. Los mismos consisten en varias capas convolucionales seguidas de capas de submuestreos, finalizando con una función de activación Softmax para la distinción de clases.

El modelo VGG16 posee 13 capas convolucionales y 3 capas Softmax mientras que VGG19 posee 16 capas convolucionales y 3 capas Softmax, por lo que se considera al modelo VGG19 una versión mejorada de su predecesor [18].

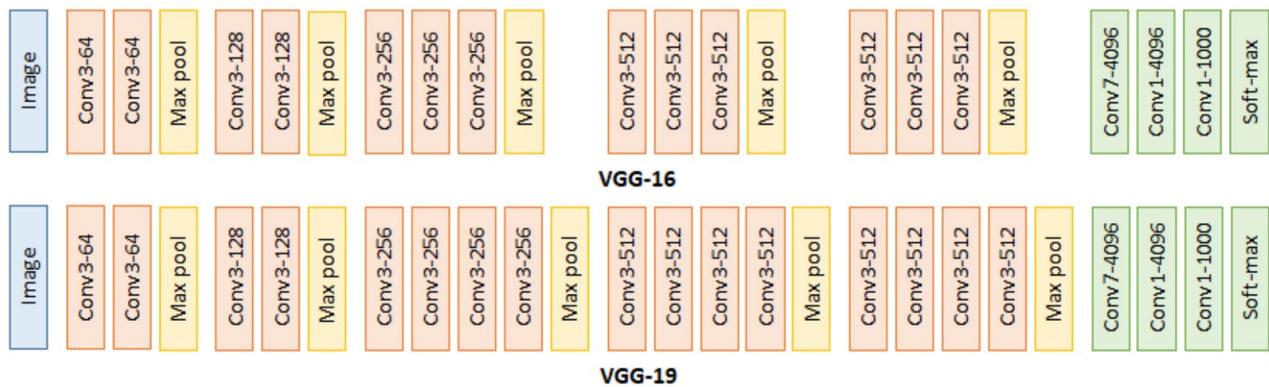


Figura 2. Comparativa de arquitecturas de VGG-16 y VGG-19 [23].

Capítulo 4: Requerimientos

El objetivo principal de este proyecto consistió en diseñar y elaborar un dispositivo que permita capturar imágenes y/o videos con un campo de detección aumentado a 360°, acoplándose al mismo un sistema de identificación de imágenes basado en especie animal. Para ello se utilizaron múltiples sensores PIR, un motor paso a paso para el posicionamiento, una interfaz gráfica de configuración y una serie de algoritmos para implementar el sistema de detección de imágenes por visión artificial.

A continuación se detallan los objetivos particulares perseguidos:

- Ser alimentado por una batería de 12V.
- Realizar una interfaz gráfica de configuración amigable para el usuario.
- Configurar y monitorear el dispositivo de captura de imágenes y video de forma remota.
- Utilizar iluminación nocturna para detecciones nocturnas sin afectar el comportamiento de los animales.
- Detectar el movimiento del objetivo y seguirlo mediante el sistema de posicionamiento, capturar imágenes y grabar al objeto.
- Almacenar las imágenes y los videos en un dispositivo de almacenamiento externo.
- Realizar un sistema de detección de imágenes por especie animal, como por ejemplo: puma, gato montés, ciervo, carpincho y vaca.
- Diseñar e implementar un gabinete estanco que soporte condiciones naturales.

El sistema desarrollado debe soportarse sobre un poste rígido, y su altura sobre el terreno puede ser variable en función del objetivo de investigación. Frecuentemente se instalan a unos 60 cm del suelo teniendo en cuenta que los animales a detectar son de tamaño mediano a grande. Un aspecto a tener en cuenta durante la instalación es la posición respecto al camino o sendero

marcado por el paso de animales previamente, dado que esto permitirá aumentar el éxito de detección de determinadas especies. Se aconseja a una distancia mínima de 3 metros de un posible sendero o camino, de manera perpendicular o formando un ángulo de 45°, preferentemente apuntando hacia el horizonte [1]. Otro de los requerimientos importante para los estudios de monitoreo de fauna en un sitio determinado, es la presencia de marcas de agua sobre las imágenes y/o videos. Por lo tanto, es indispensable contar con una sección dentro de la interfaz de configuración con una leyenda que indique a que zona del predio o lugar corresponde esa cámara y cuál es la fecha y hora de captura.

A continuación, se presentan los requerimientos funcionales y no funcionales, establecidos para el cumplimiento del objetivo principal y los particulares del proyecto.

4.1 Requerimientos funcionales

Los requerimientos funcionales hacen referencia a las acciones o funciones específicas que el sistema debe realizar, así como también, cuál debe ser su respuesta ante una entrada conocida [24].

Requerimiento Funcional	Descripción del Requerimiento
RF01	Alimentar sistema con batería de 12V
RF02	Supervisar tensiones del sistema
RF03	Medir intensidad de luz
RF04	Iluminar ambiente en condiciones de iluminación escasa
RF05	Conectar por bluetooth para Seteo inicial
RF06	Conectar a internet para acceso remoto
RF07	Configurar mediante interfaz gráfica
RF08	Detectar movimiento

RF09	Posicionar Cámara en región de detección
RF10	Capturar imágenes y grabar videos
RF11	Almacenar archivos de imágenes y videos en dispositivo externo
RF12	Almacenar archivos de imágenes y videos en carpetas específicas
RF13	Nombrar archivos de imágenes y videos con nombres específicos
RF14	Reconocer animales mediante script de visión artificial
RF15	Estar contenido en un Gabinete Estanco
RF16	Refrigerar sistema

Tabla 1. Requerimientos funcionales

4.1.1 RF01: Alimentar sistema con batería de 12V

El sistema debe ser alimentado por una batería solar de 12V, con probabilidad de picos de tensión de hasta 21V.

4.1.2 RF02: Supervisar tensiones del sistema

El sistema debe utilizar una tensión de alimentación de 12V, la cual puede variar desde 11V a 15,5V con probabilidad de picos de tensión de hasta 21V.

En caso de tensiones inferiores a 11V o superiores a 15,5V, la alimentación del sistema debe ser interrumpida hasta obtener nuevamente valores permitidos, para no dañar la batería (en caso de subtensión); ni los componentes internos (en caso de sobretensión).

La precisión en ambos valores debe ser de $\pm 2\%$.

4.1.3 RF03: Medir intensidad de luz

Se debe detectar la intensidad de la luz ambiente, para el encendido y apagado de la iluminación nocturna.

Dependiendo la intensidad de luz ambiente, el equipo cambiará su configuración entre diurna y nocturna, para el encendido/apagado de la iluminación nocturna, así como también para el guardado de las imágenes y videos en los directorios correspondientes, detallados en el RF12.

4.1.4 RF04: Iluminar ambiente en condiciones de iluminación escasa

Se debe contar con un sistema de iluminación nocturna para ser utilizado en caso de escasa iluminación ambiente. El mismo no debe perturbar el ambiente de los animales y debe tener un alcance de al menos 10 metros.

4.1.5 RF05: Conectar por bluetooth para Seteo inicial

El sistema debe poder ser accedido por bluetooth para realizar la configuración inicial del equipo cuando la conexión a internet aun no haya sido configurada.

4.1.6 RF06: Conectar a internet para acceso remoto

El sistema debe poder conectarse a internet para ser accedido remotamente, brindando la posibilidad de observar las capturas y grabaciones realizadas, comprobar el estado de funcionamiento del equipo y reconfigurar el mismo.

4.1.7 RF07: Configurar mediante interfaz gráfica

Se debe contar con una interfaz gráfica capaz de configurar:

- Resolución de imagen, posibilitando todas las permitidas por la cámara.
- Resolución de video, posibilitando todas las permitidas por la cámara.
- Cantidad de capturas por detección.
- Tiempo de grabación en segundos.
- Agregado de leyenda en imagen y video, con ubicación de equipo, fecha y hora.
- Fecha y hora de modo manual o automático.

Además, debe contar con acceso al directorio en el que se almacenan las capturas y grabaciones previamente realizadas.

4.1.8 RF08: Detectar movimiento

El sistema debe detectar movimiento, en un campo de visión horizontal de 360°, con distancia máxima de 10 metros, para objetos con alturas entre 0,2 y 2 metros.

Se define entonces a la región de detección del equipo como un círculo de radio igual a 10 metros, apoyado sobre la superficie.

4.1.9 RF09: Posicionar Cámara en región de detección

El sistema debe contar con un sistema de posicionamiento capaz de situar al mismo en la porción de la región de detección donde se realizó dicha detección, para luego realizar las capturas y posterior grabación.

4.1.10 RF10: Capturar imágenes y grabar videos

Se debe poder capturar imágenes en formato JPG, a partir de la detección de movimiento, y grabar video en formato H264, con un tiempo preseleccionado entre 1 a 90 segundos, al finalizar las capturas.

4.1.11 RF11: Almacenar archivos de imágenes y videos en dispositivo de almacenamiento externo

Se debe contar con un dispositivo de almacenamiento externo para almacenar las capturas y grabaciones realizadas.

4.1.12 RF12: Almacenar archivos de imágenes y videos en carpetas específicas

El sistema debe generar las siguientes carpetas, para la organización de las capturas y grabaciones:

- Ruta de capturas y grabaciones diurnas: ~/CyG/Diurnas
- Ruta de capturas y grabaciones nocturnas: ~/CyG/Nocturnas

4.1.13 RF13: Nombrar archivos de imágenes y videos con nombres específicos

El sistema debe nombrar los archivos de las imágenes y videos según el siguiente formato “dd_mm_aa-HH_MM_SS-(#)-X”, donde:

- dd: día
- mm: mes
- aa: año
- HH: hora
- MM: minuto
- SS: segundo
- #: número de captura

X indica el momento en el que se realizó la captura:

- X = D: Captura diurna
- X = N: Captura Nocturna

El nombre de los archivos de videos debe seguir el siguiente formato:

- dd_mm_aa-HH_MM_SS-X.

4.1.14 RF14: Reconocer animales mediante script de visión artificial

El sistema debe contar con un script que utilice métodos de visión artificial para la detección de las especies.

Los animales por detectar son:

- Puma
- Gato montés
- Ciervo
- Carpincho
- Oveja
- Vaca

Las imágenes catalogadas deben tener un nombre específico, siguiendo el siguiente formato:

- dd_mm_aa-HH_MM_SS-(#)-X-Animal.

Además, el script utilizado para correr la visión artificial debe permitir el agregado de nuevas imágenes para mejorar la precisión del modelo.

4.1.15 RF15: Estar contenido en un Gabinete Estanco

El equipo debe estar montado en un gabinete estanco, con un acceso sencillo al dispositivo de almacenamiento externo.

4.1.16 RF16: Refrigerar sistema

Dentro del compartimiento del gabinete estanco se debe incluir un sistema de refrigeración, para proteger el circuito ante temperaturas elevadas.

4.2 Requerimientos no funcionales

Los requerimientos no funcionales hacen referencia a las características o propiedades específicas, tales como restricciones o factores de calidad, que ponen límites al sistema. Las mismas indican como el sistema debe realizar las funciones o bajo qué criterios [24].

Requerimiento No Funcional	Descripción del Requerimiento
RNF01	Capacidad de almacenamiento de dispositivo externo
RNF02	Capturar imagen con tiempos de obturación menor o igual a 1 ms

Tabla 2. Requerimientos no funcionales

4.2.1 RF01: Capacidad de almacenamiento de dispositivo externo

El dispositivo de almacenamiento externo debe contar con la capacidad suficiente para almacenar 15200 imágenes en formato JPG o 5.2 horas de video en formato H264.

4.2.2 RF02: Capturar imagen con tiempo de obturación de 1 ms

El módulo de cámara debe poder utilizar un tiempo de obturación menor o igual a 1 ms para poder congelar el movimiento de los animales [25].

Capítulo 5: Desarrollo

5.1 Arquitectura

El sistema de cámara trampa desarrollado consiste en 4 etapas, donde las etapas de configuración y control, sensado y actuación serán controladas por la placa de desarrollo mientras que la etapa de clasificación de imágenes se realizará en una computadora externa al sistema.

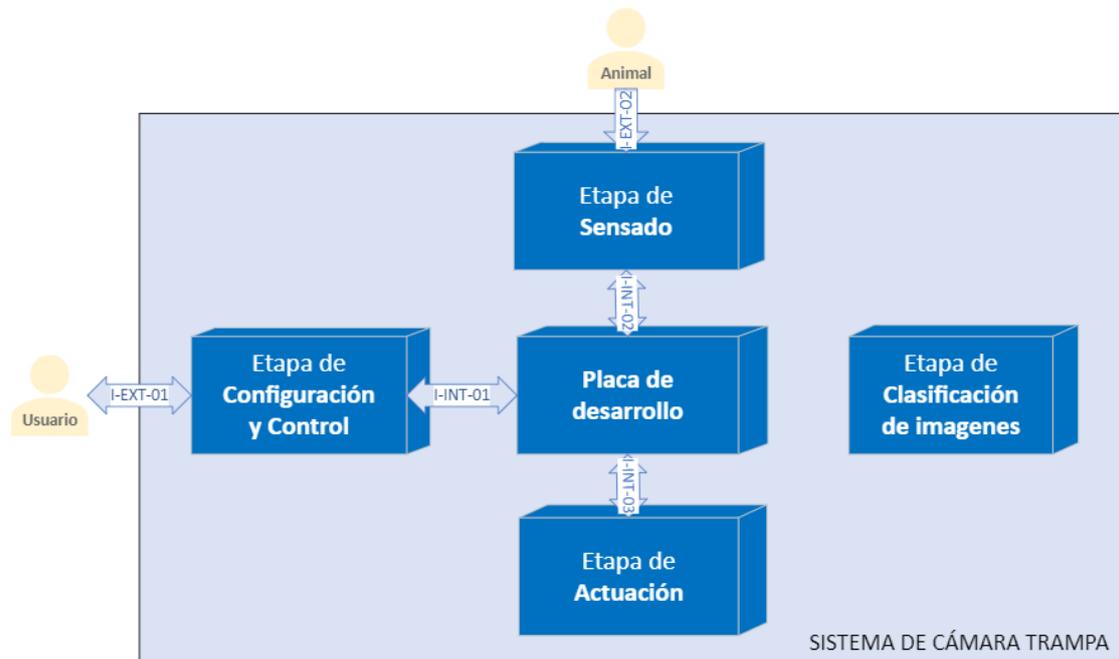


Figura 3. Partes de la arquitectura a implementar.

A continuación, se detalla el funcionamiento de cada etapa:

- **Etapa de Configuración y Control:** Consiste en la interfaz gráfica, como se detalló en el requerimiento RF07, la misma se comunica con la placa de desarrollo para setear la configuración de la cámara y la fecha y hora. Además, permite la transmisión de la imagen en tiempo real.
- **Etapa de Sensado:** El sistema sensa el movimiento de los animales, la intensidad de luz del ambiente, la tensión de la batería de la que se alimenta (según lo descrito en el requerimiento RF01) y la temperatura interna del equipo para el control de la ventilación (según lo descrito en el requerimiento RF16).

- **Etapa de Actuación:** Esta etapa consiste en la activación y/o control de 5 bloques, dentro de los que podemos nombrar: la cámara, el motor, la alimentación, la ventilación y la iluminación nocturna.
- **Etapa de Clasificación de imágenes:** Esta etapa está dividida en 2 partes. La primera consiste en el entrenamiento del modelo de visión artificial (para la cual se utilizará un modelo preentrenado), donde se requiere la mayor cantidad de imágenes posibles de las 6 especies de animales a detectar, y una computadora con placa de video para reducir el tiempo de entrenamiento. La segunda parte consiste en la utilización del modelo creado, para la clasificación de imágenes.

5.1.1 Actores externos

5.1.1.1 Actor externo: Usuario

Es la persona, quien configura y controla el sistema, accediendo remotamente a la interfaz gráfica.

5.1.1.2 Actor externo: Animal

Es el objeto en movimiento, al cual se le toman las capturas y grabaciones. Actúa sobre la etapa de sensado, al ser detectado por los sensores PIR.

5.1.2 Interfaces

Podemos identificar en la arquitectura planteada, 5 interfaces, donde 3 son internas y 2 externas.

5.1.2.1 Interfaces entre etapas

Definimos las interfaces entre el sistema y los actores externos:

- I-INT-01: Interfaz entre Etapa de Configuración y Control y Placa de desarrollo.
- I-INT-02: Interfaz entre Etapa de Sensado y Placa de desarrollo.

- I-INT-03: Interfaz entre Etapa de Actuación y Placa de desarrollo.

5.1.2.2 Interfaces con actores externos

Definimos las interfaces entre el sistema y los actores externos:

- I-EXT-01: Interfaz del Sistema con Usuario.
- I-EXT-02: Interfaz del Sistema con Animal.

5.2 Diseño

5.2.1 Diseño de Hardware

5.2.1.1 Selección de hardware para el acceso y control del sistema.

Para el desarrollo del sistema de cámara trampa, se requería una placa de desarrollo con soporte para:

- Cámara
- Conexión a internet mediante WiFi y conexión bluetooth
- Acceso remoto al equipo
- Interfaz gráfica
- Dispositivo de Almacenamiento Externo
- *Conversor analógico-digital* (ADC acrónimo del inglés, *Analog to Digital Converter*) para convertir los valores de los sensores de intensidad de luz y temperatura, que en su mayoría son analógicos.

Si bien existen placas de desarrollo basadas en microcontroladores, con soporte para cámara, conexión a internet mediante WiFi y conversores ADC, no son capaces de correr una *Interfaz Gráfica de usuario* (GUI acrónimo del inglés, *Graphical User Interface*), lo cual no cumplía con lo especificado en el requerimiento RF07.

Por este motivo, se necesitaba utilizar una placa de desarrollo basada en microprocesadores.

Existen placas de desarrollo denominadas *Computadoras de placa única* (SBC acrónimo del inglés, *Single Board Computer*), las cuales poseen una estructura similar a una computadora de escritorio, pero a diferencia de estas, integran todos los componentes necesarios para su funcionamiento en una única placa de tamaño reducido, tales como la memoria RAM, el microcontrolador, puertos de comunicación, video, audio, alimentación, conexión a internet, entre otros.

Para la realización del proyecto se optó por utilizar un SBC de la marca Raspberry Pi, debido a que las mismas presentan la mayor flexibilidad de programación, integran una gran cantidad de librerías para el desarrollo de software y poseen una comunidad amplia de desarrolladores, siendo incluso la SBC más utilizada en la educación de ingeniería e informática [26]. Además, posee gran cantidad de periféricos: USB, Conector MIPI CSI para la integración de cámara [27] [28]. También cuenta con WiFi, Bluetooth, puede ser accedido remotamente, permiten la integración de un dispositivo de almacenamiento externo mediante conexión USB y es posible ejecutar interfaces gráficas haciendo uso de sistemas operativos basados en Linux.

La única desventaja que presentaba esta placa SBC es que no integra un convertor ADC.

Existen distintas versiones de placas SBC de Raspberry Pi, desde la primera, denominada Raspberry Pi 1, hasta la más potente del mercado, denominada Raspberry Pi 4. Existen distintos modelos llamados A, A+, B y B+. Los modelos A están diseñados para tener menores costos y consumos de energía que los modelos B, los cuales son más potentes e integran mayor cantidad de periféricos.

La versión 1 en todos sus modelos se encuentra obsoleta, mientras que la versión 2, ha quedado anticuada, por lo que su reemplazo puede ser muy complicado. Se analiza entonces desde el modelo 3 en adelante.

Como se observa en la figura “Cuadro comparativo de modelos y versiones de SBCs de Raspberry Pi” (véase **Anexo 1**), las placas que poseen todas las características requeridas son:

- Raspberry Pi 3 Model B+
- Raspberry Pi 3 Model A+
- Raspberry Pi Zero Wireless

La diferencia entre el modelo 3B+ y los modelos 3A+ y Zero W, es que la primera posee el doble de memoria RAM (1GB contra 512MB) y además cuenta con un puerto Ethernet, pero las otras poseen un costo inferior.

Si bien nuestro sistema requería conexión a internet, el puerto Ethernet no era un limitante debido a que en un futuro se podría adherir un conversor USB-Ethernet (o microUSB-Ethernet en el caso del Zero W). Además, 512MB de memoria RAM eran suficientes para la aplicación a realizar.

Si bien el modelo Zero W era ideal, debido a su bajo costo y diminuto tamaño, el mismo no se encontraba en el mercado local, por lo que se optó por utilizar el modelo **Raspberry Pi 3 Model A+** (en adelante Raspberry).

Algunas de sus características más importantes son [29] :

- Procesador de 64-bit de 4 núcleos @ 1.4GHz: Broadcom BCM2837B0.
- Memoria SRAM: 512MB LPDDR2.
- WiFi: 2.4GHz y 5GHz.
- Bluetooth: 4.2/BLE.

- Alimentación: 5VDC/2.5A.
- Puertos: HDMI, USB, CSI para cámara, DSI para display.



Figura 4. Computadora de placa única Raspberry Pi 3 A+ [29].

Como se nombró con anterioridad, se requería de un conversor ADC para manipular las señales analógicas que emiten los sensores de temperatura y de intensidad de luz, pero al no poseer el Raspberry un conversor se debió adherir uno externo.

La primer opción consistía en utilizar un conversor ADC capaz de conectarse al raspberry, este dispositivo no estaba presente en el mercado local y el costo del envío encarecía en 4 veces el valor del producto.

La segunda opción constaba de utilizar un *circuito integrado programable* (PIC acrónimo del inglés, *Programmable Integrated Circuit*), el cual, según el modelo, puede integrar un conversor ADC, pero estos requieren de un programador (circuito electrónico) para transferir el código de la computadora al PIC.

La tercer opción consistía en utilizar una placa de desarrollo de la marca Arduino, basada en microcontroladores. Estas integran un conversor ADC y el código se puede transferir a través del puerto microUSB [30] .

Se seleccionó entonces la placa de desarrollo **Arduino Nano** (en adelante Arduino).

Algunas de sus características más importantes son [31] :

- Microcontrolador de 8-bit @ 16MHz: ATmega328.
- Memoria SRAM: 2KB
- Entradas analógicas: 8
- Alimentación: 5VDC
- Puerto microUSB para alimentación y transferencia de datos.

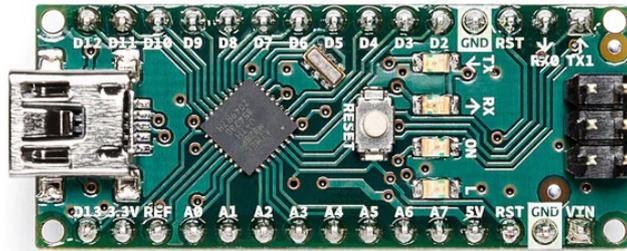


Figura 5. Placa de desarrollo Arduino Nano [31].

Una de las ventajas de utilizar el Arduino como conversor ADC, es que podía ser programado para leer el estado de los sensores y activar la alimentación, la ventilación y la iluminación nocturna, aminorando la carga de procesamiento del Raspberry.

5.2.1.2 Selección de hardware para el sensado de la iluminación

Para detectar la intensidad de luz se utilizó una fotorresistencia o *Resistencia dependiente de la luz* (LDR acrónimo del inglés, *Light Dependent Resistor*). La misma está compuesta por un semiconductor en el que al incidir luz varía su resistencia [32].

Su coeficiente de variación es alineal y negativo, indicando que su valor óhmico es mayor, cuando no incide luz sobre sus placas.

A continuación, se presenta la fórmula que describe su funcionamiento:

$$R = A * L^{\alpha} \quad (1)$$

Donde:

- R indica la resistencia, en Ohm.
- L indica la intensidad de luz, en Lux.
- A y α son constantes que depende del material y el proceso de fabricación.



Figura 6. Fotorresistencia.

5.2.1.3 Selección de hardware para el sensado de temperatura

Para el sensado de temperatura se disponía de múltiples opciones, desde termopares, sensores resistivos (termistores o detectores de temperatura resistivos), sensores analógicos electrónicos (LM35) hasta sensores digitales.

Las termocuplas son requeridas en aplicaciones donde el rango de temperaturas es muy grande, conservando la linealidad y sin dañarse, pero las mismas requieren un circuito de compensación para ser calibradas.

Por su parte, los *detectores de temperatura resistivos* (RTD acrónimo del inglés, *Resistance Temperature Detector*) presentan una muy buena linealidad y son muy precisos en la medición de temperatura.

Otro tipo de sensores son los termistores y están basados en semiconductores. Existen 2 tipos, de coeficiente positivo y negativo. Los de

coeficiente positivo son mayormente utilizados para protección, mientras que los de coeficiente negativo se utilizan para mediciones de temperatura.

Los *termistores de coeficiente negativo* (NTC acrónimo del inglés, *Negative Temperature Coefficient*) presentan la mayor precisión para rangos acotados y además son los menos costosos. Presentan como desventaja que son alineaes aunque esto puede ser solucionado con un circuito de linealización.

Los sensores analógicos LM35, son capaces de medir temperaturas desde los -55°C hasta los 150°C , para lo cual entrega como salida 10mV de tensión por cada grado Celsius medido, siendo de 0V su salida para una temperatura de 0°C .

Debido a que nuestro sistema solo debía medir 2 temperaturas límites para encender o apagar la ventilación, no se requirió de un sensor lineal, por lo que se utilizó el sensor NTC, además, el tamaño reducido del mismo era ideal para nuestro equipo.



Figura 7. Termistor NTC [33].

A continuación, se presenta la fórmula que describe su funcionamiento:

$$RT = R_0 * e^{\beta * (\frac{1}{T} - \frac{1}{T_0})} \quad (2)$$

Donde:

- R_0 indica la resistencia a la temperatura de referencia, en Ohm.
- T_0 indica la temperatura de referencia, en Kelvin.
- B indica la temperatura característica del material, en Kelvin.

- T indica la temperatura medida, en Kelvin.
- RT indica la resistencia a la temperatura T, en Ohm.

5.2.1.4 Selección de hardware para el sensado del movimiento

Como se especificó en el requerimiento RF08, se debía poder detectar movimiento a en un campo de visión horizontal de 360°, con una distancia máxima de 10 metros, para objetos con altura entre 0,2 a 2 metros. Por este motivo, se utilizaron sensores PIR, modelo Saver de la marca Optex, a continuación, se presentan sus características más importantes:

- Alimentación: 9,5 – 16V
- Cobertura: 12m x 12m con 85° de ancho.
- 78 zonas de detección.
- Altura de montaje: 1,5 a 2,4 metros (A mayor altura, mayor cobertura).
- Alarma por detección y por anti-desarme.
- Ajuste de sensibilidad (3 estados) y de estado de led indicador.
- Tiempo de alarma: 2,5 segundos
- Temperaturas de operación: -20 a 50°C



Figura 8. Sensor PIR Optex Saver.

5.2.1.5 Selección de hardware para la ventilación

Para refrigerar el sistema se utilizaron 2 ventiladores de 5V, de 40x40x10mm, uno como entrada de aire y el otro como salida.



Figura 9. Ventilador de 5V de la marca Pengda Technology.

5.2.1.6 Selección de hardware para el posicionamiento

Para el sistema de posicionamiento del equipo, se requería de un motor capaz de posicionarse en las distintas porciones de la región de detección, para la captura y grabación de videos, sin perder la ubicación entre una detección y otra. Dentro de las opciones se encontraban los Servomotores y los Motores Paso a Paso.

Si bien los servomotores son capaces de mantener el par motor sin importar la velocidad que se esté utilizando, los mismos no permanecen estáticos al momento de mantener la posición, debido a la corrección constante de errores de bucle cerrado, no permitiendo tomar fotografías o realizar videos. Los mismos están diseñados para funcionar a altas velocidades y suelen ser más utilizados en aplicaciones industriales debido a que soportan pares motor y voltajes más altos.

En contraposición, aunque en bajas velocidades los motores paso a paso tienen un buen par motor, el mismo disminuye a medida que aumenta su velocidad, pero su movimiento es más preciso, su tamaño es más compacto y son capaces de mantener su posición al no estar alimentados, por su capacidad de par de detención. En caso de aplicaciones de menor velocidad, son la solución más económica [34].

Se escogió entonces el motor paso a paso para el desarrollo del sistema de posicionamiento, siendo el mismo un NEMA 17, modelo 17HD34008-22B. Sus características son:

- Tensión nominal: 3,45V
- Corriente máxima por fase: 1,5A
- Par de retención: 3,06 Kg.cm
- Par de tope: 0,12 Kg.cm
- Ángulo por paso: 1,8°
- Pasos por vuelta: 200
- Temperatura de operación: -20 a 50°C
- Cantidad de cables: 4
- Peso: 230g
- Longitud: 34 mm
- Tamaño: 42,3 x 42,3



Figura 10. Motor Paso a Paso Nema 17 modelo 17HD34008-22B.

Para el control del motor paso a paso se requirió un driver o controlador para alimentar las bobinas del motor y que este realice el movimiento. El mismo debe entregar la corriente máxima por fase del motor quedando un sobrante del 10% para proteger el motor.

Se utilizó el driver Pololu A4988 ya que soporta corrientes de hasta 2A y micro pasos de 1/16, el mismo fue montado en un shield para disminuir la cantidad de conexiones necesarias.

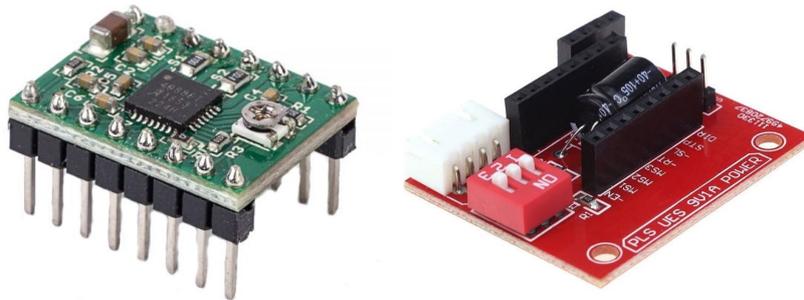


Figura 11. (a) Shield y (b) Controlador de motor paso a paso A4988 .

5.2.1.7 Selección de hardware para la iluminación nocturna

Como se indicó en el requerimiento RF04, el sistema de iluminación nocturna utilizado debía alcanzar 10 metros y no perturbar el ambiente de los animales. Por este motivo, se eligió tecnología infrarroja para iluminar la escena.

Los sensores de cámaras son capaces de detectar fotones con longitudes de onda entre 400 a 1000nm (superior a los 780nm que puede detectar el ojo humano), por este motivo, se puede observar la luz infrarroja desde una cámara [35].

Existen 2 tipos de leds infrarrojos según la longitud de onda emitida: 940nm y 850nm. Los primeros son invisibles para el ojo humano, pero las cámaras son menos sensibles a esta longitud de onda, y además, sus alcances son de la mitad que en los de 850nm.

En la siguiente figura se observa la sensibilidad relativa de los lentes de cámaras en comparación con la longitud de ondas recibida. Se puede observar como la sensibilidad para 940nm es menor que para 850nm.

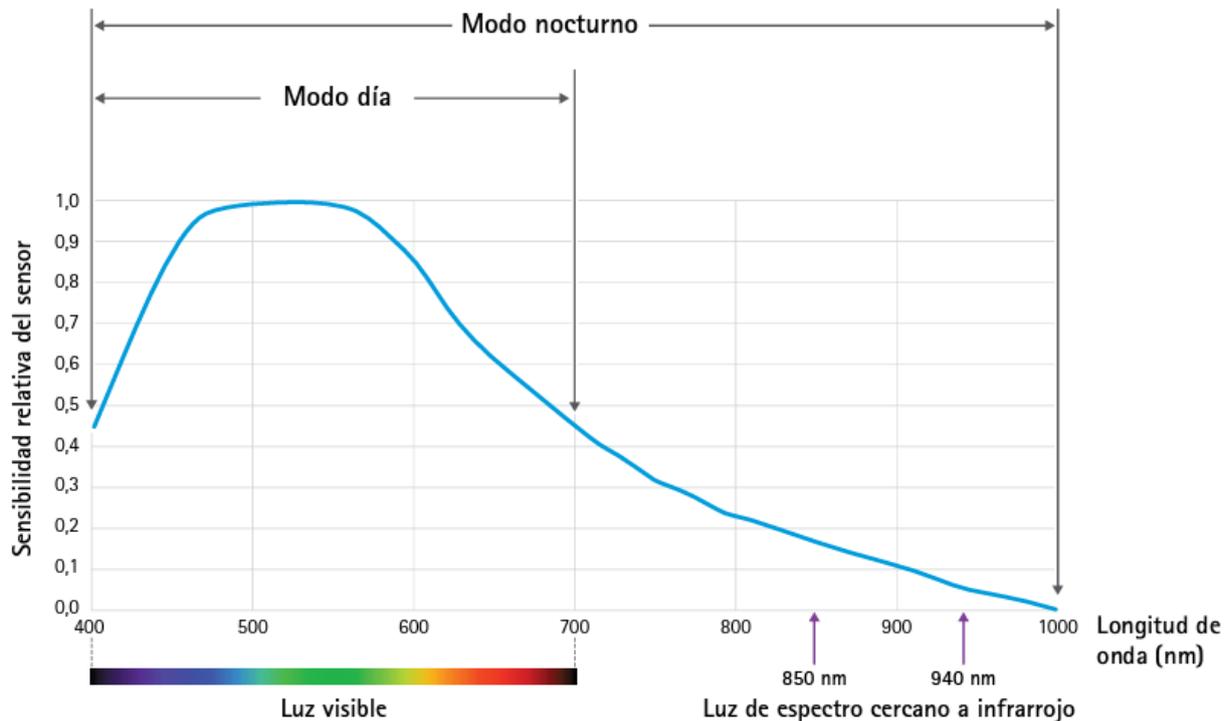


Figura 12. Sensibilidad relativa del sensor de la cámara vs longitud de onda de iluminación infrarroja [35].

Se eligieron leds infrarrojos de alta potencia de 3W, 850nm y 700mA.



Figura 13. Led High Power 3W infrarrojo de 850nm [36].

5.2.1.8 Selección de hardware para la captura de imágenes y videos

Como vimos en el apartado anterior, las cámaras digitales son capaces de detectar la luz infrarroja, las mismas integran un Filtro de Corte Infrarrojo (en adelante Filtro IR), utilizado para filtrar la luz infrarroja natural y que el color de las imágenes y videos diurnos no salga distorsionado.

Durante el día el sensor de la cámara capta la luz visible y durante la noche, al haber ausencia de luz visible, las imágenes salen oscuras. En caso de retirar el filtro IR, se observa que las imágenes diurnas toman tonalidades rosadas debido a la luz infrarroja natural del ambiente y en situación nocturna se capta mayor cantidad de luz.

Para la elección de la cámara, se debían tener en cuenta los módulos de cámara soportados por el Raspberry. Dentro de su catálogo se encontraban 5 tipos de módulos:

- Módulo V1 de 5MP.
- Módulo V1 versión NoIR (idéntico a modulo V1, pero con remoción del filtro IR)
- Modulo V2 de 8 MP.
- Módulo V2 versión NoIR (idéntico a modulo V2, pero con remoción del filtro IR)
- Módulo HQ Camera de 12,3 MP

En el caso de los módulos con filtro IR, la iluminación nocturna planteada no tendría mucho efecto, debido a que filtra gran parte de la energía emitida (una parte no es filtrada y se puede observar, como es el caso de la cámara de los celulares). Por este motivo, se debía utilizar un módulo de versión NoIR.

Para obtener imágenes de mayor calidad, se utilizó el módulo V2 versión NoIR.



Figura 14. Raspberry Pi Camera Module 2 NoIR [37].

Si bien este módulo permitía capturar imágenes nocturnas sin filtrar la luz infrarroja natural para obtener mayor luz en el ambiente, se requirió de un filtro externo que se active de día para eliminar los tonos rosados.

Por este motivo, se integró un filtro IR, como el que se muestra a continuación:



Figura 15. Filtro infrarrojo de corte [38].

5.2.1.9 Módulos conversores de tensión

Como se nombró anteriormente, el Raspberry y el Arduino se alimentan con 5V, y como se indicó en el requerimiento RF01, la batería que proporciona la alimentación para todo el sistema es de 12V. Por este motivo, se requirieron 2 módulos reductores de tensión, para regular la tensión de 12 a 5V.

Por protección se utilizó un módulo para el Raspberry, y otro para el resto del circuito.



Figura 16. Módulo conversor DC-DC Step Down.

También se necesitó un módulo elevador de tensión para alimentar el circuito serie de leds infrarrojos, los cuales requieren tensiones superiores a 12V. Se detalla su cálculo en el apartado de implementación de hardware.

5.2.1.10 Selección de hardware para el entrenamiento del modelo.

Para la realización del modelo se utilizó una computadora portátil de la marca HP modelo Victus. Sus características principales son:

- Procesador: Intel I5 11400H de 4,5 GHz
- Almacenamiento: Disco M.2 de estado sólido de 512GB
- Memoria RAM: 8 GB DDR4 de 2933 MHz
- Placa de video: Nvidia GeForce GTX 1650 de 4GB GDDR6.



Figura 17. Notebook HP Victus 16.1' GTX 1650.

Finalmente, se observa el diagrama en bloques del sistema obtenido, considerando la conexión entre cada equipo.

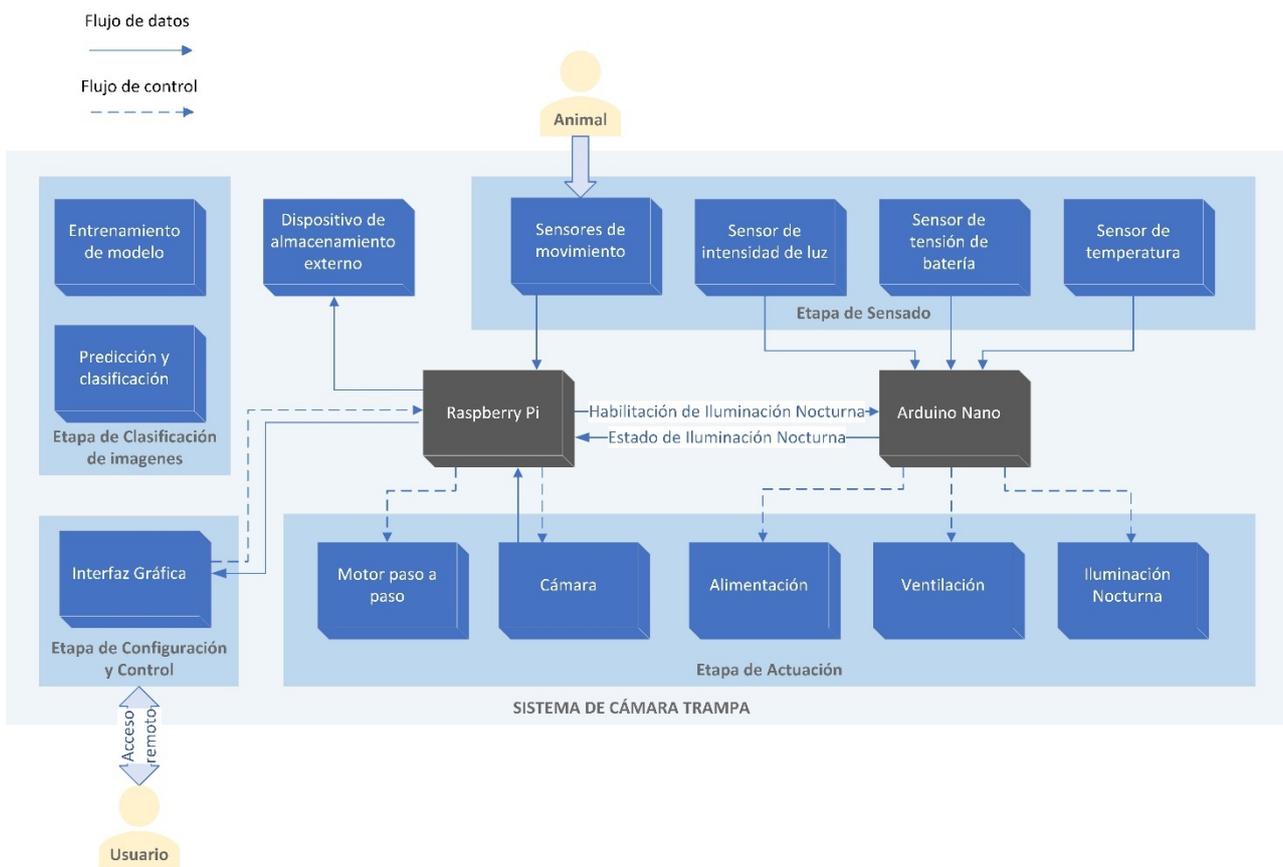


Figura 18. Diagrama en bloques final del sistema.

5.2.1.3 Selección de hardware para el sensado de voltaje.

El sensor de tensión de batería está conformado por un divisor resistivo (véase ***Sensado y Control de voltaje de batería***).

Se utilizaron resistencias de 1/2W de 5% de tolerancia.



Figura 19. Resistencia de 1/2W y 5% de tolerancia.

5.2.2 Diseño de Software

5.2.2.1 Diseño de Software de Cámara Trampa

Para realizar el diseño del software, se consideró la utilización del lenguaje Python debido a que el módulo de la cámara del Raspberry proporciona una librería denominada PiCamera, la cual consiste en una interface pura de Python, facilitando la configuración y el uso de la cámara. Además, Raspberry ya viene integrado con el lenguaje Python, permitiendo realizar secuencias de comandos en segundo plano de manera automática luego del arranque, así como también implementar una interfaz GUI para el control del hardware y software asociado [30].

Python es un lenguaje de programación de alto nivel interpretado, lo que indica que el código fuente no debe ser traducido por un compilador, en cambio, el código es traducido por un intérprete, línea por línea. Tiene por ventaja que el desarrollo del código es sencillo, pero su traducción es poco eficiente y la velocidad de ejecución es más lenta que en los lenguajes compilados.

Además, es un lenguaje orientado a objetos, con semántica dinámica y modular, permitiendo el uso de módulos para reutilizar el código. Es utilizado para el desarrollo rápido de aplicaciones y también para lenguaje de secuencias de comando. Su énfasis es la legibilidad permite reducir costos de mantenimiento, pudiendo expresar conceptos en menos líneas que en códigos como C o C++.

Otra de las cualidades de Python es su amplio uso en Machine Learning, también conocido como aprendizaje automático [39].

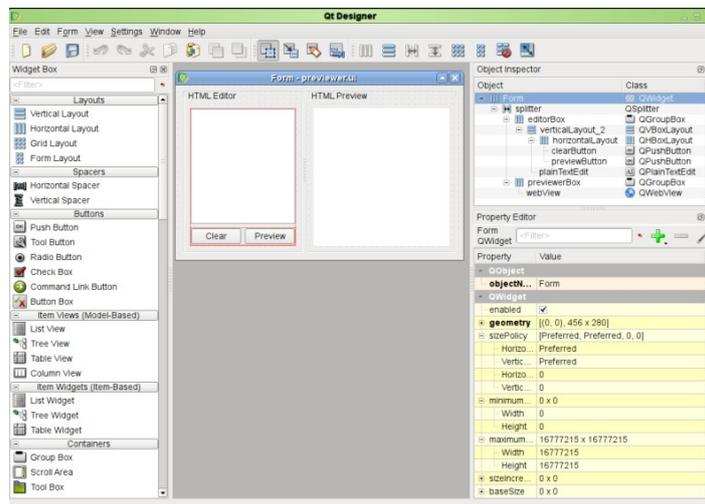


Figura 20. Lenguaje de programación Python [40].

5.2.2.2 Diseño de GUI (Interfaz Gráfica de Usuario)

Python permite el uso de múltiples de toolkits para el desarrollo de GUIs, las más conocidas son PyGTK, PyQt, Tkinter y wxPython.

Para el desarrollo de la interfaz GUI se utilizó PyQt ya que la misma permite realizar todo tipo de modificaciones del comportamiento de los widgets, y si bien su licencia no es libre (por lo que se requiere una licencia comercial, en caso de querer distribuir el código), utiliza el diseñador de GUI de QT que permite simplificar el desarrollo de la interfaz, soltando y arrastrando widgets de manera manual, así como también editar las señales de los distintos elementos [41].



Qt Designer: Diseñador de GUI de Qt [42].

5.2.2.3 Diseño de Scripts para la Catalogación de imágenes

Existen distintas herramientas para realizar modelos de aprendizaje automático, en el caso de Tensorflow o PyTorch, los mismos trabajan en niveles bajos por lo que permiten interactuar con cada parte del modelo con mayor flexibilidad, pero requieren mayor conocimiento. Por otra parte, existen otras opciones como Keras y Scikit-Learn, las cuales proveen mayor simplicidad, pero a diferencia de los primeros, carecen de características específicas.

Para los conjuntos de datos grandes, se suelen utilizar los primeros 2 ya que poseen mayores velocidades de procesamiento y rendimientos.

En este proyecto, para la codificación del script de catalogación de imágenes por visión artificial, se utilizaron los siguientes recursos:

- **Anaconda:** Es un software que provee herramientas y librerías para desarrollos científicos tales como ciencia de datos y aprendizaje automático. Integra una interfaz GUI denominada Anaconda Navigator, la cual permite acceder a los diferentes *entornos de desarrollo integrados* (IDE acrónimo del inglés, *Integrated Development Enviroment*). Existen distintos IDE según el lenguaje de programación a utilizar (Python, R, etc.), o la acción a realizar (análisis de datos, aprendizaje automático, etc.) [43].
- **Jupyter notebook:** Es uno de los IDE integrados por Anaconda. El mismo permite ejecutar bloques de código, visualizando sus resultados por separado para aumentar la legibilidad del código.
- **TensorFlow:** Es una biblioteca utilizada para el cálculo numérico a través del flujo de datos. Es utilizada para el entrenamiento de los modelos de aprendizaje automático. El mismo se puede correr en equipos que solo integren una *Unidad Central de Procesamiento* (CPU acrónimo del inglés, *Central Processing Unit*), así como también en equipos que además de CPU, integren una *Unidad Central de Procesamiento Gráfico* (GPU acrónimo del inglés, *Graphical Processing Unit*) [44].
- **Keras:** Es una librería de Python utilizada en aprendizaje profundo para la creación de capas de redes neuronales optimizando el tiempo de su programación [45].
- **OpenCV:** La biblioteca *Vision Artificial Abierta* (OpenCV acrónimo del inglés, *Open Computer Vision*), es una librería utilizada para aprendizaje

automático y visión artificial en tiempo real. La misma contiene algoritmos para distintas aplicaciones, dentro de las cuales se encuentra la clasificación de imágenes [46].

- **Numpy:** Es la librería de Python utilizada para realizar operaciones matemática a través del cálculo vectorial y matricial. La misma hace uso de matrices de N-dimensiones y es capaz de realizar cálculos elementos a elementos, operaciones matemáticas entre vectores, operaciones de algebra lineal, de transformadas y generación de números matemáticos [44] [47].
- **Scikit Learn:** Es un software de código abierto que tiene por finalidad permitir que el aprendizaje automático pueda ser utilizado por quienes no sean expertos en el área y que los costos de capacidades computacionales no sean muy elevados. Permitiéndole a investigadores de múltiples áreas, utilizar esta herramienta para mejorar sus resultados. El mismo utiliza el lenguaje Python, ampliamente aceptado por la comunidad científica [48].

5.2.3 Definición de módulos de software de Cámara Trampa

Para simplificar la lógica del programa, el mismo se subdividió en módulos.

El sistema está compuesto por 8 módulos, denominados:

- Ventana Principal de Interfaz.
- Cierre de Interfaz.
- Control.
- Inicio Script.
- Detección, captura y actualización de datos.

Al encender el equipo, se enciende el Arduino que comanda el control de sensores y actuadores. Inicialmente se testea la tensión de la batería y en caso

de que esté dentro de rangos normales (11V a 15,5V), se enciende el Raspberry, los sensores PIR, el controlador del motor paso a paso y el módulo elevador de tensión para la iluminación nocturna.

Inicialmente la iluminación nocturna se encuentra deshabilitada para ahorrar energía, debido a que no se ha iniciado el programa.

Una vez encendido el Raspberry, se lanza automáticamente la Interfaz Gráfica, la cual tiene 2 opciones: Encender automáticamente a través de un archivo de backup (en caso de existir), o esperar por el Seteo de parametros en Interfaz. Cuando exista una de las dos condiciones, se inicia el programa de la Cámara Trampa, habilita la iluminación nocturna (que se habilite no indica que se encienda, sino que en caso de escaza iluminación nocturna, se enciende y en caso contrario se mantiene apagada) y configura el sistema con los parámetros seteados en la GUI.

Luego de la configuración de parámetros, inicia la detección de movimiento y captura de imágenes y videos y prosigue con la actualización de los datos en la interfaz gráfica, así como la actualización de la imagen en tiempo real. Estos 2 módulos se mantienen en bucle infinito hasta que se cierre el programa.

En caso de que se quieran observar las capturas y grabaciones, se puede acceder mediante un botón que nos redirige a un directorio con las mismas.

A continuación, se presenta el diagrama de estados del software de cámara trampa.

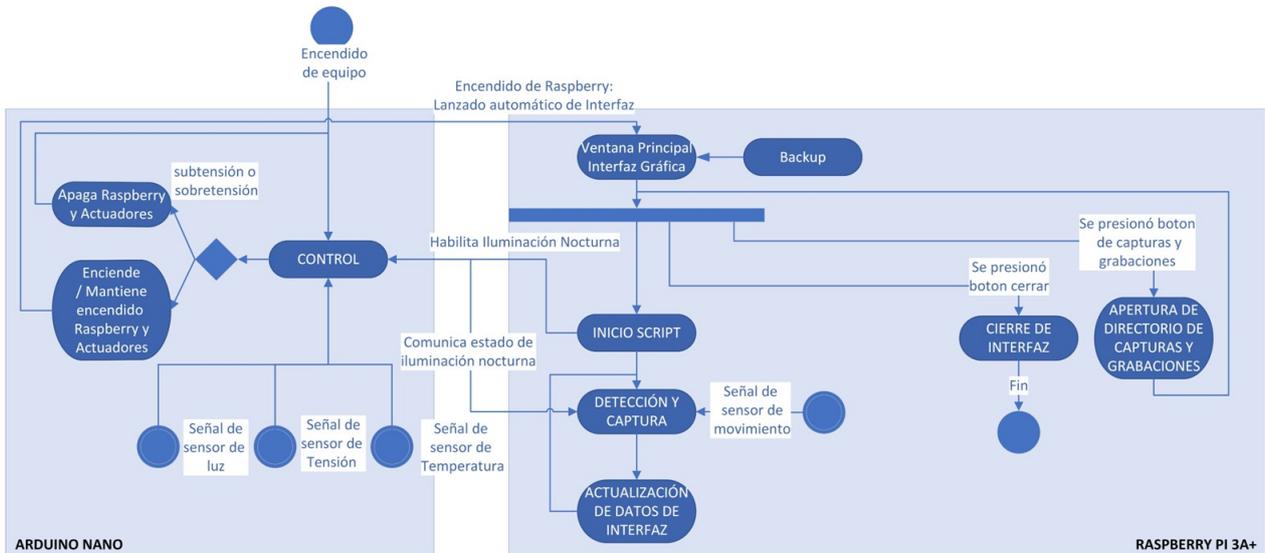


Figura 21. Diagrama de estados de Cámara Trampa.

5.2.3.1 Módulo Control

Este módulo está comandado por el Arduino, el mismo realiza las siguientes funciones:

- Leer el estado de habilitación para iluminación nocturna.
- Leer los sensores (Tensión, Temperatura e Intensidad de luz).
- Cambiar el estado de los actuadores según valores sensados.
- Cambiar el valor del puerto indicando el estado de la iluminación nocturna.

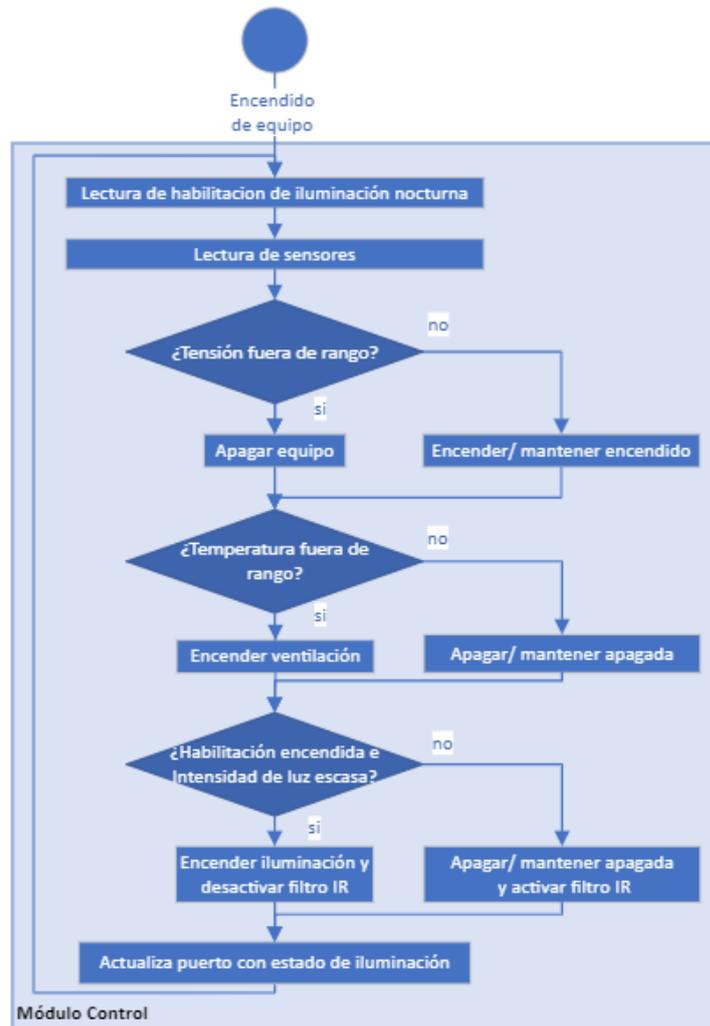


Figura 22. Diagrama de flujo de Módulo Control.

En un principio el equipo se encuentra apagado, por lo que, al encenderlo, el Arduino es alimentado y comienza a realizar su bucle infinito, iniciando el sensado de la tensión de la batería para determinar si se encuentra en un rango aceptable, y de ser el caso, enciende el Raspberry y el módulo elevador de tensión utilizado para la iluminación nocturna.

Luego, determina si se requiere conmutar o mantener el estado de la ventilación. El mismo proceso lo realiza con la iluminación nocturna, para luego escribir en el puerto GPIO, utilizado para comunicar el estado de la iluminación nocturna con el Raspberry.

Para determinar las tensiones, temperaturas e intensidades de luz mínimas y máximas se utilizan curvas de histéresis para evitar estados intermitentes.

Para el control de la tensión se utilizaron 3 tensiones como referencia, siendo la tensión mínima permitida (V_{inL}), la tensión en la que el equipo debe apagarse para proteger a la batería, de 11V; la tensión de encendido (V_{ref}), tensión de flotación de la batería, de 13.8V y la tensión máxima permitida (V_{inH}) nombrada en el requerimiento RF01, de 15.5V, para evitar quemar los sensores de movimiento PIR con los picos de tensión producidos por el cargador de la batería, siendo 16V la tensión máxima soportada por los sensores PIR.

Las tensiones V_{oH} y V_{oL} representan el estado de la salida. Si la entrada del relé de control de alimentación es V_{oH} , el relé se cierra y se alimentan las partes restantes del sistema, mientras que, si se encuentra en el estado V_{oL} , se mantienen desenergizadas.

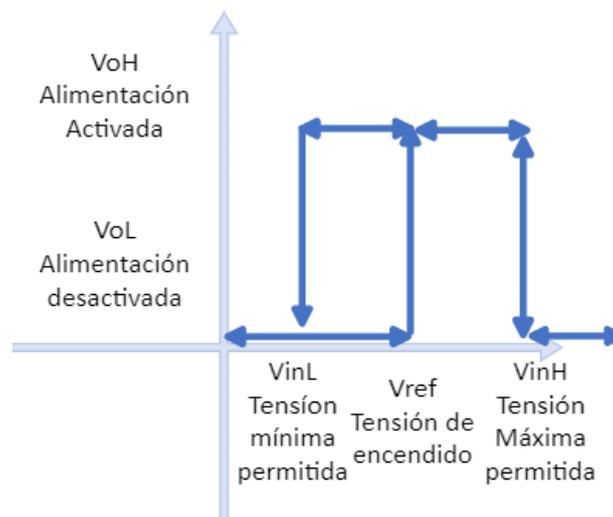


Figura 23. Curva de histéresis para control de tensión.

Este ciclo de histéresis es requerido debido a que cuando la tensión de la batería cae por debajo de 11V y se desconecta la carga, el sistema deja de demandar corriente y la tensión de la batería vuelve a aumentar, por lo que se

encendería el equipo nuevamente y la tensión volvería a caer por debajo de 11V, manteniéndose en un ciclo repetitivo que dañaría la batería.

En un principio, el equipo inicia encendido debido a que la batería puede no estar completamente cargada. En caso de que la tensión sea menor a la tensión mínima permitida, el equipo se apaga y no vuelve a encender hasta detectar una tensión de 13.8V. En caso contrario, el equipo queda encendido.

Para el control de la iluminación nocturna se utilizaron 2 tensiones como referencia, siendo la tensión de límite inferior de intensidad de luz, la tensión obtenida cuando la intensidad de luz es menor a 10 lux y la tensión superior de intensidad de luz representa la tensión obtenida cuando la intensidad de luz es igual a 25 lux. En caso de no utilizar histéresis en la iluminación nocturna, los picos de corriente generados en la conmutación podrían dañar los leds de potencia.

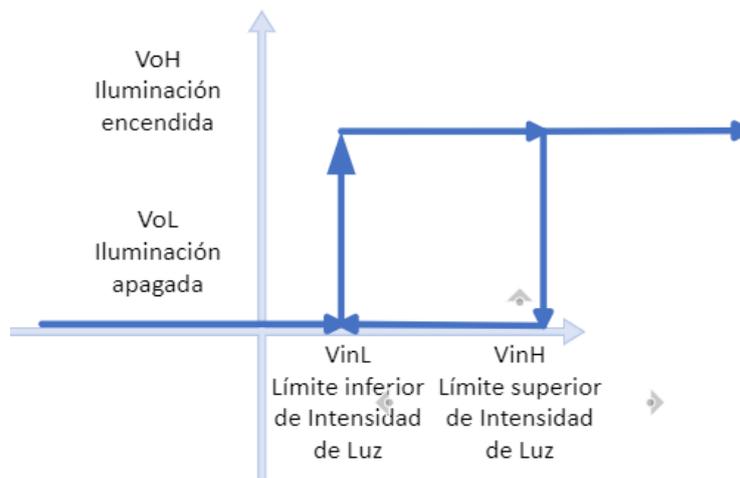


Figura 24. Curva de histéresis para control de iluminación nocturna.

Por último, en el caso del control de la temperatura se aplicó histéresis para que en un rango determinado de temperatura, la ventilación se mantenga encendida. Este rango está comprendido entre 25 y 40°C. Siendo VinL la tensión para la temperatura mínima y VinH la tensión obtenida para la temperatura máxima.

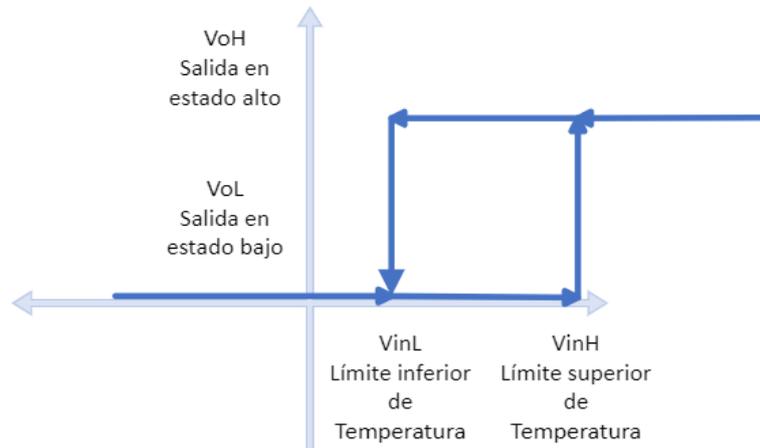


Figura 25. Curva de histéresis para control de Temperatura.

5.2.3.2 Módulo Ventana Principal de Interfaz

Para este módulo se desarrolló una interfaz gráfica utilizando PyQt5, que realiza las siguientes funciones:

- Lectura de archivo de texto de backup, para recuperar última configuración en caso de apagón.
- Seteo de parametros.
- Acceso a directorio de almacenamiento de imágenes y videos.
- Encendido y apagado del sistema por botones.
- Muestra de imagen en tiempo real.

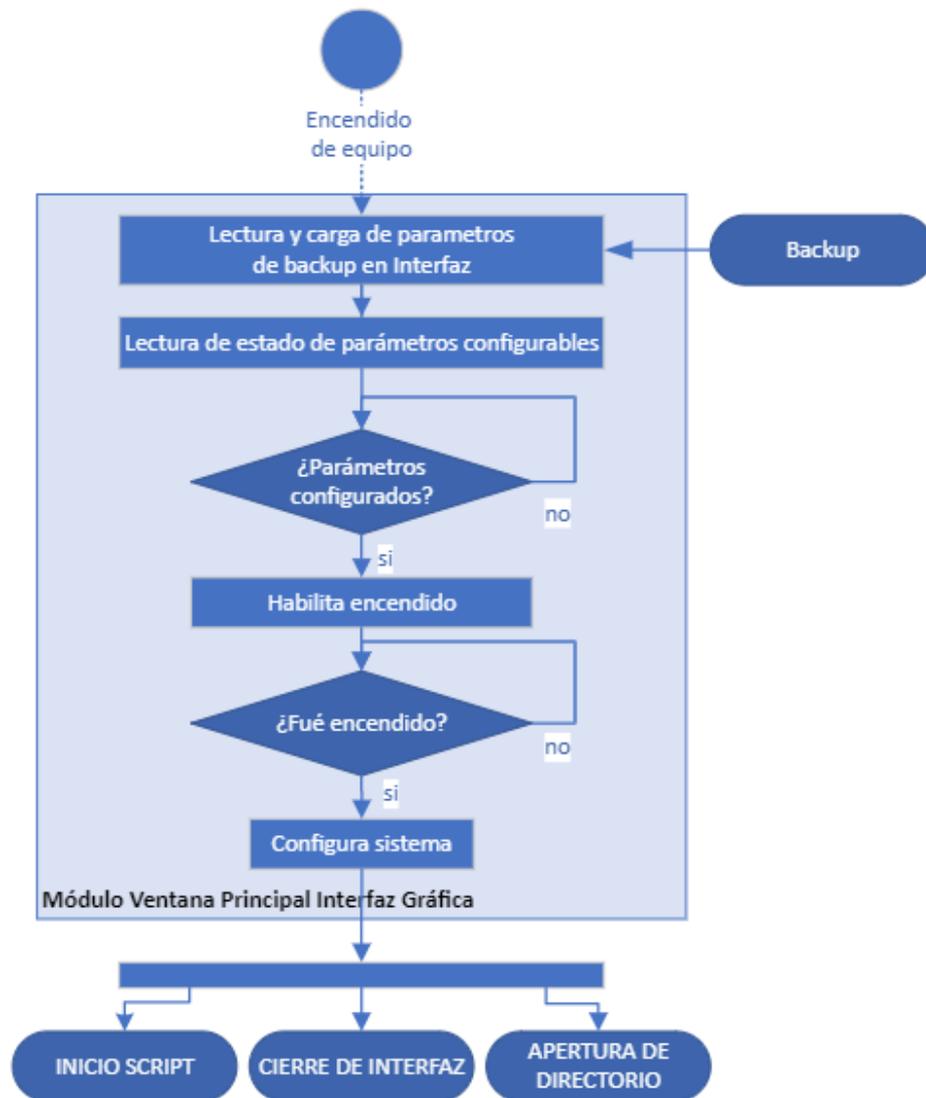


Figura 26. Diagrama de flujo de Módulo Ventana Principal de Interfaz.

Una vez habilitada la alimentación del Raspberry por parte del módulo de control, se enciende y lanza la GUI. En dicha interfaz, se encuentran distintos elementos para el ingreso de los datos de resoluciones de imagen y video, cantidad de imágenes por detección, duración del video, agregado de leyenda en imagen y video con seteo de ubicación y agregado de hora manualmente.

En caso de existir archivo de backup y que además, en la anterior configuración la fecha y hora no haya sido seteada manualmente, se cargan los parametros en la interfaz y se inicia automáticamente el programa.

En caso de que la fecha y hora haya sido seteada manualmente en la conexión previa, no se podrá iniciar el sistema ya que existe la posibilidad de que haya sido iniciado por bluetooth y no por internet, por lo que el equipo tomaría la hora del sistema, siendo la misma incorrecta ya que permanecería seteada la hora en la que se apagó el equipo. Esto sucede porque el Raspberry no posee internamente un *Reloj en Tiempo Real* (RTC acrónimo del inglés, *Real Time Clock*).

Si no existe archivo de backup, se deben setear todos los parametros correspondientes para habilitar el botón de encendido.

Una vez presionado el botón encendido, se guardan los parametros descriptos en el requerimiento RF07, en las variables correspondientes.

5.2.3.3 Módulo Inicio Script

Las funciones que provee este módulo son:

- Guardado de archivo de texto como backup con parametros seteados.
- Detección de almacenamiento externo.
- Actualización de datos en Interfaz.
- Habilitación de iluminación nocturna.
- Ejecución de rutina inicial de posición.
- Habilitación de interrupciones para sensores de movimiento.

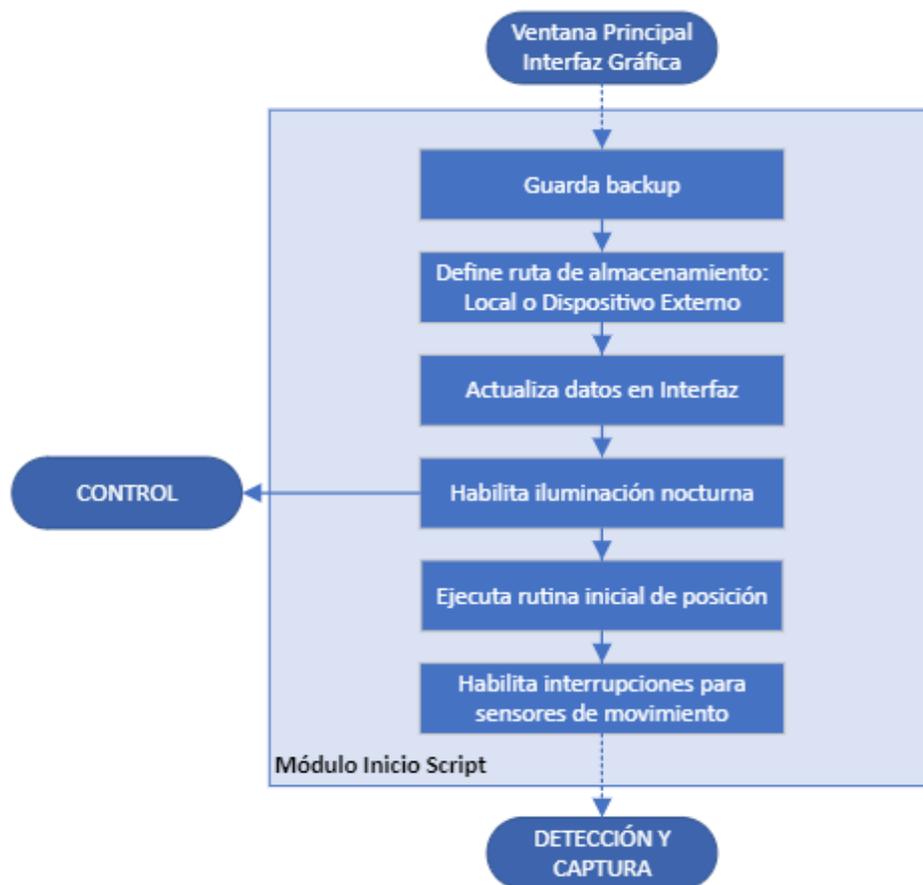


Figura 27. Diagrama de flujo de Módulo Inicio Script.

Una vez seteados los parametros en la GUI e iniciado el programa, se crea o modifica el archivo de texto de backup para que en caso de un apagón (debido a subtensión o sobretensión), se pueda reestablecer la configuración sin requerir el reseteo de parametros.

En caso de existir almacenamiento externo, se utiliza su ruta específica para la creación de los directorios nombrados en el requerimiento RF12 para el almacenamiento de imágenes y videos. Si no se encuentra ningún dispositivo de almacenamiento externo, se utiliza como almacenamiento la memoria interna del Raspberry, la cual consiste en una tarjeta SD.

Una vez determinada la ruta de almacenamiento, se actualizan los datos en la interfaz gráfica con los parametros seleccionados y se habilita la iluminación

nocturna mediante uno de los puertos GPIO (este se encuentra conectado al Arduino).

La rutina inicial de posicionamiento consiste en un procedimiento en que el motor paso a paso se dirige hacia uno de sus extremos, incidiendo contra un switch dispuesto como fin de carrera y luego retorna recorriendo 180° grados, centrándose en la posición inicial.

Una vez realizada dicha rutina, se habilitan las interrupciones para la detección de movimiento.

5.2.3.4 Módulo Detección, Captura y actualización de datos.

Las funciones que debe proveer este módulo son:

- Leer el estado de la iluminación nocturna para configurar cámara.
- Leer el estado de los sensores de movimiento.
- Enviar correo en caso de almacenamiento próximo a llenarse.
- Realizar capturas y grabación.
- Relocalizar el equipo al realizarse una detección.
- Agregar cada avistamiento en Interfaz Gráfica.
- Actualizar imagen en tiempo real en Interfaz Gráfica.

Una vez seteados todos los parametros asignados a las variables correspondientes y configurado el sistema, se procede a la detección de movimiento y captura de imágenes y videos.

Inicialmente se lee el estado de la iluminación nocturna y se configura la cámara. En condiciones de buena luz ambiente, se utiliza una velocidad de obturación de 1ms, como se indica en el requerimiento RNF02, para poder congelar objetos en movimiento, mientras que en condiciones de escasa luz

ambiente, se utiliza una velocidad de obturación menor y se configura la cámara en blanco y negro para obtener imágenes nocturnas de buen contraste.

Luego se inicia un bucle infinito en el que se leen los sensores de movimiento. Los mismos utilizan interrupciones por espera, lo que indica que al surgir una interrupción no interrumpe el sistema, sino que espera a llegar a la parte del código donde se realiza la rutina de interrupción. Es necesario realizar la lectura de esta manera, porque al ser múltiples sensores, se pueden activar a la vez, haciendo oscilar al sistema de posicionamiento.

En caso de no existir espacio libre en el dispositivo de almacenamiento, se notifica por correo electrónico y se mantiene en el bucle para continuar refrescando la imagen en tiempo real de la GUI, pero sin realizar capturas.

En caso de detección de movimiento, el sistema determina la posición del sensor del que proviene, se posiciona utilizando el motor paso a paso y captura la cantidad de imágenes seteadas en los parámetros de la GUI, luego se esperan 5 segundos y en caso de detectar otro sensor, se reposiciona, realiza nuevamente las capturas y reinicia el contador de 5 segundos. Esto sucede hasta que en un tiempo de 5 segundos no se detecte ningún objeto para luego grabar el video, con la resolución y tiempos seteados con anterioridad.

Una vez realizadas las capturas y grabaciones el equipo retorna a la posición inicial. Este procedimiento es necesario para evitar que el equipo pierda la posición ya que los motores paso a paso pueden perder pasos.

Una vez realizada la reposición se actualizan los datos de la GUI, tales como cantidad de capturas y grabaciones en los directorios utilizados, como la imagen en tiempo real.

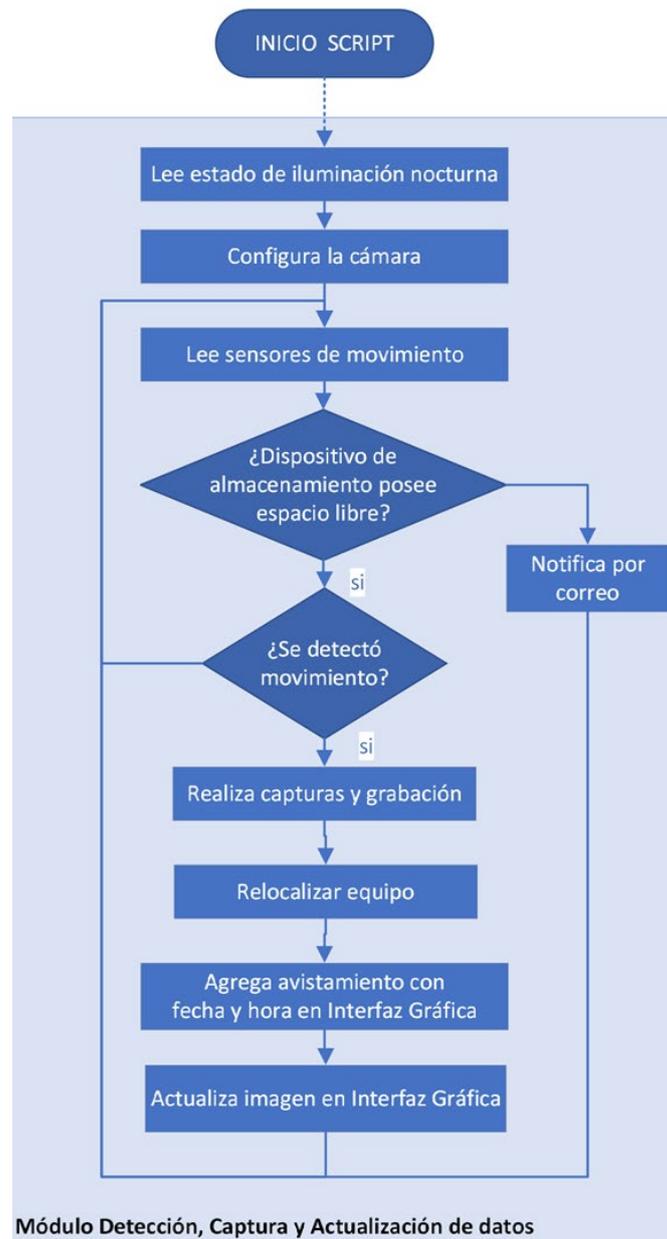


Figura 28. Diagrama de flujo de Módulo detección, captura y actualización.

5.2.3.5 Módulo Cierre de Interfaz

Este módulo provee como funciones:

- Deshabilitación de iluminación nocturna.
- Cierre de puertos de Raspberry por protección.
- Apagado de cámara.
- Borrado de backup.

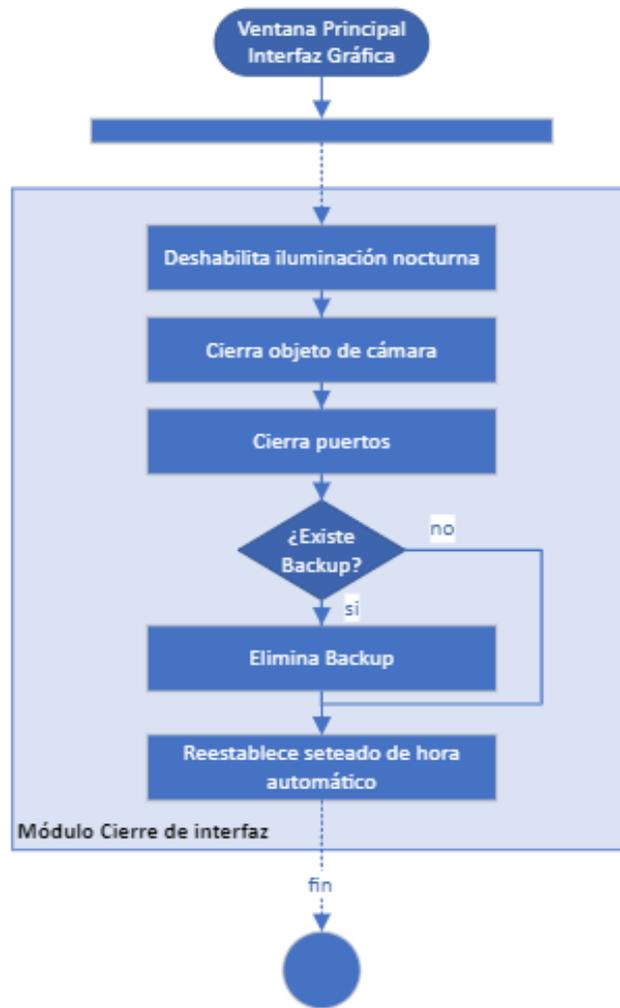


Figura 29. Diagrama de flujo de Módulo Cierre de Interfaz.

En caso de presionar el botón *cerrar* o la *cruz* en la interfaz gráfica, se deshabilita la iluminación nocturna para disminuir el consumo del equipo. Además, se cierra la cámara y los puertos GPIO, debido a que, si se vuelve a iniciar el equipo sin haberlos cerrado pueden producirse errores.

En caso de existir archivo de backup el mismo se elimina ya que este solo se utiliza en caso de cierres inesperados de la interfaz de control. Además, en caso de configurar la hora de manera manual, se reestablece el seteadado de hora automático para que la misma se vuelva a tomar desde el sistema.

5.2.4 Definición de scripts para la Catalogación de imágenes

Se realizaron 2 scripts para la catalogación de imágenes:

- Script de Entrenamiento.
- Script de Predicción y catalogación.

El funcionamiento del primer script consiste en el entrenamiento del modelo. Inicialmente se ingresan los datos por consola, tales como tipo de modelo preentrenado a utilizar (VGG16 o VGG19), tipo de set de datos (diurno o nocturno), cantidad de épocas o iteraciones, cantidad de muestras tomadas a la vez (depende de la capacidad de procesamiento de nuestro equipo), cantidad de clases de animales (en nuestro caso son 6 clases como se describe en el requerimiento RF14).

En la configuración inicial se asigna la altura y el ancho de imagen correspondiente al modelo preentrenado y los directorios desde los cuales se toman las imágenes.

En caso de que los directorios de imágenes se encuentren vacíos, se lanza una excepción por falta de datos, mientras que en caso de que se encuentren con imágenes, se generan las imágenes para el entrenamiento y la validación utilizando *Data Augmentation*, que consiste en aumentar los datos aplicando transformaciones a cada imagen, de esta manera se puede multiplicar la cantidad de datos y aumentar el rendimiento del modelo [49].

Una vez realizado el aumento de datos, se almacena el modelo y se obtienen como datos:

- Métricas de desempeño.
- Matriz de confusión.

Estos datos nos permiten determinar qué tan preciso es nuestro modelo, se analiza con más detalle en la implementación.

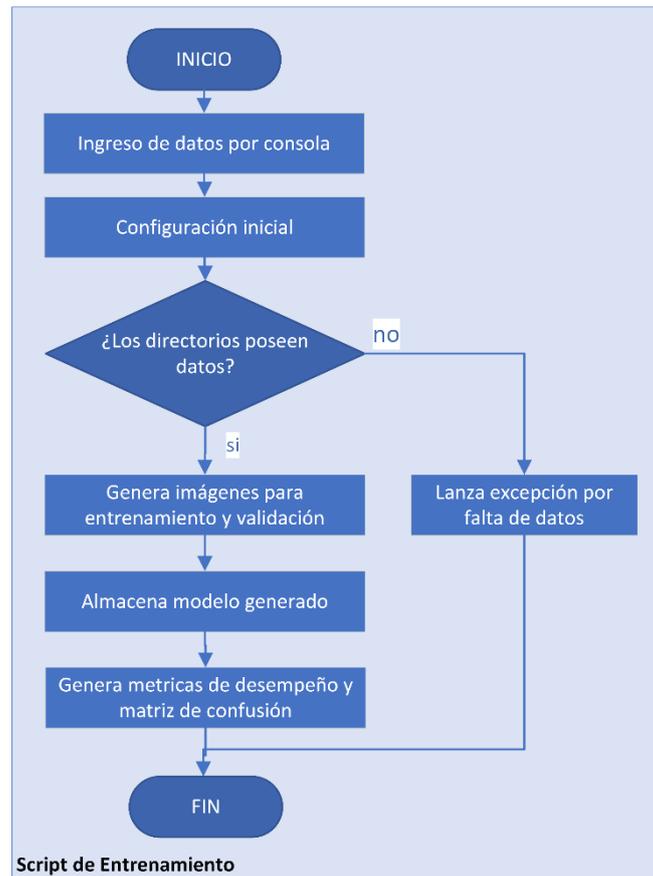


Figura 30. Diagrama de flujo de Script de entrenamiento.

El segundo programa consiste en la predicción y catalogación de imágenes. Al igual que en el primer script, se ingresa el modelo a utilizar y el tipo de set de datos a procesar y luego se realiza la configuración inicial.

En caso de no encontrar modelo generado, se lanza una excepción por falta de modelo y finaliza el programa, mientras que, si se encuentra un modelo, se leen y redimensionan las imágenes para poder ser procesadas y en caso de detectarse algún animal, se almacenan en el directorio correspondiente. En contraparte, si el porcentaje de detección es menor al requerido, se almacenan las imágenes en el directorio denominado “desconocido”.

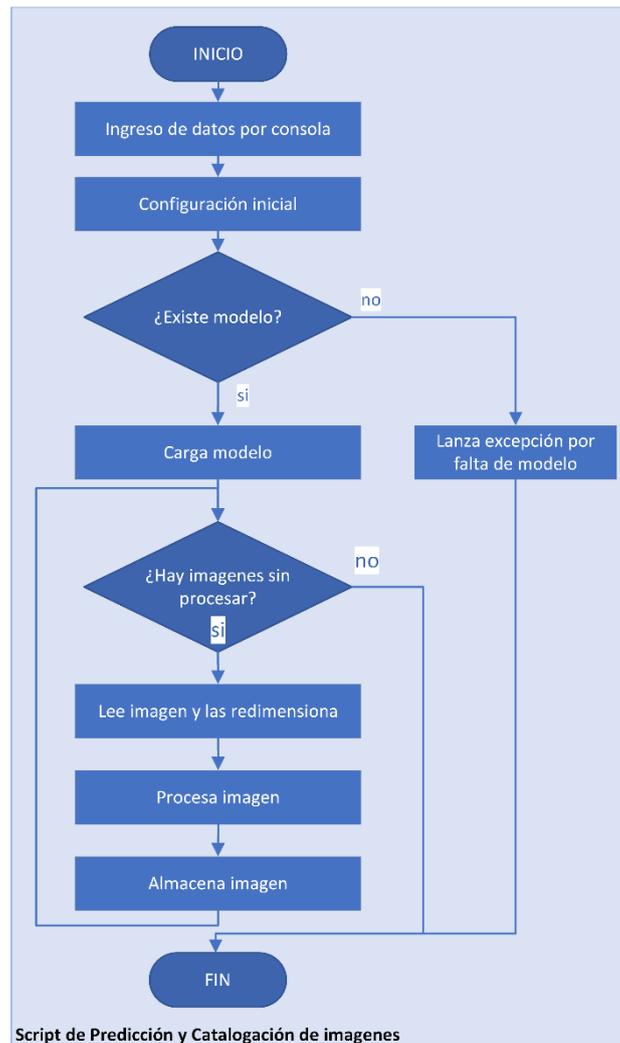


Figura 31. Diagrama de flujo de Script de Predicción y Catalogación.

5.3 Implementación

A continuación, se plasma la implementación de cada etapa del proyecto, siendo las mismas: El hardware, el software de control de la cámara trampa, los scripts utilizados para la visión artificial y la implementación del gabinete.

5.3.1 Implementación de Hardware

Para la implementación del sistema se diseñó y elaboró una placa de 1 cara para centralizar las conexiones del equipo. La misma contiene los módulos reductores de tensión, el Arduino, el driver del motor paso a paso, los módulos relé para la activación de los actuadores, las borneras necesarias para la conexión de las entradas y salidas y los fusibles para la protección de la placa y del Raspberry.

5.3.1.1 Sensado y Control de voltaje de batería

El control de voltaje se realizó mediante un divisor resistivo, conformado por una resistencia superior de 27 kOhm y una resistencia inferior de 6,8 kOhm.

Se eligieron estos valores debido a que el cargador de la batería puede generar picos de hasta 21V, los cuales pueden quemar las entradas de nuestro Arduino. Además, al utilizar los valores nombrados, la tensión obtenida es aproximadamente 5 veces menor, facilitando los cálculos.

La fórmula del divisor resistivo indica:

$$V_{out} = \frac{V_{in} \cdot R_2}{R_1 + R_2} \quad (3)$$

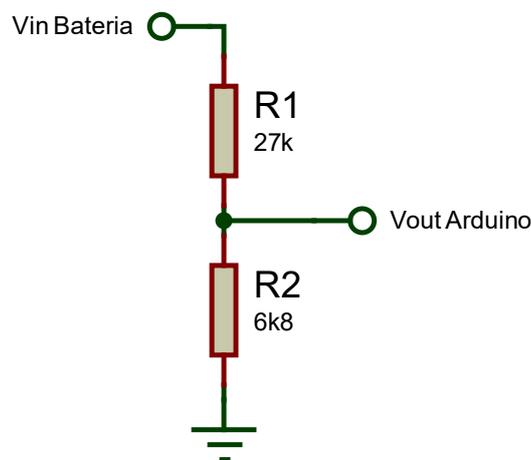


Figura 32. Sensado de tensión mediante divisor resistivo.

Las entradas analógicas del Arduino permiten hasta 5V, por lo que se necesitaría una tensión de entrada de 24,85V para llegar a la tensión máxima soportada por el Arduino, obteniendo un margen de 3,85V.

Para el control de la alimentación se utilizó un módulo relé para abrir el circuito en caso de sub o sobretensión.

5.3.1.2 Sensado de luz y Control de iluminación nocturna

Para medir la intensidad de luz también se utilizó un divisor resistivo, pero a diferencia del implementado en el control de voltaje, se dispuso el sensor LDR como resistencia R1. De esta manera, si la intensidad de luz aumenta, aumentará el voltaje de salida. De esta manera se asignaron 2 valores de tensión de cómo se describió en la curva de histéresis para iluminación nocturna.

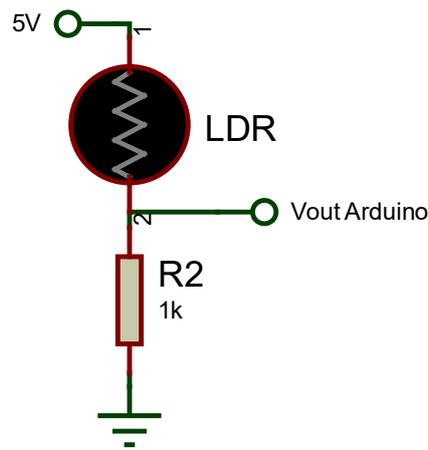


Figura 33. Sensado de intensidad de luz con circuito divisor resistivo.

A 10 lux, el LDR presenta una resistencia de 19kOhm mientras que a 25 lux presenta una resistencia de 7,5 kOhm. De este modo se obtuvieron las tensiones V_{inL} y V_{inH} del divisor resistivo:

- $V_{inL} = 0,250V$
- $V_{inH} = 0,588V$

Cuando la tensión es inferior a 0,25V se enciende la iluminación nocturna, mientras que para tensiones superiores a 0.588V se apaga. Para los valores intermedios, la iluminación nocturna se mantiene encendida.

El conversor ADC del Arduino es de 10 bits a un voltaje de operación de 5V, por lo que la resolución es de 4,88mV. Se obtiene entonces que entre V_{inH} y V_{inL} el Arduino puede detectar 69 valores distintos.

Para el control de la iluminación nocturna se utilizó un módulo relé para cerrar el circuito en caso obtener una intensidad de luz menor a 10 lux.

Además, se utilizó un módulo relé doble y un módulo simple para el control del filtro de corte infrarrojo, debido a que se requieren pulsos de distinta polaridad para la conmutación del filtro.

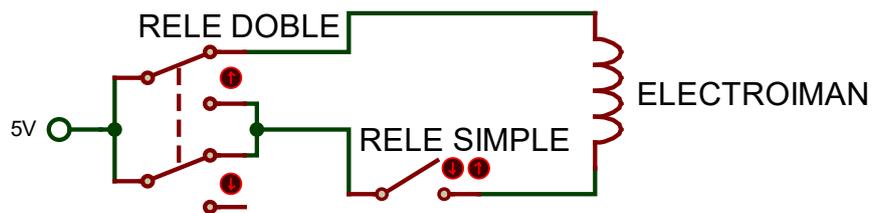


Figura 34. Circuito de control de filtro de corte infrarrojo.

El filtro IR posee un electroimán, el mismo genera un campo magnético al circular una corriente en él, atrayendo el filtro, mientras que al hacer circular la corriente en la sentido inverso, el campo magnético repelerá el filtro.

De esta manera, cuando la intensidad de la luz sea igual o mayor a 25 lux, se agrega el filtro a la cámara para evitar los tonos rosados, mientras que para intensidades menores a 10 lux, se quita el filtro para absorber la radiación infrarroja natural y la que es emitida por la iluminación nocturna.

5.3.1.3 Sensado de Temperatura y Control de ventilación

Para el sensado de temperatura se utilizó un termistor NTC de las siguientes características:

- Resistencia a temperatura de referencia (R_0) = 10 kOhm
- Temperatura característica del material (B) = 3950K

Otro dato de relevancia es la temperatura ambiente de referencia de 25°C:

- Temperatura de referencia (T_0) = 298,15K = 25°C

La ecuación de pasaje de grados Celsius a Kelvin es:

$$T [K] = T [^{\circ}C] + 273,15 \quad (4)$$

Haciendo uso de la ecuación (2), de la Resistencia a la temperatura de referencia y de la ecuación (4), se obtiene la siguiente tabla:

T[°C]	20	25	30	35	40	45	50	55	60
RT [kOhm]	12,54	10	8,04	6,51	5,30	4,35	3,59	2,98	2,49

Tabla 3. Resistencia de Termistor NTC según temperatura medida.

Obteniéndose entonces la resistencia que presenta el termistor para una determinada temperatura.

Se utilizó nuevamente un divisor resistivo, disponiendo al termistor NTC como R1, para que al aumentar la temperatura, aumente la tensión de salida, debido a que el termistor NTC posee un coeficiente negativo, por lo que al aumentar la temperatura, disminuye su resistencia.

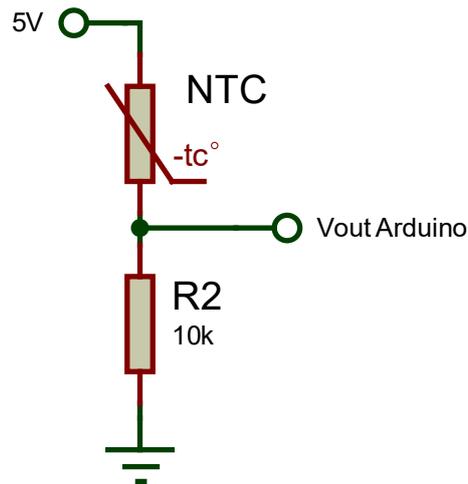


Figura 35. Sensado de temperatura mediante divisor resistivo con NTC.

Utilizando una resistencia de 10k y una tensión de alimentación de 5V, se obtuvo la siguiente tabla:

T [°C]	20	25	30	35	40	45	50	55	60
RT [kOhm]	12,54	10	8,04	6,51	5,30	4,35	3,59	2,98	2,49
Vout [V]	2,22	2,5	2,77	3,03	3,27	3,48	3,68	3,85	4

Tabla 4. Tensión de salida de divisor resistivo según temperatura medida.

Se utilizó una temperatura mínima de 25°C y una temperatura máxima de 40°C, siendo entonces:

- $V_{inL} = 2,5V$
- $V_{inH} = 3,27V$

En caso de temperaturas superiores a 40°C, se cierra el circuito de ventilación mediante un módulo relé.

5.3.1.4 Circuito de Iluminación nocturna

El circuito de iluminación nocturna consistió en una resistencia limitadora de corriente y un arreglo de leds de potencia en serie.

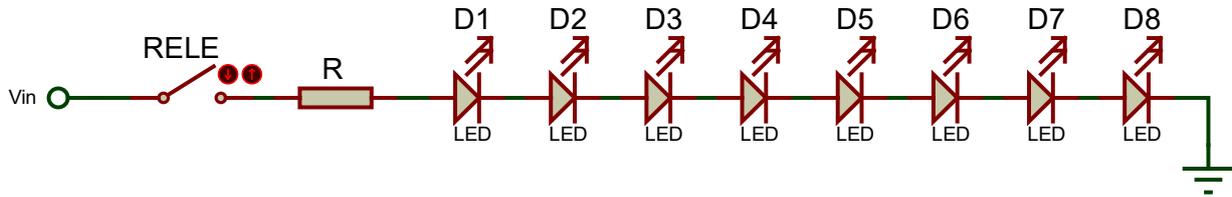


Figura 36. Circuito de iluminación nocturna.

Se utilizaron 8 leds infrarrojos de 3W y 850nm, sus características son:

- Corriente directa (I_d): 700mA
- Tensión directa (V_d): 1,85[V]

Por ley de ohm tenemos que, para una resistencia de 12 Ohm, la caída de tensión es:

$$V_R = R * I = 12 \text{ Ohm} * 0,7A = 8,4V$$

La caída de tensión de cada led funcionando con una corriente de 700mA es de 2V, por lo que se requiere una tensión de 23,2V para alimentar todo el circuito.

$$V_S = V_R * V_{dmax} * n = 8,4V * 1,85V * 8 = 23,2V$$

La corriente disipada por la resistencia a dicha corriente es:

$$P_R = V_R * I = 8,4V * 0,7A = 5,88W$$

Se precisó entonces de una resistencia comercial de 12 Ohm y 10W.

La potencia de los leds a 1,85V es:

$$P_{led} = V_{dmax} * I = 1,85V * 0,7A = 1,295W$$

Obteniéndose como potencia total de todo el arreglo:

$$P_{serie} = P_{led} * n = 1,295A * 8 = 10,36W$$

Para alimentar el sistema se utilizó el módulo conversor de continua, elevador de tensión, que soporta una tensión de salida máxima de 35V y una corriente máxima de 3A.

Por último, el módulo relé es controlado por el Arduino. Al obtener una intensidad de luz menor a 10 lux, cierra el circuito y se enciende la iluminación nocturna.

5.3.1.5 Sensado de Movimiento

Los sensores PIR utilizan lógica normalmente cerrada, para que, en caso de corte de cables, el circuito se abra y se active la alarma.

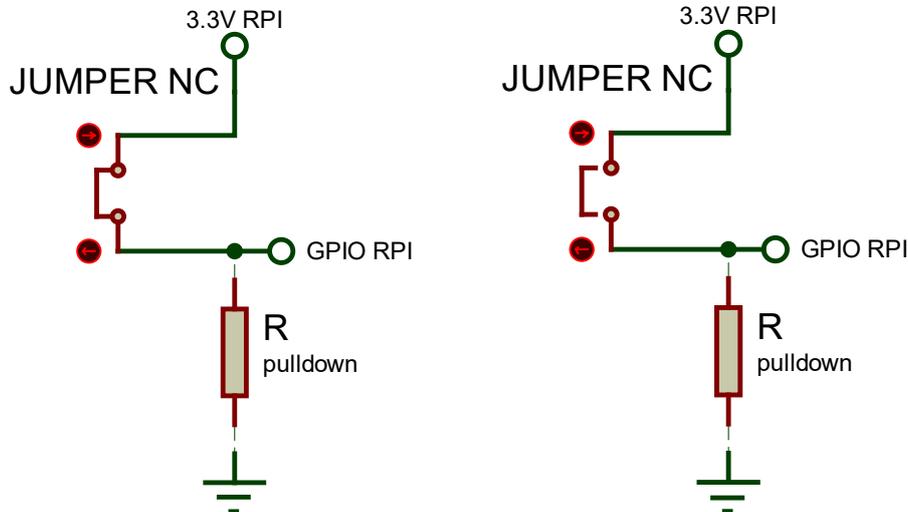


Figura 37. (a) Circuito de Alarma desactivada y (b) activada sensores PIR.

Por este motivo, para la detección de movimiento se conectó uno de los bornes de alarma de cada sensor a la tensión de 3,3V de Raspberry, y con 4 puertos GPIO se conectó el otro borne de la alarma. Para simplificar el cableado se utilizaron las resistencias de *pull down* del Raspberry.

Los sensores PIR son alimentados con 12V, por lo que al cerrarse el circuito de control de voltaje de batería, se conectan directos a la misma.

En el manual del sensor PIR de la marca Optex podemos observar su cobertura:

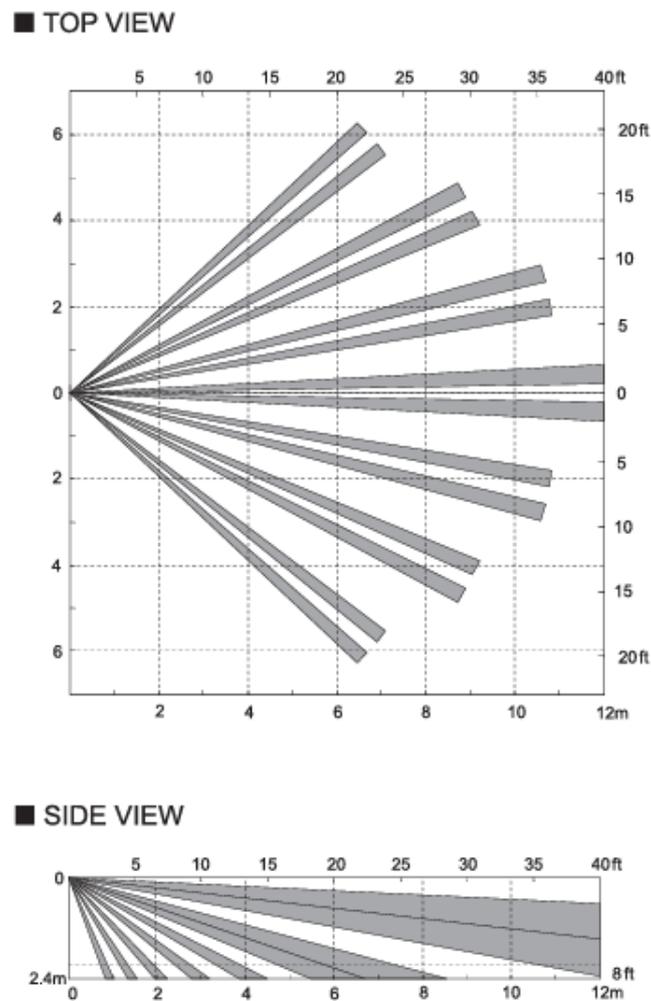


Figura 38. (a) Vista Superior y (b) Vista de Lado de cobertura [50].

Para obtener una cobertura de 10 metros, se calculó el siguiente ángulo de detección:

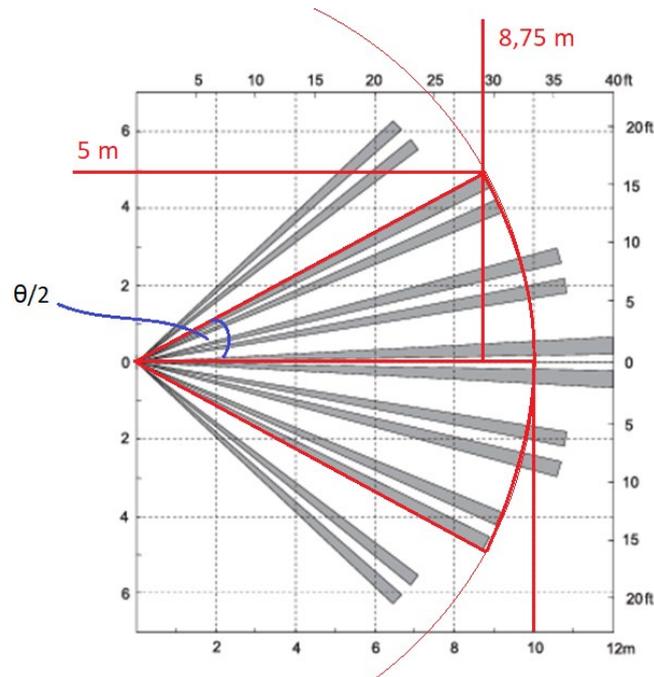


Figura 39. Ángulo de detección de sensor PIR.

A través de la razón trigonométrica de la tangente, tenemos que:

$$\tan\left(\frac{\theta}{2}\right) = \frac{5 \text{ m}}{8,75 \text{ m}}$$

Siendo el valor del ángulo de cobertura:

$$\theta = 2 * \text{atan}\left(\frac{5 \text{ m}}{8,75 \text{ m}}\right) = 59,49^\circ \approx 59.5^\circ$$

Para obtener la mayor cobertura posible en los 360°, se dispusieron los sensores a 45° de la posición inicial, con una separación entre sensor y sensor de 90°.

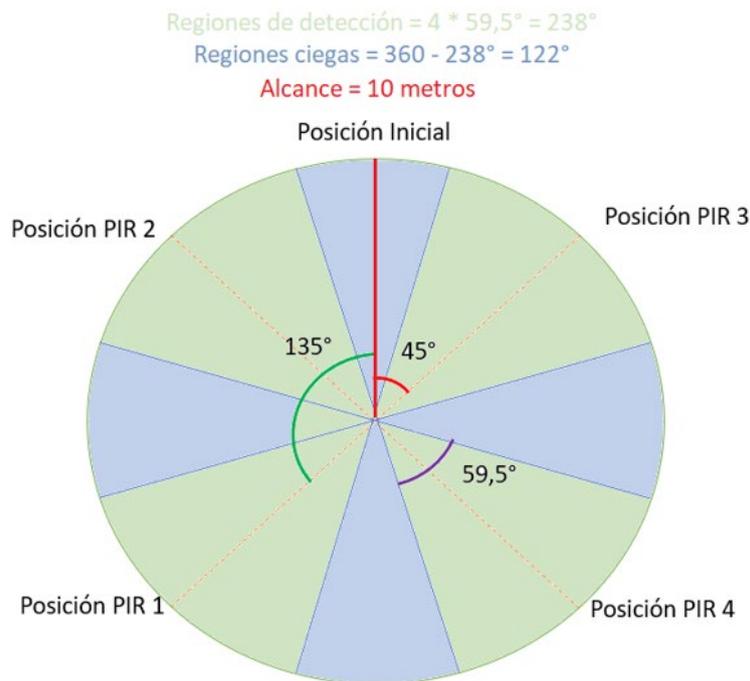


Figura 40. Ubicación de sensores, regiones de detección y cobertura.

5.3.1.6 Control de motor

Para el control del motor se utilizó el driver Pololu A4988. Inicialmente se ajustó la corriente máxima de protección mediante una medición indirecta haciendo uso del preset dispuesto en el controlador y el valor resistivo de una resistencia de montaje superficial, denominada R_s , siendo su valor resistivo de 0,1 Ohm.

La corriente máxima del controlador está dada por:

$$I_{max} = \frac{V_{ref}}{8 * R_s} \quad (5)$$

Siendo V_{ref} la diferencia de potencial entre la carcasa del preset y GND.

Conociendo la corriente máxima de nuestro motor (1,5A), se dejó un margen de 25% para proteger el motor y evitar sobrecalentamiento. Por lo que nuestra corriente máxima es de 1,125 A, obteniendo de la ecuación 5 una tensión de referencia de 0,9V.

En el circuito propuesto, el módulo se conectó mediante las entradas VMOT y GND a la batería, la misma suministra la corriente necesaria para mover el motor a través de los pines 1A, 2A, 1B y 2B; a través de los pines VDD y GND se alimenta la etapa de control del driver 4988; Los pines Reset y Sleep van cortocircuitados para inhibir sus funciones; los pines MS1, MS2 y MS3 son utilizados para realizar micro pasos por lo que no se conectaron y para el control de dirección y de paso, se conectaron los pines STP y DIR al Raspberry para controlar la posición del motor.

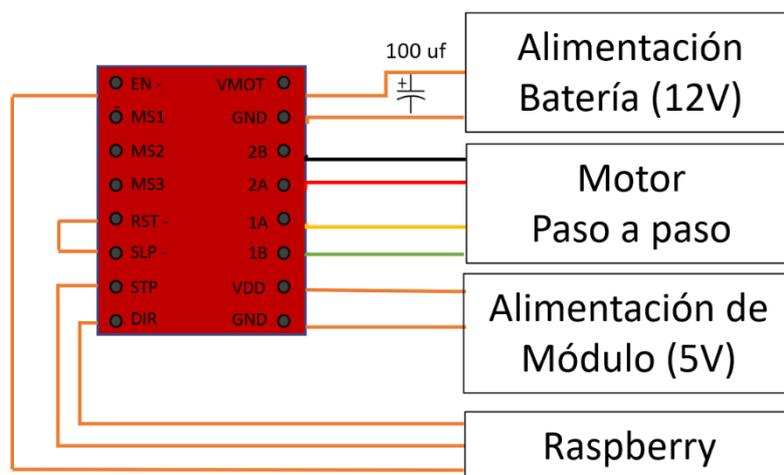


Figura 41. Circuito de control de motor paso a paso.

Para el switch de fin de carrera se utilizó la siguiente conexión:

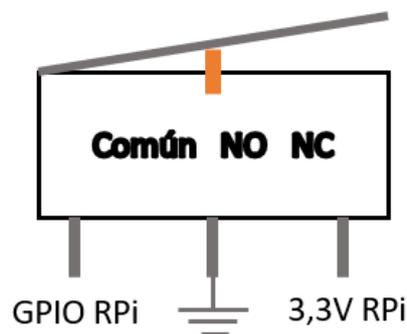


Figura 42. Conexión de switch de fin de carrera.

5.3.1.7 Circuito final

Para la explicación del esquema circuital completo (véase **Anexo 2**), se utiliza un esquema circuital simplificado, presentado en la siguiente figura, el cual se encuentra dividido en bloques.

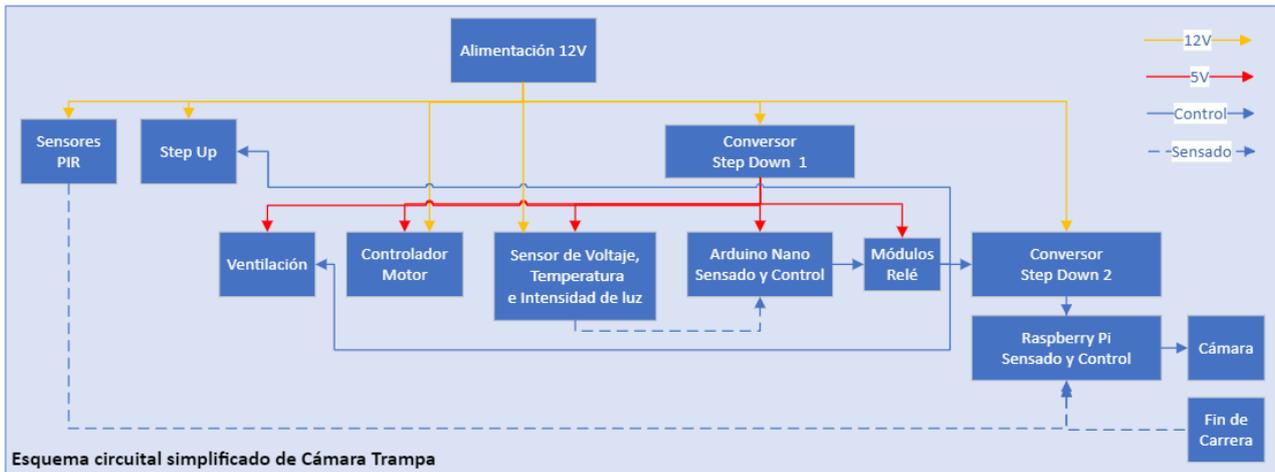


Figura 43. Esquema circuital simplificado de Cámara Trampa.

Bloque Alimentación 12V:

Este bloque está compuesto por la batería utilizada para alimentar el sistema, el fusible de protección del sistema completo, de 5A, y un switch de encendido/apagado.

Este bloque alimenta a los 4 sensores PIR, al convertor Step Up de la iluminación nocturna, y al controlador del motor A4988.

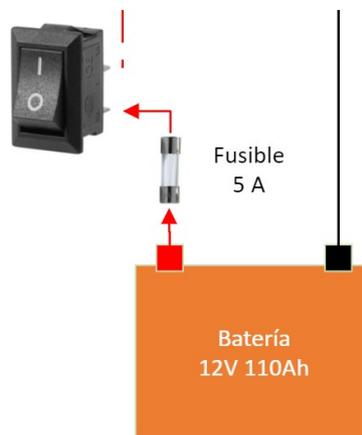


Figura 44. Bloque de alimentación de circuito final.

Bloque conversor Step Down 1:

El mismo está compuesto por el conversor DC-DC Step Down, utilizado para alimentar el Arduino Nano, los sensores (voltaje, temperatura e intensidad de luz), los relés (para la activación de la alimentación del Raspberry, la ventilación, la iluminación nocturna, el filtro de corte infrarrojo), los ventiladores y el circuito de control del Controlador de motor A4988, este último requiere de 5 y 12V para su funcionamiento.

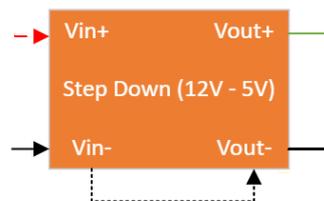


Figura 45. Bloque de conversor Step Down de Arduino y módulos relé.

Bloque Arduino Nano – Sensado y Control:

El Arduino nano se encuentra alimentado por el Conversor Step Down 1. El mismo sensa las salidas de los sensores de voltaje, temperatura e intensidad de luz y controla los relés. En caso de obtener una tensión en el rango establecido, se conmuta el relé y alimenta el resto del circuito, compuesto por los sensores PIR, el Conversor DC-DC Step Down 2 para la alimentación del Raspberry y el Conversor DC-DC Step Up para la iluminación nocturna.

Del mismo modo, al obtener valores de temperatura o intensidad de luz fuera del rango permitido, se conmuta la salida de los relés establecidos para la activación de sus actuadores.

Los relés controlados por el Arduino nano son:

- Módulo Relé de voltaje para conmutar la alimentación.
- Módulo Relé de pulso para conmutar filtro IR.
- Módulo Relé de cambio de polaridad para conmutar filtro IR.
- Módulo Relé para conmutar iluminación Nocturna.
- Módulo Relé para conmutar ventilación.

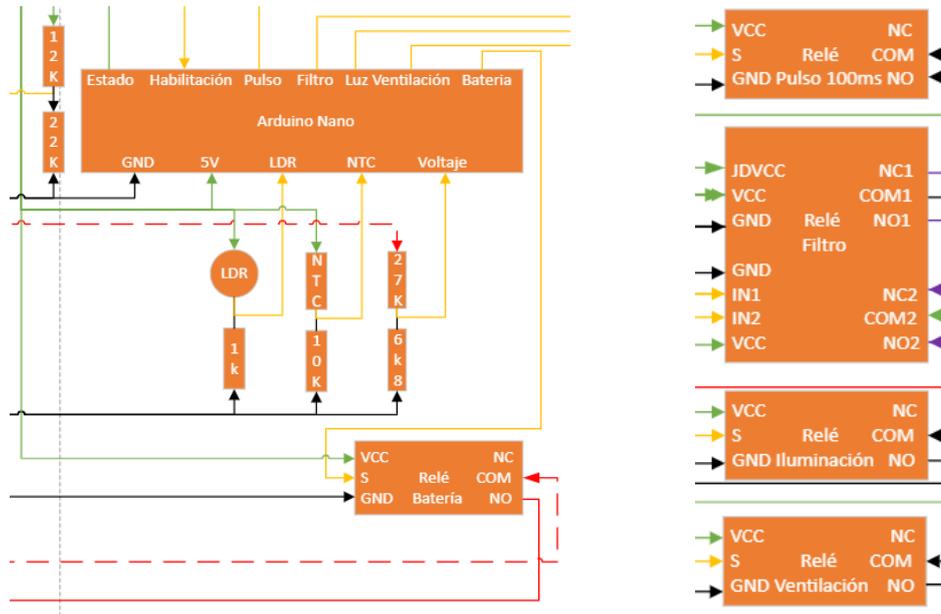


Figura 46. Bloque Arduino Nano – Sensado y control.

Bloque convertor Step Down 2:

Este bloque está compuesto por el Convertor DC-DC Step Down para alimentar el Raspberry. Además integra un fusible de 2A, debido a que al energizar al Raspberry desde sus pines GPIO, se pierde la protección del fusible integrado, el cual se encuentra conectado al puerto microUSB de alimentación.

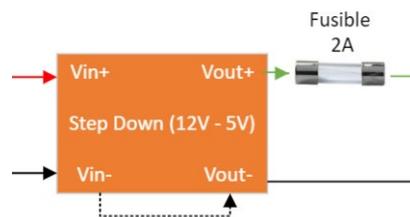


Figura 47. Bloque de convertor Step Down 2 para alimentar Raspberry.

Bloque Raspberry – Sensado y Control

El Raspberry es el encargado de controlar el módulo de cámara y el controlador del motor A4988, así como también, de sensar el estado del switch de fin de carrera y de los 4 sensores de movimiento PIR.

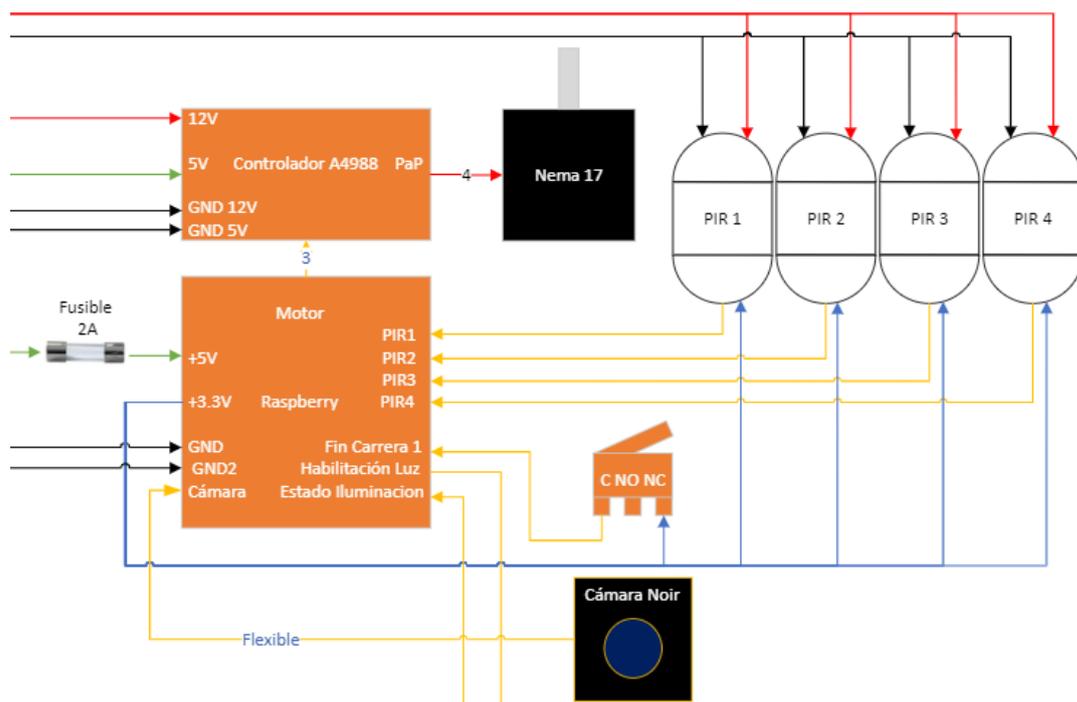


Figura 48. Bloque Raspberry – Sensado y control.

El fin de carrera está conectado directo al Raspberry debido a que no consume corriente. Además, se utilizaron 2 puertos GPIO para la comunicación de los estados con el Arduino. Este último utiliza tecnología TTL, por lo que reconoce señales superiores a 2,4V como un valor lógico alto, y considerando que los puertos de Raspberry utilizan 3,3V, la detección de estados es efectiva.

J8:		Fusible		Alimentación regulador 5V	
Señal PIRs	3V3	(1)	(2)	5V	Fan RPI
No usar como entradas	GPIO2	(3)	(4)	5V	
	GPIO3	(5)	(6)	GND	
	GPIO4	(7)	(8)	GPIO14	
	GND	(9)	(10)	GPIO15	
	GPIO17	(11)	(12)	GPIO18	
	GPIO27	(13)	(14)	GND	
Fin de Carrera 1	GPIO22	(15)	(16)	GPIO23	
	3V3	(17)	(18)	GPIO24	
	GPIO10	(19)	(20)	GND	
	GPIO9	(21)	(22)	GPIO25	
	GPIO11	(23)	(24)	GPIO8	
	GND	(25)	(26)	GPIO7	
	GPIO0	(27)	(28)	GPIO1	
	GPIO5	(29)	(30)	GND	
PIR4	GPIO6	(31)	(32)	GPIO12	Estado Iluminación
PIR3	GPIO13	(33)	(34)	GND	GND Nano
PIR2	GPIO19	(35)	(36)	GPIO16	Habilitacion Luz Nano
PIR1	GPIO26	(37)	(38)	GPIO20	Direccion PaP
Enable PaP	GND	(39)	(40)	GPIO21	Step PaP

Figura 49. Conexión de puertos GPIO de Raspberry.

5.3.1.8 Placa PCB

Para obtener mayor organización y reducir el cableado se realizó una *placa de circuito impreso* (PCB acrónimo del inglés, *Printed Circuit Board*) utilizando KiCAD.

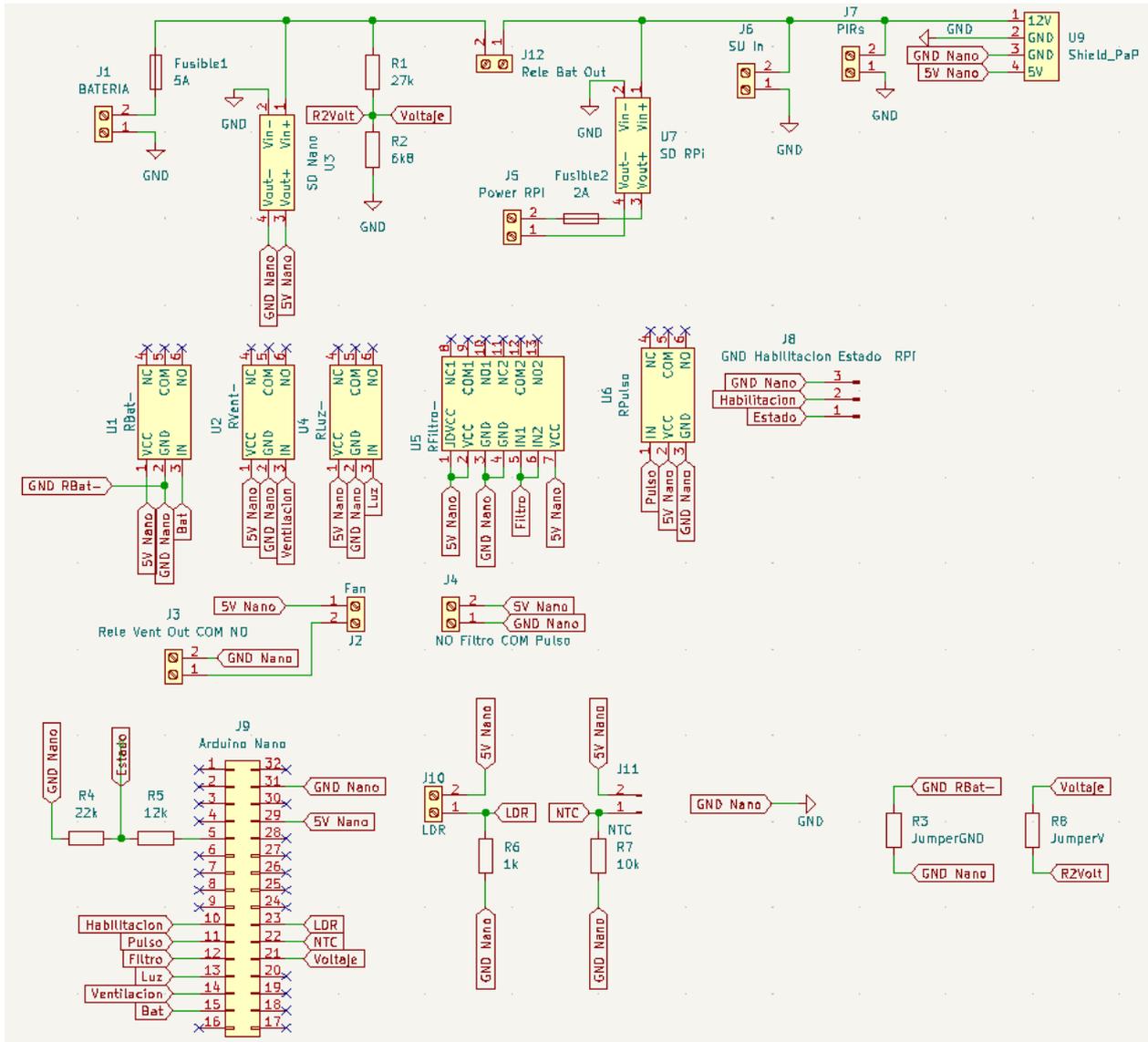


Figura 50. Diagrama esquemático de conexiones de cámara trampa.

Se obtuvo el siguiente diseño para la placa PCB:

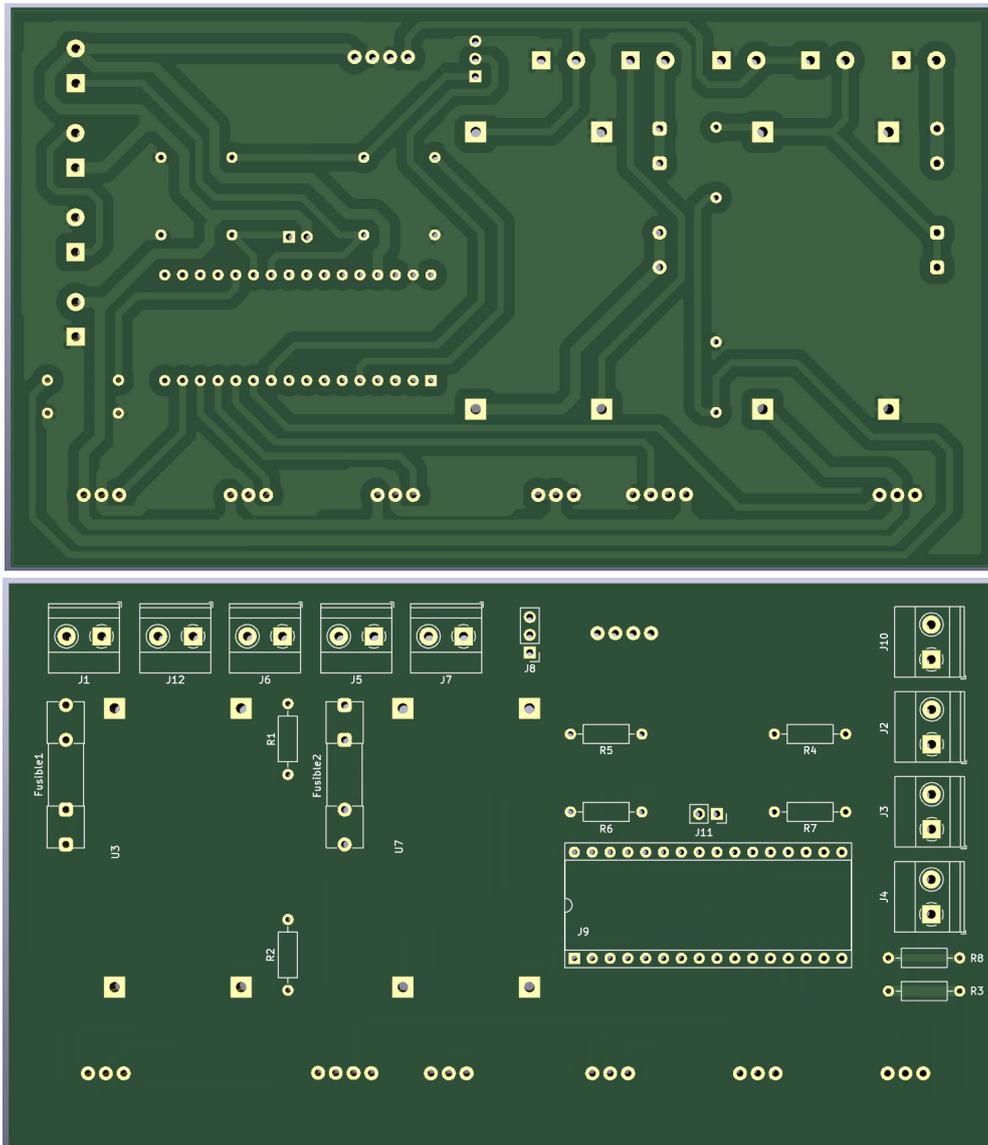


Figura 51. (a) Vista superior y (b) inferior de diseño Placa PCB.

Una vez quemada la placa y soldado los componentes se obtuvieron los siguientes resultados:

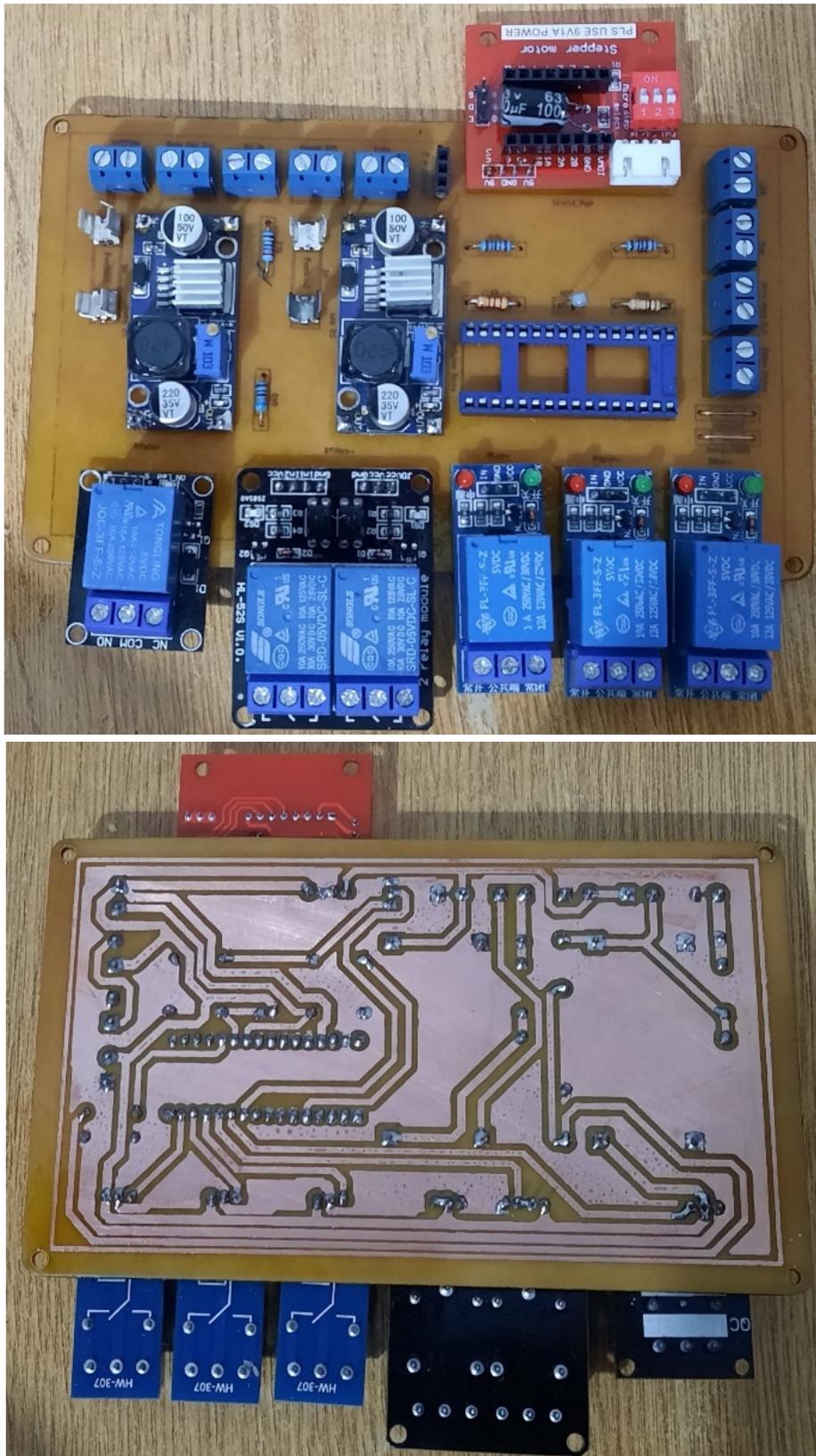


Figura 52. (a) Vista superior y (b) inferior de Placa PCB.

5.3.2 Implementación de Software

5.3.2.1 Implementación de software de Cámara Trampa

Siguiendo el diseño de diagrama de estados de cámara trampa y del requerimiento RF07, se realizó la siguiente interfaz gráfica.

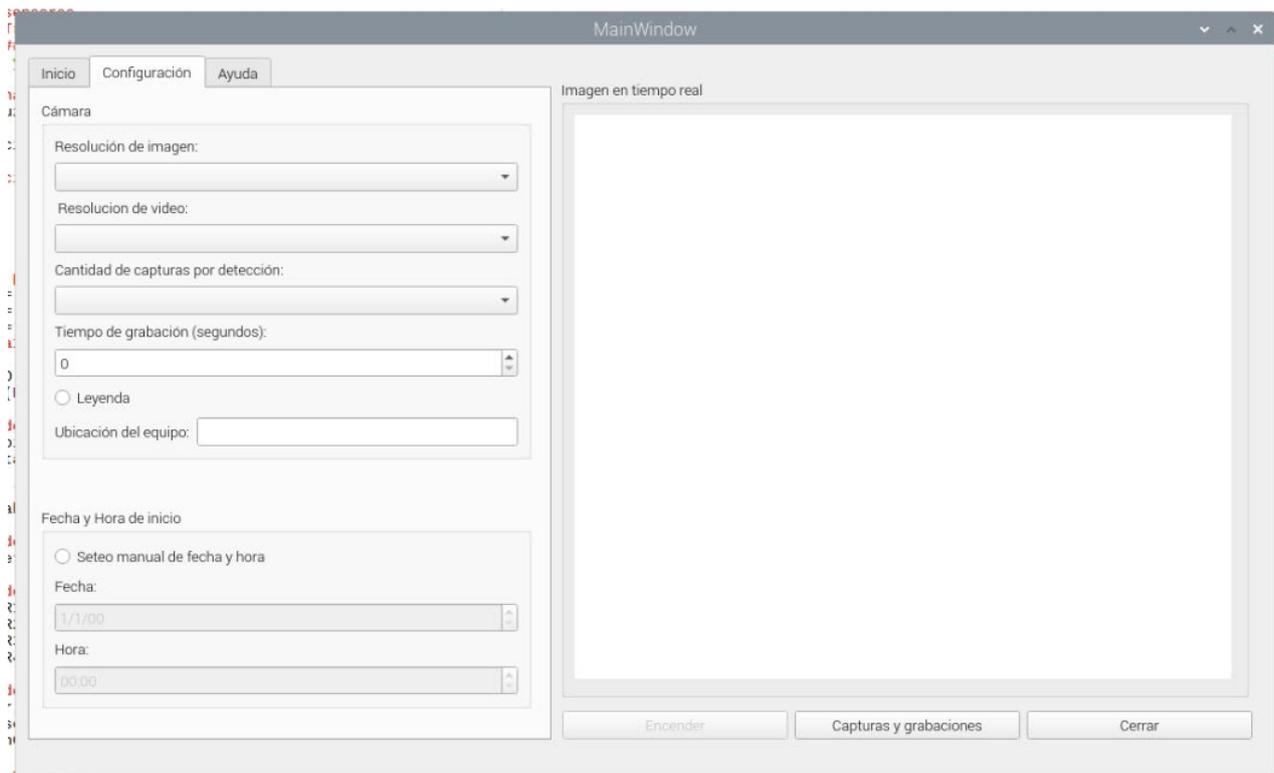


Figura 53. Interfaz Gráfica de Cámara Trampa.

En la pestaña de Ayuda, se agregaron las instrucciones de uso de la interfaz. Si bien sería ideal poder cambiar las resoluciones, sin importar que el sistema este encendido, la librería PiCamera utilizada para el control de la cámara no permite el cierre y reapertura de esta (luego de la reconfiguración), además, la cámara es utilizada por 2 librerías, PiCamera y OpenCV, por lo que un cambio de resolución genera errores debido a que no se vacía el buffer de la cámara.

La librería OpenCV es utilizada para la muestra de imagen en tiempo real a través del label (cuadro en blanco de la interfaz gráfica de la Figura 51).

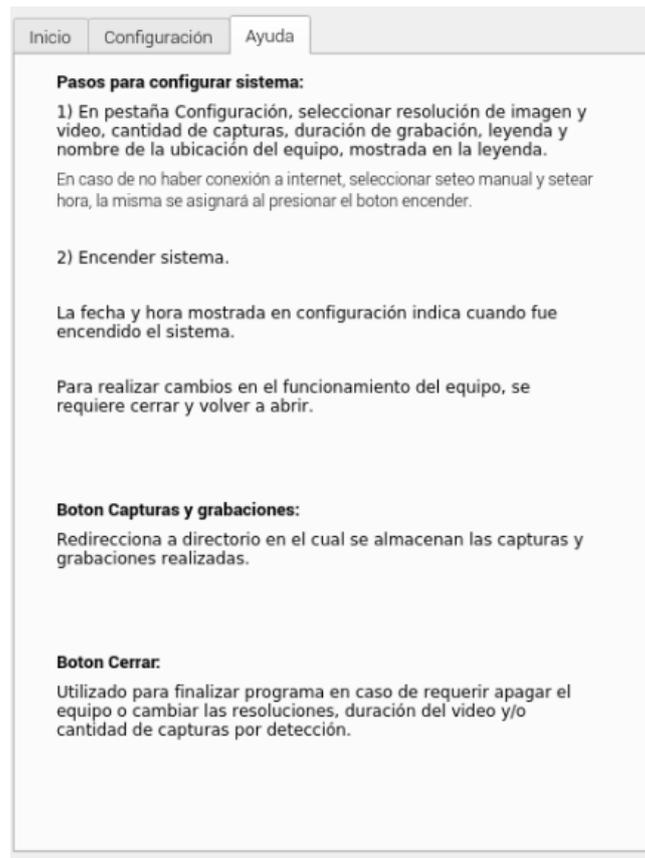


Figura 54. Sección de ayuda de Interfaz Gráfica.

En la siguiente figura se puede observar que al encender el programa, se actualiza el label con la imagen en tiempo real.

Cada vez que se realiza una detección, la misma es cargada según su fecha y hora de avistamiento. Al mismo tiempo, se contabilizan las capturas y grabaciones diurnas y nocturnas en los directorios correspondientes.

Ademas, la interfaz integra una sección de configuraciones establecidas en la que se muestran: la resolución de imagen y video, la duración de video, la cantidad de capturas a realizar por detección y la ubicación del equipo.

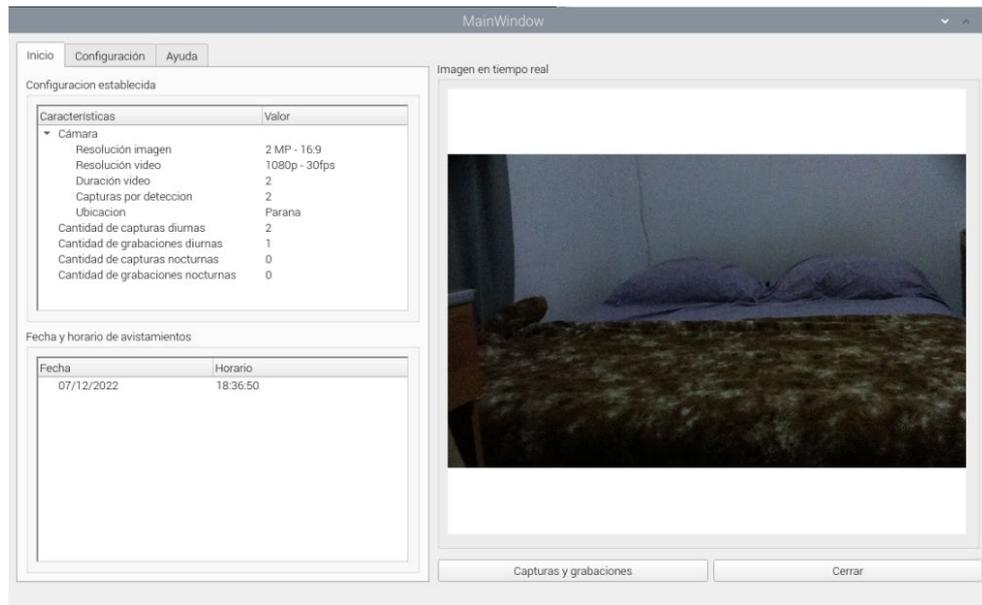


Figura 55. GUI de Cámara Trampa con visión nocturna desactivada.

En caso de que la intensidad de luz sea menor a la establecida, la cámara se reconfigura y se enciende la iluminación nocturna, el resultado es el siguiente:

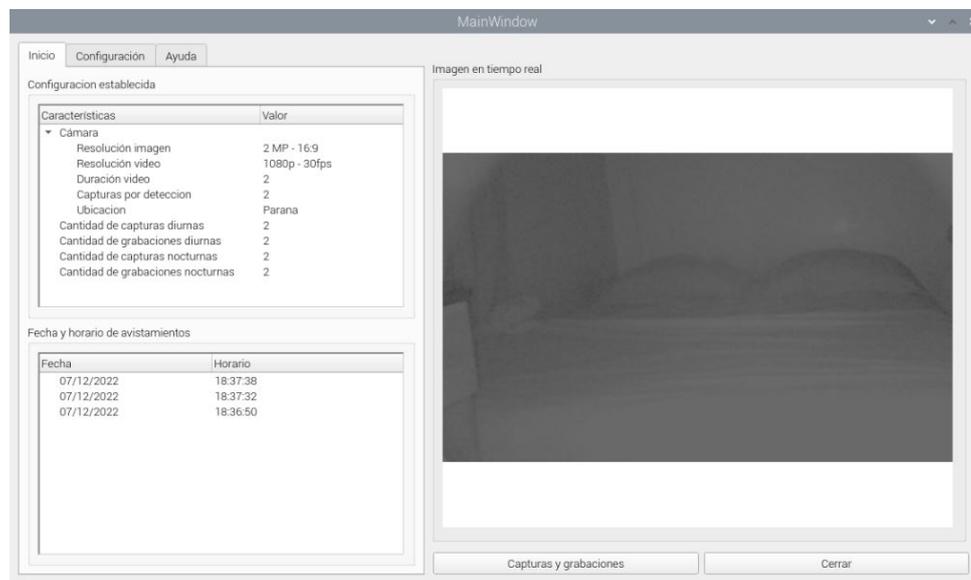


Figura 56. GUI de Cámara Trampa con visión nocturna activada.

También se observa que las nuevas detecciones han sido agregadas a las capturas y grabaciones nocturnas.

En caso de presionar el botón de capturas y grabaciones, se abre un navegador de archivos, en la carpeta de las capturas y grabaciones.

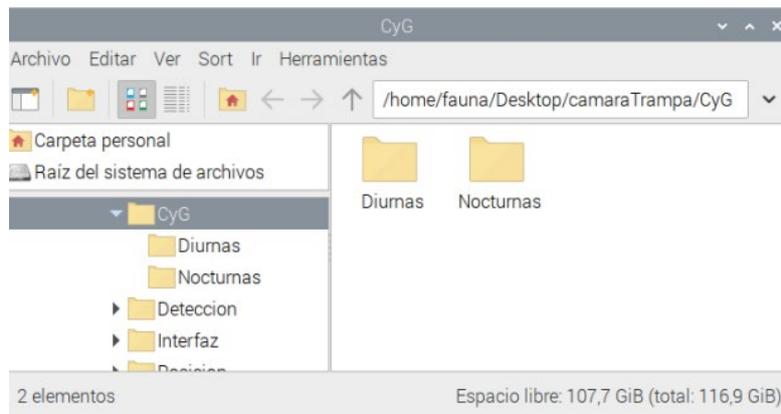


Figura 57. Navegador de archivos en directorio de capturas y grabaciones.

5.3.2.2 Implementación de scripts para la Catalogación de imágenes

Para el modelo de visión artificial, se requirió la descarga de la mayor cantidad posible de imágenes de los animales a detectar, según el requerimiento RF14, para obtener un modelo más preciso.

De las imágenes obtenidas se apartaron 10 de cada especie para el testeo y el resto se dividieron en 75% para el entrenamiento y 25% para la validación.

Las cantidades de imágenes obtenidas fueron las siguientes:

Especie	Entrenamiento	Validación	Test	Total
Carpincho	475	103	10	588
Ciervo	471	125	10	606
Gato montés	334	85	10	429
Puma	521	140	10	671
Oveja	342	74	10	426
Vaca	482	123	10	615

Figura 58. Imágenes utilizadas en el entrenamiento del modelo.

Existen distintas razas de cada especie animal. Para la creación del modelo se agregaron solamente las que se encuentran en Argentina.

Ciervo	Oveja	Vaca	Gato montés	Carpincho	Puma
Blastoceros dichotomus	Corriedale	Angus	Leopardus geoffroyi geoffroyi	Hydrochoerus hydrochaeris	Puma concolor cabrae
Mazama rufina	Merino	Hereford	Leopardus geoffroyi euxanthus	Hydrochoerus isthmus	Puma concolor puma
Mazama guazoubira	Merino Argentino	Braham	Leopardus geoffroyi leucobaptus	Hydrochoerus gaylordi	Puma concolor hudsoni
Mazama americana	Romney Marsh	Brangus	Leopardus geoffroyi paraguayae	Hydrochoerus ballesterensis	Puma concolor anthonyi
Hippocamelus bisulcus	Lincoln	Braford	Leopardus geoffroyi salinarum	Hydrochoerus gracilis	-
Pudu pudu	Polwerth o Ideal	Limousin	-	-	-
Hippocamelus antisensis	Karakul	Criollo	-	-	-
Ozotoceros bezoarticus	Hampshire Down	Holstein	-	-	-
Axis axis	Texel	Jersey	-	-	-
Cervus elaphus	Dorset Horn	-	-	-	-
Dama dama	Ost Friesan Mischalfe	-	-	-	-
Rangifer tarandus	corino y cormo	-	-	-	-

Figura 59. Razas de las distintas especies a detectar.

Inicialmente se recortaron y distribuyeron las imágenes en los directorios correspondientes. Luego se utilizaron 2 scripts programados en Python, donde el primero se utiliza para el entrenamiento y el otro para la predicción.

Entrenamiento

Inicialmente se asignan la cantidad de muestras por actualización, número de iteraciones, cantidad de clases, modelo a utilizar y tipo de imágenes (diurnas o nocturnas). Luego, el script realiza el siguiente proceso:

1. Realiza Data Augmentation para ampliar el set de imágenes.
2. Reduce el tamaño de las imágenes a 224 x 224, siendo este el tamaño máximo soportado por los modelos VGG-16 y VGG-19.
3. Genera modelo utilizando ultima capa y conservando los pesos del modelo preentrenado utilizado, permitiendo agilizar el proceso y aumentar precisión.
4. Muestra las métricas de desempeño y la matriz de confusión.

La cantidad de muestras depende de la capacidad de procesamiento de la computadora donde se realice el entrenamiento. La computadora utilizada soporta un máximo de 16 muestras por iteración. Se realizaron 500 iteraciones para obtener una buena precisión.

Se puede observar que en la validación se obtuvieron porcentajes cercanos al 95% para ambos modelos, observándose un aumento de precisión en el modelo VGG-19.

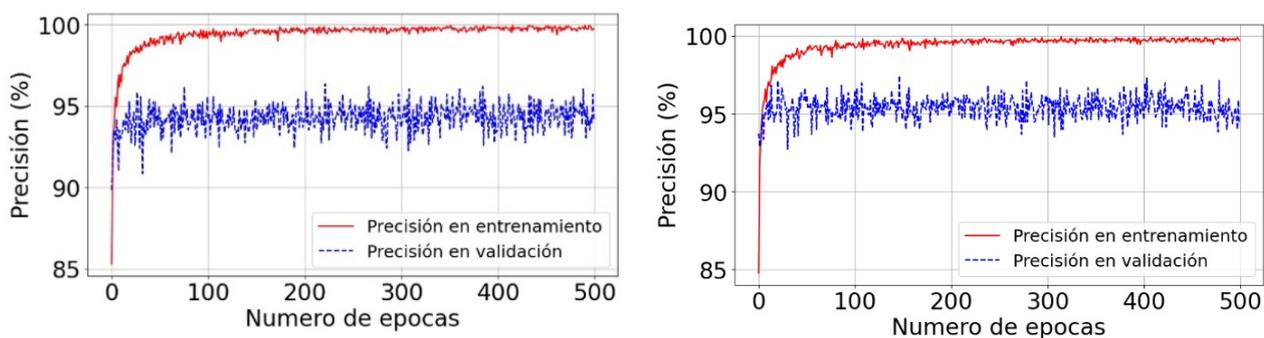


Figura 60. Métricas de desempeño para (a) VGG-16 y (b) VGG-19 .

También se obtuvieron las matrices de confusión, las cuales indican a su izquierda, la clasificación real de la imagen, y en su parte inferior indica cuales fueron las predicciones realizadas.

De esta manera podemos ver en la matriz de confusión para VGG-16, que, para 10 imágenes de carpinchos, 9 fueron clasificadas correctamente mientras que 1 fue clasificada incorrectamente como oveja.

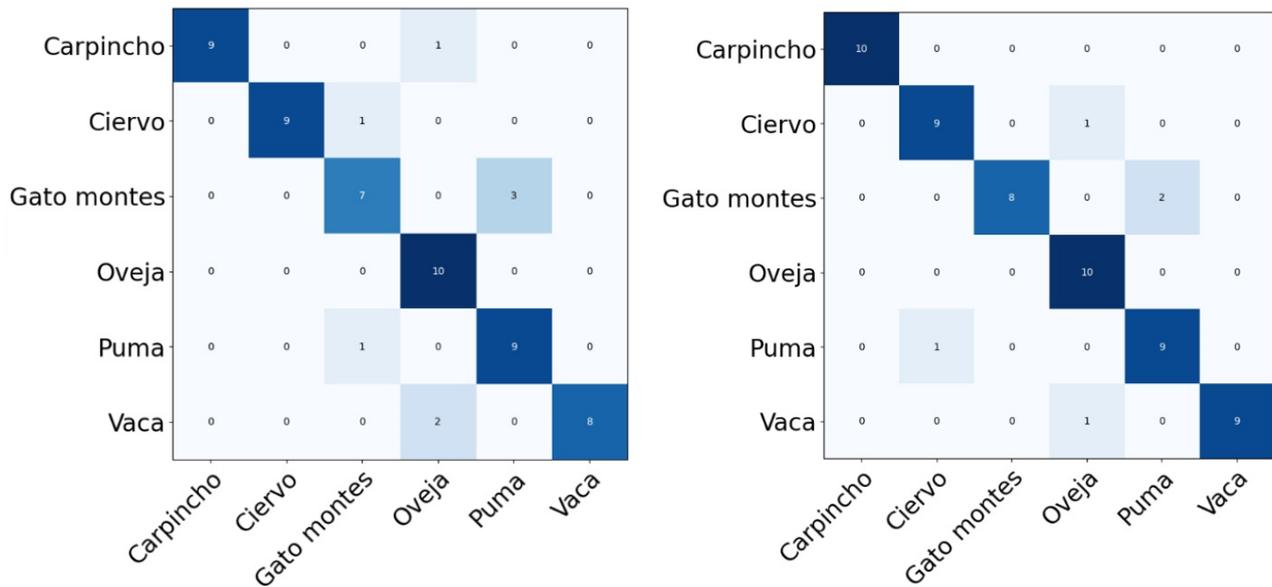


Figura 61. Matriz de confusión para (a) VGG-16 y (b) VGG-19 .

En el modelo VGG-16 se clasificaron de manera correcta 52 imágenes de 60, siendo 8 las clasificaciones incorrectas, mientras que el modelo VGG-19 clasificaron 55 de las mismas imágenes de manera correcta y 5 incorrectas.

Estos resultados fueron obtenidos al correr el script de entrenamiento, siendo “.ipynb” la extensión del archivo. El nombre de esta extensión indica que es un *cuaderno interactivo de python* (IPYNB acrónimo del inglés, *Interactive Python Notebook*). Este cuaderno funciona como registro, conteniendo el código en Python del script, así como también, las entradas y salidas del programa, las funciones matemáticas, la representación de imágenes y texto.

Predicción:

El funcionamiento del script de predicción consiste en:

1. Asignar cantidad de muestras por actualización, número de iteraciones, cantidad de clases, modelo a utilizar y tipo de imágenes (diurnas o nocturnas).
2. Cargar el modelo generado por el script de entrenamiento.
3. Leer las imágenes del directorio especificado.

4. Procesar las imágenes, comenzando por una reducción de tamaño a 224 x 224.
5. Realizar las predicciones.
6. Si las predicciones realizadas son superiores a una constante denominada porcentaje de certeza, las imágenes son renombradas y relocalizadas en el directorio del animal detectado, en caso de que ninguna de las predicciones de las categorías supere el valor del porcentaje de certeza, la imagen se renombra y se relocaliza en el directorio de objetos desconocidos.

Para la elección del porcentaje de certeza se realizaron múltiples pruebas, con el fin de obtener los mejores resultados de predicción, haciendo uso de 100 imágenes no pertenecientes al set de datos. Se busca obtener la mayor cantidad de imágenes clasificadas, con el menor porcentaje de error.

Porcentaje de certeza > 0.9995:

Modelo	Carpincho	Ciervo	Gato montes	Oveja	Puma	Vaca	X
VGG-19	0	0	3	0	0	0	30
VGG-16	0	2	2	1	0	0	33

Tabla 5. Errores e imágenes desconocidas para % de certeza > 0,9995.

Porcentaje de certeza > 0.999:

Modelo	Carpincho	Ciervo	Gato montes	Oveja	Puma	Vaca	X
VGG-19	0	0	3	0	1	1	28
VGG-16	2	2	2	1	0	0	28

Tabla 6. Errores e imágenes desconocidas para % de certeza > 0,999.

Porcentaje de certeza > 0.995:

Modelo	Carpincho	Ciervo	Gato montes	Oveja	Puma	Vaca	X
VGG-19	0	0	4	0	1	0	21
VGG-16	2	2	2	1	1	0	22

Tabla 7. Errores e imágenes desconocidas para % de certeza > 0,995.

Porcentaje de certeza > 0.99:

Modelo	Carpincho	Ciervo	Gato montes	Oveja	Puma	Vaca	X
VGG-19	0	0	4	0	1	0	16
VGG-16	2	2	2	1	1	0	21

Tabla 8. Errores e imágenes desconocidas para % de certeza > 0,99.

Porcentaje de certeza > 0.95:

Modelo	Carpincho	Ciervo	Gato montes	Oveja	Puma	Vaca	X
VGG-19	0	0	4	0	1	0	13
VGG-16	2	5	3	2	2	0	10

Tabla 9. Errores e imágenes desconocidas para % de certeza > 0,95.

Porcentaje de certeza > 0.9:

Modelo	Carpincho	Ciervo	Gato montes	Oveja	Puma	Vaca	X
VGG-19	1	1	4	0	1	1	10
VGG-16	2	5	3	3	3	0	8

Tabla 10. Errores e imágenes desconocidas para % de certeza > 0,9.

Se puede observar que a medida que se disminuye el porcentaje de certeza, mas imágenes son clasificadas, pero si disminuimos mucho este valor las clasificaciones son mayormente erróneas. Se consideró entonces utilizar los siguientes valores:

Certeza (%) VGG-19 = 0,95 (5 errores en 87 imágenes clasificadas)

Certeza (%) VGG-16 = 0,99 (8 errores en 87 imágenes clasificadas)

Para calcular la precisión de la catalogación de imágenes, se calcula el porcentaje de imágenes clasificadas. Siendo, para el modelo VGG19:

- Total de imágenes clasificadas: 87
- Errores = 5

$$\frac{\text{Total de imagenes clasificadas} - \text{Errores}}{\text{Total de imagenes clasificadas}} * 100\% \quad (6)$$

Finalmente, la precisión obtenida para el modelo VGG-19 es de 94,25% mientras que para el modelo VGG-16 es de 90,80%.

5.3.3 Consumo: Elección de batería y panel solar

Para determinar que batería y panel solar utilizar, se estudió el consumo del equipo en 2 situaciones: iluminación nocturna encendida y apagada.

Inicialmente se determinó la corriente requerida en cada situación, dando como resultado:

- Corriente con iluminación nocturna encendida: 3,30 [A]
- Corriente con iluminación nocturna apagada: 0,75 [A]

Considerando que el equipo funciona a 12V, la potencia demandada por el equipo en ambas situaciones era:

- Potencia con iluminación nocturna encendida: 39,6 [W]
- Potencia con iluminación nocturna apagada: 9,0 [W]

Se debió considerar para la correcta elección de la batería y el panel solar, la duración de día más corta de la región en la que se instalará el equipo.

La página web del Servicio de Hidrografía Naval de Argentina [51] proporciona datos acerca de la horas de salida y puesta del sol, pudiéndose así determinar la duración del día. A través de este recurso se obtiene que la **duración de día más acotada** se dará en la fecha de 01/07/2023, la misma será de **11h 1m 1s**, siendo el crepúsculo matutino a las 07:26 y el crepúsculo vespertino a las 18:32.

Se obtuvo de esta manera el gráfico de consumo diario de nuestro equipo en la condición más desfavorable:



Figura 62. Consumo de la cámara trampa en condición más desfavorable.

Haciendo uso de la calculadora online provista por electroimpulso [52], se calcula la batería y el panel solar. Los datos requeridos para el cálculo fueron:

- Potencia máxima: 39,6 [W]
- Tiempo de consumo máximo por día: 13 horas
- Tensión del banco de baterías: 12V
- Profundidad de descarga admisible: 55% (máximo recomendable para no dañar la batería).

Se obtuvo:

- Capacidad de banco de baterías necesario: 76,8 Ah
- Consumo diario: 42,25Ah.

Para el cálculo de la capacidad de generación solar se necesita conocer la radiación solar de la zona. En la época de julio, la ciudad de Paraná, Entre Ríos presenta una radiación diaria de 3 kWh/m² (dato obtenido de la página web del grupo de estudio de la radiación solar, GERSolar [53]).

Esto indica que un panel solar de 100W que recibe 3 horas pico de sol al día en julio, generará 300Wh de energía al día.

Considerando un consumo máximo de 39,6 Wh durante 13 horas, se obtuvo un consumo diario de 514,8 Wh/día.

En la región de la instalación se obtienen 3 horas efectivas de radiación solar, siendo la potencia del panel solar a utilizar:

$$\text{Potencia panel solar: } \frac{514,8 \frac{\text{Wh}}{\text{día}}}{3 \frac{\text{h}}{\text{día}}} = 171,6 \text{ w}$$

Se recomienda entonces utilizar una batería solar de 80Ah y 1 panel solar de 180W.

5.3.4 Implementación de Gabinete

Como indica el requerimiento RF15, se utilizó un gabinete estanco con soportes impresos en 3D para poder acceder al dispositivo de almacenamiento sin requerir desarmar o desconectar cables.



Figura 63. División del circuito en 2 niveles para simple acceso al almacenamiento.



Figura 64. Gabinete de cámara trampa con conectores estancos, portaleds y porta cámara.

Además, se utilizó un gabinete estanco, el cual fue modificado para soportar los sensores de movimiento PIR para así poder detectar en 360° y ser soportado desde un poste.



Figura 65. Gabinete de sensores PIR.

5.4 Interfaz Humano-Maquina

5.4.1 Uso de Cámara Trampa

Una vez realizada la instalación y configuración de programas en el Raspberry (véase **Anexo 3**), se procede a ingresar remotamente al Raspberry utilizando la aplicación VNC Viewer. El mismo puede ser accedido a través del servidor de VNC Viewer o por Bluetooth.

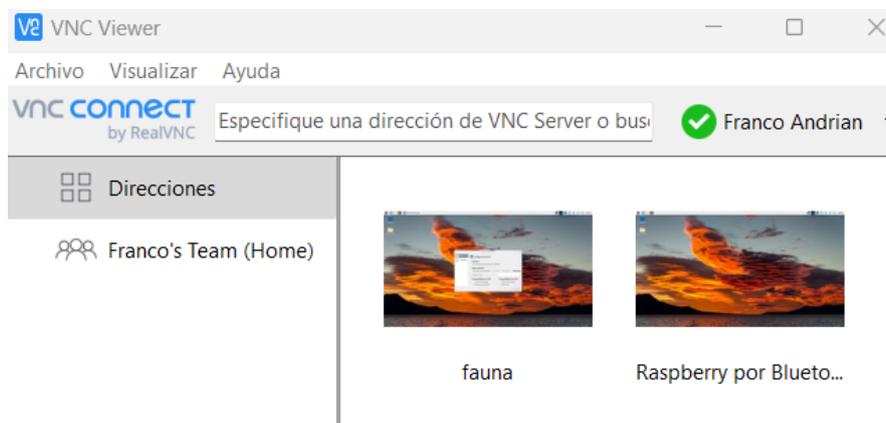


Figura 66. Redes creadas para el ingreso remoto.

Al ingresar mediante VNC Viewer al escritorio remoto, podemos observar que la interfaz se ha abierto de manera automática, en el arranque:

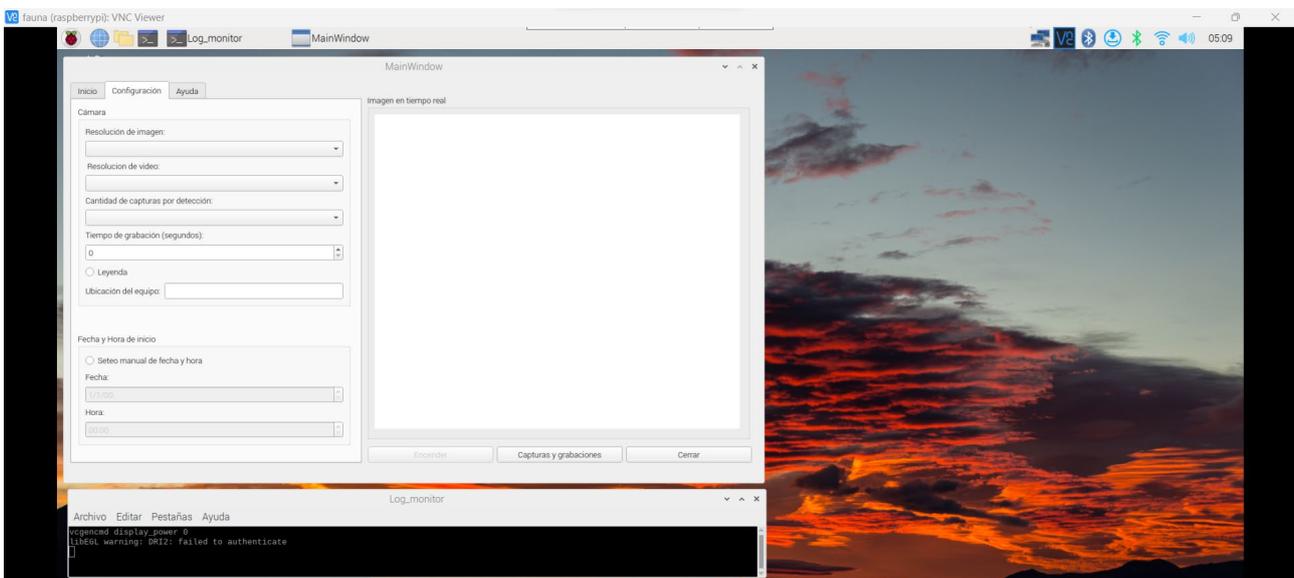


Figura 67. Apertura automática de interfaz Gráfica.

Luego solo resta rellenar los campos obligatorios y encender el equipo:

- Resolución de imagen.
- Resolución de video.
- Cantidad de capturas por detección.
- Tiempo de grabación.

Los campos de leyenda, Ubicación y seteo de fecha y hora son opcionales.

La conexión por bluetooth se utilizará en caso de no tener acceso a internet. En este caso, el sistema no podrá configurar su hora de manera automática, por lo cual se debe setear a través de la interfaz. En caso de que el equipo se apague, al reencenderse se lanzará la interfaz, pero no se iniciará el programa sin importar si existe o no archivo de backup, debido a que no se sabe con exactitud cuanto tiempo estuvo apagado.

5.4.2 Uso de scripts de Vision Artificial

Una vez realizada la instalación y configuración de programas en la computadora (véase **Anexo 4**) se ingresa a ventana de comandos Anaconda Prompt y se ingresan los siguientes comandos:

```
conda activate GPU  
cd {ruta de archivos de entrenamiento y predicción}  
Jupyter Notebook
```

Estos comandos lanzarán la interfaz de Jupyter Notebook desde el navegador web predeterminado, como se muestra en la siguiente figura:

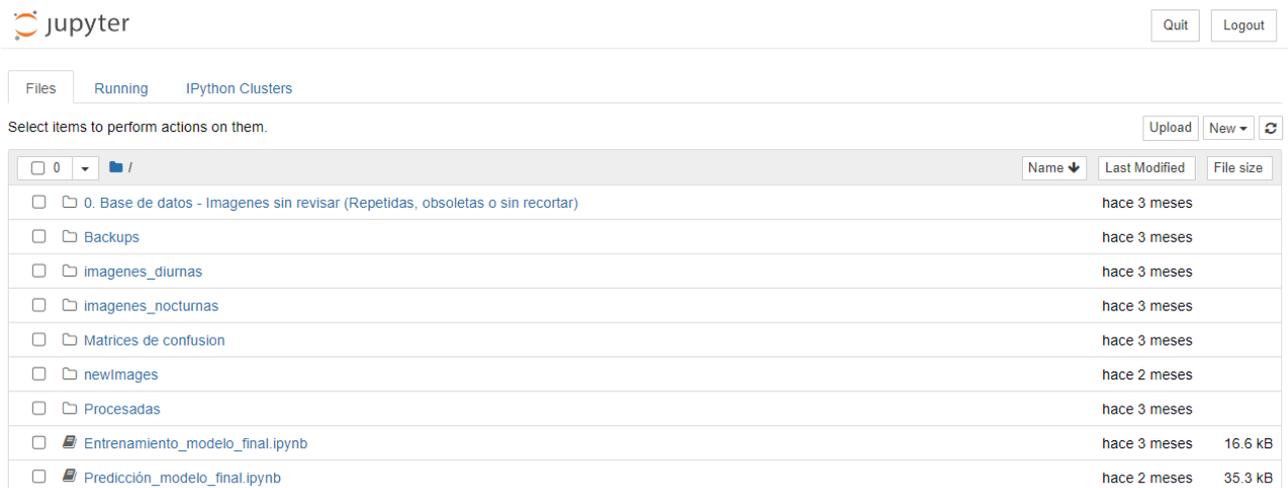


Figura 68. Interfaz Web de Jupyter Notebook.

Luego se ingresa al archivo a utilizar:

- Para el entrenamiento: Entrenamiento_modelo_final.ipynb
- Para la predicción: Predicción_modelo_final.ipynb

Para correr el código se selecciona el icono RUN.



La segunda opción para procesar el código consiste en realizar los mismos pasos descritos con anterioridad, a excepción del comando **Jupyter notebook** que será reemplazado por:

```
ipython {nombreArchivo.ipynb}
```

De esta manera se corre el código desde la consola.

5.4.2.1 Agregado de imágenes a set de datos para reentrenamiento.

Agregando nuevas capturas a las distintas clases descritas en el punto 5.3.3.2, podemos aumentar la precisión de nuestro modelo.

A continuación se detallan los pasos para el agregado:

1. Recortar las imágenes nuevas según el animal que se desea catalogar.

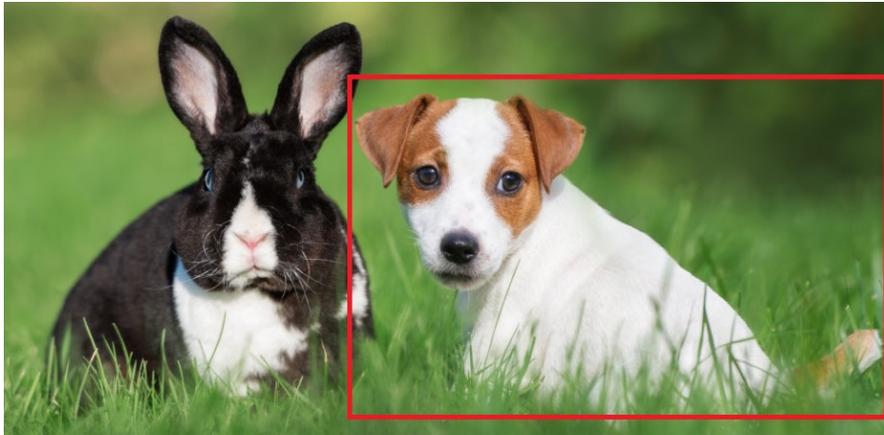


Figura 69. descripción de recorte de imagen [51].

2. Dividir el 100% de las imágenes obtenidas en 75% para entrenamiento y 25% para validación, para cada animal.
3. Ingresar a la ruta del modelo según las imágenes a agregar:
 - Imágenes diurnas: ~/modelo/imagenes_diurnas
 - Imágenes nocturnas: ~/modelo/imagenes_nocturnas
4. Agregar imágenes a los directorios de entrenamiento y validación correspondientes, según la ruta a utilizar.
5. Correr el script de entrenamiento de modelo, con los pasos del inciso 5.4.2.

5.5 Validación y verificación

En este apartado se detallan las pruebas utilizadas para verificar el funcionamiento del equipo. Los casos de prueba son los procedimientos utilizados para determinar si se cumplen los requerimientos funcionales estipulados.

Se presenta a continuación la matriz de trazabilidad del sistema, la misma consiste en una tabla de 2 entradas donde se ordenan los casos de prueba y los requerimientos funcionales del sistema. La misma permite determinar que requerimientos se testean en cada caso de prueba, y en caso de falla, se deben comprobar nuevamente cada uno de los requerimientos de la prueba.

Req. → Pruebas ↓	RF01	RF02	RF03	RF04	RF05	RF06	RF07	RF08	RF09	RF10	RF11	RF12	RF13	RF14	RF15	RF16
CP01	X	X														
CP02			X	X												
CP03					X											
CP04						X										
CP05							X			X						
CP06								X	X							
CP07											X	X	X			
CP08														X		
CP09															X	
CP10																X

Tabla 11. Matriz de trazabilidad de requerimientos funcionales del sistema.

Para el testeo de los distintos casos de prueba, se utilizaron las siguientes herramientas:

Herramientas Pruebas	Tester	Fuente de alimentación	Aplicación Luxómetro	PC	Dispositivo de almacenamiento externo
CP01	X	X			
CP02			X	X	
CP03				X	

CP04				X	
CP05				X	
CP06				X	
CP07				X	X
CP08				X	
CP10				X	

Tabla 12. Herramientas utilizadas para cada caso de prueba.



Figura 70. Tester o multímetro digital.



Figura 71. Fuente de tensión continua regulable.

El dispositivo de almacenamiento externo puede ser un pendrive o un disco rígido conectado a través de un conversor USB-SATA. La computadora utilizada se detalló en el apartado “Selección de hardware para Entrenamiento de modelo”.

5.5.1 Casos de prueba

Se describen a continuación los casos de prueba utilizados:

5.5.1.1 Caso de prueba N° 01: Alimentación del sistema.

Se comprobará que para tensiones comprendidas entre la tensión de encendido y la tensión máxima, el equipo se encenderá y a su vez, se alimentará el Arduino, el cual encenderá el módulo conversor Step Up, el Raspberry, los sensores PIR y el controlador del motor.

Una vez encendido el equipo, se verificará que al suministrar tensiones continuas inferiores y superiores a las permitidas se apagará el conversor Step Up, el Raspberry y los sensores PIR.

5.5.1.2 Caso de prueba N° 02: Iluminación nocturna.

Se verificará que para intensidades de luz menores al límite inferior seteado, se encenderá la iluminación nocturna. Además, La cámara debe ser capaz de detectar la iluminación nocturna artificial hasta una distancia de 10 metros, en condiciones de escasa iluminación.

5.5.1.3 Caso de prueba N° 03: Conexión mediante Bluetooth.

Se comprobará que el sistema es capaz de emparejarse por bluetooth y setear la hora del sistema de manera manual.

5.5.1.4 Caso de prueba N° 04: Conexión mediante WiFi.

Se comprobará que el sistema es capaz de conectarse a internet por WiFi para el acceso remoto.

5.5.1.5 Caso de prueba N° 05: Configuración mediante GUI.

Se constatará que el sistema permite configurar la resolución de la cámara, la resolución de video, la cantidad de capturas por detección, el tiempo de

grabación en segundos, el agregado de leyenda con la ubicación del equipo en imagen y video y la asignación manual y automática de la fecha y hora.

Ademas, las imágenes deben ser guardadas en formato JPG y los videos en formato H264.

5.5.1.6 Caso de prueba N° 06: Detección de movimiento y reposicionamiento del sistema.

Se verificará que los sensores detectan movimiento hasta 10 metros de alcance, en los 360°, exceptuando los puntos ciegos detallados en el apartado de sensado de movimiento. Una vez detectado el movimiento, el sistema de posicionamiento debe localizar la cámara en la región detectada.

5.5.1.7 Caso de prueba N° 07: Almacenado de archivos.

Se comprobará el almacenado de las imágenes y videos obtenidos, en el dispositivo de almacenamiento externo. Luego se verificará la creación de los directorios en el dispositivo de almacenamiento utilizado, para el almacenado de imágenes y videos, diurnos y nocturnos y se determinará si el sistema nombra correctamente las imágenes y videos, según lo requerido.

5.5.1.8 Caso de prueba N° 08: Catalogado de imágenes.

Se comprobará que el script de predicción es capaz de catalogar imágenes de las especies solicitadas. Tambien se verificará el nombre asignado.

5.5.1.9 Caso de prueba N° 09: Gabinete Estanco.

Se comprobará que el sistema este contenido en un gabinete estanco que permita rapido acceso al dispositivo de almacenamiento.

5.5.1.10 Caso de prueba N° 10: Refrigeración del sistema.

Se comprobará que la ventilación se enciende al aumentar la temperatura del equipo y que, al disminuir, se apaga para el ahorro de energía.

Se describe mediante la siguiente tabla, el escenario, las condiciones previas y la descripción de casos y sub-casos, donde:

- **Escenario:** Determina las herramientas y equipos a utilizar en cada caso y en que entorno serán desarrolladas las pruebas.
- **Condiciones previas:** Se determinan las condiciones iniciales que se deben cumplir antes de iniciar el testeo del caso.
- **Descripción de casos y sub-casos:** Indica el procedimiento a seguir para el testeo del caso.

Prueba	Escenario	Condiciones previas	Descripción de casos y sub-casos
CP01	Sistema, Tester, Fuente de tensión,	Sistema ON,	<p>CP01-1: Se alimenta el sistema con 10,9V, el mismo se enciende y se apaga por sub-tensión.</p> <p>CP01-2: Se alimenta el sistema con 15,6V, el mismo se enciende y se apaga por sobretensión.</p> <p>CP01-3: Se alimenta el sistema con 12V, el mismo se enciende y se mantiene encendido.</p> <p>CP01-4: Se aumenta la tensión con el equipo encendido hasta 15,6V y el equipo se apaga.</p> <p>CP01-5: Se reduce la tensión con el equipo encendido hasta 10,9V y el equipo se apaga.</p>

CP02	Sistema, App. Luxómetro, Computadora, Exterior,	Sistema ON, Mínima luz ambiente, Demarcación de circunferencia de radio 10m	<p>CP02-1: Se alimenta el sistema, se corre el programa y se espera a que la intensidad disminuya hasta 9 lux y la iluminación nocturna se enciende.</p> <p>CP02-2: Se alimenta el sistema, se corre el programa y se espera a que la intensidad aumente hasta 26 lux y la iluminación nocturna se apaga.</p> <p>CP02-3: Se alimenta el sistema, y en condición de iluminación nocturna encendida, se observa buena iluminación en demarcación de distancia de 10 metros, en la imagen en tiempo real de interfaz.</p>
CP03	Sistema, Computadora,	Sistema ON, BT de RPI ON BT de PC y RPI emparejado, Red de VNC creada,	<p>CP03-1: Se alimenta el sistema, desde la PC se une a la red de área personal por Bluetooth generada por Raspberry y se conecta remotamente desde VNC Viewer, luego en Raspberry se abre la GUI.</p>
CP04	Sistema, Computadora, Switch,	Sistema ON, Red WiFi config. en RPi, Red de VNC creada,	<p>CP04-1: Se alimenta el sistema y desde la PC se conecta remotamente desde VNC Viewer a través del servidor de VNC.</p>
CP05	Sistema, Computadora,	Sistema ON,	<p>CP05-1: Se alimenta el sistema, se abre la GUI y se configura:</p> <p>Res. img. : 8MP Res. Vid. :1080p 30fps Cant. Capturas: 1 Duración vid.: 1 s Leyenda: Activada Ubicación: Parana Fecha manual: Activada Fecha: 01/01/23 Hora: 12:12</p>

			<p>Se obtiene por detección: 1 img. de 8MP de res. En formato JPG y 1 vid. de 1 segundo de res. 1080p 30fps, en formato H264.</p> <p>El sistema debe cambiar su hora por la seteada y cada imagen y video debe presentar una leyenda con la ubicación, fecha y hora y numero de imagen o video en directorio.</p> <p>CP05-2: Encendido el equipo, se cierra y se vuelve a abrir la GUI y se configura: Res. Img. : 0,3MP Res. Vid. :480p 90fps Cant. Capturas: 5 Duración vid.: 90 s Leyenda: Desactivada Fecha manual: Desactivada</p> <p>Se obtiene por detección: 5 img. de 0,3 MP de res. En formato JPG y 1 vid. De 90 segundos de res. 480p, en formato H264.</p>
CP06	Sistema, Computadora, Persona gateando,	Sistema ON, Área de circunferencia de radio 10m descubierta con puntos ciegos y regiones de detección demarcadas.	<p>CP06-1: Se alimenta el sistema y se corre el programa. La persona se posiciona en un punto ciego, a 10 metros del equipo y recorre la circunferencia de radio 10 metros hacia la derecha, hasta llegar al segundo punto ciego.</p> <p>Una vez ingresada la persona a la región de detección, se enciende el led del sensor PIR y se localiza el equipo en dicha posición.</p>

			El procedimiento se realizará 4 veces, recorriendo finalmente una circunferencia completa.
CP07	Sistema, Computadora, Almacenamiento externo,	Sistema ON, Capturas de img. y vid. previas,	CP07-1: Se conecta el dispositivo de almacenamiento externo por el puerto USB al Raspberry, se alimenta el sistema y se corre el programa. Luego se realiza un movimiento delante de uno de los sensores para realizar las capturas y grabaciones correspondientes, una vez actualizada la interfaz con el avistamiento, se ingresa a través del botón de capturas y direcciones de la GUI al directorio del dispositivo externo, en donde se observan los directorios creados para las capturas diurnas y nocturnas, dentro de dichos directorios se encuentran las imágenes y videos obtenidos, con los nombres establecidos. En caso de no estar conectado el dispositivo de almacenamiento, el directorio de capturas y grabaciones se crea en la ruta: ~/desktop/camaraTrampa
CP08	Sistema, Computadora,	Computadora ON Set de imágenes,	CP08-1: Se agregan las imágenes al directorio de imágenes a procesar y se corre el programa de predicción. Las imágenes son catalogadas y relocalizadas de acuerdo con el animal detectado.
CP09	Sistema,	Sistema OFF,	CP09-1: Con el equipo apagado, se abre el gabinete de la cámara trampa y se accede al dispositivo de almacenamiento externo sin necesidad de mover elementos ni desconectar cables.

CP10	Sistema, Computadora,	Sistema ON,	<p>CP10-1: Se abre compartimento y se alimenta el sistema. Luego se acerca un elemento caliente al sensor de temperatura NTC, sin tocarlo. Al aumentar la temperatura los ventiladores se encienden.</p> <p>CP10-2: Una vez encendidos los ventiladores, se espera hasta que se reduzca la temperatura y los mismos se apagan.</p>
------	-----------------------	-------------	--

Tabla 13. Escenario, precondiciones y casos/sub-casos de cada prueba.

5.5.2 Resultados de los casos de prueba

A continuación, se presenta la tabla de resultados esperados y obtenidos de cada prueba, donde se enumeran las fallas en caso de haberlas.

Prueba	Resultados esperados	Resultados obtenidos	Test N°		
			1	2	3
CP01	<p>CP01-1: Sistema se enciende y se apaga.</p> <p>CP01-2: Sistema se enciende y se apaga.</p> <p>CP01-3: Sistema se enciende y se mantiene encendido.</p> <p>CP01-4: Sistema inicia encendido y se apaga.</p> <p>CP01-5: Sistema inicia encendido y se apaga.</p>	<p>CP01-1: Cumple.</p> <p>CP01-2: El sistema se enciende y apaga al ser alimentado con 15,7V.</p> <p>CP01-3: Cumple.</p> <p>CP01-4: Cumple.</p> <p>CP01-5: El sistema se apaga al disminuir la tensión a 11,2V.</p>	F A L L A	P A S A	

CP02	<p>CP02-1: Intensidades de luz < 10 lux encienden la iluminación nocturna.</p> <p>CP02-2: Intensidades de luz iguales o mayores a 25 lux apagan la iluminación nocturna.</p> <p>CP02-3: Cámara detecta iluminación nocturna a 10 metros del sistema.</p>	<p>CP02-1: El sistema enciende la iluminación al detectarse 9 lux en la app. Luxómetro.</p> <p>CP02-2: El sistema enciende la iluminación al detectarse 24 lux en la app. Luxómetro.</p> <p>CP02-3: Cumple.</p>	F A L L A 1	P A S A	
CP03	CP03-1: Acceso remoto por conexión Bluetooth exitoso.	CP03-1: El equipo se conecta mediante Bluetooth y muestra la interfaz.	P A S A		
CP04	CP04-1: Acceso remoto por conexión WiFi exitoso.	CP04-1: El equipo se conecta mediante WiFi y se muestra la interfaz.	P A S A		

CP05	<p>CP05-1: Se obtienen por detección: 1 img. de 8MP de res. En formato JPG y 1 vid. de 1 segundo de res. 1080p 30fps, en formato H264, con una leyenda con ubicación, fecha y hora seteada y número de detección.</p> <p>La fecha y hora del sistema cambia a 01/01/23 - 12:12.</p> <p>CP05-2: Se obtienen por detección: 5 img. de 0,3 MP de res. En formato JPG y 1 vid. De 90 segundos de res. 480p, en formato H264.</p>	<p>CP05-1: Se obtuvo una imagen de nombre "01_01_23-12_12_21-(1)-D", de resolución 8 MP y formato JPG y un video de nombre "01_01_23-12_12_21-D" de resolución 1080p en formato H264, con duración de 1 segundo. La hora fue seleccionada manualmente y se agregó la leyenda correspondiente.</p> <p>CP05-2: Se obtuvieron 5 imágenes de nombre "20_02_23-22_25_37-(1)-D.jpg" a "20_02_23-22_25_37-(5)-D.jpg", de resolución 0.3 MP, en formato JPG y un video de nombre "20_02_23-22_25_37-D.h264" resolución 480p, en formato h264 de 90 segundos de duración.</p>	F A L L A 1	P A S A	
CP06	CP06-1: Los 4 sensores detectan movimiento y el sistema se posiciona correctamente en las 4 posiciones.	C06-1: Cumple	F A L L A 1	F A L L A 2	P A S A
CP07	CP07-1: Se obtienen directorios de captura diurna y nocturna en el directorio CyG creado en el dispositivo de almacenamiento externo. En su interior posee las imágenes y grabaciones en el formato indicado en el requerimiento RF13.	CP07-1: Cumple	P A S A		

CP08	CP08-1: Las imágenes procesadas se relocalizan en los directorios asociados a las especies de los animales.	CP08-1: Cumple	P A S A		
CP09	CP09-1: El acceso al dispositivo de almacenamiento es sencillo, sin necesidad de desconexión o desplazamiento de elementos.	CP09-1: Cumple	P A S A		
CP10	CP10-1: Al calentar el sensor de temperatura se enciende la ventilación. CP10-2: Encendida la ventilación, luego de disminuir la temperatura, la misma se apaga.	CP10-1: Cumple CP10-2: Cumple	P A S A		

Tabla 14. Resultados obtenidos de la ejecución de cada caso de prueba.

En la siguiente tabla se enlistan las fallas obtenidas en cada prueba y su respectiva solución:

Prueba	Falla y solución
CP01	FALLA N°1: Conversor ADC de Arduino nano funcionaba de manera alineal al ser alimentado a 5V, la mayor linealidad se encontró alimentando el módulo a 4.58V. Se solucionó ajustando la tensión del conversor Step Down que alimenta al Arduino nano y los módulos Relé, e ingresando la tensión de alimentación de 4.58V como referencia analógica.
CP02	FALLA N°1: Al igual que en la prueba CP01, la no linealidad del conversor ADC generaba que la iluminación nocturna se encienda en valores distintos a los calculados. Misma solución que CP01.
CP05	FALLA N°1: Videos grabados con librería PiCamera de Python para la cámara de Raspberry Pi no duraban el tiempo requerido y tenían pérdidas de fotogramas. Se modificó la tasa de bits y el perfil de codificación.

CP06	<p>FALLA N°1: Múltiples detecciones en el mismo sensor generaban que el seguimiento del objetivo sea incorrecto, se solucionó aumentando el tiempo de espera entre señales de entrada.</p> <p>FALLA N°2: Múltiples detecciones en el mismo sensor generaban un tiempo de espera innecesario. Se solucionó evitando que el contador se reinicie al detectar por segunda vez el mismo sensor.</p>
------	---

Tabla 15. Fallas de los casos de prueba y sus soluciones.

5.5.3 Pruebas de campo

Se realizó la siguiente prueba:

- Instalación de equipo en un poste con una altura superior a 2,4m con un área despejada de 15 metros de radio.

En esta prueba se verificó:

- Actuación rápida ante una detección.
- Seguimiento del objetivo.
- Cobertura máxima de los sensores de movimiento.
- Congelamiento del movimiento.

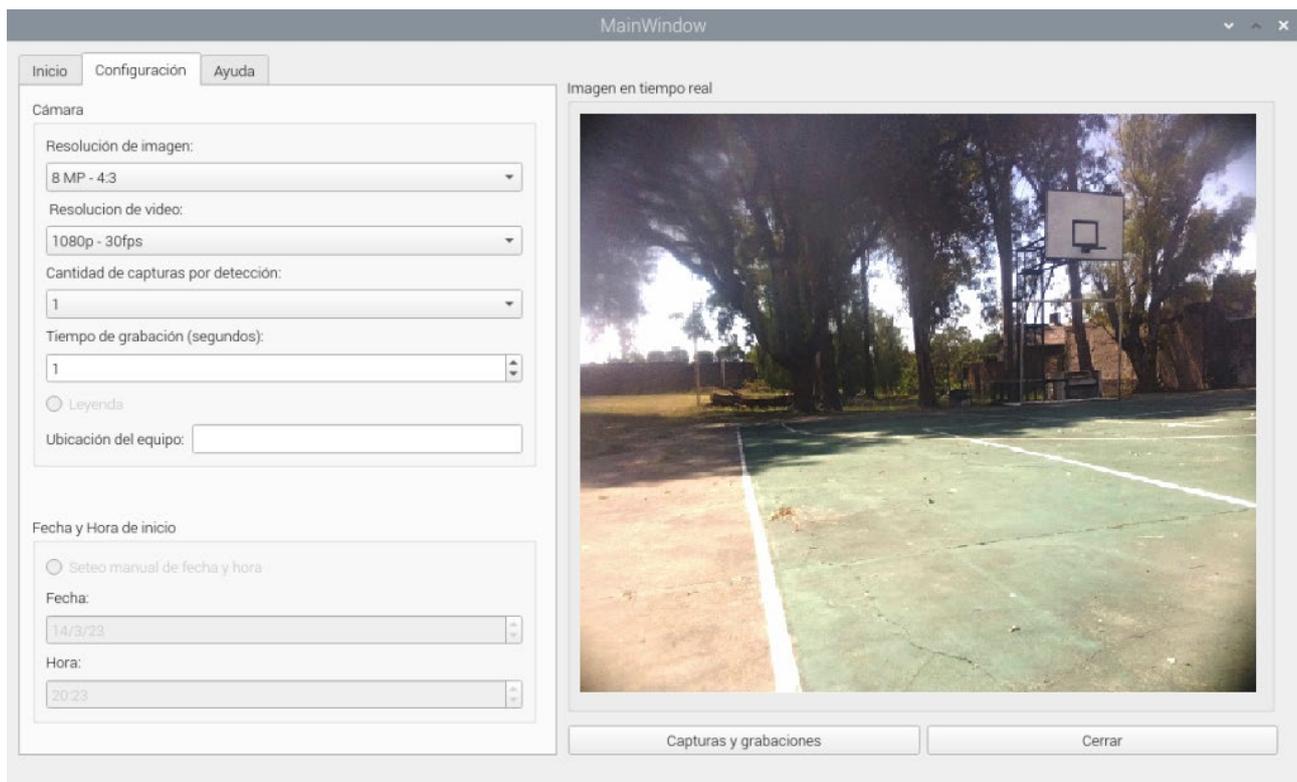


Figura 72. Configuración de sistema establecida.

5.5.3.1 Prueba en área despejada de radio 15 metros.

En esta prueba se instaló el gabinete de los sensores de movimiento a una altura de 2,4 metros y el gabinete de la cámara trampa a una altura de 0,6 metros, medida desde el suelo hasta el lente de la cámara.

Para la realización de la prueba de campo se instaló el sistema de cámara trampa en el poste de la cancha de Básquet de la UTN Regional Paraná.



Figura 73. Sistema de cámara trampa, montado en poste.

La prueba se realizó sobre el área despejada, siendo el objeto por detectar, una persona caminando a una velocidad aproximada de 4 km/h, sobre una circunferencia de radio 10 m, siendo el poste el centro de la circunferencia.

Bajo estas condiciones se esperaba:

- Que se detecte el movimiento hasta 10 metros de distancia.
- Que el sistema detecte el movimiento y se posicione en la región de detección.
- Que al posicionarse el sistema capture la/s foto/s y espere el tiempo determinado (5 segundos), y que en caso de no detectar movimiento nuevamente, realice la grabación de video.
- Que al posicionarse el sistema capture la/s foto/s, espere el tiempo determinado (5 segundos), y que en caso de detectar movimiento nuevamente se repositone, capture nuevamente la/s foto/s y reinicie el contador. Este procedimiento se debería realizar hasta que no se detecte movimiento en el tiempo establecido, finalizando con la grabación de video.

Los resultados obtenidos fueron satisfactorios, obteniéndose:

- Seguimiento del objetivo a velocidad de caminata constante.
- Detección de movimiento a distancias mayores de 15 metros.
- Congelamiento de movimiento.

Se presentan a continuación las imágenes asociadas a la prueba:



Figura 74. Detección y captura a 10 metros, demarcada con cinta blanca.



Figura 75. Detección y captura a 15 metros.



Figura 76. Detección y captura a 7 metros, caminando rapido.



Figura 77. Seguimiento del objetivo hacia la derecha.



Figura 78. Seguimiento del objetivo hacia la izquierda.

5.5.3.2 Prueba de iluminación nocturna

A continuación, se observa la distancia máxima de la iluminación nocturna, siendo de 15 metros la distancia entre la cámara y la pared blanca del fondo.



Figura 79. Imagen diurna: 8MP, ISO 100, velocidad de obturación 500.



Figura 80. Imagen nocturna: 8MP, ISO 800, velocidad de obturación 300000.

Capítulo 6: Resultados

Se obtuvo como resultado un prototipo funcional de cámara trampa, el cual mediante un motor paso a paso y un engranaje impreso en 3D puede posicionar el equipo en las 4 zonas de detección dispuestas en los 360°, permitiendo ampliar el campo de detección y por consiguiente, aumentando la probabilidad de detecciones, capturas y grabaciones de los animales presentes en el área.

Por su parte, el modelo de visión artificial logró una precisión del 94,25% para el modelo VGG-19, y una precisión del 90,80% para el modelo VGG-16.

Tanto la iluminación nocturna como los sensores de movimiento superan los 10 metros requeridos, lo que permite aumentar la cantidad de detecciones mientras el modelo de visión artificial podrá reconocer con mayor facilidad los patrones presentes en cada objeto, en fotografías nocturnas.

Se realizó un análisis FODA para determinar las cualidades del equipo.

Fortalezas

- Protección ante subtensión y sobretensión para evitar disminución de vida útil de batería.
- Reconfiguración automática ante apagado inesperado del equipo en funcionamiento.
- Iluminación nocturna potente, permitiendo capturas nocturnas con un rango visible mayor a 10 metros.
- Acceso remoto, el cual permite configurar y observar el funcionamiento del equipo sin tener que estar presente en el área de instalación, el cual muchas veces es de difícil acceso.
- Utilización de batería y panel solares permite que el equipo trabaje de manera ininterrumpida durante semanas o meses, evitando tener que ir al terreno de instalación a reemplazar las baterías constantemente.

Además, es una opción más amigable con el medioambiente, ya que las pilas alcalinas utilizadas en las cámaras trampa comerciales producen elevados niveles de contaminación.

- Almacenamiento de imágenes y videos organizados por fecha, hora y parte del día (diurna o nocturna).
- Expansión de memoria utilizando dispositivo de almacenamiento externo.
- Aviso mediante correo electrónico indicando que el almacenamiento está próximo a llenarse (95% del espacio lleno).
- Conexión mediante bluetooth y WiFi.

Oportunidades

- No existe ninguna solución en el mercado de cámaras trampa que integre un sistema de posicionamiento para ampliar el campo de visión.
- Son pocas las cámaras trampa comerciales que admiten controlar el dispositivo de manera remota.
- Las cámaras trampa comerciales presentan especificaciones dudosas: Los sensores de movimiento no poseen el alcance especificado, la iluminación nocturna no cubre la longitud deseada, la duración de las baterías es inferior a la indicada.

Debilidades

- El elevado peso y tamaño del equipo no permite elevar la velocidad de posicionamiento ya que se producen vibraciones y generan pérdidas de pasos en el motor. Por este motivo, si un animal corre por delante del sensor, no saldrá en la captura o saldrá recortado.
- La disposición de los sensores de movimiento, del motor y de la batería, por fuera del gabinete de la cámara trampa, aumenta la cantidad de cables, lo cual perjudica al posicionamiento del equipo cuando el cableado no se encuentra correctamente sujetado.

- Para aumentar la eficiencia del modelo de visión artificial, las imágenes agregadas al entrenamiento del modelo deben ser previamente recortadas.

Amenazas

- El equipo debe ser instalado en la intemperie por lo cual la exposición a fenómenos meteorológicos puede causar daños irreparables.

Capítulo 7: Análisis de Costos

A continuación, se indican los costos en dólares, de los materiales y componentes utilizados.

Componente/Dispositivo	Descripción	Cant.	Precio Unitario	Precio Final
Raspberry Pi 3A (Tarjeta SD + Fuente)	Computadora para controlar sistema	1	\$ 122,67	\$ 122,67
Cámara Raspberry Pi V2 NoIR	Módulo de cámara nocturna	1	\$ 35,56	\$ 35,56
Regulador Step Down	Alimentación para Arduino y raspberry	2	\$ 3,79	\$ 7,57
Regulador Step Up	Alimentación para arreglo de leds	1	\$ 4,53	\$ 4,53
Sensor PIR Optex Saver	Sensores de movimiento	4	\$ 9,33	\$ 37,33
LED IR 3W 850nm con predisipador	LED para iluminación nocturna	8	\$ 3,20	\$ 25,60
Disipadores estrella para LED		8	\$ 2,67	\$ 21,33
Motor Paso a Paso Nema 17	Control de posición	1	\$ 24,00	\$ 24,00
Driver Motor A4988	Controlador de motor	1	\$ 1,84	\$ 1,84
Shield A4988	Soporte para montaje de Controlador	1	\$ 1,71	\$ 1,71
Ventilador 5V	Ventilación de equipo	2	\$ 2,48	\$ 4,95
Sensor LDR	Sensor de luz	1	\$ 1,19	\$ 1,19
Microswitch	Interruptor para fin de carrera	1	\$ 0,40	\$ 0,40
Conectores hembra		1	\$ 1,33	\$ 1,33
Grasa siliconada		1	\$ 0,92	\$ 0,92
Arduino Nano	Microcontrolador para lectura de sensores	1	\$ 10,67	\$ 10,67
Borneras		9	\$ 0,27	\$ 2,40
Fusibles		2	\$ 0,09	\$ 0,17
Portafusibles		4	\$ 0,17	\$ 0,66
Placa PCB simple faz 15 x 20 cm		1	\$ 3,55	\$ 3,55
Gabinete estanco Genrod	Gabinete para sensores de movimiento	1	\$ 3,20	\$ 3,20
Adhesivo de conducción térmica	Adhesivo para pegar LEDs	1	\$ 2,18	\$ 2,18
Módulo Relay 2 canales 5V 10A		1	\$ 3,81	\$ 3,81
Módulo Relay 1 canal 5V 10A		4	\$ 1,71	\$ 6,85
Conector estanco de 10 pines	Conexión entre sensores y cámara trampa	1	\$ 5,06	\$ 5,06
Conector estanco de 6 pines	Conexión entre motor y cámara trampa	1	\$ 4,18	\$ 4,18
Conector estanco de 2 pines	Conexión entre batería y cámara trampa	1	\$ 1,89	\$ 1,89
Abrazadera acero inoxidable	Ajuste de soportes a poste	4	\$ 0,51	\$ 2,06
Pintura blanca 3 en 1 - 240cc	Pintura para proteger soportes metálicos	3	\$ 0,46	\$ 1,39
Poxiran	Pegamento para sensores de movimiento	1	\$ 0,44	\$ 0,44
Precintos de nylon	Precintos para soportar cables a poste	1	\$ 0,30	\$ 0,30
Silicona	Sellado de gabinetes	1	\$ 0,37	\$ 0,37
tubo termocontraible (x metro)		2	\$ 0,78	\$ 1,55
Prensacables		4	\$ 0,33	\$ 1,33
Cable LAN cat. 5E (x metro)	Conexión entre switch y Raspberry	10	\$ 0,22	\$ 2,15
Soportes de gabinetes		1	\$ 12,95	\$ 12,95
			TOTAL:	\$ 358

Tabla 16. Costo final del proyecto.

Capítulo 8: **Discusión y Conclusión**

El sistema implementado cumple con los requisitos propuestos, los cuales fueron verificados y validados a través de los casos de prueba.

El mayor desafío en la elaboración del dispositivo fue el sistema de posicionamiento integrado por el switch de fin de carrera, el motor de pasos, el engranaje que soporta al gabinete de la cámara y el soporte del motor.

Las dificultades presentadas fueron:

- Al no estar el soporte completamente a nivel, el motor no realizaba los pasos correspondientes, debido a que el peso del equipo se encontraba desbalanceado, para solucionarlo se modificó la posición de las abrazaderas de sujeción.
- El cableado dispuesto en la parte inferior del gabinete de la cámara trampa se trababa con el soporte del motor. Se solucionó cambiando de lugar la salida del cableado.

Las contribuciones del equipo son las siguientes:

- Ampliación del campo de visión de la cámara trampa.
- Reducción de jornadas de catalogación de imágenes.
- Ampliación del área captada en situación nocturna.
- Reducción de la contaminación ambiental.
- Evitación de pérdida de capturas por almacenamiento lleno.
- Evitación de traslados innecesarios para cambios de configuración del sistema.
- Control de temperatura y la tensión del equipo para evitar daños.

El uso de un sistema de posicionamiento permite ampliar la cantidad de capturas de animales, sin la necesidad de aumentar el número de cámaras

presentes en el terreno de estudio, lo cual permite reducir los costos del estudio. Muchas cámaras trampa comerciales aumentan el campo de visión utilizando lentes gran angulares, los mismos generan distorsión, dificultando la identificación de las especies por parte del operario y dificultando la utilización de modelos de visión artificial.

Además, gran parte de las cámaras trampa comerciales presentan especificaciones engañosas, como el alcance de los sensores y de la iluminación nocturna, las resoluciones permitidas, el retardo entre disparos, la duración de las baterías, entre otras.

Otra característica importante que presentan muy pocas cámaras trampa es el control a distancia y la verificación del funcionamiento a través de internet, las cuales son funcionalidades útiles para investigadores que poseen múltiples equipos en distintos terrenos en estudio.

8.1 Trabajos Futuros

Serían de gran utilidad, la utilización de comunicación inalámbrica entre el equipo y los sensores de movimiento para reducir el cableado del equipo. Además, la integración de un módulo GPS permitiría el ajuste automático de la hora del sistema cuando el mismo se encuentre sin conexión a internet.

Capítulo 9: Literatura Citada

- [1] MMA & Ambiente, ONU & CONAF, & LAGOS, Nicolas., MANUAL DE USO DE TRAMPAS CÁMARA para el monitoreo de carnívoros nativos y exóticos, Santiago: Equipo Proyecto GEF Corredores Biológicos de Montaña, 2022.
- [2] A. y. E. P. G. Díaz-Pulido, Manual de fototrampeo: una herramienta de investigación para la conservación de la biodiversidad en Colombia, Instituto de Investigación de Recursos Biológicos Alexander von Humboldt, 2012.
- [3] J. & B. A. García-Grajales, «Las cámaras trampa y su avance tecnológico en favor de la conservación,» *Ciencias Marinas*, vol. XXII, nº 65, pp. 53-61, 2018.
- [4] E. L. H. -. P. K. S.-P. J. F. M. -. R. N. M. R. Á. R. -. Hurtado, «Fototrampeo: Descubriendo lo que no podemos ver,» *ECOFRONTERAS*, vol. 21, nº 61, pp. 26-29, 2017.
- [5] C. A. d. I. T. H. B. R. M. H. Z. y. G. C. Chávez, Manual de fototrampeo para estudio de fauna silvestre. El jaguar en México como estudio de caso, Ciudad de México: Universidad Nacional Autónoma de México, 2013.
- [6] A. W. C. D. J. P. & A. L. Dustin J. Welbourne, «How do passive infrared triggered camera traps operate and why does it matter? Breaking down common misconceptions,» *Remote Sens Ecol Conserv*, vol. II, nº 2, pp. 77-83, 2016.
- [7] O. R. W. & P. Glover-Kapfer, «Camera-trapping for conservation: a guide to best-practices,» WWF Conservation Technology Series, Woking, United Kingdom, 2017.
- [8] B. K. A. A. N. J. W. A. F. Seidlitz Anke, «Animal detections increase by using a wide-angle camera trap model but not by periodically repositioning camera traps within study sites,» *Pacific Conservation Biology*, nº 28, pp. 25-35, 2022.
- [9] Z. J. M. S. J. D. Fan Jinlong, «Wide-Angle Image Rectification: A Survey,» *International Journal of Computer Vision*, vol. 130, pp. 747-776, 2022.
- [10] F. E. A. N. M. R. W. C. M. S. R. K. Delisle Zackary J., «Next-Generation Camera Trapping: Systematic Review of Historic Trends Suggests Keys to Expanded Research Applications in Ecology and Conservation,» *Frontiers in Ecology and Evolution*, vol. 9, 2021.
- [11] P. a. C. S. K. a. S. V. K. a. G. T. N. Singh, «Motion detection and tracking using microwave sensor for eliminating illegal mine activities,» de 2018 *3rd International Conference on Microwave and Photonics (ICMAP)*, Dhanbad, India, 2018.

- [12] M. Adamu, «Design and Development of an Ultrasonic Motion Detector,» *International Journal of Security, Privacy and Trust Management*, vol. 2, 2013.
- [13] H. M. A. P. C. B. C. K. Tirupal Talari, «The Influence of Ultrasound for the Protection of Animals on Highways,» vol. 5, pp. 15-20, 2020.
- [14] OMRON, «Technical Explanation for Photoelectric Sensors».
- [15] S. C. K. S. G. a. P. C. Priya Sharma, «Neural Network Based Image Classification for Animal Intrusion Detection System,» *International Journal of Progressive Research In Science and engineering*, vol. 1, nº 4, pp. 1-7, 2020.
- [16] G. K., *An introduction to neural networks*, London: CRC Press, 2018.
- [17] J. J. E. O. G. G. D. R. J. Carlos J., «Desarrollo de un modelo de regresión con redes neuronales artificiales para estimar la resistencia rotórica de un motor de inducción,» 2022.
- [18] A. M. Á., «Clasificación de imágenes usando redes neuronales convolucionales en Python,» Sevilla, 2019.
- [19] R. P. P. D. P.-R. J. Alvear P. Vanessa, «Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura (Internet of Things and Artificial Vision, Performance and Applications: Literature Review,» *Enfoque*, vol. 8, pp. 244-256, 2017.
- [20] I. Rubio, G. Fondato y L. Esnaola, «Clasificación de objetos en imágenes usando redes convolucionales,» de *XXVII Jornadas de Jóvenes Investigadores de la Asociación de Universidades Grupo Montevideo – AUGM*, São Carlos, Brasil, 2019.
- [21] A. M. Castejón, «Aplicación de Deep Learning para problemas de clasificación de imágenes,» Madrid, 2019.
- [22] A. S. González, «Estudio comparativo de modelos de clasificación automática de señales de tráfico,» Pamplona, España, 2020.
- [23] S. G. M. S. Tawfeeq M., «Automatic Medical Images Segmentation Based on Deep Learning Networks,» 2020.
- [24] G. Martín de las Puebas Encinas, «Definición de Requisitos Funcionales bajo Especificación IEEE para un Sistema de Ingeniería,» Madrid, 2019.
- [25] W. A. Gardner M., *Manual básico de fotografía al aire libre*, Madrid: Ediciones Desnivel, 2005.
- [26] B. H. Álvarez A. J., «Understanding the role of single-board computers in engineering and computer science education: A systematic literature review,» *Computer Applications in Engineering Education*, vol. 30, pp. 304-329, 2022.
- [27] M. J. D. D. E. Álvarez J. L., «Analysis of Single Board Architectures Integrating Sensors Technologies,» *Sensors*, vol. 21, nº 8, 2021.

- [28] P. N. P., «A comparative analysis of Raspberry pi Hardware with Aduino, Phidgets, Beaglebone black and Udo», *International Research Journal of Engineering and Technology*, vol. 3, nº 6, 2016.
- [29] Raspberry Pi, «Raspberry Pi 3 Model A plus», [En línea]. Available: <https://www.raspberrypi.com/products/raspberry-pi-3-model-a-plus/>. [Último acceso: 01 12 2022].
- [30] J. J. W., «Broad-scale applications of the Raspberry Pi: A review and guide for biologists», *Methods in Ecology and Evolution*, vol. XXII, nº 9, pp. 1562-1579, 2021.
- [31] Arduino.cc, «Arduino nano», [En línea]. Available: <https://store.arduino.cc/products/arduino-nano>. [Último acceso: 01 12 2022].
- [32] P. A. R., *Sensores y Acondicionadores de señal*, Cuarta ed., Marcombo, 2005.
- [33] Digi-Key Electronics, «digikey.com», [En línea]. Available: <https://www.digikey.com/en/products/detail/epcos-tdk-electronics/B57891M0103K000/3500546>. [Último acceso: 02 12 2022].
- [34] Kollmorgen, «www.kollmorgen.com», 08 Junio 2016. [En línea]. Available: https://www.kollmorgen.com/sites/default/files/Servo_or_Stepper_08_05_16_WhitePaper_FINAL.pdf. [Último acceso: 2022 12 02].
- [35] Axis Communications, «IR in surveillance Day-and-night cameras and OptimizedIR», Lund, Suecia, 2018.
- [36] DEMASLED, «dled.com.ar», [En línea]. Available: <https://www.dled.com.ar/led-high-power-infrarrojo-850nm-500mw-3v-3w-700ma-hp3ir850/>. [Último acceso: 02 12 2022].
- [37] Raspberry Pi, «Raspberry Pi Camera Module 2 NoIR», [En línea]. Available: <https://www.raspberrypi.com/products/pi-noir-camera-v2/>. [Último acceso: 02 12 2022].
- [38] AliExpress, «TREEYE-filtro de corte IR M12 de 5,0 megapíxeles, conmutador doble ICR, IR-CUT», [En línea]. Available: <https://es.aliexpress.com/item/1005001859424438.html>. [Último acceso: 02 12 2022].
- [39] a. o. Sanner Michel F., «Python - the fastest growing programming language», *International Research Journal of Engineering and Technology*, vol. IV, nº 12, pp. 354-357, 2017.
- [40] Python, «The Python Logo», [En línea]. Available: <https://www.python.org/community/logos/>. [Último acceso: 05 12 2022].
- [41] P. G., «The Python Papers», *The Python Papers*, vol. III, nº 1, pp. 26-37, 2008.

- [42] Qt, «Getting to know Qt Designer,» [En línea]. Available: <https://doc.qt.io/qt-6/designer-to-know.html>. [Último acceso: 12 12 2022].
- [43] R. M. R. M. T. C. K. Rolon Mérette D., «Introduction to Anaconda and Python: Installation and setup,» *Python for research in psychology*, vol. XVI, nº 5, pp. 55-511, 2016.
- [44] D. S. B. M. T. V. L. G. Á. H. I. M. P. H. L. Nguyen G., «Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey,» *Artificial Intelligence Review*, vol. LII, nº 1, pp. 77-124, 2019.
- [45] M. N. Kumar, «Understanding and Working with Keras,» de *Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras*, Berkeley, CA, Apress, 2018, pp. 31-43.
- [46] G. J. A. G. Arévalo V., «La Librería de visión artificial opencv. Aplicación a la docencia e investigación,» *Base Informática*, vol. XL, pp. 61-66, 2004.
- [47] M. K. Wes, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*, O'Reilly Media, Inc., 2012.
- [48] B. L. L. G. G. O. P. F. M. A. Varoquaux G., «Scikit-learn: Machine learning without learning the machinery,» *GetMobile: Mobile Computing and Communications*, vol. XIX, nº 1, pp. 29-32, 2015.
- [49] P. J. D. O. Srasa C. C. G., «Técnicas de aumento de datos para imágenes aéreas y evaluación de rendimiento en modelos de deep learning,» *Revista Universidad Católica de Oriente*, vol. XXXI, nº 45, pp. 100-115, 2020.
- [50] Optex, «Optex-europe,» [En línea]. Available: <https://www.optex-europe.com/es/productos/deteccion-de-intrusiones/rxc-st>. [Último acceso: 07 12 2022].
- [51] Zooplus, «Zooplus Magazine,» [En línea]. Available: <https://www.zooplus.es/>. [Último acceso: 31 01 2023].
- [52] tp-link, «TL-SF1008D Switch de sobremesa con 8 puertos a 10/100 Mbps,» [En línea]. Available: <https://www.tp-link.com/es/home-networking/soho-switch/tl-sf1008d/>. [Último acceso: 08 12 2020].
- [53] Yodeck, «Raspberry Pi Comparison,» 02 Julio 2019. [En línea]. Available: <https://www.yodeck.com/docs/creating-a-yodeck-player/raspberry-pi-comparison/>. [Último acceso: 01 12 2022].

Capítulo 10: Anexos

Anexo 1: Cuadro comparativo de versiones de Raspberry Pi

Family	Model A										Zero			Compute		
	Model B					Model A					Zero			Compute		
Introduction	The Raspberry Pi Model B is the flagship Raspberry Pi Model, with the fastest CPU and most connectivity features. The latest revision (Raspberry Pi 5 - Model B) features a 4.4GHz Quad Core processor, and on-board 2x15 Pin WiFi & Bluetooth. The Model B is the largest Pi available with the approximate size of a credit card.										The Raspberry Pi Zero is an ultra low cost and ultra small variant of the Raspberry Pi. It comes with an unpopulated GPIO header, and only a single micro USB port for connectivity. The Raspberry Pi Zero W (Wireless) includes on-board WiFi and Bluetooth.			The Compute Module features a 200 Pin 1202x-SDMMC layout, and is designed for engineers who need Raspberry Pi functionality but need a smaller form factor for embedded designs.		
	Model	Raspberry Pi Model B	Raspberry Pi Model B+	Raspberry Pi 2 Model B	Raspberry Pi 2 Model B+	Raspberry Pi 3 Model B	Raspberry Pi 3 Model B+	Raspberry Pi 4 Model A	Raspberry Pi 4 Model A+	Raspberry Pi 5 Model A	Raspberry Pi 5 Model A+	Raspberry Pi Zero v1.2	Raspberry Pi Zero v1.3	Raspberry Pi Zero Wireless	Compute Module 3	Compute Module 3 Lite
Release Date	Mar-12	Jun-14	Oct-15	Feb-16	Mar-16	Mar-16	Feb-17	Nov-17	Nov-18	Nov-18	Nov-18	May-19	Feb-17	Jan-17	Jan-17	
Latest Version	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Legacy (In Production)	Latest Version	Latest Version	Latest Version	Latest Version	
RRP (USD)	\$35.00	\$35.00	\$35.00	\$35.00	\$35.00	\$35.00	\$25.00	\$25.00	\$25.00	\$25.00	\$5.00	\$5.00	\$10.00	\$30.00	\$30.00	
SOC Type	Broadcom BCM2835 ARMv6 (23-bit) (64/23-bit)										Broadcom BCM2835 ARMv6 (23-bit) (64/23-bit)			Broadcom BCM2835 ARMv6 (23-bit) (64/23-bit)		
Core Architecture	Broadcom VideoCore IV										Broadcom VideoCore IV			Broadcom VideoCore IV		
No. of Cores	4										4			4		
GPU	700 MHz Single Core ARMv6										700 MHz Single Core ARMv6			700 MHz Single Core ARMv6		
CPU Clock	1 GHz										1 GHz			1 GHz		
RAM	512 MB										512 MB			512 MB		
USB Ports	2 x USB										1 x Micro USB (OTG)			1 x Micro USB (OTG)		
Ethernet	10/100Mbit/s Ethernet Port										No			No		
Wi-Fi	No										No			On Board WiFi 802.11n		
Bluetooth®	No										No			On Board Bluetooth 2.0/4.1		
Video Output	HDMI 3.5mm Composite DSI (for LCD)										HDMI 3.5mm Composite DSI (for LCD)			HDMI 3.5mm Composite DSI (for LCD)		
Audio Output	I2S 3.5mm Jack										I2S 3.5mm Jack			I2S 3.5mm Jack		
Camera Input	15 Pin CSI										15 Pin CSI			15 Pin CSI		

Figura 81. Modelos y versiones de Raspberry Pi [53].

Anexo 3: Instalación y configuración de programas en Raspberry

Para poder utilizar el Raspberry se le debe instalar un Sistema Operativo. Existen múltiples versiones en la página oficial de Raspberry Pi. En este caso se utilizó la opción “Raspberry Pi OS with desktop”.

El sistema operativo se debe instalar en la tarjeta SD del Raspberry utilizando el programa BalenaEtcher, este mismo escribe imágenes en dispositivos de almacenamiento.

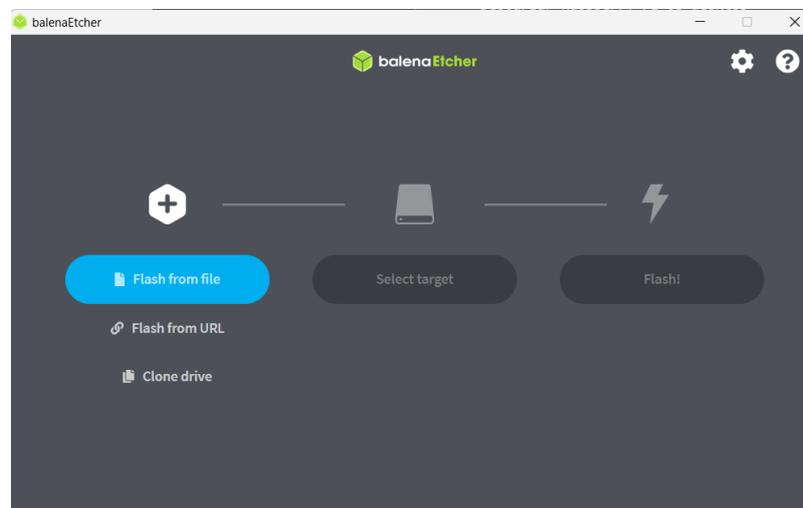


Figura 83. Captura de interfaz de software BalenaEtcher.

Una vez instalada la imagen en la SSD, se conecta una pantalla por HDMI y el teclado al Raspberry por USB y se realiza la configuración inicial de fecha y hora, la configuración de red y se actualiza.

Finalizada la actualización, se accede a la ventana de comandos y se agrega el comando **sudo raspi-config** para configurar el sistema.

Ingresando al apartado de Interface Options, se configuran los siguientes parámetros:

- Legacy Camera: Enable
- SSH: Enable

- VNC: Enable
- Remote GPIO: Enable

Se procede entonces a reiniciar el sistema, para luego realizar la instalación de las siguientes librerías:

Librerías de Vnc Viewer para acceso remoto:

```
sudo apt-get install -y realvnc-vnc-server realvnc-vnc-viewer
```

Librerías para software de Cámara Trampa

```
sudo apt install python3-opencv  
sudo apt-get install libatlas-base-dev  
sudo apt-get install libhdf5-dev  
sudo apt-get install libhdf5-serial-dev  
sudo apt-get install libjasper-dev  
sudo apt-get install libqtgui4  
sudo apt-get install libqt4-test  
pip install -U numpy
```

Librerías para conexión Bluetooth:

```
sudo apt install network-manager-gnome blueman
```

Configuración de VNC Viewer:

- i. Se instala el programa VNC Viewer en la computadora, y se registra una cuenta la página oficial de RealVNC y en la pestaña Account se activa una cuenta Home.

- ii. Se ingresa en VNC Viewer desde la computadora y se inicia sesión con la cuenta creada. Una vez realizado este proceso, aparecerá una nueva sección dentro de la aplicación de VNC, con el nombre Home.
- iii. Desde Raspberry se ingresa al icono blanco de VNC Viewer de la barra de tareas con click derecho, se selecciona aplicación de licencias y se inicia sesión.



Figura 84. Icono de VNC Viewer.

- iv. Se observa la red creada:

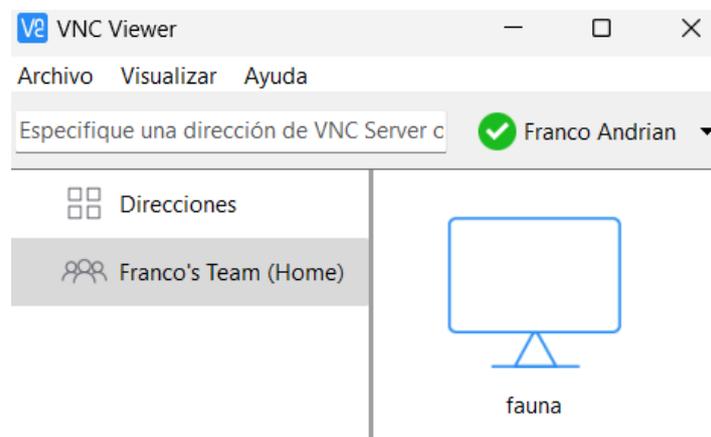


Figura 85. Nueva red para acceso por servidor de VNC Viewer.

Configuración de inicio automático de software de Cámara Trampa:

Para la configuración del lanzado automático de la aplicación una vez encendido el sistema, se escriben los siguientes comandos en consola:

- i. Cambio de permisos a interfaz de cámara trampa:

```
sudo chmod 755 /home/fauna/Desktop/camaraTrampa/InterfazRPi.pyw
```

- ii. Se crea el directorio y los archivos “comandosIniciales.sh” y “arranque.sh” y se modifican sus permisos:

```
sudo mkdir /home/fauna/Autorun
```

```
cd /home/fauna/Autorun
sudo chmod 777 /home/fauna/Autorun
sudo cat > comandosIniciales.sh
sudo chmod +x comandosIniciales.sh
sudo cat > arranque.sh
sudo chmod +x arranque.sh
```

- iii. Desde la interfaz del sistema, se ingresa al archivo comandosIniciales.sh y se escriben las siguientes líneas:

```
#!/bin/bash
sudo raspi-config nonint do_wifi_country AR
echo vcgencmd display_power 0
python /home/fauna/Desktop/camaraTrampa/InterfazRPi.pyw
```

- iv. Nuevamente desde la interfaz del sistema, se accede al archivo arranque.sh y se escriben las siguientes líneas:

```
#!/bin/bash
Lxterminal -t Log_monitor --command="/bin/bash -c
'/home/fauna/Autorun/comandosIniciales.sh; /bin/bash'"
```

- v. Desde consola, se abre el archivo "autostart" de LXDE-pi:

```
sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
```

- vi. Se agrega la siguiente línea al final del archivo:

```
@Lxterminal -e /home/fauna/Autorun/arranque.sh
```

- vii. Se dan permisos especiales al directorio media y se crea el directorio fauna:

```
cd
sudo chmod 777 /media
cd /media
mkdir fauna
```

Finalizada la configuración, se reinicia el sistema para comprobar que la interfaz se lance correctamente.

Configuración de bluetooth:

Al instalar las librerías de bluetooth, se puede encontrar un nuevo icono de bluetooth blanco y azul, siendo este el programa instalado.



Figura 86. Icono de bluetooth Blueman.

- i. Activamos el Bluetooth en el dispositivo a conectar con el Raspberry.
- ii. Damos click derecho en el icono azul de bluetooth y realizamos la siguiente configuración:

Adaptadores → Configuración de Visibilidad: Siempre visible

Servicios locales → Red → Servicios: Punto de acceso de la red (NAP)

→ Ajustes NAP: dnsmasq

→ Dirección IP: 10.1.1.1

→ Compatibilidad PAN y DUN: Blueman

Dispositivos → Buscar: Seleccionamos dispositivo, los emparejamos y en pestaña de conexión marcamos Punto de acceso de la red.

- iii. Ingresamos a VNC Viewer desde la computadora y se crea una nueva conexión desde Archivo → Nueva conexión, y se ingresan los siguientes datos:

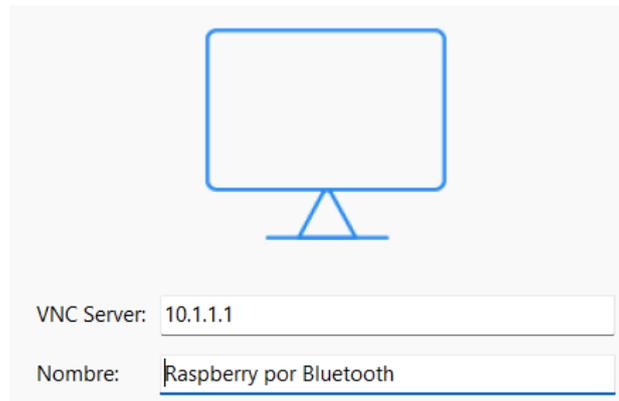


Figura 87. Creación de nueva red para acceso por bluetooth.

Para crear la conexión se debe ingresar la contraseña de la cuenta de VNC Viewer detallada anteriormente.

- iv. Se empareja la computadora con el Raspberry, y haciendo click derecho en el icono de bluetooth se selecciona “Unirse a una red de área local”, lo cual abre una pestaña de dispositivos e impresoras.
- v. En el icono de Raspberry se da click derecho → conectar mediante → Punto de acceso. Una vez realizada dicha configuración, se accede al raspberry a través de VNC Viewer con la red creada anteriormente.

Anexo 4: Instalación y configuración de programas en computadora

Para la instalación y gestión de paquetes en la computadora en la que se realizará el entrenamiento y la catalogación de imágenes, se utiliza el programa Anaconda, el cual se descarga desde su página oficial.

Con la instalación de Anaconda, se instala Anaconda Navigator y Anaconda Prompt.

Anaconda Navigator es la interfaz gráfica de usuario utilizada para administrar los paquetes sin tener que utilizar líneas de comando, mientras que Anaconda Prompt es la consola de comando integrada.

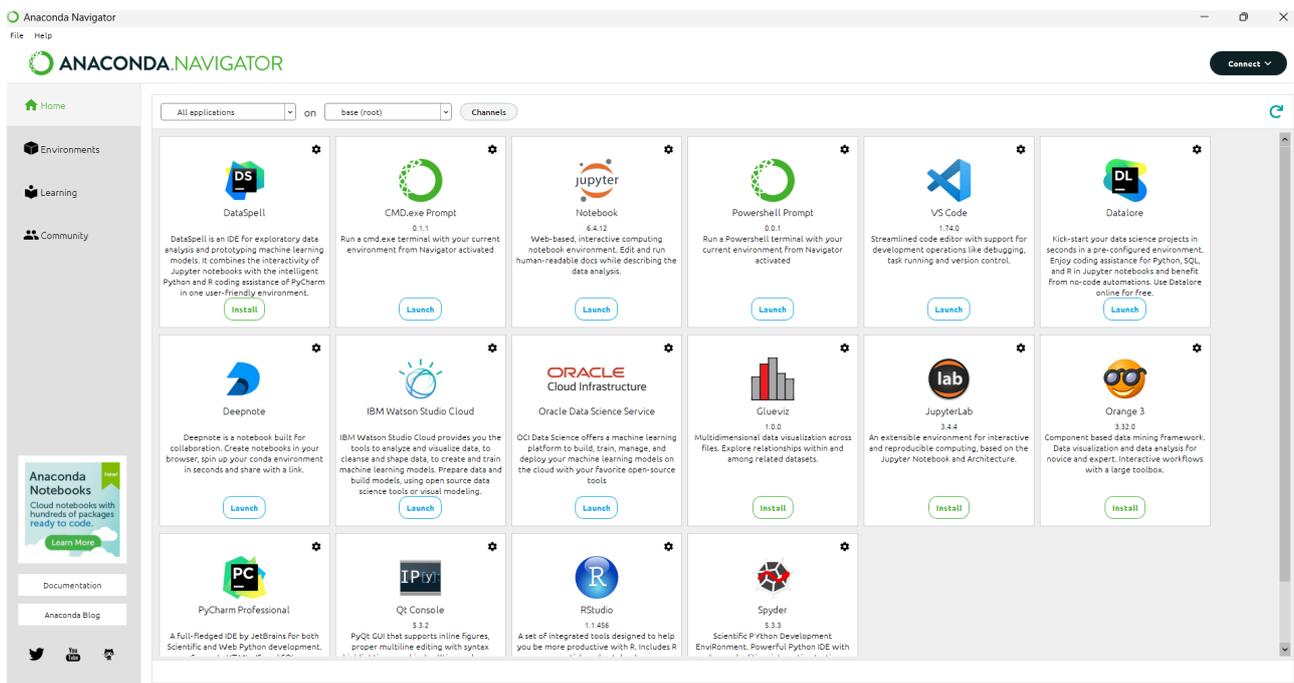


Figura 88. Anaconda Navigator: Interfaz Gráfica de Anaconda.

A continuación se enlistan los pasos para la instalación y configuración de los complementos necesarios para el entrenamiento del modelo de visión artificial haciendo uso de la placa de video integrada:

- i. Se ingresa a la consola Anaconda Prompt y se instalan las librerías requeridas:

```
pip install --upgrade jupyterhub
pip install --upgrade --user nbconvert
conda create -n gpu python==3.7.7
conda activate gpu
conda install ipykernel
python -m ipykernel install --user --name gpu --display-name "gpu"
conda install tensorflow-gpu==2.1.0 cudatoolkit=10.1
pip install tensorflow==2.1.0
pip install jupyter
pip install keras==2.3.1
pip install numpy scipy Pillow cython matplotlib scikit-image opencv-
python h5py imgaug IPython[all]
pip install sklearn
pip3 install -U scikit-learn scipy matplotlib --user
pip install mlxtend
```

- ii. Se ingresa al archivo `saving.py` y se borran todas las apariciones de `".decode('utf-8')"`, el mismo se encuentra en la siguiente ruta:

```
"~\anaconda3\envs\gpu\Lib\site-packages\keras\engine\saving.py"
```

- iii. Se testea si se detecta la placa con los siguientes comandos:

```
python
import tensorflow as tf
tf.__version__
tf.test.gpu_device_name()
```

- iv. Se obtiene como resultado una descripción similar a la siguiente:

```
2022-12-08 04:37:14.569077: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1241] Created TensorFlow device
(/device:GPU:0 with 2903 MB memory) -> physical GPU (device: 0, name: NVIDIA GeForce GTX 1650, pci bus id: 00
00:01:00.0, compute capability: 7.5)
'/device:GPU:0'
```

Figura 89. Detección de placa de video NVIDIA GeForce GTX 1650.

El valor de cudatoolkit puede variar según la placa de video utilizada. Los pasos para la instalación de cudatoolkit son los siguientes:

1. Actualizar controlador Geforce Game Ready (Versión: 516.59 WHQL).
2. Según la versión de del controlador, se obtienen las versiones de CUDNN (7.6) y de CUDA (10.1) a instalar.

La librería de Arquitectura Unificada de Dispositivos de Computo (CUDA acrónimo del inglés, Compute Unified Device Architecture) es una librería genérica para realización de cálculos con la placa de video, mientras que la biblioteca de Red Neuronal Profunda CUDA de NVIDIA (CUDNN acrónimo del inglés, NVIDIA CUDA Deep Neural Network) posee primitivas para la aceleración de GPU en aplicaciones de redes neuronales.