



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE TELECOMUNICACIONES**

**Tema:**

---

**SISTEMA ELECTRÓNICO DE CONTROL Y MONITORIZACIÓN DE  
VEHÍCULOS EN CASO DE ROBO PARA LA EMPRESA CONFECCIONES  
GALAR MEDIANTE VISIÓN ARTIFICIAL**

---

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la  
obtención del título de Ingeniero en Telecomunicaciones

**ÁREA:** Electrónica

**LÍNEA DE INVESTIGACIÓN:** Tecnología de la información y Sistemas de  
control

**AUTOR:** Fernando Paul Lema Galarza

**TUTOR:** Ing. Julio Enrique Cuji Rodríguez, Mg.

**Ambato - Ecuador**

**febrero – 2024**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del trabajo de titulación con el tema: SISTEMA ELECTRÓNICO DE CONTROL Y MONITORIZACIÓN DE VEHÍCULOS EN CASO DE ROBO PARA LA EMPRESA CONFECCIONES GALAR MEDIANTE VISIÓN ARTIFICIAL desarrollado bajo la modalidad Proyecto de Investigación por el señor Fernando Paul Lema Galarza, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

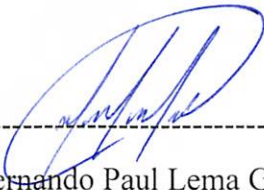
-----  
Ing. Julio Enrique Cuji Rodríguez, Mg.

**TUTOR**

## AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA ELECTRÓNICO DE CONTROL Y MONITORIZACIÓN DE VEHÍCULOS EN CASO DE ROBO PARA LA EMPRESA CONFECCIONES GALAR MEDIANTE VISIÓN ARTIFICIAL es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



Fernando Paul Lema Galarza

C.C. 1804383204

AUTOR

## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



Fernando Paul Lema Galarza

C.C. 1804383204

AUTOR

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Fernando Paul Lema Galarza, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA ELECTRÓNICO DE CONTROL Y MONITORIZACIÓN DE VEHÍCULOS EN CASO DE ROBO PARA LA EMPRESA CONFECCIONES GALAR MEDIANTE VISIÓN ARTIFICIAL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

-----  
Ing. Elsa Pilar Urrutia Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----  
Ing. Edgar Córdova Córdova, Mg.  
PROFESOR CALIFICADOR

-----  
Ing. Jesús Guamán Molina, Mg.  
PROFESOR CALIFICADOR

## DEDICATORIA

*Este proyecto de investigación está dedicado con profundo agradecimiento y cariño a dos pilares fundamentales en mi vida, mi padre Enrique y mi querida madre Gloria. Su presencia constante y apoyo incondicional han sido mi mayor fortaleza a lo largo de este trayecto académico.*

*A mi querido hermano Diego, le dedico este logro con gratitud y afecto. Su compañía y aliento han constituido un núcleo esencial en mi vida, inspirándome en este camino de aprendizaje.*

*Expreso mi dedicación a mis abuelitos, Blanca y Segundo, quienes han tenido un papel significativo en mi vida.*

*A mis tíos y primos, les agradezco por su cariño y alegría, iluminando mi camino y llenando mi corazón de felicidad.*

**Fernando**

## AGRADECIMIENTO

*En primer lugar, expreso mi agradecimiento a Dios por otorgarme salud, sabiduría y la fuerza necesaria para afrontar los desafíos y continuar persiguiendo mis aspiraciones. Quiero dedicar un agradecimiento infinito a mis padres, en especial a mi Madre Gloria quien con su amor incondicional y ejemplo de superación han constituido la base sobre la cual he edificado mi camino hacia el éxito. Cada paso que he dado ha estado respaldado por la certeza de que ella estaba presente para respaldarme y alentarme en cada elección.*

*A mi hermano Diego, les agradezco por ser mi sólido apoyo a lo largo de mi trayectoria universitaria. Él es mi mayor inspiración en la vida, y siempre me ha apoyado en mis momentos más difíciles.*

*A mis abuelitos, tíos y primos, les agradezco su apoyo, me han alentado desde pequeño y siempre estuvieron en cada paso de mi vida.*

*A mis amigos, les estoy agradecido por proporcionarme la fortaleza y la motivación necesarias para avanzar en este camino, en especial a Vinicio quien es un querido amigo, así como por su respaldo incondicional, gracias a ustedes he pasado momentos muy alegres.*

*A una de las personas especiales en mi vida, Susana le estoy agradecido por su apoyo y motivación en el transcurso de estos años y de los que vendrán.*

*Quiero expresar mi gratitud de manera especial al Ing. Julio Cují, mi tutor de tesis. Además de ser mi tutor fue quien me guio y me respaldo para seguir adelante con su orientación y conocimiento a lo largo de este proceso de investigación que han sido cruciales para alcanzar este logro, le debo mi gratitud eterna.*

**Fernando**

## ÍNDICE GENERAL DE CONTENIDOS

<b>APROBACIÓN DEL TUTOR</b> .....	<b>ii</b>
<b>AUTORÍA</b> .....	<b>iii</b>
<b>DERECHOS DE AUTOR</b> .....	<b>iv</b>
<b>APROBACIÓN DEL TRIBUNAL DE GRADO</b> .....	<b>v</b>
<b>DEDICATORIA</b> .....	<b>vi</b>
<b>AGRADECIMIENTO</b> .....	<b>vii</b>
<b>ÍNDICE GENERAL DE CONTENIDOS</b> .....	<b>viii</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>xiv</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>xvi</b>
<b>ÍNDICE DE ANEXOS</b> .....	<b>xx</b>
<b>RESUMEN EJECUTIVO</b> .....	<b>xxi</b>
<b>ABSTRACT</b> .....	<b>xxii</b>
<b>CAPÍTULO I. MARCO TEÓRICO</b> .....	<b>1</b>
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema .....	1
1.2 Antecedentes investigativos .....	3
1.3 Fundamentación teórica .....	6



1.3.1 Vehículos .....	6
1.3.2 Robos	6
1.3.3 Robo a vehículos .....	6
1.3.4 Indicadores de robo de vehículos .....	7
1.3.5 Métodos de robos a vehículos .....	8
1.3.6 Sistema de Monitorización.....	8
1.3.7 Sistema de Control .....	9
1.3.8 Procesamiento de imágenes .....	9
1.3.9 Sistemas de detección de robo a vehículos .....	11
1.3.10 Visión Artificial .....	11
1.3.11 Características de la Visión Artificial .....	12
1.3.12 Seguimiento de Objetos Mediante Visión Artificial.....	14
1.3.13 Procesamiento Digital de Señales .....	15
1.3.14 Integración de la visión artificial en el procesamiento digital de señales. ....	16
1.3.15 Fundamentos del procesamiento digital de imágenes y videos .....	17
1.3.16 Redes Neuronales.....	18
1.3.17 Redes Neuronales Convolucionales.....	18
RELU .....	20
POOLING.....	21
Capas Conectadas.....	22
1.4 Objetivos .....	23

1.4.1 Objetivo general.....	23
1.4.2 Objetivos específicos .....	23
<b>CAPÍTULO II. METODOLOGÍA .....</b>	<b>24</b>
2.1 Materiales.....	24
2.2 Métodos.....	24
2.2.1 Modalidad de la investigación .....	24
2.2.2 Población y muestra .....	25
2.2.3 Recolección de información.....	25
2.2.4 Procesamiento y análisis de datos .....	25
<b>CAPÍTULO III.- RESULTADOS Y DISCUSIÓN .....</b>	<b>27</b>
3.1 Desarrollo de la propuesta.....	27
3.2 Equipos de seguridad electrónica para la prevención de robo que existen en el mercado automotriz.....	27
3.2.1 Equipos de seguridad electrónica existentes .....	27
3.2.2 Equipos de seguridad electrónica para vehículos en el Ecuador.....	30
3.2.3 Expresión corporal .....	31
3.3 Arquitectura.....	33
3.4 ETAPA 1.- Adquisición .....	37
3.5 Desarrollo del sistema.....	37

3.5.1 Adquisición de Imágenes .....	37
3.5.2 Selección de cámara.....	38
3.5.3 Comparativa de Cámaras .....	41
3.6 ETAPA 2.- Procesamiento .....	42
3.6.1 Librerías o Bibliotecas requeridas.....	42
3.6.2 Selección del Microcontrolador .....	44
3.6.3 Selección del Enrutador .....	46
3.6.4 Selección del respaldo (Baterías).....	48
3.6.5 Periféricos de entrada y salida.....	50
3.6.6 Selección de la tecnología inalámbrica .....	51
3.7 ETAPA 3.- Entrenamiento .....	52
3.7.1 Software .....	52
3.7.2 Comparativa de cada Software.....	52
3.7.3 Algoritmo de la red neuronal .....	54
3.7.4 Selección del servidor .....	57
3.7.5 Comparativa de plataformas del Servidor.....	58
3.7.6 Selección del protocolo de comunicación.....	60
3.7.7 Selección de Data Base .....	61
3.7.8 Selección de la herramienta de visualización.....	62
3.8 ETAPA 4.- Visualización en la Interfaz Grafica .....	62

3.8.1	Instanciación en la Nube .....	62
3.8.2	Activación del servidor de Cloud de Microsoft Azure .....	64
3.8.3	Acceder al tráfico de los puertos en Azure .....	66
3.8.4	Instalación del Servidor Mosquitto .....	67
3.8.5	Instalación de Apache Server.....	68
3.8.6	Instalación de la base de datos usando PHP MyAdmin.....	69
3.8.7	Instalación, creación y acceso a Node-RED .....	70
3.9	Conexión con PHP Myadmin.....	71
3.9.1	Ajuste de Node-Red .....	72
3.9.2	Configuración de Node-RED para la conexión a la base de datos .....	73
3.9.3	Configuración de Node-Red para registrar la información generada por la Raspberry Pi 3 Modelo B+.....	74
3.10	Entrenamiento de la red.....	79
3.10.1	Redes Neuronales Convolucionales.....	82
3.10.2	Programación del algoritmo en Python.....	92
3.11	Funcionamiento del sistema .....	93
3.12	Evaluación de las pruebas de detección de robo .....	95
3.12.1	Resultados .....	97
3.12.2	Presupuesto .....	100
<b>CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>103</b>
4.1	Conclusiones .....	103

4.2 Recomendaciones.....	104
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>105</b>
<b>ANEXOS .....</b>	<b>109</b>

## ÍNDICE DE TABLAS

Tabla 1. Indicadores de robo de vehículos en Ecuador .....	7
Tabla 2. Etapas de la arquitectura de sistemas electrónicos para el procesamiento de videos enfocada a la visión artificial .....	11
Tabla 3. Aplicaciones de la integración de la visión artificial en el procesamiento digital de señales .....	16
Tabla 4. Fundamentos del procesamiento digital de imágenes .....	17
Tabla 5. Gestos corporales en caso de ser el asaltante .....	31
Tabla 6. Gestos corporales en caso de ser la víctima .....	32
Tabla 7. Problemas de captura de video e imagen .....	33
Tabla 8. Características principales de cada cámara .....	41
Tabla 9. Librerías utilizadas en el sistema .....	43
Tabla 10. Comparación de diferentes tipos de microcontroladores .....	44
Tabla 11. Parámetros técnicos de cada enrutador .....	47
Tabla 12. Comparación de diferentes tipos de baterías .....	49
Tabla 13. Tecnologías Inalámbricas .....	51
Tabla 14. Selección de software para procesamiento de imágenes. ....	52
Tabla 15. Comparación de las características técnicas de cada software que estará enfocada a la Visión Artificial. ....	53
Tabla 16. Algoritmos de la red neuronal .....	55
Tabla 17. Comparativa de las características principales de cada plataforma de Cloud Computing .....	59

Tabla 18. Parámetros Técnicos de cada protocolo de comunicación.....	60
Tabla 19. Parámetros Técnicos de cada Data Base.....	61
Tabla 20. Imágenes recolectadas.....	80
Tabla 21. Capas del modelo de la Red Convolutiva .....	84
Tabla 22. Comparación de los resultados de las pruebas de las imágenes .....	90
Tabla 23. Entrenamiento a 100 épocas. ....	91
Tabla 24. Matriz de confusión del sistema.....	97
Tabla 25. Descripción detallada de los costos de los materiales utilizados en la construcción de sistemas electrónicos.....	102

## ÍNDICE DE FIGURAS

Figura 1. Diagrama de Bloques Sistema de Control .....	9
Figura 2. Etapas de los sistemas electrónicos para el procesamiento de imagen y video.....	10
Figura 3. Visión Artificial.....	12
Figura 4. Sistema de Procesamiento de Señales. ....	16
Figura 5. Estructura de la Red Neuronal Convolutacional .....	19
Figura 6. Operaciones de la convolución.....	20
Figura 7. Propagación de la activación RELU.....	21
Figura 8. Dimensiones de operación de Pooling.....	22
Figura 9. Max Pooling .....	22
Figura 10. Capas de la Red Neuronal Conectada.....	23
Figura 11. Arquitectura .....	34
Figura 12. Arquitectura de adquisición de datos.....	34
Figura 13. Arquitectura de procesamiento de datos del sistema.....	34
Figura 14. Arquitectura de entrenamiento y almacenamiento en la nube.....	35
Figura 15. Arquitectura de servicio y visualización en la interfaz gráfica.....	35
Figura 16. Diagrama de bloques del sistema .....	36
Figura 17. Diagrama de flujo de la etapa de adquisición.....	38
Figura 18. Cámara web digital D-Bugg M26 .....	39
Figura 19. Cámara Pi Noir V2 .....	39



Figura 20. Cámara Logitech C920 HD PRO WEBCAM .....	40
Figura 21. Cámara SV3C HX Series .....	41
Figura 22. Instalación de las cámaras .....	42
Figura 23. Bibliotecas utilizadas en el código de programación .....	44
Figura 24. Tarjeta Raspberry Pi 3+ Modelo B.....	45
Figura 25. Implementación de la Tarjeta Raspberry Pi 3+ Modelo B en el automóvil. .....	46
Figura 26. Instalación del enrutador TP Link .....	48
Figura 27. Batería Hoovo.....	49
Figura 28. Implementación de la batería Hoovo.....	50
Figura 29. Periféricos de entrada y salida .....	51
Figura 30. Periféricos de entrada y salida .....	56
Figura 31. Periféricos de entrada y salida .....	57
Figura 31. Suscripción a Azure Students .....	63
Figura 32. Creación de la Máquina Virtual en Azure .....	63
Figura 33. Validación de los datos para la máquina virtual.....	64
Figura 34. Creación de Ubuntu Server.....	65
Figura 35. Autenticación para la clave pública SSH.....	66
Figura 36. Recurso creado.....	66
Figura 37. Habilitación del puerto SSH en Azure.....	67
Figura 38. Activación del servidor de MariaDB .....	68

Figura 39. Pagina oficial de Apache activada.....	69
Figura 40. Activacion de PHP MyAdmin .....	70
Figura 41. Instalación, creación y acceso a Node-RED.....	71
Figura 42. Almacenamiento de la grabación por, usuario, hora, fecha.....	71
Figura 43. Almacenamiento del historial de grabaciones mediante la Raspberry Pi 3 B+ .....	72
Figura 44. Interfaz de Node-RED .....	73
Figura 45. Nodo “form” .....	73
Figura 46. Enlace a la base de datos .....	74
Figura 47. Nodo Function .....	74
Figura 48. Nodo Conexión a Mysql.....	74
Figura 49. Vinculación del Nodo Mqtt y Nodo json.....	75
Figura 50. Creación de tablas y redes .....	75
Figura 51. Tabla de la lista para las cámaras en la Interfaz .....	76
Figura 52. Visualización de usuario, hora y fecha .....	76
Figura 53. Tabla de la lista para las cámaras en la Interfaz .....	76
Figura 54. Diagrama de flujo del sistema de monitorización .....	78
Figura 55. Imágenes recolectadas .....	79
Figura 56. Imágenes recolectadas .....	80
Figura 57. Código de Python para convertir el video en frames.....	81
Figura 58. Diagrama de distribución para cada uno de los casos establecidos.....	82

Figura 59. Arquitectura VGG .....	83
Figura 60. Capas del algoritmo .....	84
Figura 61. Matriz de correlación .....	88
Figura 62. Curva de pérdidas del entrenamiento y evaluación de la imagen RGB ...	89
Figura 63. Curva de exactitud del entrenamiento y evaluación de la imagen RGB ..	89
Figura 64. Curva de pérdidas del entrenamiento y evaluación de la imagen en escala de grises.....	89
Figura 65. Curva de exactitud del entrenamiento y evaluación de la imagen en escala de grises.....	90
Figura 66. Matriz de confusión de 100 épocas. ....	91
Figura 67. Librería Keras y Tensorflow.....	92
Figura 68. Robo a vehículo con arma blanca (Día) .....	94
Figura 69. Robo a vehículo con arma blanca (Noche).....	94
Figura 70. Persona transcurriendo cerca del automóvil.....	95
Figura 71. Detección de robo con arma blanca (noche) Cámara 1 .....	95
Figura 72. Detección de robo con arma blanca (día) Cámara 1 .....	96
Figura 73. Detección de robo con arma blanca (día) Cámara 2.....	96
Figura 74. Monitoreo y control de las cámaras en tiempo real mediante la interfaz gráfica.....	97
Figura 75. Especificaciones Técnicas de Raspberry pi 3 model B+ .....	109

## ÍNDICE DE ANEXOS

Anexo A. Raspberry Pi 3 Modelo B+ .....	109
Anexo B. Cámara SV3C HX Series.....	110
Anexo C. Router Wi-Fi de doble banda AC1200 .....	111
Anexo D. Módulo SIM808 .....	112
Anexo E. Batería Lipo HOOVO .....	113
Anexo F. Diagrama del dispositivo de Monitoreo .....	113
Anexo G. Empresa Confecciones GALAR.....	114
Anexo H. Frames de la simulación de robo .....	117
Anexo I. Código en Python del sistema de control y monitorización.....	118
Anexo J. Interfaz gráfica del sistema de control y monitorización.....	123

## RESUMEN EJECUTIVO

El presente proyecto de investigación, se implementa un sistema electrónico de control y monitorización de vehículos contra robos para la empresa "Confecciones GALAR" mediante el uso de visión artificial. El sistema aborda la problemática de la seguridad vehicular con una solución innovadora que detecta y monitorea posibles robos en las instalaciones de la empresa, ofreciendo respuestas rápidas y eficientes ante situaciones de riesgo. La implementación se basa en cámaras de seguridad equipadas con algoritmos de visión artificial que detectan automáticamente actividades sospechosas, especialmente aquellas relacionadas con intentos de robo de vehículos. El sistema analiza el comportamiento de las personas en el área de estacionamiento y reconoce patrones asociados a actividades delictivas.

La arquitectura electrónica del sistema incluyó cámaras de alta resolución y un sistema de procesamiento de imágenes en tiempo real. Se utilizan técnicas de aprendizaje profundo para identificar y clasificar eventos específicos, como la presencia no autorizada cerca de los vehículos. La interfaz de usuario proporciona visualización en tiempo real y alertas automáticas ante situaciones sospechosas. Se abordan aspectos técnicos como la selección de algoritmos de visión artificial, la configuración del hardware y la integración efectiva en el entorno operativo de "Confecciones GALAR". La evaluación del sistema muestra una efectividad del 95% en situaciones simuladas y reales, destacando su potencial para reducir los riesgos asociados a robos de vehículos con un bajo margen de error.

**Palabras claves:** Monitoreo, cámaras, visión artificial, vehículos, red neuronal, entrenamiento, microcontrolador, seguridad.

## ABSTRACT

The present research project, an electronic vehicle control and monitoring system against theft is implemented for the company "Confecciones GALAR" using artificial vision. The system addresses the issue of vehicle security with an innovative solution that detects and monitors potential thefts on the company's premises, providing quick and efficient responses to risky situations. The implementation relies on security cameras equipped with artificial vision algorithms that automatically detect suspicious activities, especially those related to attempts to steal vehicles. The system analyzes people's behavior in the parking area and recognizes patterns associated with criminal activities.

The electronic architecture of the system includes high-resolution cameras and a real-time image processing system. Deep learning techniques are used to identify and classify specific events, such as unauthorized presence near vehicles. The user interface provides real-time visualization and automatic alerts in suspicious situations. Technical aspects, such as the selection of artificial vision algorithms, hardware configuration, and effective integration into the operational environment of "Confecciones GALAR," are addressed. The system evaluation demonstrates a 95% effectiveness in simulated and real situations, highlighting its potential to reduce risks associated with vehicle thefts with a low margin of error.

**Keywords:** Monitoring, cameras, artificial vision, vehicles, neural network, training, microcontroller, security.

## **CAPÍTULO I. MARCO TEÓRICO**

### **1.1 Tema de investigación**

SISTEMA ELECTRÓNICO DE CONTROL Y MONITORIZACIÓN DE VEHÍCULOS EN CASO DE ROBO PARA LA EMPRESA GALAR CONFECCIONES MEDIANTE VISIÓN ARTIFICIAL

#### **1.1.1 Planteamiento del problema**

Actualmente la inseguridad y la delincuencia en el país y el mundo constituyen un problema para las entidades encargadas de este tipo de percances, tal es así, que la policía o grupos de seguridad no abarcan la totalidad de la Ciudad de Ambato, ya sean zonas urbanas o rurales. Existen diversos casos en los que las personas pueden ser afectadas de manera física, robos a sus viviendas o vehículos y en los peores casos llegar a tener pérdidas humanas. Por tales motivos, los moradores de dichas zonas normalmente se sienten vulnerables y recurren a la autoprotección, solicitando sistemas de seguridad, monitorización o video vigilancia. [1]

En Ecuador, el departamento de Logística de Hunter receipto 406 denuncias de robos de vehículos en el país, de los cuales cerca del 70% fueron recuperados por la compañía hasta el momento, lo que representa un monto de USD 9'364.263.06. En el primer semestre del año 2022 hubo un total de 147 denuncias, frente a las 406 que se registraron hasta el momento en el 2023. El transporte pesado; es decir, los camiones que transportan mercadería, los vehículos 4x4 y las camionetas son los automotores más vulnerables y atractivos para los intereses del crimen organizado. Adicionalmente, del total de hurtos se estima que un 50% responde a vehículos de uso particular y el otro 50% a vehículos pesados o de uso empresarial. Respecto a las zonas de mayor incidencia de robos vehiculares reportados, en la provincia del Guayas (Guayaquil) ocupa el primer lugar, con el 36% del total de robos, seguido de Los Ríos (Quevedo), con el 23%, Santo Domingo con el 19%, y Pichincha con el 8%. “Las cifras evidencian que Guayaquil, hasta el momento, es la ciudad más conflictiva”, remarca el especialista de Hunter. Agrega que le siguen Quevedo, Santo Domingo y Quito. [2]

En Tungurahua, el robo de vehículos y accesorios de estos, son parte de la lista de delitos que se cometen con más frecuencia. Existen varios casos que se han registrado en la provincia. En 2021, 350 personas denunciaron haber sido víctimas del robo de partes y accesorios de sus automotores, mientras que, en 2022, la cifra fue de 280 casos. En las zonas de la provincia, el 67% de las víctimas encuestadas estuvieron solas en el momento del delito, principalmente en robos y asaltos a sus vehículos. [3]

Ambato es uno de los cantones en donde más casos de delincuencia son reportados. No existe un punto determinado en la ciudad donde se enfoque con frecuencia el cometimiento de estos delitos, porque las bandas operan a partir de las circunstancias, manifestó la autoridad policial. Según las estadísticas establecidas por la Policía Nacional a lo que respecta a robos de bienes, accesorios y autopartes, durante 2022, se registraron 311 ilícitos, mientras que, en lo que va de este año, 297 hechos ilegales se anotaron. Marcando un decremento en cuanto a lo que a robos de autopartes se refiere. [4]

Los sistemas de seguridad actuales instalados en los vehículos nuevos, son muy costosos y dependen del sistema de batería principal, en donde los atracadores desconectan el mismo y se queda sin monitoreo el automotor, además tienen un alto consumo de energía. Así mismo se debe cancelar una cuota mensual para su constante monitorización, lo cual que dificulta su adquisición. [4]

La dependencia de la tecnología actual de visión artificial en vehículos monitoreados presenta un desafío significativo debido a su bajo consumo eléctrico. Esta característica, aunque beneficiosa en términos de eficiencia energética, crea una vulnerabilidad en la continuidad del monitoreo. Los vehículos que dependen del sistema eléctrico principal para el monitoreo están expuestos a la pérdida de ubicación en caso de fallo eléctrico. Este problema se agrava aún más en situaciones donde la batería principal falla, se corta o se manipula, lo que resulta en la incapacidad de mantener un monitoreo constante y confiable del automotor. La falta de un sistema de respaldo efectivo en estas circunstancias representa un riesgo significativo para la integridad del monitoreo vehicular, ya que la interrupción del suministro eléctrico principal puede tener consecuencias perjudiciales en la capacidad de rastreo y vigilancia del vehículo. [5]



## 1.2 Antecedentes investigativos

Los siguientes antecedentes investigativos contribuirán para trayectoria evolutiva en la aplicación de la visión artificial a la seguridad vehicular, las soluciones en los sistemas de rastreo basados en GPS, asimismo, sobre cuestiones relacionadas con la integración de sistemas electrónicos en vehículos para maximizar la efectividad de las soluciones de seguridad.

En la investigación desarrollada por Christian López, en el año 2021, en la Universidad San Francisco de Quito USFQ, con el tema “Diseño e implementación de un prototipo de vehículo autónomo a escala con capacidades de visión artificial”, se detalla la implementación y desarrollo de un prototipo autónomo (nivel de autonomía de nivel 3 - 4) a escala, que utiliza una red neuronal y un mando Dualshock®4 para su accionamiento. El mecanismo del vehículo consta con dos principales modos de interacción: un modo manual (en el que se utiliza el mando para el movimiento y acciones dentro de la plataforma) y un modo automático, mediante la tecnología Neural Computer Stick 2 (NCS2). El código se estructuró para monitorear los sensores secuencialmente, es decir que la primera interacción será con un objeto de frente, para luego analizar el mejor curso de acción dependiendo de la proximidad de la obstrucción hacia el sensor de la izquierda o de la derecha. También se aprecian los voltímetros colocados para medir el voltaje de las baterías que alimentan los motores y el Raspberry, la utilidad de estos es para evitar una descarga profunda que dañe las baterías. Se presenta una vista completa de todo el vehículo en la cual se aprecia la pantalla Raspberry que permite monitorear los procesos. [6]

En la investigación desarrollada por Mario X. Bonilla, en el año 2019, en la Universidad Técnica de Ambato, con el tema “Sistema autónomo de monitoreo de señales fisiológicas con gestión de emergencias para seguridad vial de ciclistas amateur”, El trabajo presenta un sistema de emergencia para monitorear el estado de salud del ciclista con datos que puede ser visualizados en una HMI situadas en la estación fija secundaria del sistema usando la tecnología Wifi y GPRS, además el sistema contiene un wearable en la indumentaria del conductor con la cual procesa datos como temperatura, oximetría de pulso y frecuencia cardiaca, en el caso que sufra un accidente. A su vez se monitoreó el nivel de oxígeno en la sangre, con el objetivo

de impedir que exista deficiencia de oxígeno en el cuerpo, ayudando a prevenir la acumulación de ácido láctico en los músculos que están sujetos a esfuerzo físico. El sistema está integrado en un Jersey, que es la indumentaria para ciclistas, el cual ayudará con la activación de las alertas de emergencia si el deportista sufre un accidente o si sus variables fisiológicas monitoreadas cambian repentinamente, las alertas contienen los datos de los sensores y ubicación del deportista y son enviadas a través de un SMS hacia un terminal receptor donde la información será almacenada y hacia un familiar para que el mismo acuda al rescate del deportista. [7]

Así mismo se tiene el artículo científico “Application and Design of Wireless Community Alarm System Based on nRF24L01 Module” de Wu Chan. En el año de 2019, desarrollado en la Universidad Jiangnan en Wuhan, se realiza un sistema de alarma inalámbrico residencial utilizando el módulo nRF24L01, con el cual, se construye una red inalámbrica con una transmisión bidireccional y un microcontrolador utilizado con una máquina(host) para la monitorización central. Los usuarios pueden enviar señales de alarma a través de una submáquina (subhost) de usuario y luego la máquina central (host) recibe las señales y muestra la dirección de donde se activó la alerta, para finalmente enviar una advertencia. En donde, en base al módulo nRF se concluye que el uso del canal dúplex completo, permite obtener una comunicación bidireccional, por ende, un sistema de alarma inalámbrico completo de bajo costo, rentable y sin necesidad de cables. Se verifico que el sistema inalámbrico, supera a la alarma cableada original, la cual es de alto costo, difícil construcción está sujeta a interferencias. [8]

En la investigación desarrollada por Joseph Hernández en el año 2019, en su proyecto “Sistema de monitoreo GPS para los vehículos de la escuela de capacitación para conductores profesionales de Píllaro”, desarrollado en la Universidad Técnica de Ambato, realiza un sistema de geolocalización de vehículos utilizando la red móvil. El dispositivo está compuesto por un módulo GPS/GPRS los cuales permiten obtener la ubicación del vehículo y conectarse a internet respectivamente, se envía los datos a un servidor web, en donde se procesa y se almacena en una base de datos para posteriormente ser mostrados en un sitio web. Además, cuenta con un pulsador en caso de emergencia, el cual, al ser pulsado genera una alerta al supervisor de la escuela, y se podrá actuar de manera inmediata y eficiente dando soporte, asistencia médica y/o

mecánica, así como a los estudiantes de la escuela. Por lo tanto, en los resultados se evidencia una buena precisión del GPS en donde se compara con un teléfono inteligente para verificar su funcionalidad, concluyendo que el módulo GPS GY-NEO6MV tiene una alta precisión tanto en latitud como en longitud, funcionando con un margen de error mínimo. Lo cual permite localizar vehículos con mayor precisión y poca energía. [9]

En la investigación desarrollada por Grecia Pico en el año 2019, en su proyecto “Sistema avanzado de asistencia al conductor empleando visión artificial en vehículos de transporte público”, realizado en la Universidad Técnica de Ambato. El trabajo presenta el análisis, desarrollo y creación de un prototipo de sistema avanzado de asistencia al conductor empleando visión artificial en vehículos de transporte público para la prevención de accidentes de tránsito en la ciudad, además se analizaron las tecnologías empleadas en sistemas avanzados de asistencia al conductor (SAAC) vigentes en el mercado, como también se estudiaron los algoritmos aplicados con visión artificial en SAAC. El prototipo basa su funcionamiento en la utilización de software libre mediante la programación en Open CV, Python y el apoyo de herramientas para el reconocimiento de patrones empleando técnicas de visión artificial como los algoritmos predefinidos de los descriptores Haar Cascades, con el uso de dichos elementos se realiza el procesamiento necesario para la detección de obstáculos definidos como peatones, líneas de carril y la activación de alertas visuales y sonoras en caso de que se presenten infracciones en el recorrido de los autobuses, para el reconocimiento de estos obstáculos se utiliza una cámara propia de Raspberry pi. En base a las pruebas realizadas por el sistema en la sección se peatones en un rango de distancia entre uno y diez metros la cantidad de peatones detectada es igual a la cantidad de peatones presentes en la vía a partir de los quince metros se va perdiendo esta característica de reconocimiento y dando paso a la presencia tanto de falsos positivos como de falsos negativos, disminuyendo la confiabilidad del sistema de detección, el tiempo de respuesta del sistema ante eventos individuales varía entre 1s y 1.5s, la respuesta del sistema completo varía entre los 2s y 3s considerando la distancia entre el automotor y el obstáculo. [10]

### **1.3 Fundamentación teórica**

El Presente marco teórico abordará aspectos como el sistema de monitoreo y control aplicados fundamentalmente a la visión artificial, tales como técnicas de procesamiento de imágenes, resaltando su aplicación específica en la detección y seguimiento de situaciones relacionadas con el robo de vehículos.

#### **1.3.1 Vehículos**

Un vehículo es un medio de transporte diseñado para el desplazamiento de personas o mercancías de un lugar a otro. Puede ser propulsado por diferentes formas de energía, como motores de combustión interna, motores eléctricos o propulsión humana, y puede adoptar diversas formas, como automóviles, camiones, motocicletas, bicicletas, entre otros. Los vehículos son instrumentos fundamentales en la movilidad moderna, proporcionando una variedad de opciones para satisfacer necesidades de transporte tanto individuales como colectivas. [11]

#### **1.3.2 Robos**

El robo es un delito contra la propiedad que implica la sustracción ilegítima de bienes, pertenencias o activos de una persona o entidad, con la intención de apropiarse de ellos de manera indebida y sin el consentimiento del propietario. Este acto delictivo puede llevarse a cabo mediante el uso de la fuerza, intimidación, amenazas o aprovechando situaciones en las que el propietario no está presente. El robo puede abarcar una amplia gama de situaciones, desde pequeños hurtos hasta actos más complejos y planificados, y puede afectar tanto a individuos como a empresas. Las leyes que rigen el robo varían en función de la jurisdicción, pero en general, el acto de sustraer propiedad ajena con intenciones ilícitas constituye un delito penal. [11] [12]

#### **1.3.3 Robo a vehículos**

El robo a vehículos es un delito específico que implica la apropiación ilegal de un automóvil u otro tipo de vehículo sin el consentimiento del propietario. Este acto delictivo puede llevarse a cabo de diversas maneras, como el forzamiento de

cerraduras, el uso de dispositivos electrónicos para el arranque sin llave, o incluso la amenaza directa o el uso de la fuerza física contra el propietario. El objetivo principal del robo a vehículos es obtener el control y la posesión del vehículo con el fin de utilizarlo de manera indebida, revenderlo o desmantelarlo para la venta de partes. Este delito no solo causa pérdidas económicas significativas a los propietarios, sino que también puede tener implicaciones en términos de seguridad pública y movilidad. Las leyes y sanciones asociadas al robo de vehículos varían según la jurisdicción, pero en general, se considera un delito grave que puede conllevar consecuencias legales severas para los perpetradores. [11]

### 1.3.4 Indicadores de robo de vehículos

Los indicadores más comunes de robos a vehículos pueden variar y estar influenciados por diversos factores, incluyendo el contexto socioeconómico, la ubicación geográfica y las condiciones específicas de seguridad en un área determinada. [13] [12]

En la Tabla 1 se muestra los indicadores más frecuentes de robo en Ecuador:

Tabla 1. Indicadores de robo de vehículos en Ecuador [13]

Indicadores	Características
<b>Oportunidad y falta de seguridad</b>	Muchos robos a vehículos ocurren debido a la falta de medidas de seguridad adecuadas. Estacionar en áreas poco iluminadas o sin vigilancia, dejar el vehículo sin cerrar o con las ventanas abiertas son situaciones que pueden ser aprovechadas por los delincuentes.
<b>Desarrollo de bandas organizadas</b>	En algunas ocasiones, los robos a vehículos son perpetrados por bandas organizadas que se dedican a este tipo de delitos de manera sistemática. Estas bandas pueden tener estrategias sofisticadas para el robo y la posterior comercialización de vehículos o piezas.
<b>Altos índices de pobreza y desempleo</b>	Factores socioeconómicos, como la pobreza y el desempleo, pueden contribuir al aumento de la delincuencia, incluidos los robos a vehículos. La necesidad económica puede llevar a algunas personas a buscar oportunidades delictivas.
<b>Tráfico ilícito de vehículos</b>	En ocasiones, los vehículos robados son utilizados para el tráfico ilícito de bienes, personas o drogas. Esto puede aumentar la incidencia de robos a vehículos, ya que los delincuentes buscan vehículos específicos para sus actividades ilícitas.
<b>Falta de conciencia y educación en seguridad</b>	La falta de conciencia sobre medidas de seguridad adecuadas y la educación pública sobre la importancia de proteger los vehículos pueden contribuir a que los propietarios no tomen precauciones necesarias para evitar el robo.

### **1.3.5 Métodos de robos a vehículos**

#### ***a. Robos mediante Rotura de Vidrio***

Una modalidad común de robo a vehículos en Ecuador involucra la rotura de vidrios. Los delincuentes optan por esta técnica cuando buscan acceder al interior del vehículo de manera rápida y discreta. Estacionamientos poco vigilados o áreas urbanas con baja iluminación suelen ser lugares propicios para este tipo de incidentes. La ausencia de medidas preventivas, como sistemas de alarma o cámaras de seguridad, puede facilitar la ejecución de robos por medio de la rotura de vidrios. [13] [12]

#### ***b. Robos mediante Forzamiento de la Puerta del Vehículo***

El forzamiento de la puerta del vehículo es otra táctica empleada por ladrones para acceder al interior de un automóvil. Esto puede implicar el uso de herramientas como palancas o destornilladores para manipular cerraduras y entrar al vehículo sin levantar sospechas. Este método a menudo se lleva a cabo en lugares menos concurridos, donde la actividad delictiva puede pasar desapercibida durante un tiempo más prolongado. [13]

#### ***c. Acercamiento de Personas Sospechosas al Vehículo.***

El acercamiento de personas sospechosas a los vehículos es una estrategia previa al robo. Delincuentes pueden merodear alrededor de automóviles, observando la presencia de objetos de valor en su interior o evaluando la posibilidad de realizar un robo. La falta de conciencia por parte de los propietarios sobre su entorno y la ausencia de medidas preventivas, como el cierre adecuado de ventanas, pueden convertir estos momentos en oportunidades para los ladrones. [13] [12]

### **1.3.6 Sistema de Monitorización**

Los sistemas de monitorización ayudan a llevar un seguimiento continuo de los valores recopilados por los sensores que forman la red, esto proporciona una interfaz amigable, para que la información pueda ser visualizada, independientemente del dispositivo donde este alojado esta interfaz, esta información puede ser utilizada para presentarla con un análisis grafico o estadístico ya dependerá del uso que se pretenda dar a la

información recolectada se los caracteriza por ser sistemas pasivos ya que el manejo de la información y el control de la misma, depende del personal competente.

### 1.3.7 Sistema de Control

Los sistemas de control a su vez conocidos como sistemas reactivos, ya que el análisis y posterior control de la información se realiza de forma automática, ayuda a que la información antes receptada sea puesta en función para que realicen una acción establecida dependiendo el uso que se deba dar a la misma. [14]

En la Figura 1 se muestra el diagrama de bloques del sistema de control.

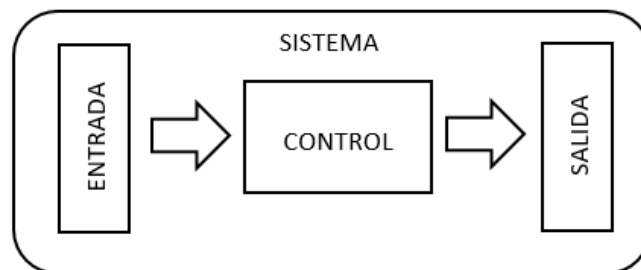


Figura 1. Diagrama de Bloques Sistema de Control [14]

### 1.3.8 Procesamiento de imágenes

Los sistemas electrónicos para el procesamiento de imágenes son dispositivos diseñados específicamente para adquirir, procesar y analizar imágenes de manera eficiente y precisa. Estos sistemas suelen incluir componentes como sensores de imagen, circuitos de captura y conversión analógico-digital, unidades de procesamiento de imágenes y dispositivos de almacenamiento. Los sistemas electrónicos emplean algoritmos y técnicas avanzadas para realizar diversas operaciones, como filtrado, segmentación, reconocimiento de patrones y mejora de calidad de imagen. [11]

**a. Etapas Procesamiento de imágenes y videos.**

Comprende una serie de componentes y etapas que trabajan en conjunto para analizar y procesar imágenes de manera eficiente y efectiva.

En la Figura 2 se muestra las etapas de los sistemas electrónicos para el procesamiento de imágenes y videos.

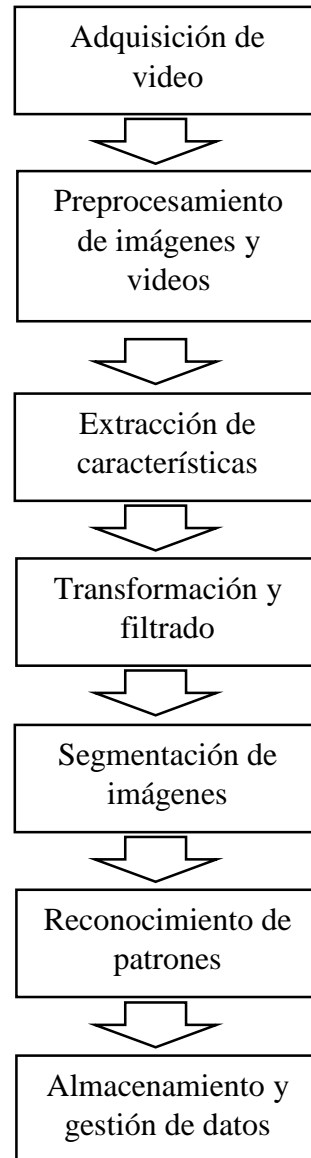


Figura 2. Etapas de los sistemas electrónicos para el procesamiento de imagen y video.

Además, permite obtener información valiosa a partir de los datos visuales.



En la Tabla 2 se muestra las Etapas de la arquitectura de sistemas electrónicos para el procesamiento de videos enfocada a la visión artificial.

Tabla 2. Etapas de la arquitectura de sistemas electrónicos para el procesamiento de videos enfocada a la visión artificial [15]

<b>Etapa</b>	<b>Descripción</b>
<b>Adquisición de videos</b>	Captura de videos utilizando cámaras, escáneres u otros dispositivos de entrada.
<b>Preprocesamiento de imágenes y videos</b>	Mejora de la calidad y claridad de los videos mediante ajustes de contraste, eliminación de ruido, etc.
<b>Extracción de características</b>	Identificación y extracción de características relevantes de los videos, como bordes, texturas y formas.
<b>Transformación y filtrado</b>	Aplicación de técnicas de transformación y filtrado para modificar y mejorar los videos.
<b>Segmentación de videos</b>	Separación de regiones de interés u objetos del fondo mediante algoritmos de segmentación.
<b>Reconocimiento de patrones</b>	Separación de regiones de interés u objetos del fondo mediante algoritmos de segmentación.
<b>Almacenamiento y gestión de datos</b>	Almacenamiento y gestión eficiente de imágenes y datos relacionados en sistemas de almacenamiento adecuados.

### 1.3.9 Sistemas de detección de robo a vehículos

Los sistemas de detección de robos a vehículos son diferentes herramientas diseñadas para identificar y alertar sobre al usuario o dueño del vehículo sobre una persona sospechosa acercándose al vehículo, estos pueden variar desde alertas sonaras hasta sistemas más sofisticados como sistemas GPS. Estos sistemas utilizan diferentes técnicas y sensores para detectar y prevenir el robo de vehículos. [16] [17]

### 1.3.10 Visión Artificial

Para hacer uso de la visión artificial se es necesario contar con unidad lógica programables (CPUs), transductores de entrada (cámara de video) y sistemas para extraer información significativa a partir de imágenes digitales, videos y otras entradas visuales, y tomar medidas o realizar recomendaciones en función de esa información. Si la inteligencia artificial (IA), permite a los ordenadores pensar, la visión artificial les permite ver, observar y comprender. [18]

La visión artificial permite la mejora de las siguientes capacidades:

- La visión humana: A pesar que la vista humana tiene ventaja de toda una vida de contexto para reconocer objetos, a qué distancia están, si se están moviendo y si hay algo mal en una imagen, la visión artificial permite enfocar el estudio en un punto específico con la finalidad de captar mayor información [18]
- Entrenar a las máquinas para realizar funciones, en menor tiempo con cámaras, datos y algoritmos en lugar de retinas, nervios ópticos y una corteza visual. Como un sistema entrenado para inspeccionar productos o ver un activo de producción puede analizar miles de productos o procesos por minuto, detectando defectos o problemas imperceptibles, puede superar rápidamente las capacidades humanas. [18]
- Disminuir la participación humana en el proceso de fabricación aumentado la producción y reduciendo los tiempos de manufactura, lo que permitirá el incremento de ingresos económicos a las diferentes industrias. [18]

En la Figura 3 se muestra el lente de la cámara que se enfocara para la visión artificial.



Figura 3. Visión Artificial [11]

### 1.3.11 Características de la Visión Artificial

La visión artificial ha ganado un espacio cada vez más grande en los procesos productivos en el ámbito de las líneas de producción altamente automatizadas, en dispositivos para el reconocimiento facial o de incidentes a través de cámaras de vigilancia, en vehículos de conducción autónoma y en tecnologías militares como el uso de vehículos aéreos no tripulados como los UAVs, drones o misiles guiados. Lo que ha permitido este rápido desarrollo de los sistemas es la integración de sistemas de IA, los cuales en base a los diferentes algoritmos de reconocimiento permiten tener

una mejor respuesta. Las redes neuronales artificiales son una de la técnica más usadas en el reconocimiento de patrones ya que estas trabajan con capas internas que permiten mejorar el tiempo de procesamiento. [15]

Las redes neuronales artificiales cumplen con las siguientes características [19]:

- Cada unidad posee un estado para la activación
- Las unidades tienen conexiones internas, las cuales tienen asociado un peso que decide la acción en la señal de entrada.
- Tienen una función de activación que determina la salida de acuerdo con la entrada actual y la anterior
- El bias es un parámetro externo que posee cada unidad
- Se debe generar un aprendizaje o método de enseñanza para adquirir información
- Las redes neuronales trabajan con señales de entrada y de error para realizar ajustes aproximando y mejorar la eficiencia del sistema.

Mediante la Ecuación 1 se hace referencia a un sistema de entradas ponderadas mediante una función de activación y un umbral para producir una salida para una red de visión artificial: [19]

$$y = f(\sum_k w_k x_k) + \theta \quad (1)$$

Donde:

$f(*)$  = *Función no lineal simple*

$x_k$  = *Entradas externas*

$w_k$  = *Pesos de los enlaces de conexión*

$\theta$  = *Bias de ajuste*

En las redes de una capa, las entradas están directamente conectadas a las salidas mediante pesos, en donde las salidas no interactúan, por lo que una red con n salidas puede ser analizada como n redes individuales. [19]

Mediante la Ecuación 2 se considera las entradas como una función en el dominio del tiempo, lo que permite analizar cómo evolucionan en relación con el tiempo: [19]

$$y(t) = \sum_k w_k(t) x_k(t) + \theta(t) \quad (2)$$

La fórmula representa la salida  $y(t)$  de un sistema, donde  $w_k(t)$  son los pesos asociados a las entradas  $x_k(t)$  en el tiempo  $t$ , y  $\theta(t)$  es el umbral en el tiempo  $t$ . La salida  $y(t)$  se calcula como la suma ponderada de las entradas  $x_k(t)$  multiplicadas por sus respectivos pesos  $w_k(t)$ , junto con el umbral  $\theta(t)$ . Este cálculo permite entender cómo el sistema responde en un instante de tiempo específico en función de las entradas y sus pesos asociados. [19]

### 1.3.12 Seguimiento de Objetos Mediante Visión Artificial

Puesto de una manera sencilla, ubicar un objeto específico dentro de una secuencia de imágenes, es a lo que se conoce como “tracking” o (rastreo, seguimiento). Pese a que el concepto suena sencillo y concreto, rastrear un objeto consolida varias ideas diferentes similares en concepto, pero variantes en técnica. Se detalla a continuación [20]:

- Densidad de Flujo Óptico: este tipo de algoritmos permiten determinar el vector de dirección de un píxel específico dentro de una imagen.
- Caudal de Flujo Óptico: este tipo de algoritmos, permiten realizar el seguimiento de los puntos característicos de una imagen.
- Filtros de Kalman: un algoritmo de visión bastante popular, que permite determinar la posición de un objeto en base a información de movimiento previa.
- Varianza Máxima: estos algoritmos permiten determinar los picos de una función de densidad, lo cual resulta útil para realizar tracking.
- Localizadores de Único Objeto: estos algoritmos encuadran el objeto de interés dentro de un rectángulo.

- Localizadores de Múltiples Objetos: cuando se cuenta con un algoritmo de detección de alta velocidad, no se localiza uno, sino varios objetos.

El seguimiento de objetos posee varias aplicaciones de interés tanto como para procesos de automatización como de investigación. Dos ejemplos exitosos de rastreo con dominios específicos son el seguimiento de rostros y de personas. La detección de objetos también encuentra en escenario potencial de alto interés para su aplicación en su incorporación de sistemas de navegación de vehículos. [20]

Sin embargo, realizar el rastreo de un objeto genérico sigue siendo una tarea desafiante. El seguimiento de objetos en general es implementado mediante el uso de algoritmos de seguimiento en conjunto con el uso de clasificadores o entrenamiento del sistema. [20]

### **1.3.13 Procesamiento Digital de Señales**

El procesamiento digital de señales abarca técnicas y algoritmos utilizados para manipular, analizar y transformar señales en formato digital mediante el uso de herramientas informáticas. Proporciona una amplia variedad de operaciones, como filtrado, modulación, compresión, análisis espectral y detección de patrones. Estas técnicas permiten mejorar la calidad de las señales, extraer información relevante y facilitar el análisis de los datos digitales. [21]

Un procesador digital de señales conocido por sus siglas en inglés, Digital Signal Processing (DSP) adquiere señales del mundo real, como voz, video, temperatura, presión o posición, y las procesa a alta velocidad y en tiempo real después de haber sido digitalizadas. El DSP toma estas señales analógicas, las convierte en formato digital y aplica algoritmos y operaciones matemáticas para manipularlas, analizarlas y transformarlas según los requerimientos específicos. Gracias a su capacidad de procesamiento rápido, el DSP permite realizar tareas complejas en tiempo real, lo que lo convierte en una herramienta esencial en aplicaciones como comunicaciones, control industrial, procesamiento de audio y video, entre otros. [21] [22]

En la Figura 4 se menciona el sistema de procesamiento de señales.

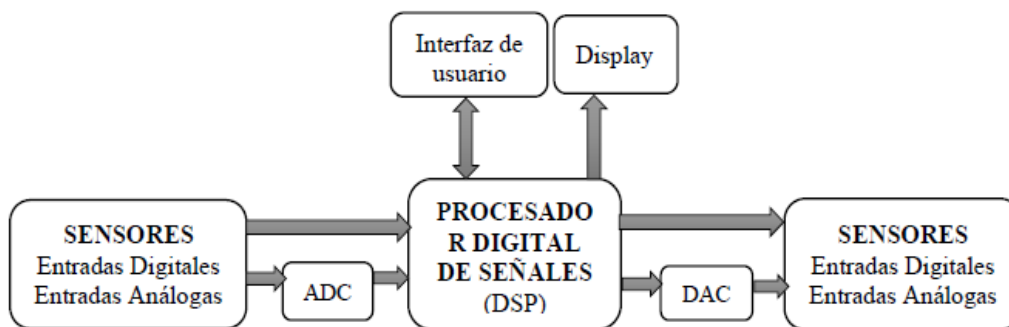


Figura 4. Sistema de Procesamiento de Señales. [22]

### 1.3.14 Integración de la visión artificial en el procesamiento digital de señales.

La integración de la visión artificial en el procesamiento digital de señales (DSP) es un avance tecnológico significativo que ha transformado la manera en que se interpreta y utiliza información visual en diversos campos. La visión artificial implica el desarrollo de algoritmos y sistemas que permiten a las máquinas analizar y comprender imágenes o videos de manera similar a la percepción humana. Cuando se fusiona con el DSP, este enfoque brinda una capacidad adicional para procesar y extraer información valiosa de datos visuales [22].

En la Tabla 3 se muestra las aplicaciones de la integración de la visión artificial en el procesamiento digital de señales.

Tabla 3. Aplicaciones de la integración de la visión artificial en el procesamiento digital de señales [22]

Aplicaciones	Características
<b>Reconocimiento de Objetos</b>	En la aplicación en Seguridad Vehicular, la visión artificial integrada en el PDS puede identificar objetos relevantes en entornos de seguridad vehicular, como reconocer vehículos, peatones o señales de tráfico, contribuyendo a sistemas avanzados de asistencia al conductor y seguridad.
<b>Seguimiento de Movimiento</b>	Para la aplicación en Vigilancia y Monitoreo, la visión artificial puede utilizarse para el seguimiento de objetos en tiempo real, como el monitoreo de áreas de seguridad críticas. Integrada con el PDS, permite analizar patrones de movimiento, identificar comportamientos sospechosos y activar alertas de seguridad.
<b>Detección de Anomalías</b>	En la aplicación en Mantenimiento Predictivo, la combinación de visión artificial y PDS puede identificar anomalías visuales en equipos o infraestructuras, facilitando el mantenimiento predictivo al detectar problemas antes de que se vuelvan críticos.
<b>Procesamiento de Video</b>	En sistemas de seguridad y vigilancia, la visión artificial integrada con el PDS permite el análisis eficiente de flujos de video,

	identificando eventos relevantes, como intrusiones o comportamientos anómalos.
--	--

### 1.3.15 Fundamentos del procesamiento digital de imágenes y videos

El procesamiento digital de imágenes y videos se fundamenta en principios matemáticos y probabilísticos, pero la toma de decisiones respecto a las técnicas empleadas a menudo involucra la intuición y el análisis subjetivo por parte de los especialistas. Esta elección suele basarse en juicios visuales y experiencia previa. Por lo tanto, adquirir un conocimiento básico sobre la percepción humana resulta relevante y necesario para comprender mejor cómo los seres humanos interpretan y procesan las imágenes, lo que a su vez puede enriquecer el desarrollo y selección de técnicas de procesamiento digital más efectivas y adecuadas. [23]

El procesamiento digital de imágenes se refiere al conjunto de técnicas y algoritmos utilizados para manipular y analizar imágenes digitales. [23]

En la Tabla 4 se muestra los Fundamentos del procesamiento digital de imágenes.

Tabla 4. Fundamentos del procesamiento digital de imágenes [23]

<b>Fundamentos</b>	<b>Características</b>
<b>Representación de imágenes y videos</b>	Las imágenes y videos se representan digitalmente utilizando una cuadrícula de píxeles, donde cada píxel almacena información sobre el color y la intensidad de esa ubicación.
<b>Muestreo y cuantización</b>	El muestreo implica la captura de muestras de la imagen en puntos discretos de la cuadrícula de píxeles. La cuantización consiste en asignar valores discretos a las muestras tomadas, lo que permite la representación digital.
<b>Mejora de imágenes</b>	Este proceso incluye técnicas para mejorar la calidad de las imágenes, como el ajuste del contraste, la reducción de ruido, la nitidez y la mejora del detalle.
<b>Transformaciones espaciales</b>	Las transformaciones espaciales son operaciones que se aplican directamente a los píxeles de una imagen. Estas operaciones incluyen el ajuste del brillo y el contraste, el cambio de escala, la rotación y la filtración espacial, que pueden resaltar o suprimir características específicas de una imagen.
<b>Segmentación de imágenes</b>	La segmentación implica dividir la imagen en regiones u objetos más significativos, utilizando criterios como el color, la textura, el contraste o la forma.

<b>Extracción de características</b>	La extracción de características consiste en identificar y extraer atributos específicos de una imagen que son relevantes para una tarea particular. Esto puede incluir la detección de bordes, la textura, el color, la forma u otras características descriptivas de la imagen.
<b>Reconocimiento de patrones</b>	Se utilizan algoritmos y técnicas de aprendizaje automático para identificar y clasificar patrones en las imágenes, lo que permite la detección de objetos, reconocimiento facial, seguimiento de objetos, entre otros.
<b>Muestreo y cuantización</b>	El muestreo implica la captura de muestras de la imagen en puntos discretos de la cuadrícula de píxeles. La cuantización consiste en asignar valores discretos a las muestras tomadas, lo que permite la representación digital de la imagen.

### 1.3.16 Redes Neuronales

Las redes neuronales son modelos computacionales inspirados en el funcionamiento del cerebro humano. Están compuestas por capas de neuronas artificiales interconectadas, donde cada neurona procesa información y transmite señales a otras neuronas. Estas redes aprenden a través de la adaptación de los pesos de las conexiones entre las neuronas, permitiendo reconocer patrones complejos y realizar tareas como clasificación, reconocimiento de imágenes, procesamiento de lenguaje natural y más. Las redes neuronales han demostrado ser efectivas en el campo del aprendizaje automático y la inteligencia artificial, y su capacidad para aprender y generalizar a partir de ejemplos los convierte en una herramienta poderosa en diversas aplicaciones. [24]

### 1.3.17 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son redes artificiales inspiradas en la corteza visual del cerebro y compuesta por capas celulares simples y complejas. Estas redes están diseñadas para realizar tareas de clasificación directamente a partir de imágenes, texto o sonidos. Son particularmente efectivos para detectar patrones en imágenes, permitiendo el reconocimiento de objetos, rostros y escenas aprendiendo directamente datos de imágenes e identificando patrones de clasificación. [25]

Estas redes aprenden extrayendo capas de características no lineales de las imágenes. La complejidad de esta jerarquía aumenta a través de las capas, desde la detección de



bordes y sombras en la capa inicial, hasta características más complejas (como caras u objetos) en capas más profundas. Las últimas capas de la red utilizan las características resultantes para realizar tareas de clasificación o regresión. [25]

Ademas, utiliza una matriz tridimensional como imagen de entrada. Las dos primeras dimensiones de estas matrices representan el largo y el ancho de la imagen en píxeles, mientras que la tercera dimensión representa el número de canales de la imagen. Dispone de 3 canales para imágenes RGB y 1 canal para imágenes en blanco y negro. Estas matrices se introducen en una red que consta de capas convolucionales, capas de agrupación y redes neuronales completamente conectadas. [25]

En la Figura 5 se muestra la estructura de la red neuronal convolucional.

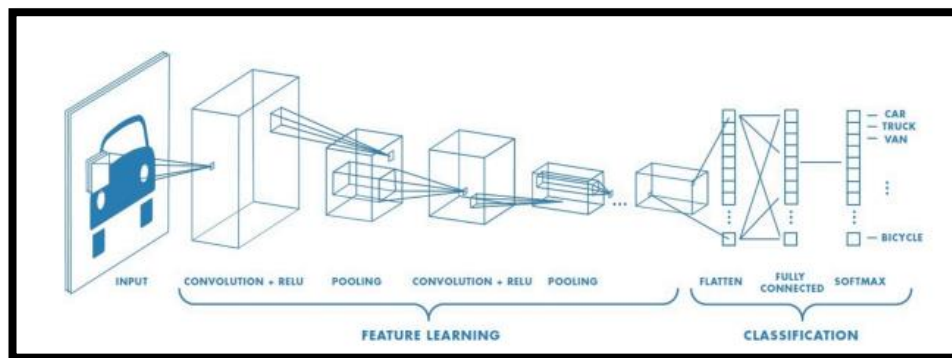


Figura 5. Estructura de la Red Neuronal Convolucional [25]

#### *a. Capas de una Red Neuronal convolucional*

Una red neuronal convolucional es un sistema compuesto por capas de nodos o neuronas interconectadas, las cuales se encargan de procesar información. El número de capas presentes en una red neuronal varía según la complejidad de la función que se intenta modelar. La forma más sencilla de una red neuronal es la de alimentación directa, que puede constar de una o varias capas, donde la información fluye en una dirección, de entrada, a salida, sin retroalimentación. Si bien las capas de entrada y salida siempre están presentes, las capas ocultas pueden o no estar incluidas, dependiendo de la configuración de la red. [26]

La capa convolucional actúa como un extractor de características para aprender características de la imagen de entrada. Estas capas procesan la imagen combinándolas

con "filtros" o "núcleos". Esta combinación implica multiplicar la matriz del filtro por la matriz de la imagen de entrada y luego sumar todos los elementos de la matriz resultante para obtener un valor único. Arrastrar el filtro sobre la imagen y realizar una operación de convolución genera valores que forman un "mapa de características". [25]. En la Figura 6 se muestra las operaciones de la convolución que actúan como diferentes filtros.

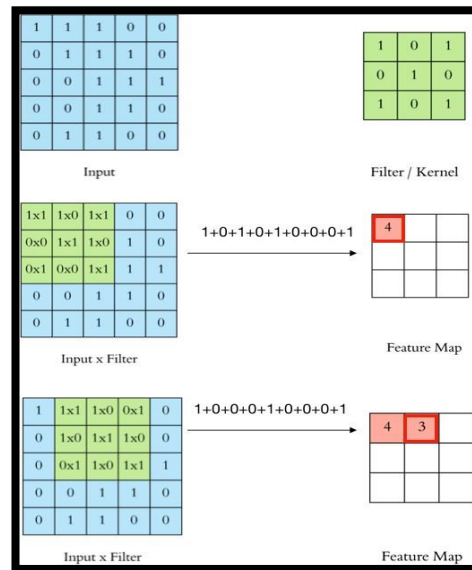


Figura 6. Operaciones de la convolución [25]

En el proceso de entrenamiento, la red neuronal utiliza un algoritmo de descenso de gradiente para aprender los valores óptimos de la matriz de filtro. Estos valores permiten a la red extraer características significativas como textura, bordes y formas del mapa de características de entrada. A medida que aumenta la cantidad de filtros aplicados a la entrada, aumenta la capacidad de la CNN para extraer diferentes características. [25]

## RELU

La función de activación "RELU" (Unidad lineal dirigida) es una operación que es no lineal e introduce no linealidad en el modelo para que pueda aprender funciones no lineales. Está determinada por la siguiente Ecuación 3: [25]

$$f(x) = \max(0, x) \quad (3)$$

Esta función de activación devuelve cero para entradas negativas, mientras que los valores positivos permanecen constantes, lo que acelera enormemente el proceso de entrenamiento al evitar la propagación de gradientes negativos. [25]

En la Figura 7 se muestra la propagación de la activación de RELU.

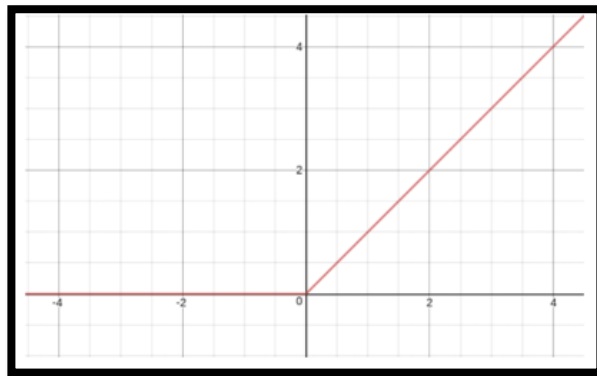


Figura 7. Propagación de la activación RELU [25]

## **POOLING**

La fusión (POOLING) es una operación que reduce el tamaño espacial, la longitud y el ancho de un mapa de objetos sin afectar su profundidad. Esta capa generalmente se aplica después de la capa de convolución o después de la función de activación. La operación de agrupación reduce la cantidad de parámetros, lo que ayuda a controlar el sobreajuste de la red. Además, asegura la invariancia espacial del modelo. [25]

En la Figura 8 se muestra las dimensiones de operación de Pooling.

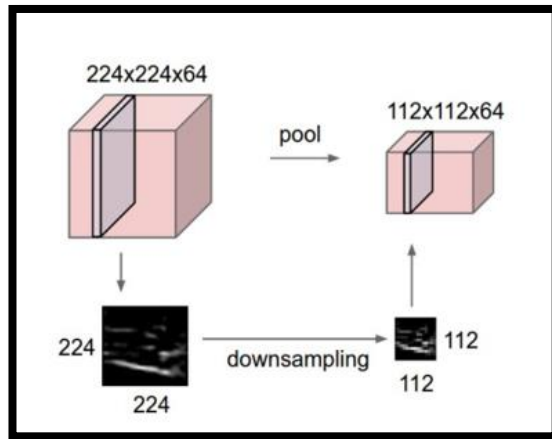


Figura 8. Dimensiones de operación de Pooling [25]

El algoritmo más comúnmente utilizado para la operación de Pooling es el "Max Pooling". Max Pooling funciona de manera análoga a la convolución, deslizándose sobre el mapa de características mediante ventanas de un tamaño determinado. Para cada ventana, el valor máximo se selecciona y se envía a un nuevo mapa de características, mientras que todos los demás valores son descartados. Esto permite preservar las características más importantes. [25]

En la Figura 9 se muestra el metodo de operación Max Pooling.

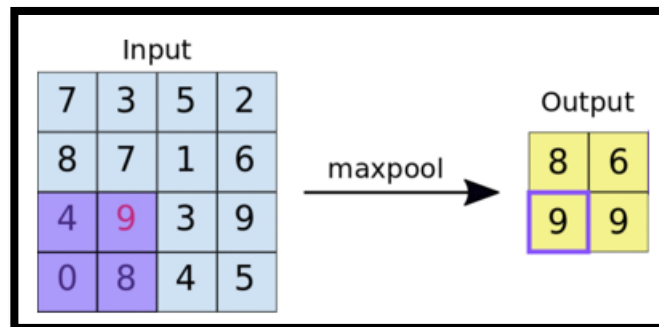


Figura 9. Max Pooling [25]

### Capas Conectadas

La capa neuronal completamente conectada se encuentra al final de la red neuronal convolucional y realiza tareas de clasificación basadas en las características obtenidas mediante la convolución. Por lo tanto, la entrada a estas capas es una versión aplanada del mapa de características obtenido en la etapa de convolución. [25]

En la Figura 10 se muestra las capas de la red neuronal conectadas.

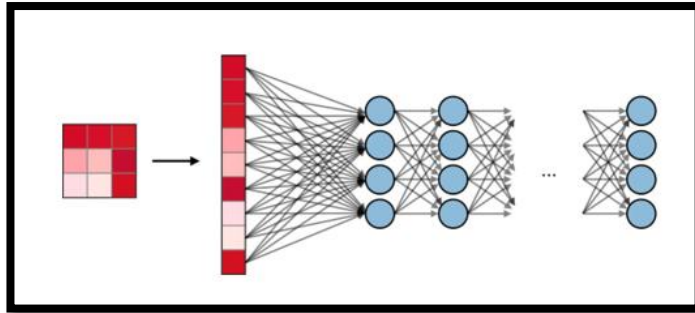


Figura 10. Capas de la Red Neuronal Conectada [25]

## 1.4 Objetivos

### 1.4.1 Objetivo general

Implementar un sistema electrónico de control y monitorización de vehículos en caso de robo para la Empresa GALAR Confecciones mediante visión artificial.

### 1.4.2 Objetivos específicos

- Analizar los equipos de seguridad electrónica para la prevención de robo que existen en el mercado automotriz.
- Establecer los parámetros para dispositivos de visión artificial empleados para la seguridad de vehículos.
- Diseñar el sistema para el monitoreo y control de vehículos.

## **CAPÍTULO II. METODOLOGÍA**

### **2.1 Materiales**

El dispositivo realizado en el presente proyecto de investigación cuenta con sistemas de control y monitorización de vehículos en situaciones antirrobo que utilizan visibilidad artificial requieren de varios materiales básicos para su efectiva implementación. Entre los elementos claves se incluyen cámaras de vigilancia de alta resolución para capturar imágenes claras y detalladas, sensores de movimiento para iniciar la detección de eventos sospechosos y placas de desarrollo como Raspberry Pi para procesar y analizar la información recopilada. Además, para establecer una conexión se requieren componentes como un módulo WiFi o un módem celular que permitan la transferencia de datos a través de Internet. El uso de una base de datos para almacenar registros de eventos y una fuente de energía constante (mediante baterías recargables o conexiones eléctricas) son esenciales para el funcionamiento continuo del sistema.

La implementación de algoritmos de visión por computadora, posiblemente utilizando redes neuronales convolucionales (CNN), también es esencial para la detección y clasificación efectiva de situaciones de robo.

### **2.2 Métodos**

#### **2.2.1 Modalidad de la investigación**

El presente proyecto se sustentó mediante una investigación aplicada debido a que se utilizó los conocimientos que fueron adquiridos durante los años de carrera y permitieron el análisis de aspectos económicos, recursivos y temporales para proponer una solución mediante un sistema utilizando visión artificial.

Se empleó investigación bibliográfica, ya que se buscó soluciones mediante la recopilación de información de datos bibliográficos mediante libros, tesis, revistas, anuarios, publicaciones científicas, fichas bibliográficas, fichas nemotécnicas e informes estadísticas.

Igualmente, se empleó la investigación de campo, para obtener la información relacionada a los procesos utilizados para la verificación del producto final.

### **2.2.2 Población y muestra**

La población comprende todos los vehículos propiedad de la empresa "Confecciones GALAR" que se encuentran estacionados en sus instalaciones. Esto incluye automóviles de empleados, vehículos de carga y cualquier otro medio de transporte de la empresa.

La muestra para este proyecto consiste en un subconjunto representativo de vehículos dentro de la población mencionada. Se eligió los vehículos principales para el transporte de mercadería, así como sus modelos y ubicaciones en el área de estacionamiento de la empresa para garantizar una diversidad adecuada. La muestra se seleccionó de manera estratégica para abarcar condiciones variables y maximizar la eficacia del sistema en diferentes escenarios.

### **2.2.3 Recolección de información**

Para la recolección de información se llevó a cabo una investigación exhaustiva de la información, combinando métodos de investigación de campo y revisión bibliográfica. La investigación de campo se utilizó específicamente para evaluar el proceso de monitoreo y control de los vehículos en situaciones reales, mientras que la investigación bibliográfica se centró en identificar y analizar las mejores prácticas y técnicas existentes para el manejo del sistema. Esta combinación de enfoques permitió obtener una comprensión integral del tema y garantizar la adopción de estrategias efectivas en el desarrollo del sistema.

### **2.2.4 Procesamiento y análisis de datos**

Para el procesamiento y análisis de datos se realizaron los siguientes pasos:

- Revisión de la información recopilada.

- Estudio de las propuestas de solución planteadas para el monitoreo y control con visión artificial.
- Planteamiento de la propuesta de solución.
- Control y verificación de los datos obtenidos mediante el testeado del dispositivo.



## **CAPÍTULO III.- RESULTADOS Y DISCUSIÓN**

### **3.1 Desarrollo de la propuesta**

Con el fin de cumplir cada uno de los objetivos propuestos de sistema electrónico de control y monitorización de vehículos en caso de robo para la empresa “Confecciones GALAR” mediante visión artificial, lo cual permite al usuario controlar y monitorizar mediante cámaras que estarán colocados en lugares que cubren el campo de visualización del vehículo y detectar la presencia de individuos que pretendan sustraerse o robar el automóvil.

Para el desarrollo del sistema se estableció diversos métodos de adquisición de imágenes, los cuales permitieron identificar diversos tipos de robo a vehículos mediante la visión artificial. Los algoritmos de visión artificial empleados lograron establecer una detección idónea que permite detectar a personas sospechosas. También, se realizó una interfaz que permitirá visualizar a cada instante al automóvil, esta interfaz es amigable con el usuario.

La evaluación del entrenamiento de algoritmos arrojó resultados aceptables con una precisión de 0,95 y un rendimiento del 95%. Esto demuestra un rendimiento excelente en la identificación y clasificación de casos positivos y negativos en el conjunto de datos proporcionado. Sin embargo, es importante señalar que la efectividad del sistema puede variar con diferentes conjuntos de datos y situaciones de prueba, por lo que se necesitan más pruebas para garantizar su confiabilidad en diferentes entornos del mundo real.

### **3.2 Equipos de seguridad electrónica para la prevención de robo que existen en el mercado automotriz.**

#### **3.2.1 Equipos de seguridad electrónica existentes**

A continuación, se realizó un análisis para la información sobre los equipos de seguridad electrónicos existentes en el mercado.

### ***a. Sistemas de Alarmas***

Las alarmas desempeñan una función crucial al disuadir situaciones de peligro o amenaza mediante la emisión de tonos o sonidos codificados distintivos. Estas alarmas son pasivas, lo que significa que responden a impactos o a la interrupción de la señal o circuito del interruptor que llevan las puertas. Es necesario ajustar la sensibilidad a golpes o impactos, ya que algunas alarmas vienen preconfiguradas con una sensibilidad alta, lo que puede provocar falsas alarmas al pasar cerca de vehículos pesados. [27]

Además, se pueden configurar varias funciones preventivas contra robos mediante la alarma, como la apertura del maletero eléctrico desde el control, el cierre de las lunas eléctricas, el apagado remoto del vehículo, el bloqueo del motor por puerta, el reconocimiento del conductor, el bloqueo del motor por llave o contacto al encender el vehículo, y la activación automática del sistema de alarma. La configuración adecuadamente estas funciones es esencial para evitar falsas alarmas y garantizar un rendimiento efectivo del sistema de seguridad del vehículo. [27]

### ***b. Sistemas de Alarma de Doble Vía Digital***

Los sistemas de seguridad, como la "Alarma Digital de doble vía", se destacan como una de las mejores opciones para garantizar la seguridad de un vehículo. En algunos casos, dependiendo del fabricante, se ofrece la posibilidad de contar con dos controles digitales de pantalla para monitorear el estado del vehículo. También existe la opción de tener un control analógico estándar junto con un control digital, aunque la preferencia general tiende a inclinarse hacia el uso de dos controles digitales. [27]

Este modelo de alarma digital de doble vía ofrece características de seguridad excepcionales. Por ejemplo, la capacidad de alcanzar un rango de más de 500 metros es una ventaja significativa. A través del control digital, los usuarios pueden recibir notificaciones mediante vibración y sonidos en caso de que el vehículo sufra un impacto o si alguna puerta, maletero o capó es abierto. Estos eventos se presentan claramente en la pantalla LCD a través de sonidos, vibración e imágenes. [27]

Las alarmas digitales de doble vía permiten encender el vehículo a distancia mediante el control, ofreciendo incluso la opción de programar el encendido y apagado del motor. También brindan la posibilidad de apagar el motor a distancia mediante el control en caso de robo, lo que añade una capa adicional de seguridad al sistema. [27]

#### ***c. Dispositivos de Seguridad "Cortacorriente"***

Existen diversas variantes de sistemas de corte de corriente o bloqueadores de motor, y un ejemplo de estos es el sistema de bloqueo de motor conocido como "Toque Pluma" o "Anti Atraco", fabricado en Perú. En el mercado, siempre se encuentra clientes que buscan la máxima seguridad para sus vehículos, así como aquellos que prefieren opciones más básicas y económicas. En este contexto, el sistema de seguridad "Toque Pluma" destaca como una alternativa económica y accesible para aquellos usuarios que buscan una solución de seguridad más asequible. [27]

#### ***d. Dispositivo Antirrobo a Distancia o Inmovilizador***

Este constituye uno de los sistemas de seguridad vehicular más antiguos, y está disponible en varios diseños y marcas. Los sistemas de inmovilización o bloqueo de motor a distancia, también conocidos como anti asalto por distancia, se caracterizan por ser dispositivos totalmente silenciosos, ya que al activarse no emiten ningún sonido audible. [27]

Estos sistemas proporcionan medidas efectivas contra robos a distancia, actuando como un antirrobo y un inmovilizador de moto al mismo tiempo, gracias a su tecnología de transceptor. [27]

#### ***e. Cámaras de CCTV***

Las cámaras de monitoreo se utilizan ampliamente en sistemas de seguridad y vigilancia en automóviles, hogares y negocios. Las imágenes de alta definición de las cámaras modernas permiten la identificación de personas y el control total sobre la ubicación monitoreada. Hay soluciones en tecnología IP y HDCVI que se adaptan a proyectos de diferentes costos, estructuras y tamaños, junto con grabadoras digitales y software y aplicaciones que complementan el sistema. Además, también existen

modelos compactos, Plug & Play, para las personas que desean seguridad de sus activos: son los kits o sistemas de monitoreo más básicos. [27]

### **3.2.2 Equipos de seguridad electrónica para vehículos en el Ecuador**

En la actualidad, Ecuador experimenta una creciente adopción de equipos de seguridad electrónica como parte integral de sus sistemas de vigilancia. Los sistemas de cámaras de video vigilancia son ampliamente utilizados para monitorear espacios públicos y privados, contribuyendo significativamente a la prevención y detección de delitos. Estos sistemas suelen integrar tecnologías de reconocimiento facial y análisis de comportamiento para mejorar la eficacia en la identificación de personas y eventos sospechosos. [11]

Los sistemas de control de acceso electrónicos también han ganado popularidad en Ecuador, especialmente en instalaciones corporativas y gubernamentales. Estos equipos utilizan tarjetas de proximidad, biométricos o dispositivos móviles para gestionar de manera eficiente y segura el acceso a edificios y áreas restringidas, brindando un mayor control sobre la seguridad física de los espacios. [11]

Los sistemas de alarma electrónica, conectados a centrales de monitoreo, son esenciales para la protección de hogares y negocios. Estos dispositivos detectan intrusiones, incendios y otros eventos de emergencia, enviando notificaciones instantáneas a las autoridades o propietarios. La combinación de sensores avanzados y sistemas de comunicación modernos ha mejorado significativamente la capacidad de respuesta y la fiabilidad de estos equipos. [11]

En el ámbito vehicular, los sistemas de rastreo GPS y las alarmas antirrobo son comunes en Ecuador. Estos dispositivos contribuyen a la seguridad de vehículos mediante la localización en tiempo real y la detección de actividades sospechosas, lo que ayuda a reducir los índices de robo de autos en el país. [11]

La ciberseguridad también ha cobrado importancia, especialmente con el aumento de la conectividad. Las empresas y organizaciones ecuatorianas implementan medidas avanzadas para proteger sus redes y sistemas informáticos, evitando ciberataques y preservando la integridad de la información sensible. [11]

### 3.2.3 Expresión corporal

Las expresiones corporales conocidas también como gestos corporales constituyen una parte integral de la comunicación humana y se pueden interpretar de diversas maneras. En este caso durante un robo pueden variar dependiendo de si la persona está perpetrando el acto o si es la víctima. En el caso del asaltante, algunos gestos pueden incluir movimientos bruscos, posturas agresivas, miradas furtivas, manos ocultas o en los bolsillos, y una actitud general de nerviosismo o tensión. Estos gestos pueden ser señales de que la persona está preparada para cometer un acto delictivo y puede representar un peligro potencial. Por otro lado, la víctima de un robo puede exhibir gestos corporales de miedo, sumisión o ansiedad. Estos gestos pueden incluir encogerse, miradas nerviosas, temblores, manos temblorosas o levantadas en señal de rendición, y una expresión facial de miedo o pánico. La víctima también puede mostrar signos de congelamiento o parálisis debido al miedo extremo.

En la Tabla 5, se muestra los gestos corporales en caso de ser el asaltante.

Tabla 5. Gestos corporales en caso de ser el asaltante

<b>Expresiones Corporales</b>	<b>Descripción</b>
<b>Movimientos bruscos</b>	Se puede detectar por Movimientos repentinos y rápidos, como gestos agresivos o movimientos de manos bruscos, que pueden indicar una actitud impulsiva o agresiva por parte del asaltante.
<b>Posturas agresivas</b>	Cuando adoptan posturas físicas que denotan dominio y agresividad, como una postura erguida, hombros anchos, pecho hacia adelante o un paso firme y decidido hacia la víctima.
<b>Miradas furtivas</b>	Dirige miradas evasivas o inquietas que buscan evitar el contacto visual directo, a menudo acompañadas de movimientos de ojos rápidos, que pueden indicar nerviosismo o intenciones ocultas.
<b>Manos ocultas o en bolsillos</b>	Mantiene las manos fuera de la vista o en los bolsillos, lo cual puede sugerir que el asaltante está ocultando algo o preparándose para un acto delictivo sin revelar sus intenciones.
<b>Uso de objetos sospechosos</b>	La detección de personas llevando objetos inusuales o sospechosos, como palancas, herramientas de corte, o bolsas grandes y vacías en situaciones donde no se justifiquen, podría indicar actividades delictivas.
<b>Actitud general de nerviosismo o tensión</b>	Exhibe signos de nerviosismo, como temblor, sudoración, respiración rápida o una expresión facial tensa, que pueden ser indicativos de que el asaltante está bajo presión o anticipando una confrontación.

De la misma forma, las expresiones corporales que presenta una persona al ser víctima de un ataque se pueden caracterizar en función de sus gestos. En la Tabla 6, se muestra los gestos corporales en caso de ser la víctima.

Tabla 6. Gestos corporales en caso de ser la víctima

<b>Gestos Corporales</b>	<b>Descripción</b>
<b>Encogerse</b>	Adopta una postura encogida o encorvada, con los hombros caídos y el cuerpo hacia atrás, indicando sumisión o miedo ante una situación amenazante.
<b>Miradas nerviosas</b>	Dirige miradas inquietas o nerviosas, con movimientos rápidos de los ojos que buscan constantemente el entorno, mostrando ansiedad o preocupación por la situación.
<b>Temblores corporales</b>	Experimenta temblores corporales, que pueden manifestarse como movimientos sutiles o visibles de las extremidades, manos o voz, indicando nerviosismo o miedo intenso.
<b>Manos temblorosas o levantadas en señal de rendición</b>	Las manos tiemblan de forma visible o se levantan en señal de rendición, mostrando sumisión o una clara intención de no representar una amenaza, en un intento por evitar la confrontación.
<b>Expresión facial de miedo o pánico</b>	Presenta una expresión facial marcada por el miedo o el pánico, con los ojos abiertos, cejas levantadas, boca entreabierta o una mirada vidriosa, mostrando un estado emocional de gran temor.

En algunos casos las cámaras no podrán detectar de mejor manera los rasgos corporales, por lo cual dificultaría el reconocimiento de ser un asalto o no.

La detección de gestos corporales a través de la cámara se pueden identificar situaciones de robo puede enfrentar varios desafíos. Uno de los problemas principales es la variabilidad en las condiciones de iluminación, ya que en entornos oscuros o con cambios bruscos de luz, la calidad de la imagen puede deteriorarse, lo que dificulta la identificación precisa de gestos. Además, la presencia de obstáculos oclusivos en el campo de visión de la cámara, como personas u objetos, puede obstruir la visualización de los gestos corporales relevantes. Otra limitación importante es la diversidad en la apariencia y el comportamiento humano, lo que dificulta el desarrollo de algoritmos de detección robustos que puedan generalizarse a diferentes personas y situaciones. Además, la velocidad de procesamiento de las cámaras y la capacidad de análisis en tiempo real pueden ser insuficientes para capturar y procesar gestos rápidos o sutiles,

lo que puede resultar en una detección inexacta o tardía de situaciones de robo. Por último, las preocupaciones relacionadas con la privacidad y la ética pueden surgir debido a la grabación y análisis continuo de las imágenes de las personas, lo que puede generar resistencia por parte de la sociedad.

En la Tabla 7, se muestra los problemas que tendría la cámara al momento de capturar el video e imagen.

Tabla 7. Problemas de captura de video e imagen

<b>Problema</b>	<b>Descripción</b>
<b>Iluminación deficiente</b>	La iluminación inadecuada puede dificultar la detección precisa de gestos corporales, especialmente en entornos oscuros.
<b>Variabilidad en la apariencia humana</b>	Las diferencias en la apariencia física de las personas pueden dificultar la creación de algoritmos de detección robustos.
<b>Obstrucciones oclusivas</b>	Objetos o personas que obstruyen parcialmente la visión pueden interferir con la detección precisa de gestos corporales.
<b>Ruido de fondo</b>	La presencia de ruido visual o distracciones en la escena puede dificultar la identificación clara de gestos corporales.
<b>Limitaciones en la resolución de imagen</b>	Una resolución de imagen insuficiente puede afectar negativamente la capacidad de la visión artificial para detectar gestos.
<b>Movimientos rápidos</b>	Los movimientos rápidos o abruptos pueden ser difíciles de seguir y detectar con precisión para los algoritmos de visión artificial.
<b>Variabilidad cultural</b>	Los gestos corporales pueden variar según la cultura, lo que puede dificultar la creación de algoritmos universalmente efectivos.
<b>Falta de etiquetado de datos adecuado</b>	La falta de datos de entrenamiento etiquetados adecuadamente puede afectar la capacidad de los modelos para aprender y generalizar.
<b>Problemas de privacidad</b>	La detección de gestos corporales plantea preocupaciones éticas y de privacidad que deben abordarse de manera adecuada.

### 3.3 Arquitectura

En la Figura 11, se muestra el esquema general del sistema de control y monitorización que está conformado por 4 etapas, las cuales se explicaran a continuación:

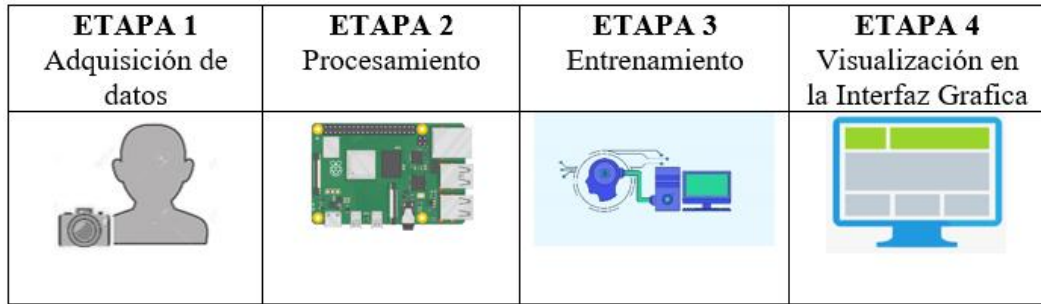


Figura 11. Arquitectura

**Etapa 1.-** El proceso comienza con la recopilación de imágenes o videos en tiempo real de las cámaras integradas en el sistema de video vigilancia. Las cámaras estarán dispuestas de tal forma que puedan abarcar la mayor área posible, con la finalidad de recolar información del lugar en el que se encuentran los vehículos de la empresa CONFECIONES GALAR.

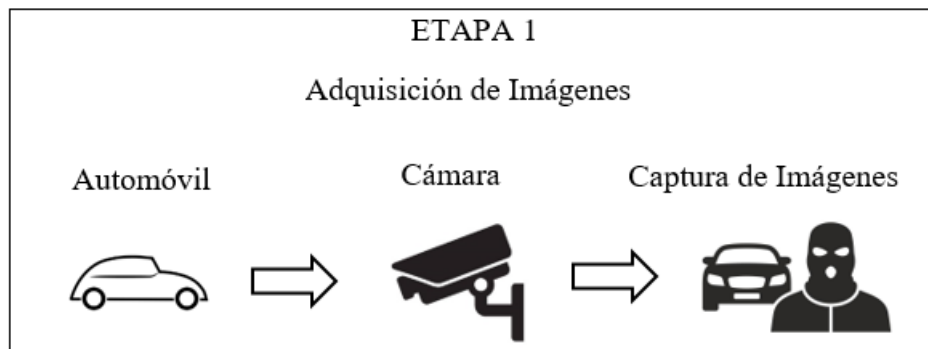


Figura 12. Arquitectura de adquisición de datos

**Etapa 2.-** Los datos recogidos se envían a un microcontrolador (Raspberry Pi 3 Modelo B+) para su procesamiento.

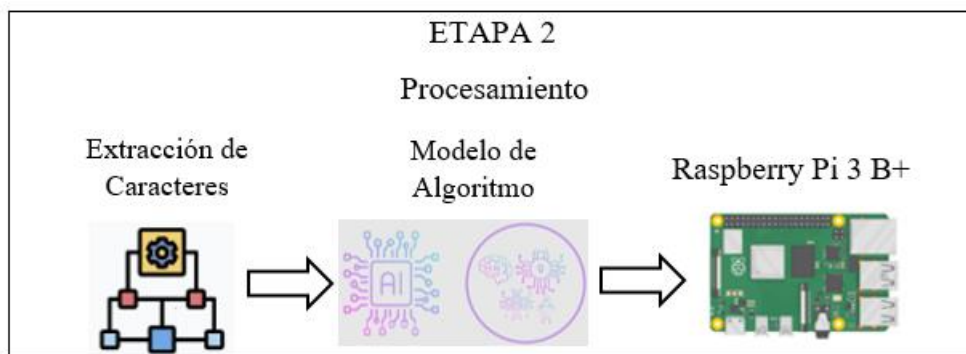


Figura 13. Arquitectura de procesamiento de datos del sistema



**Etapa 3.-** Se utilizaron técnicas de visión artificial, específicamente redes neuronales, para entrenar un modelo con la capacidad de identificar con precisión a las personas que intentaban robar o sustraerse un vehículo, el modelo implementado emplea un servicio de alojamiento en la nube, lo que permitió un uso eficiente y efectivo.

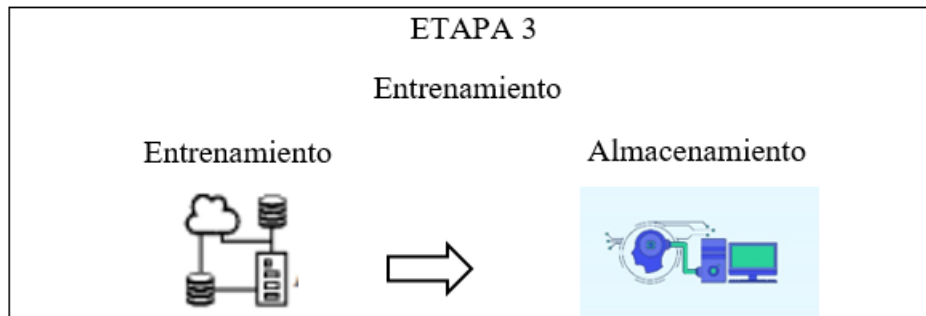


Figura 14. Arquitectura de entrenamiento y almacenamiento en la nube

**Etapa 4.-** El modelo controla y monitorea constantemente el vehículo y puede detectar personas que intentan robarlo. Los resultados se pueden visualizar a través de una interfaz intuitiva y fácil de usar.

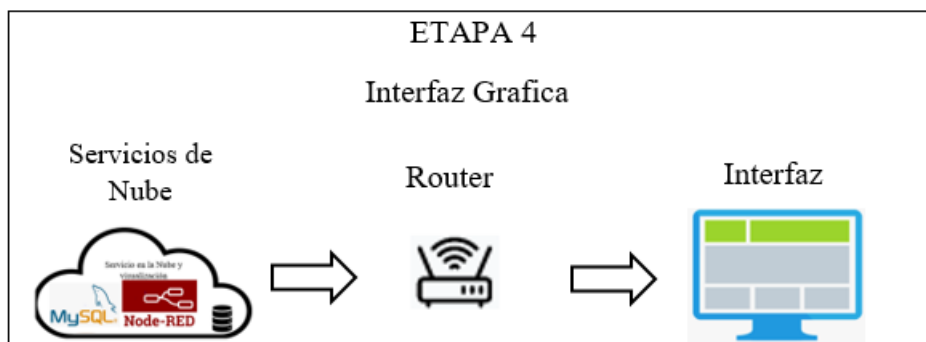


Figura 15. Arquitectura de servicio y visualización en la interfaz gráfica

**b. Diagrama de bloques**

En la Figura 16, se muestra los detalles de los bloques del sistema implementado.

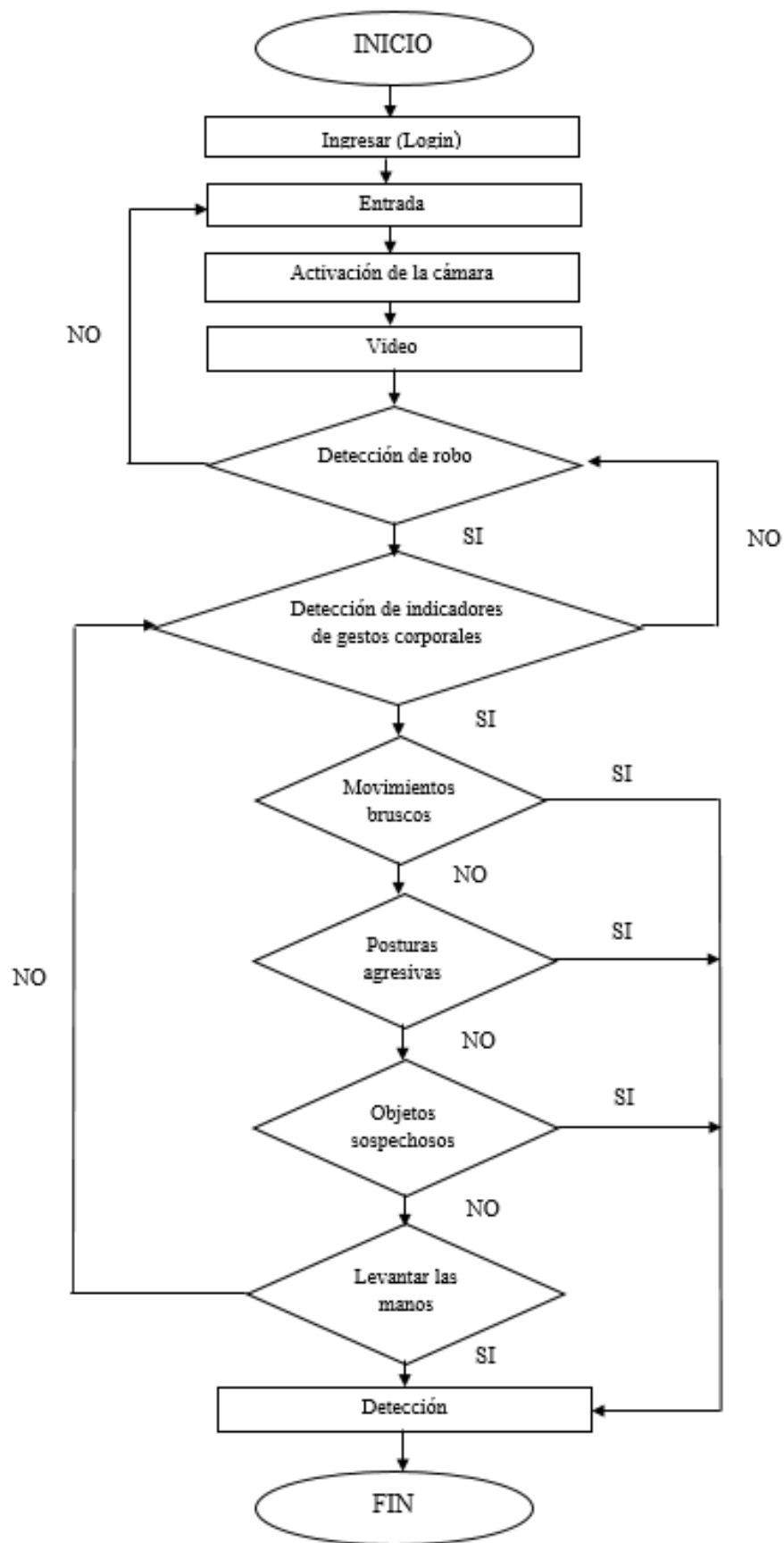


Figura 16. Diagrama de bloques del sistema

### **3.4 ETAPA 1.- Adquisición**

Todos los componentes electrónicos necesarios para construir el sistema han sido cuidadosamente seleccionados para garantizar que cumplan con los requisitos básicos para lograr los resultados deseados.

### **3.5 Desarrollo del sistema**

#### **3.5.1 Adquisición de Imágenes**

Al tomar fotografías con una cámara, hay varios aspectos a considerar, como la configuración de iluminación, la calidad de la resolución de la cámara, la distancia del sujeto y, lo más importante, el objetivo principal es extraer datos valiosos de la imagen.

Cuando se trata de sistemas, hay algunos elementos a considerar:

El sistema de iluminación es un componente extremadamente importante, ya que una imagen de entrada adecuada optimiza las posibilidades de procesamiento posterior. Cuanto más optimizado sea el esquema de iluminación, mayor será la eficiencia del sistema de reconocimiento.

En cuanto a las cámaras y lentes, su función es utilizar sensores electrónicos para capturar y enfocar imágenes de una escena y convertir los cambios de luz en señales eléctricas. Hay una variedad de cámaras disponibles y la elección puede basarse en las necesidades específicas de la aplicación, ya sea analógica o digital.

Cabe señalar que no es apropiado utilizar cámaras de baja resolución para analizar imágenes desde el espacio.

En la Figura 17, se muestra el diagrama de flujo de la etapa de adquisición.



Figura 17. Diagrama de flujo de la etapa de adquisición

La importancia de la cámara en la visión artificial radica en su capacidad para capturar datos visuales del entorno, los cuales son fundamentales para que los algoritmos de visión artificial puedan procesar, analizar y comprender el mundo que los rodea. Las cámaras proporcionan imágenes y videos que contienen una rica variedad de información visual, incluidas formas, colores, texturas y movimientos.

### 3.5.2 Selección de cámara

#### Cámara web digital D-Bugg M26

Una cámara web digital es un dispositivo de grabación de imágenes y videos que se conecta a la computadora o dispositivo compatible mediante USB. Estas cámaras están destinadas a facilitar la comunicación en línea y permitir la captura de imágenes en tiempo real. En el campo de la visión artificial (VA), las cámaras web digitales se pueden utilizar como fuentes de entrada de video para aplicaciones de reconocimiento de objetos y visión por computadora.

En la Figura 18 se muestra la Cámara web digital D-Bugg M26.



Figura 18. Cámara web digital D-Bugg M26 [28]

### **Cámara Pi Noir V2**

La cámara Pi Noir V2 está diseñada específicamente para ser usadas con las placas Raspberry Pi y es compatible con Jetson Nano. Equipada con un sensor de imagen Sony IMX219 de 8 megapíxeles, esta cámara se diferencia de las cámaras tradicionales en que está optimizada para capturar imágenes en condiciones de poca luz y en ausencia de luz visible. Una característica especial de la cámara Pi Noir V2 es que no tiene filtro de infrarrojos, lo que la hace ideal para aplicaciones de visión nocturna y vigilancia en ambientes oscuros. En el campo de la inteligencia artificial, la cámara Pi Noir V2 se suele utilizar en proyectos de visión por computadora que requieren capturar imágenes en entornos oscuros.

En la Figura 19 se muestra la Cámara Pi Noir V2.

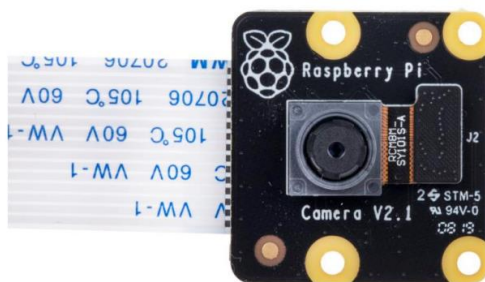


Figura 19. Cámara Pi Noir V2 [29]

### **Cámara Logitech C920 HD PRO WEBCAM**

Esta cámara está diseñada con elementos ópticos y sensores de alta calidad para capturar imágenes claras y de alta resolución. En inteligencia artificial, las cámaras Logitech se utilizan como dispositivos de entrada de vídeo para aplicaciones de

reconocimiento de objetos y visión por computadora. Las cámaras Logitech son perfectas para una variedad de tareas como reconocimiento facial, seguimiento de objetos, análisis de sentimientos y más.

En la Figura 20 se muestra la Cámara Logitech C920 HD PRO WEBCAM.



Figura 20. Cámara Logitech C920 HD PRO WEBCAM [30]

### **Cámara SV3C HX Series**

Esta cámara exterior WiFi con resolución de 1080P está diseñada para funcionar con una red WiFi de 2,4 GHz. Ofrece funciones como detección de movimiento, notificaciones por correo electrónico, acceso remoto, apagado inteligente de infrarrojos para visión diurna y nocturna y resistencia a la intemperie con grado de protección IP66. Fácil de instalar, es una excelente opción para vigilancia interior y exterior. Esto garantiza una grabación precisa de alertas y eventos de movimiento, y permite a los usuarios ajustar el rango de detección y la sensibilidad de la cámara según sus preferencias a través de la aplicación móvil o la PC. Además, también da la opción de establecer un horario específico para la detección de movimiento y activarlo solo en ciertos momentos del día.

En la Figura 21 se muestra la Cámara SV3C HX Series



Figura 21. Cámara SV3C HX Series [31]

### 3.5.3 Comparativa de Cámaras

En la Tabla 8, se muestra las características principales de la selección de la cámara.

Tabla 8. Características principales de cada cámara [28] [29] [30] [31]

TIPO	Webcam D-Bugg M26	Pi Noir V2	Logitech C920 HD PRO WEBCAM	Cámara SV3C HX Series
<b>Tamaño</b>	19 x 7,6 x 12,2 cm	Alrededor de 25 × 24 × 11,5 mm	21 x 7,6 x 15,2 cm	7,87 x 2,76 x 2,56 pulgadas
<b>Resolución fija</b>	90 fps 4096 px x 2160 Px	8 megapíxeles	15.0 megapíxeles	5.0 megapíxeles HD
<b>Sensor</b>	No	Sony IMX219	-	Motion Detection
<b>Compatibilidad con la Raspberry pi</b>	No	Si	Si	Si
<b>Profundidad de campo</b>	No	Aproximadamente 10 cm a ∞	71 mm	Zoom
<b>Longitud focal</b>	28 mm	4,74 mm	-	360 grados
<b>Costo</b>	\$ 15.00	\$ 25.00	\$ 60.00	\$ 70.00

Después de considerar los requerimientos específicos del proyecto, se decidió elegir el módulo de Cámara SV3C HX Series, debido a su velocidad para realizar capturas y procesamiento de imagen, así como por su compatibilidad con el microordenador seleccionado.

En la Figura 22 se muestra la instalación de las cámaras de seguridad.



Figura 22. Instalación de las cámaras

### **3.6 ETAPA 2.- Procesamiento**

#### **3.6.1 Librerías o Bibliotecas requeridas**

Las librerías o bibliotecas de Python son conjuntos de funciones predefinidas y módulos que facilitan el desarrollo de software al proporcionar herramientas y utilidades listas para usar. Estas bibliotecas abarcan una amplia gama de funcionalidades y áreas de aplicación, desde el procesamiento de datos y la manipulación de archivos hasta el desarrollo de aplicaciones web y el aprendizaje automático. Al aprovechar las librerías de Python, los desarrolladores pueden acelerar el proceso de desarrollo, reducir la cantidad de código necesario y beneficiarse de la experiencia colectiva de la comunidad de programadores. Algunas de las librerías más populares y ampliamente utilizadas en Python incluyen NumPy para computación numérica, pandas para manipulación y análisis de datos, Matplotlib para visualización de datos, TensorFlow. Es de gran ayuda la extensa colección de librerías disponibles, Python se ha convertido en uno de los lenguajes de programación más versátiles y poderosos para una variedad de aplicaciones.



En la Tabla 9, se muestra las librerías utilizadas en el sistema

Tabla 9. Librerías utilizadas en el sistema

<b>Bibliotecas</b>	<b>Características</b>
<b>tkinter</b>	Esta biblioteca permite tiene como finalidad, representar una interfaz gráfica de usuario (GUI) estándar de Python y proporciona widgets y herramientas para la creación de aplicaciones, además, se utiliza para crear ventanas, botones, etiquetas, cuadros de texto y otros elementos de la interfaz de usuario.
<b>pil</b>	Proporciona herramientas para abrir, manipular y guardar imágenes en varios formatos, además de ser una biblioteca que permite operaciones básicas de procesamiento de imágenes como recorte, rotación, redimensionamiento, conversión de formatos, etc.
<b>cv2</b>	OpenCV se utiliza para la visión por computadora y procesamiento de imágenes, es una biblioteca que proporciona una amplia gama de herramientas y algoritmos para tareas como detección de objetos, seguimiento de objetos, reconocimiento facial, calibración de cámara, además, ofrece una implementación rápida de algoritmos de visión por computadora mediante el uso de optimizaciones de bajo nivel.
<b>os</b>	Esta librería proporciona funciones para trabajar con archivos, directorios y otras operaciones relacionadas con el sistema de archivos, también, permite crear, eliminar, copiar y mover archivos y directorios, así como acceder a variables de entorno y ejecutar comandos del sistema.
<b>Tensorflow.keras.models</b>	TensorFlow permite construir y entrenar modelos de aprendizaje profundo que proporcionan una interfaz de alto nivel para crear y manipular modelos de redes neuronales, además incluye funciones para cargar modelos preentrenados, compilar modelos con diferentes configuraciones de optimización y pérdida, y realizar inferencia con modelos entrenados.
<b>Tensorflow.keras.layers</b>	Esta biblioteca proporciona una amplia gama de capas predefinidas que se pueden combinar para construir arquitecturas de redes neuronales complejas e incluye capas convolucionales, capas de agrupamiento, capas totalmente conectadas, capas de regularización.
<b>numpy</b>	Es una biblioteca que proporciona estructuras de datos eficientes y funciones para realizar operaciones matemáticas en matrices y arreglos multidimensionales, siendo utilizado ampliamente en aplicaciones de ciencia de datos, aprendizaje automático, procesamiento de señales, entre otros y ofrece una amplia gama de funciones matemáticas y herramientas para trabajar con datos numéricos de manera eficiente.

- ***Bibliotecas utilizadas en el código de programación***

En la Figura 23, se muestra las bibliotecas utilizadas en el código de programación.

```

import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk
import cv2
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
import numpy as np
import os





```

Figura 23. Bibliotecas utilizadas en el código de programación

### 3.6.2 Selección del Microcontrolador

En la Tabla 9 se muestra la comparación de los diferentes tipos de microcontroladores.

Tabla 10. Comparación de diferentes tipos de microcontroladores [32] [33] [34] [35]

Tarjeta				
Modelo	<b>Brix Celeron BPCE 3455C</b>	<b>Raspberry Pi 4</b>	<b>Jetson Nano</b>	<b>RASPBerry PI 3+ Modelo B</b>
Microprocesador	Intel Celeron 3455U	Broadcom BCM2711	NVIDIA Tegra X1	Broadcom BCM2837B0
Conector	HDMI, USB, RJ45, otros	HDMI, USB, GPIO, otros	HDMI, USB, GPIO	HDMI, USB, GPIO, otros
Comunicación	Ethernet, Wi-Fi, Bluetooth	Ethernet, Wi-Fi, Bluetooth, GPIO	Ethernet, Wi-Fi, Bluetooth	Ethernet, Wi-Fi, Bluetooth, GPIO
Pines	N/D	40 GPIO pins	40-pin GPIO header	40-pin GPIO header
Voltaje de E/S	N/D	3.3V	3.3V	3.3V
Voltaje de entrada (nominal)	N/D	5V	5V	5V
Corriente de consumo	N/D	Variable	Variable	Variable
Conector de fuente de alimentación	N/D	USB-C	Barrel jack	Micro USB
Memoria	N/D	2GB, 4GB, 8GB RAM	4GB RAM	1GB RAM
Procesador principal	CPU x86_64	CPU ARM Cortex-A72	CPU ARM Cortex-A57	CPU ARM Cortex-A53
Dimensiones	N/D	85mm x 56mm x 17mm	100mm x 80mm	85mm x 56mm x 17mm
Temperatura	N/D	0°C to 50°C	0°C to 50°C	0°C to 70°C
Precio	195\$	230\$	210\$	180\$

El microcontrolador seleccionado debe tener unas dimensiones reducidas, ya que se instala en el vehículo y está diseñado para ocupar el menor espacio posible. Además, requiere un consumo energético mínimo y puede soportar altas temperaturas.

Actualmente, existen muchos tipos de microcontroladores diseñados para diferentes aplicaciones, por lo que sólo se describirán brevemente algunos de ellos. Se debe verificar la capacidad de memoria del microcontrolador ya que necesita procesar la información del módulo GPS y ejecutar el programa utilizando pesos de red neuronal para un monitoreo continuo.

Se eligió la placa Raspberry Pi 3 Modelo B+ para este proyecto basándose en varias consideraciones importantes, lo que la convierte en la mejor opción para un sistema antirrobo de vehículos, ya que logra el equilibrio adecuado entre rendimiento y consumo de energía, proporcionando la potencia necesaria para ejecutar de manera eficiente algoritmos de visión por computadora sin comprometer la eficiencia energética. Además, sus funciones de conectividad, que incluyen Wi-Fi y Bluetooth integrados, facilitan la comunicación inalámbrica dentro del sistema. La versatilidad de Raspberry Pi y su gran comunidad de desarrollo y soporte facilitan la integración de los diversos módulos y sensores necesarios para el monitoreo de vehículos. Finalmente, su coste asequible lo convierte en una opción atractiva para implementaciones a gran escala, proporcionando proyectos con una buena relación calidad-precio.



Figura 24. Tarjeta Raspberry Pi 3+ Modelo B [35]



Figura 25. Implementación de la Tarjeta Raspberry Pi 3+ Modelo B en el automóvil.




### 3.6.3 Selección del Enrutador

Un enrutador es un dispositivo básico de red informática responsable de dirigir y gestionar el flujo de datos entre diferentes redes o subredes informáticas. El dispositivo utiliza la información de la dirección IP para determinar la mejor ruta para entregar datos desde el origen al destino. Actúa como puerta de enlace entre redes, facilitando la comunicación eficiente y el intercambio de información entre dispositivos conectados a la red.

Además de las funciones básicas de enrutamiento, algunos enrutadores modernos incluyen funcionalidad de firewall, filtrado de contenido y funciones de seguridad para proteger la red de amenazas externas.

En la Tabla 11 se muestra los parámetros técnicos de cada enrutador.

Tabla 11. Parámetros técnicos de cada enrutador

Enrutador			
Parámetros	Router HP	Router Huawei	Router TP-Link
<b>Velocidad inalámbrica</b>	1200 Mbps	1500 Mbps	1000 Mbps
<b>Estándares inalámbricos</b>	802.11ac, 802.11n	802.11ax, 802.11ac	802.11ac, 802.11n
<b>Puertos LAN</b>	4 puertos Gigabit Ethernet	3 puertos Gigabit Ethernet	4 puertos Gigabit Ethernet
<b>Puertos WAN</b>	1 puerto Gigabit Ethernet	1 puerto Gigabit Ethernet	1 puerto Gigabit Ethernet
<b>Seguridad</b>	WPA3, cifrado AES	WPA3, cifrado AES	WPA3, cifrado AES
<b>Características especiales</b>	QoS, Beamforming	MU-MIMO, IPv6 support	Guest Network, Parental Control
<b>Interfaz de usuario</b>	Web-based, intuitiva	Huawei HiLink, fácil de usar	TP-Link Tether, amigable
<b>Característica</b>	Router HP	Router Huawei	Router TP-Link
<b>Precio</b>	35\$	40\$	27\$

Los enrutadores HP son una buena opción para los usuarios que buscan un rendimiento sólido y velocidades inalámbricas de hasta 1200 Mbps. Utiliza los estándares inalámbricos 802.11ac y 802.11n, así como funciones como QoS y formación de haces para garantizar una conectividad eficiente y mejorar la experiencia del usuario.

Los enrutadores Huawei se distinguen por tecnologías avanzadas con velocidades inalámbricas de hasta 1500 Mb/s y soporte para los estándares 802.11ax y 802.11ac. Equipado con funciones como MU-MIMO y compatibilidad con IPv6, proporciona un rendimiento excelente y es ideal para entornos domésticos o empresariales.

Se eligió el enrutador TP-Link por su equilibrio entre rendimiento y funcionalidad. Tiene una velocidad inalámbrica de 1000 Mbps y estándares 802.11ac y 802.11n para una conexión confiable. Sus características especiales, como red de invitados y controles parentales, lo convierten en una opción versátil para una variedad de necesidades.

En la Figura 26 se muestra la instalación del enrutador TP Link.





Figura 26. Instalación del enrutador TP Link

#### **3.6.4 Selección del respaldo (Baterías)**

El sistema de seguimiento debe funcionar de forma continua. Por lo tanto, si se corta la alimentación principal del vehículo, se espera que el sistema cambie automáticamente a la batería de respaldo. Por este motivo, es importante elegir una batería con la autonomía necesaria para que el sistema pueda funcionar durante al menos 24 horas sin interrupción. Para ello, se comparó diferentes tipos de respaldos energéticos (baterías) para dispositivos electrónicos, con el objetivo de elegir el más adecuado a las necesidades del proyecto.

Tabla 12. Comparación de diferentes tipos de baterías [36] [37] [38]

<b>Módulo</b>			
<b>Modelo</b>	<b>Batería Ni-Cd</b>	<b>Batería Li-Ion</b>	<b>Bateria Hoovo</b>
<b>Capacidad</b>	Baja	Alta	Muy Alta 6200 mAh
<b>Servicios</b>	Alto Más de 1000 cargas.	Regular Entre 300 a 500 cargas	Regular Entre 500 a 750 cargas
<b>Tiempo de carga</b>	Rápido	Moderado	Moderado
<b>Descripción</b>	Este tipo de baterías son muy resistentes a caídas o al calor	Son delicadas y sensibles al calor.	Son delicadas y sensibles al calor.
<b>Precio</b>	10\$	50\$	40\$

Las baterías de níquel-cadmio (Ni-Cd) y las baterías ionizadas de Litio (Li-Ion) son más robustas, pero significativamente más grandes que las baterías de níquel-cadmio. Además, no ofrecen grandes capacidades de almacenamiento y por tanto no contribuyen al objetivo que se plantea implementar, que es conseguir el dispositivo más compacto posible con autonomía para todo el día.

Después de considerar varias opciones de baterías, se determinó que las baterías Hoovo eran la mejor opción. Esto se debe a su alta resiliencia, que es un requisito básico para los sistemas autónomos, y a una vida útil típica de alrededor de 4 años. Estas ventajas, combinadas con su tamaño compacto, peso ligero y resistencia a los efectos de la memoria, lo convierten en una opción popular para proyectos de dispositivos electrónicos y portátiles.

En la Figura 27 se muestra la Bateria Hoovo.



Figura 27. Bateria Hoovo [38]

En la Figura 28 se muestra la implementación de la batería Hoovo.



Figura 28. Implementación de la batería Hoovo

### 3.6.5 Periféricos de entrada y salida

- Monitor: Dispositivo que presenta información visual, como imágenes o texto, a un usuario. La conexión se realiza a través de una tarjeta gráfica, también llamada adaptador o tarjeta gráfica [39].
- Mouse: Dispositivo de entrada para operar una interfaz de computadora. Sus funciones principales son apuntar, seleccionar o mover datos y archivos y navegar por diferentes aplicaciones [39].
- Teclado: Dispositivo de entrada con el que un usuario interactúa para ingresar datos instantáneamente en una computadora [39].

En la Figura 29 se muestra los periféricos de entrada y salida





Figura 29. Periféricos de entrada y salida

### 3.6.6 Selección de la tecnología inalámbrica

En la Tabla 13, se menciona a detalle las características principales de cada una de las tecnologías inalámbricas.

Tabla 13. Tecnologías Inalámbricas [40]

Parámetros	Wi-Fi	Bluetooth	ZigBee
Estándar	802.11	802.15.1	802.15.4
Frecuencia de la señal	2.4 – 5 GHz	2.4 GHz	2.4 GHz, 915 MHz
Ancho de banda	2 – 100 Mbps	1 Mbps	20 – 250 kbps
Numero de nodos por red	255	65000	7
Distancia de transmisión sin interferencias	50 -100 m	1 - 10 m	1 – 75 m
Memoria	100 kb		32 – 60 kb
Topología	Bus, anillo, estrella y árbol	Punto a punto, estrella y árbol.	Punto a punto, estrella, malla, árbol y clúster

Actualmente existen varias tecnologías inalámbricas viables para implementar este proyecto. La opción Bluetooth queda excluida ya que el alcance está limitado a 10 metros. También se rechaza el uso de la tecnología ZigBee debido a los elevados costes asociados a la misma.

Se decidió utilizar la tecnología Wi-Fi debido a su facilidad de conectividad a Internet y su amplia variedad de dispositivos compatibles.

### 3.7 ETAPA 3.- Entrenamiento

#### 3.7.1 Software

El rendimiento del sistema está estrechamente relacionado con la programación, instalación y configuración precisa del software en el dispositivo para realizar las funciones requeridas.

En la Tabla 14, se muestra la selección de software para procesamiento de imágenes.

Tabla 14. Selección de software para procesamiento de imágenes [41].

Software	Características
<b>Matlab</b>	Se trata de un software de programación y entorno de desarrollo empleado en aplicaciones de inteligencia y visión artificial (IA y VA). Ofrece una variedad extensa de herramientas y funciones especializadas destinadas al análisis, procesamiento y representación visual de datos, consolidándose como una herramienta potente para la creación de algoritmos de VA.
<b>Python</b>	Es considerado uno de los lenguajes más conocidos y utilizados en el campo del aprendizaje automático. Incluye una variedad de bibliotecas y marcos que simplifican la implementación de algoritmos de procesamiento de datos y aprendizaje automático, incluidos TensorFlow, Keras, PyTorch y scikit-learn.
<b>Keras</b>	Esta interfaz de programación está diseñada para aplicaciones (API) de código abierto desarrollada en Python. A menudo se utiliza como una interfaz de alto nivel para crear y entrenar modelos de redes neuronales. Su desarrollo tuvo como objetivo facilitar la experimentación y crear rápidamente modelos de aprendizaje profundo.

#### 3.7.2 Comparativa de cada Software

Para la comparativa de software es necesario un proceso para evaluar y seleccionar la mejor opción para satisfacer las necesidades y requisitos del proyecto. Este proceso implica analizar diversas características y funcionalidades de cada software, además de su facilidad de uso, rendimiento, escalabilidad, costo, soporte técnico y compatibilidad con otras tecnologías.

En la Tabla 15, se muestra las características técnicas de cada software que estará enfocada a la Visión Artificial.

Tabla 15. Comparación de las características técnicas de cada software que estará enfocada a la Visión Artificial [42].

<b>CARACTERÍSTICAS TÉCNICAS</b>	<b>MATLAB</b>	<b>PYTHON</b>	<b>KERAS</b>
<b>Redes Neuronales</b>	Ofrece herramientas para diseñar y entrenar redes neuronales, pero menos versátil que Python.	Ampliamente utilizado en Python, especialmente con TensorFlow y PyTorch, para desarrollar y entrenar redes neuronales.	Interfaz de alto nivel construida sobre TensorFlow, facilitando el desarrollo de redes neuronales.
<b>Procesamiento de Imágenes y Videos</b>	Ofrece el Image Processing Toolbox para procesamiento de imágenes.	OpenCV proporciona una amplia gama de funciones para procesamiento de imágenes y visión por computadora.	Se integra con OpenCV y TensorFlow para procesamiento de imágenes y visión por computadora.
<b>Procesamiento de Texto</b>	MATLAB tiene funciones para procesamiento de texto, pero no es su enfoque principal.	Amplio soporte para procesamiento de texto en Python, con bibliotecas como NLTK y spaCy.	Enfocado principalmente en visión por computadora, pero se puede combinar con herramientas de procesamiento de texto en Python.
<b>Procesamiento de Datos</b>	Herramientas para manipulación de datos, pero menos versátil que las bibliotecas de Python.	Python es líder en procesamiento de datos, con bibliotecas como NumPy y Pandas.	Integrado con Python y TensorFlow para aprovechar las capacidades de procesamiento de datos.
<b>Aprendizaje Automático</b>	Posee herramientas para aprendizaje automático, pero menos enfocado que Python.	Ampliamente utilizado en aprendizaje automático, con bibliotecas como scikit-learn y TensorFlow.	Interfaz de alto nivel que se basa en TensorFlow para aprendizaje profundo.
<b>Visión por Computadora</b>	Herramientas específicas para visión por computadora, incluyendo cajas de herramientas y funciones.	OpenCV es una biblioteca líder para visión por computadora en Python.	Keras se integra con OpenCV y TensorFlow, permitiendo visión por computadora a través de modelos de aprendizaje profundo.
<b>Comunidad y Soporte</b>	Amplia comunidad, especialmente en entornos académicos e industriales. Soporte técnico de MathWorks.	Enorme comunidad y abundancia de recursos en Python. Soporte a través de foros, documentación.	Ampliamente adoptado y respaldado por la comunidad de aprendizaje profundo

En el campo de la visión artificial, MATLAB, Python y Keras desempeñan papeles importantes, cada uno con sus propias ventajas y enfoques.

MATLAB proporciona una amplia gama de herramientas especializadas y útiles cajas de herramientas especializadas, que incluyen: Una caja de herramientas para realizar tareas específicas de procesamiento de imágenes.

Por el contrario, Python se ha convertido en el lenguaje dominante en la comunidad de visión por computadora, aprovechando bibliotecas como OpenCV y TensorFlow para implementar soluciones sólidas y versátiles. Construido sobre TensorFlow,

Keras proporciona una interfaz de alto nivel que simplifica el diseño y el entrenamiento de modelos de aprendizaje profundo, incluidos los modelos diseñados para visión por computadora.

La opción más adecuada es Keras como lenguaje de programación que permite a los usuarios aprovechar el poder y la flexibilidad de TensorFlow junto con una interfaz intuitiva que facilita el desarrollo eficiente de modelos avanzados para aplicaciones de visión artificial.

### **3.7.3 Algoritmo de la red neuronal**

Los algoritmos de una red neuronal son fundamentales en el campo del aprendizaje automático y la inteligencia artificial. Estos algoritmos son responsables de entrenar la red neuronal para que pueda aprender y reconocer patrones complejos en los datos de entrada. Desde algoritmos clásicos como el Perceptrón y el Algoritmo de Retropropagación hasta enfoques más modernos como las Redes Neuronales Convolucionales (CNN) y las Redes Neuronales Recurrentes (RNN), cada uno tiene sus propias características y aplicaciones específicas.

En la Tabla 16, se especifica las características para el desarrollo de la red neuronal, se han utilizado específicamente cuatro algoritmos de visión artificial.

Tabla 16. Algoritmos de la red neuronal

Algoritmos	Características
<b>Red Neuronal Convolutiva (CNN)</b>	Este algoritmo es particularmente eficaz en aplicaciones de visión por computadora. Las capas convolucionales se pueden utilizar para obtener información sobre los patrones y características presentes en una imagen, lo que resulta beneficioso para la detección de objetos y el reconocimiento de patrones en imágenes relacionadas con vehículos.
<b>Red Neuronal Recurrente (RNN)</b>	Las redes neuronales recurrentes (RNN) son adecuadas para procesar secuencias de datos y son herramientas útiles en situaciones donde se deben considerar relaciones temporales. En el campo de la seguridad de los vehículos, podría utilizarse para analizar la secuencia de eventos al monitorear comportamientos sospechosos.
<b>Autoencoder</b>	Un codificador automático es un tipo de red neuronal diseñada para reconstruir datos de entrada reduciendo su dimensionalidad. Su aplicación se enfoca en detectar patrones anómalos dentro de la información capturada por el sistema de cámaras, que pueden indicar un posible intento de robo.
<b>Redes Generativas Adversarias (GAN)</b>	Las GAN son conocidas por generar datos realistas. En el contexto de la seguridad vehicular, podrían utilizarse para generar imágenes realistas de vehículos y escenarios, lo que podría ser útil en sistemas de entrenamiento y simulación para mejorar la detección de robos.

En el ámbito del desarrollo de Machine Learning (ML), varias tecnologías y bibliotecas juegan un papel crucial en la construcción y despliegue de modelos de IA efectivos. En este proyecto, se emplearon OpenCV, cada una con capacidades únicas que se adaptan a diferentes aspectos del ciclo de desarrollo de algoritmos de ML, desde el entrenamiento de modelos hasta la gestión de datos y la optimización del procesamiento de imágenes y videos. A continuación, describo sus aspectos más relevantes de las librerías más usadas para el reconocimiento de objetos:

**CUDA:** Desarrollada por NVIDIA, CUDA es una plataforma de computación paralela que transforma la velocidad de procesamiento de datos en proyectos de IA. Su capacidad para aprovechar la potencia de las unidades de procesamiento gráfico (GPU) es esencial para acelerar tanto el entrenamiento como la inferencia de modelos de IA.

**Ultralytics:** Es una biblioteca de código abierto que contribuye al aprendizaje automático y al desarrollo de IA. Ultralytics ofrece modelos de IA preentrenados, como YOLOv5, que son altamente valiosos en tareas de gran relevancia, como la detección de objetos, segmentación y clasificación de imágenes.

OpenCV: Es una biblioteca de visión por computadora de código abierto que se destaca por su versatilidad en proyectos de IA relacionados con el procesamiento de imágenes y videos. Ofrece un conjunto completo de herramientas que van desde la detección y seguimiento de objetos hasta la segmentación de imágenes y la calibración de cámaras, lo que contribuye a una amplia variedad de aplicaciones de IA.

Roboflow: Es una plataforma integral de gestión de datos de IA. Su función principal es etiquetar y organizar conjuntos de datos de imágenes y videos, simplificando el proceso de preparación de datos. Además, proporciona herramientas para el entrenamiento y la inferencia de modelos de IA, así como facilita su implementación en dispositivos locales.

Estas tecnologías y bibliotecas combinadas proporcionan un conjunto robusto de herramientas para el desarrollo de modelos de ML e IA, desde la adquisición y preparación de datos hasta el entrenamiento y despliegue de modelos, cubriendo una amplia gama de aplicaciones en el campo del aprendizaje automático y la visión por computadora.

En la Figura 30, se presenta el diagrama de entrenamiento de YOLOV5.

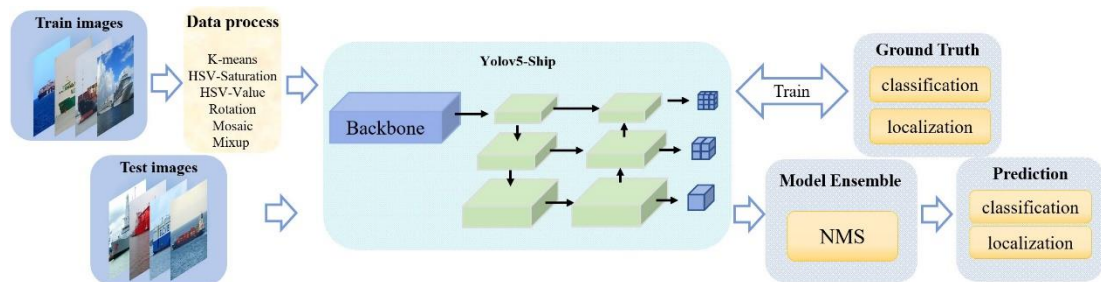


Figura 30. Periféricos de entrada y salida

El conjunto de datos, comúnmente conocido como dataset, es esencial para cada algoritmo de entrenamiento. En esta fase, se llevó a cabo la recopilación del conjunto de datos necesario para la investigación, identificando fuentes abiertas de imágenes relacionadas con el tema de estudio, que incluyen armas de fuego (pistolas y revólveres), armas blancas (cuchillos y machetes), y ejemplos de situaciones no violentas (personas en escenarios cotidianos sin armas) para casos negativos de detección, enriqueciendo así la calidad y diversidad del conjunto de datos. En la Figura

30 se muestran ejemplos de imágenes correspondientes a las dos clases de los datasets recopilados. Una vez obtenidas las imágenes, se procedió al procesamiento, el cual incluyó la corrección de posibles imperfecciones, normalización de tamaños y formatos, así como la eliminación de datos irrelevantes o ruidosos, con el fin de homogeneizar los datos. Seguidamente, se llevó a cabo la etiquetación de cada imagen en el conjunto de datos, indicando si contenía armas o personas, con el objetivo de mantener una cierta calidad en los datos de entrenamiento, Figura 31.

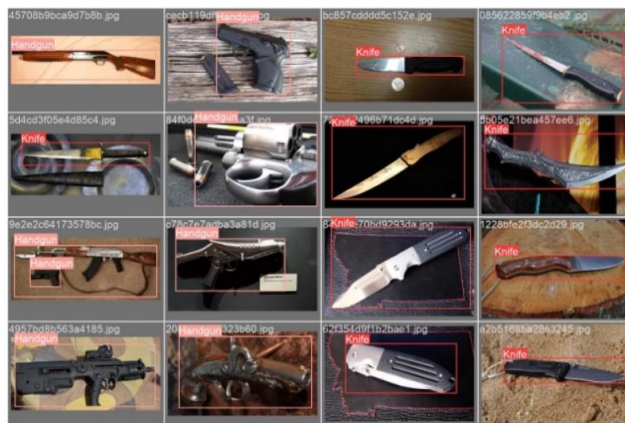


Figura 31. Periféricos de entrada y salida

### 3.7.4 Selección del servidor

Antes de elegir e implementar los servidores encargados de recoger los datos del responsable del tratamiento, se realizó un estudio detallado de las plataformas más populares para la creación de servidores web en la nube. A continuación, se proporciona un resumen de las características principales de cada una de estas plataformas.

#### *c. Cloud Computing*

La tecnología de computación en la nube ha cambiado la forma en que se almacenan, procesan y acceden a los datos y la información. En lugar de depender de recursos locales como servidores y discos duros, la computación en la nube se basa en una red de servidores remotos y potentes distribuidos en diferentes ubicaciones geográficas [43].

#### *d. Microsoft Azure*

Microsoft Azure es una plataforma de computación en la nube que proporciona una variedad de servicios y herramientas para desarrollar, implementar y administrar aplicaciones de visión por computadora (VA). Esto incluye servicios de infraestructura, almacenamiento, bases de datos y análisis de datos, así como servicios VA especializados como Azure Machine Learning y Cognitive Services. Estos servicios permiten a los usuarios crear, entrenar e implementar modelos de aprendizaje automático e implementar aplicaciones inteligentes que aprovechan tecnologías como el reconocimiento de voz, la visión por computadora y el procesamiento del lenguaje natural [44].

#### *e. Amazon EC2 (AWS)*

AWS se posiciona como una plataforma líder de servicios en la nube y ofrece una variedad de soluciones para aplicaciones de visión por computadora (VA). Entre sus servicios destacan opciones como cómputo escalable, almacenamiento, base de datos y análisis de datos, así como servicios específicos para VAs como Amazon SageMaker y Amazon Rekognition. Estas herramientas permiten a los usuarios desarrollar, entrenar e implementar modelos VA mientras aprovechan de manera eficiente la infraestructura escalable de AWS. Además, AWS proporciona servicios avanzados de automatización, orquestación y procesamiento de datos que permiten un análisis eficaz de grandes cantidades de información, potencian el VA y facilitan la toma de decisiones informadas [45].

### **3.7.5 Comparativa de plataformas del Servidor**

En la Tabla 17, se menciona a detalle cada una de las características principales de cada plataforma de Cloud Computing.



Tabla 17. Comparativa de las características principales de cada plataforma de Cloud Computing.

<b>Parámetros</b>	<b>Microsoft Azure</b>	<b>Amazon Web Services (AWS)</b>
<b>Almacenamiento</b>	Azure Blob Storage Azure Disk Storage Azure Files	Amazon S3 (Simple Storage Service) Amazon EBS (Elastic Block Store) Amazon Glacier
<b>Máquinas Virtuales</b>	Azure ofrece Azure Virtual Machines.	AWS proporciona Elastic Compute Cloud (EC2)
<b>Bases de datos</b>	Azure SQL Database Azure Cosmos DB (base de datos NoSQL) Azure Synapse Analytics (data warehouse) Azure Database for MySQL/PostgreSQL	Amazon RDS (Relational Database Service) Amazon DynamoDB (base de datos NoSQL) Amazon Redshift (data warehouse) Amazon Aurora (base de datos relacional).
<b>Sistemas operativos</b>	Windows Server Linux macOS FreeBSD.	Windows Server Linux macOS FreeBSD

Luego de una evaluación exhaustiva de las opciones disponibles, se decidió elegir Microsoft Azure como plataforma de servidor para este proyecto. Azure está en funcionamiento desde 2010 y cuenta con una gama más amplia de nubes en comparación con Google Cloud Platform, que se lanzó en 2011 y ha ganado importantes clientes como Airbus, Coca-Cola, HTC y Spotify, entre otros.

Los costos varían según el uso de la plataforma. Sin embargo, Microsoft Azure tiene beneficios adicionales: cuando se registra con la dirección de correo electrónico de una institución, puede obtener servicios gratuitos como facilitar el aprendizaje, investigación, diseño, desarrollo, pruebas y demostraciones de aplicaciones sin fines de lucro, más \$100 por año. Para esto es particularmente relevante para la implementación del proyecto. Por lo tanto, se eligió la plataforma en la nube Microsoft Azure porque satisface las necesidades específicas del proyecto en términos de flexibilidad y versatilidad.

### 3.7.6 Selección del protocolo de comunicación

Se realizó un análisis para seleccionar el protocolo de comunicación adecuado en el campo de la computación en la nube y el Internet de las cosas (IoT). MQTT es un protocolo que permite el intercambio de datos en tiempo real entre múltiples clientes a través de un punto central, lo que permite una transferencia de datos eficiente. Por el contrario, CoAP permite la transferencia uno a uno de información de estado entre clientes y servidores. En comparación con MQTT, HTTP genera una mayor sobrecarga de establecimiento de conexión. Esto se debe a que MQTT permite enviar y recibir múltiples mensajes en una sola sesión TCP, lo cual es una optimización del protocolo.

En la Tabla 18 se menciona los parámetros técnicos de cada protocolo de comunicación.

Tabla 18. Parámetros Técnicos de cada protocolo de comunicación

Parámetros	HTTP	MQTT	CoAP
Arquitectura	Cliente/Servidor	Cliente/Broker	Cliente/Sevidor o Cliente/Broker
Puerto predeterminado	80/443 (TSL/SSI)	18883/8883 (TSL/SSL)	5683 (UDP Port)/5684 (DLTS)
Aplicaciones	Web	Automatizaciones a nivel del hogar y empresarial	Redes inteligentes, automatización de edificios y casas inteligentes.
Calidad de servicio (QoS)/Confiabilidad	Limitado (a través del protocolo de transporte - TCP)	QoS 0: como máximo una vez (dispara y olvida) QoS 1: al menos una vez. QoS 2: exactamente una vez	Mensaje no confiable (similar a por lo menos una vez). Mensaje confiable (similar a como máximo una vez).
Consumo de energía	Requiere una mayor potencia/energía consumida, fue mucho mayor que con MQTT.	MQTT es más eficiente	Es más eficiente en términos de energía
Conectividad	Uno-a-uno	Uno-a-uno, uno a muchos y muchos-a-muchos	Uno a uno y muchos a muchos.
Consumo de banda ancha	Implica un mayor ancho de banda	Consume más banda ancha	Implica el ancho de banda más bajo

### 3.7.7 Selección de Data Base

Se tomó la decisión de implementar un sistema que clasifique y organice los datos en función de las matrículas de los automóviles y asigne un identificador único a cada vehículo. Aunque PostgreSQL se considera el sistema de gestión de bases de datos más sólido del mercado, no es tan popular como MySQL y tiene menos recursos de aprendizaje, como tutoriales y cursos, en comparación con MySQL.

En la Tabla 19 se menciona los parámetros técnicos de cada Data Base.

Tabla 19. Parámetros Técnicos de cada Data Base

Parámetros	Oracle	PostgreSQL	MySQL
<b>Licencia</b>	Propietaria con opciones comerciales	PostgreSQL License (Open Source)	GNU General Public License (GPL)
<b>Descripción</b>	Sistema RDBMS ampliamente utilizado en entornos empresariales. Destaca por rendimiento, escalabilidad y funciones avanzadas.	RDBMS robusto, conocido por manejar grandes cantidades de datos, con funciones avanzadas y extensibilidad.	RDBMS ampliamente utilizado, rápido y eficiente, adecuado para aplicaciones web y sistemas de información.
<b>Compatibilidad con Node-RED</b>	Conectividad con Node-RED	Nodo específico para Node-RED	Soporte para Node-RED
<b>Lenguaje de Programación</b>	SQL, PL/SQL, Java, Python, entre otros	SQL, PL/pgSQL, PL/Tcl, PL/Perl, PL/Python, entre otros.	SQL, PHP, Java, Python, entre otros.
<b>Arquitectura de Implementación</b>	Cliente-servidor	Cliente-servidor orientada a objetos	Cliente-servidor
<b>S.O Compatible</b>	Múltiples, incluyendo Windows y Linux	Múltiples, incluyendo Windows y Linux	Múltiples, incluyendo Windows y Linux
<b>Parámetros</b>	Oracle	PostgreSQL	MySQL

En otro sentido, Oracle es uno de los repositorios más utilizados en el entorno empresarial, aunque los costes, incluidos los precios de las licencias, son muy elevados. En comparación, MySQL es un sistema de base de datos extremadamente popular y utilizado en todo el mundo. Tiene mucha información y tutoriales, lo que lo hace adecuado para proyectos pequeños y medianos.

Se eligió MySQL por su facilidad para recuperar datos de dispositivos electrónicos. Además, se integra directamente con Node-RED, facilitando la visualización dinámica de los parámetros del sistema de monitoreo.

### **3.7.8 Selección de la herramienta de visualización**

Se eligió Node-RED como herramienta de visualización de los datos recopilados por el sistema de registro automático. Primero, Node-RED se puede integrar de manera óptima con la base de datos MySQL elegida para almacenar información del vehículo. Esta compatibilidad simplifica la conexión entre ambos sistemas y garantiza una gestión eficaz de los datos.

Además, Node-RED es una plataforma de programación visual completamente gratuita. Esto significa que su uso no requiere costes adicionales y está al alcance de cualquier persona interesada. Además, es intuitivo y fácil de usar, lo que lo convierte en la manera perfecta de transmitir el flujo de información desde dispositivos electrónicos. Los datos se pueden mostrar dinámicamente en tiempo real, lo que facilita la interpretación de los resultados.

En resumen, en este trabajo de investigación se implementó Node-RED porque es una buena opción para visualizar los datos recopilados por los sistemas de registro.

## **3.8 ETAPA 4.- Visualización en la Interfaz Grafica**

### **3.8.1 Instanciación en la Nube**

Se empleó Azure Virtual Machines, un servicio que simplifica la virtualización eliminando el hardware físico, se implementó una máquina virtual que aloja los servicios Node-RED, Mosquitto y MariaDB utilizando un servidor Ubuntu.

En la Figura 30 se muestra los datos de la suscripción a Azure Students.

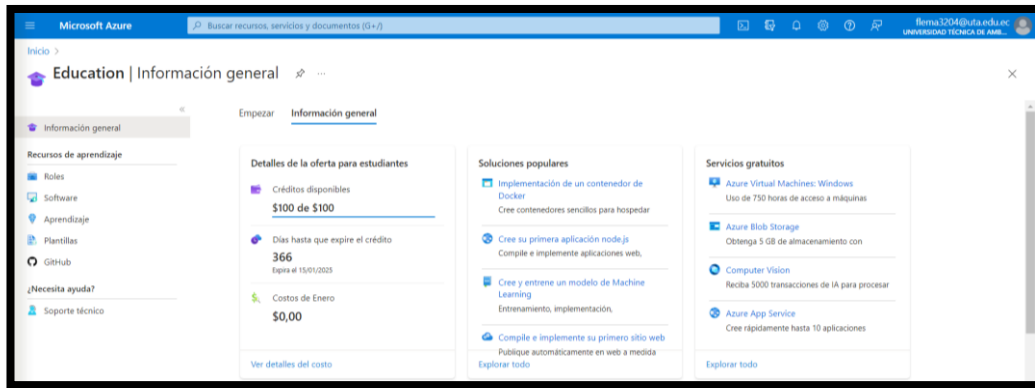


Figura 32. Suscripción a Azure Students

Para acceder a los servicios de Azure, se necesitó crear un conjunto de recursos, como se muestra en la Figura 31.

Figura 33. Creación de la Máquina Virtual en Azure

En la Figura 32 se denomina el grupo de recursos y pretender una región, se procedió a validar la exactitud de la explicación ingresada.

Validación superada	
<b>Datos básicos</b>	
Suscripción	Azure subscription 1
Grupo de recursos	(nuevo) ConfeccionesGALAR_group
Nombre de máquina virtual	ConfeccionesGALAR
Región	West Central US
Opciones de disponibilidad	No se requiere redundancia de la infraestructura
Tipo de seguridad	Máquinas virtuales de inicio seguro
Habilitar arranque seguro	Sí
Habilitar vTPM	Sí
Supervisión de integridad	No
Imagen	Ubuntu Server 20.04 LTS - Gen2
Arquitectura de VM	x64
Tamaño	Standard B2als v2 (2 vcpu, 4 GiB de memoria)
Tipo de autenticación	Contraseña
Nombre de usuario	ConfeccionesGALAR
Puertos de entrada públicos	SSH
Azure de acceso puntual	No

Figura 34. Validación de los datos para la máquina virtual

Después de finalizar la validación de los datos para la máquina virtual, los servicios de Azure están listos para su utilización.

### 3.8.2 Activación del servidor de Cloud de Microsoft Azure

El servidor está alojado en Microsoft Azure como una máquina servidor Ubuntu virtual con capacidades Node-RED y Mosquitto Broker. Node-RED facilita la comunicación eficiente con el hardware y se encarga de recopilar métricas que se envían desde la placa Raspberry Pi a través de una conexión WiFi al Mosquitto Broker en el mismo servidor. La información recopilada se almacena en la base de datos MariaDB.

Estos datos se mostrarán en el nodo Node-RED del panel. El sistema operativo elegido es Ubuntu Server 20.04 LTS - Gen2, el cual es adecuado para el desarrollo de este proyecto.

En la Figura 33 se muestra la creación de la máquina virtual Ubuntu Server.

**Crear una máquina virtual** ...

**⚠** Al cambiar opciones básicas se pueden restablecer las selecciones realizadas. Revise todas las opciones antes de crear la máquina virtual.

Imagen \* ⓘ    
[Ver todas las imágenes](#) | [Configurar la generación de máquinas virtuales](#)

Arquitectura de VM ⓘ  Arm64   
 x64

Ejecución de Azure Spot con descuento ⓘ

**i** Se encuentra en el período de evaluación gratuita. Los costos asociados a esta máquina virtual pueden cubrirse con los créditos restantes de la suscripción. [Más información](#) ⓘ

Tamaño \* ⓘ    
[Ver todos los tamaños](#)

Habilitar hibernación (versión preliminar) ⓘ    
**i** Para habilitar la hibernación, es necesario registrar la suscripción. [Más información](#) ⓘ

Figura 35. Creación de Ubuntu Server

Para la implementación de un servidor web, se debe definir las suscripciones, grupos de recursos, zonas y regiones necesarias, así como la configuración de la memoria RAM y la cantidad de núcleos de CPU. En este proceso, teniendo en cuenta los servicios básicos que son suficientes para la implementación del proyecto, se decidió asignar 4 GB de RAM y dos núcleos de CPU.

En la Figura 34 se menciona la autenticación para el acceso al servidor web se realiza vía SSH utilizando una clave pública.

**Cuenta de administrador**

Tipo de autenticación ⓘ  Clave pública SSH   
 Contraseña

**i** Ahora, Azure genera automáticamente un par de claves SSH y le permite almacenarlo para usarlo en el futuro. Es una forma rápida, sencilla y segura de conectarse a la máquina virtual.

Nombre de usuario \* ⓘ  ✓

Origen de clave pública SSH  ▼

Nombre de par de claves \*  ✓

**Reglas de puerto de entrada**

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos \* ⓘ  Ninguno   
 Permitir los puertos seleccionados

Figura 36. Autenticación para la clave pública SSH

Después de instalar la máquina virtual, puede consultar las especificaciones del servidor web y su dirección IP pública. Esta información es esencial para evaluar el rendimiento del servidor y garantizar el acceso desde cualquier lugar con conexión a Internet. También simplifica el monitoreo y ajustes remotos de servidores, brindando importantes beneficios para el avance y sustentación del proyecto.

En la Figura 35 se muestra la creación de los recursos para la máquina virtual.

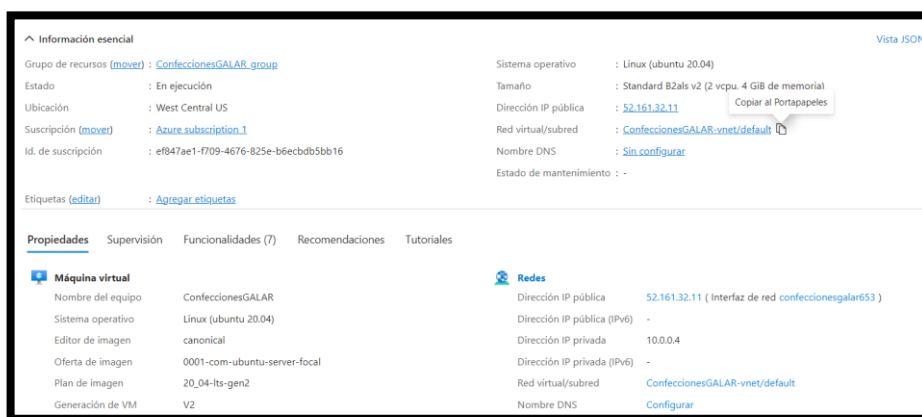


Figura 37. Recurso creado

Una vez que se tiene la clave de acceso SSH, se usó PUTTY para establecer una conexión segura al servidor virtual. Una vez conectada, se puede acceder a la máquina virtual para realizar todas las configuraciones necesarias para garantizar el correcto funcionamiento de los servicios Node-RED, Mosquitto y MariaDB. Este enfoque ayuda a mejorar la seguridad y confiabilidad del proyecto al permitir un monitoreo efectivo de los datos provenientes del sistema utilizando la cámara del dispositivo.

### 3.8.3 Acceder al tráfico de los puertos en Azure

Para utilizar los servicios Node-RED, Mosquitto, MariaDB y Apache, se deben habilitar los puertos correspondientes en la política de seguridad de la máquina virtual de Azure.

En la Figura 36 se muestra la habilitación de los puertos SSH en Azure.



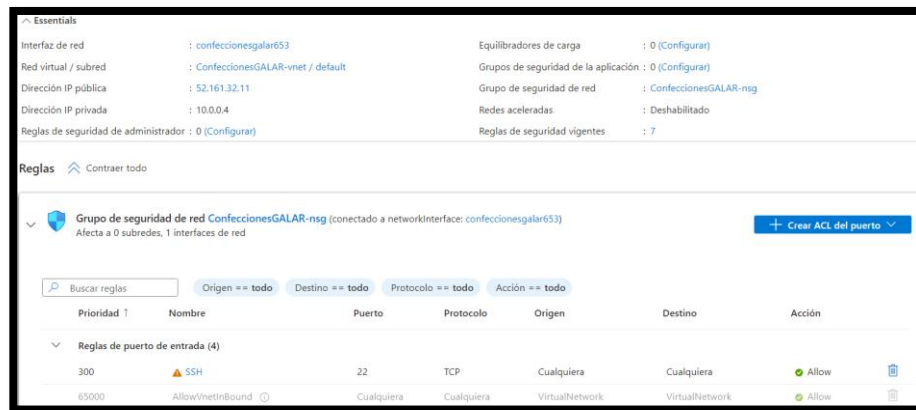


Figura 38. Habilitación del puerto SSH en Azure

### 3.8.4 Instalación del Servidor Mosquito

Mosquito Broker, propiedad de la Fundación Eclipse, es un corredor de mensajería de código abierto. Funciona mediante el protocolo MQTT (Message Queue Telemetry Transport), que permite a los dispositivos publicar y suscribirse a mensajes más fácilmente mediante temas. Mosquito es conocido por su peso ligero, lo que lo hace adecuado para dispositivos de bajo consumo y servidores de alto rendimiento.

Para instalar los paquetes de Ubuntu, se debe seguir los siguientes pasos, tomando en cuenta el uso para cada comando:

1. Se inició actualizando los paquetes de Ubuntu mediante los siguientes comandos:
  - `sudo apt-get update`
  - `sudo apt-get upgrade`
2. Posteriormente, se procedió a instalar el Broker de Mosquito desde los repositorios actualizados, utilizando el siguiente comando:
  - `sudo apt-get install mosquitto mosquitto-clients`
3. Finalmente, se empleó el siguiente comando para confirmar la correcta instalación y ejecución del servicio:
  - `service mosquitto status`

```
• mariadb.service - MariaDB 10.3.37 database server
  Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
  Active: active (running) since Sun 2022-12-11 15:17:55 UTC; 56min ago
  Docs: man:mysqld(8)
        https://mariadb.com/kb/en/library/systemd/
  Process: 708 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/
  Process: 739 ExecStartPre=/bin/sh -c systemctl unset-environment _MSREP_STA
  Process: 771 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && V
  Process: 1017 ExecStartPost=/bin/sh -c systemctl unset-environment _MSREP_S
  Process: 1019 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/
  Main PID: 871 (mysqld)
  Status: "Taking your SQL requests now..."
  Tasks: 31 (limit: 1076)
  Memory: 113.5M
  CGroup: /system.slice/mariadb.service
          └─871 /usr/sbin/mysqld
```

Figura 39. Activación del servidor de MariaDB

Por último, se estableció una contraseña y se llevó a cabo una pequeña configuración mediante el siguiente comando:

- `sudo mysql_secure_installation`

Durante este proceso, se asignó una contraseña y se realizaron ajustes breves.

### 3.8.5 Instalación de Apache Server

Apache es un servidor web considerablemente utilizado para alojar todos los archivos necesarios para mostrar un sitio web. La instalación de Apache es esencial para la administración de la base de datos MariaDB y se logra realizando los siguientes pasos:

Se instaló el software Apache utilizando el comando:

- `sudo apt-get install apache2`

Luego, se habilitó y reinició el servicio de Apache con los siguientes comandos:

- `sudo systemctl enable apache2`
- `sudo systemctl restart apache2`

Para comprobar si el servicio Apache funciona correctamente, ingrese la dirección IP del servidor (20.69.29.239) en cualquier navegador. Si la configuración del servidor es correcta, la página mostrará un mensaje como se muestra en la imagen.

En la Figura 38 se muestra la activación de la página oficial de Apache.

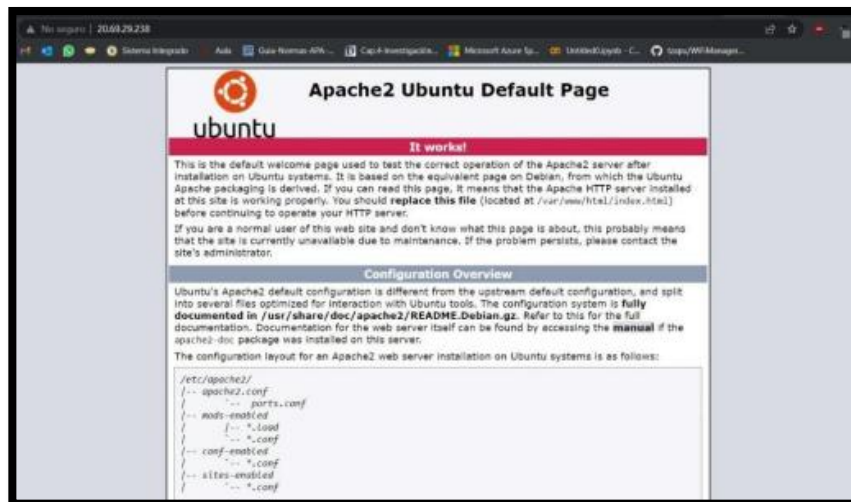


Figura 40. Pagina oficial de Apache activada

### 3.8.6 Instalación de la base de datos usando PHP MyAdmin

Este software ofrece una interfaz gráfica de usuario fácil de usar para la administración de MariaDB. La instalación se realizó utilizando el siguiente comando:

- `sudo apt install phpMyAdmin`

Para acceder a phpMyAdmin en Ubuntu Server, se requirió realizar una configuración adicional en el servidor Apache. Se editó el archivo de configuración con el siguiente comando:

- `sudo nano /etc/apache2/apache2.conf`

Luego, se abrió un archivo donde se añadió la siguiente línea de código:

- `Include /etc/phpmyadmin/apache.conf`

Las configuraciones se guardaron con la combinación de teclas "Ctrl + x", seguido de la tecla "y" para confirmar. Después de guardar el archivo, se reinició el servicio de Apache con el siguiente comando:

- `sudo service apache2 reload`

Para confirmar el correcto funcionamiento del servicio de phpMyAdmin, se accedió a la dirección correspondiente en el navegador:

- <http://20.69.29.239/phpmyadmin>

En la Figura 39 se muestra la activación de la base de datos PHP MyAdmin

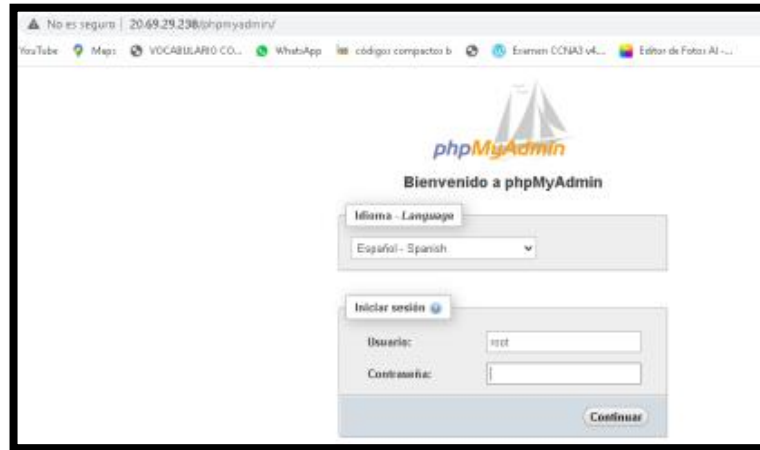


Figura 41. Activación de PHP MyAdmin

### 3.8.7 Instalación, creación y acceso a Node-RED

La instalación de Node-RED en un servidor Ubuntu se realiza usando la terminal PUTTY usando el siguiente comando:

Primero, se instala Node.js y Node-RED usando la siguiente línea de comando:

- `curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -`
- `sudo apt-get install -y nodejs build-essential`
- `sudo npm install -g --unsafe-perm node-red`

Para iniciar Node-RED automáticamente cada vez que se inicia el servidor, se usa PM2 con el siguiente comando:

- `sudo npm install -g --unsafe-perm pm2`
- `pm2 start `which node-red` -- -v`
- `pm2 save`
- `pm2 startup`

En la Figura 40 se muestra la instalación, creación y el acceso a Node-RED.

```
[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /usr/bin/node-red in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	▣	status	cpu	memory
0	node-red	fork	0	online	0%	25.9mb

Figura 42. Instalación, creación y acceso a Node-RED

### 3.9 Conexión con PHP Myadmin

Para registrar los datos adquiridos por el sistema, necesita una base de datos que pueda almacenar todos los valores de manera oportuna. Para ello se utiliza el administrador de la base de datos phpMyAdmin. El primer paso es crear una base de datos llamada "Confecciones\_GALAR". Se crearon las tablas necesarias en la base de datos para almacenar estos valores.

La tabla que se creara se almacenara los datos para el control y monitorización del vehículo en el garaje.

En la Figura 41 se muestra el Almacenamiento de la grabación por fecha, batería, entre otros, en la base de datos

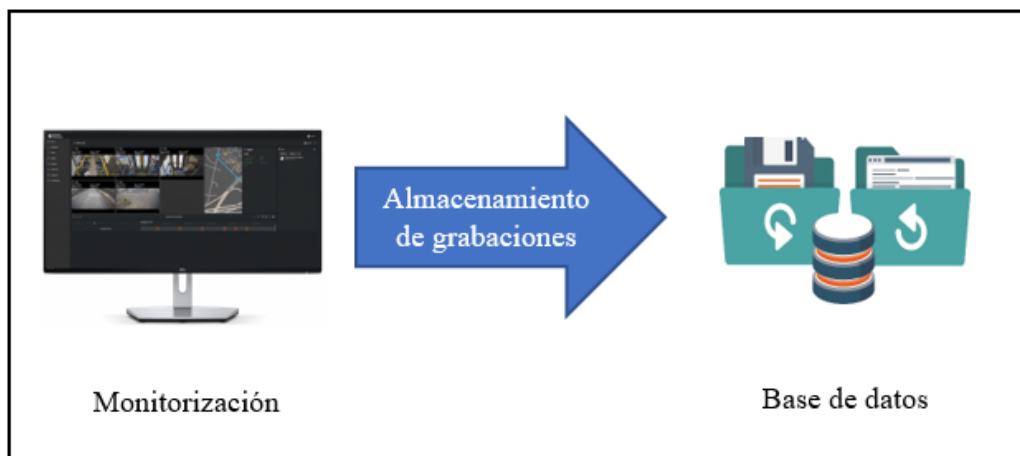


Figura 43. Almacenamiento de la grabación por, usuario, hora, fecha.

En este caso, la tabla almacena el historial y datos de la Raspberry Pi. Esta tabla registra datos relacionados con las grabaciones de vehículos obtenidas con las cámaras a través del sistema de registro automático.

En la Figura 42 se muestra el almacenamiento del historial de grabaciones mediante la Raspberry Pi 3 B+



Figura 44. Almacenamiento del historial de grabaciones mediante la Raspberry Pi 3 B+

### 3.9.1 Ajuste de Node-Red

Node-RED es un software versátil de código abierto que se puede implementar en varios sistemas operativos, como Windows, Linux y servidores en la nube. Su enfoque de programación basado en bloques proporciona una experiencia altamente interactiva y su capacidad para interactuar con hardware lo hace ideal para aplicaciones de Internet de las cosas (IoT). Su interfaz visual facilita la visualización y el control inmediato. Basado en JavaScript, Node-RED puede administrar de manera flexible varias aplicaciones y crear funciones personalizadas según las necesidades del usuario.

Para acceder a la interfaz de Node-RED, simplemente use la dirección IP de su servidor y el puerto predeterminado de Node-RED, por ejemplo: <http://20.69.29.239:1880>.

En la Figura 43 se muestra la Interfaz de Node-RED

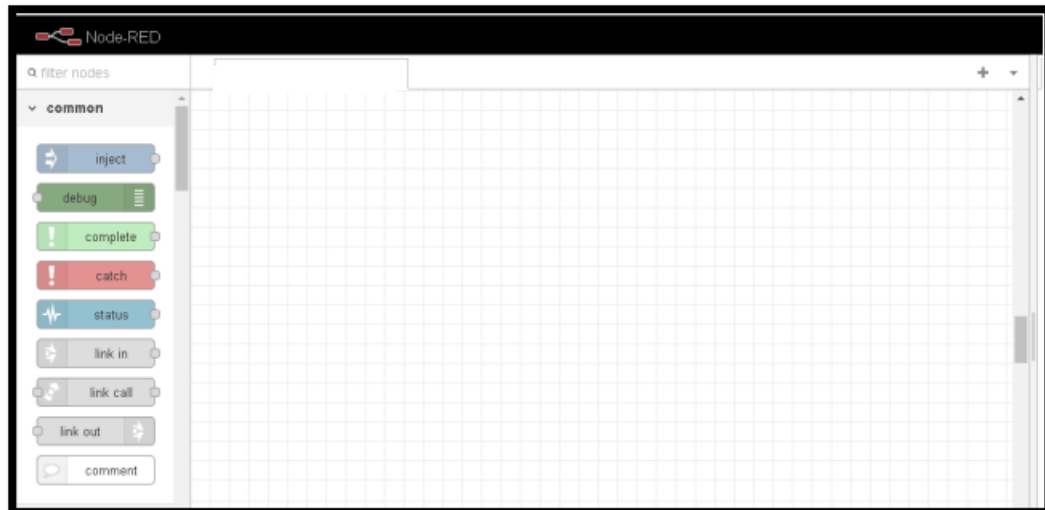


Figura 45. Interfaz de Node-RED

### 3.9.2 Configuración de Node-RED para la conexión a la base de datos

Para almacenar los datos provenientes de las grabaciones de las cámaras, se emplea un sistema de base de datos donde se ingresan los valores relevantes. Dado que se trata de un sistema de monitoreo, es esencial contar con campos como nombre, fecha, dirección, historial y modelo/año de la cámara. Para llevar a cabo esta operación, se utiliza el nodo "form" en Node-RED.

En la Figura 44 se muestra el nodo form.

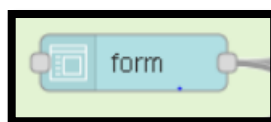


Figura 46. Nodo "form"

La Figura 44 exhibe un nodo que simplifica la introducción de los datos de los usuarios. Para almacenar estos valores en la base de datos, se utilizaron los nodos "function" y "mysql". La inserción de los datos se llevó a cabo mediante la sentencia SQL "INSERT INTO" id (nombre, fecha, dirección, historial y modelo/año de la cámara)".

En la Figura 45 se muestra la conexión entre estos nodos y se muestra un mensaje de confirmación (Conectado), que significa que se ha establecido la conexión a la base de datos MariaDB

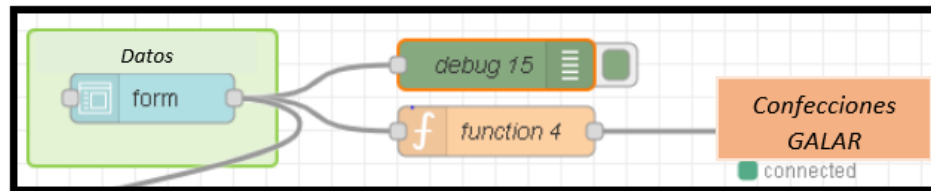


Figura 47. Enlace a la base de datos

### 3.9.3 Configuración de Node-Red para registrar la información generada por la Raspberry Pi 3 Modelo B+

Para iniciar una conexión a la interfaz, se implementa un nodo funcional y este nodo se configura con una declaración SQL que recopila todos los valores requeridos y los guarda en el repositorio.

En la Figura 46 se muestra el nodo function.



Figura 48. Nodo Function

Para insertar valores desde Node-RED a la base de datos se utiliza el nodo "mysql", lo que simplifica esta conexión tras la configuración anterior. Esta configuración incluye especificar el nombre de la base de datos y la información del usuario al que se conecta.

En la Figura 47 se muestra el nodo de conexión a Mysql.



Figura 49. Nodo Conexión a Mysql



Node-RED utiliza el nodo "MQTT in" para obtener datos que se guardarán más adelante. Este nodo te permite recuperar todos los datos de la placa Raspberry Pi 3 modelo B+. Dado que los datos del mapa están en formato JSON, es necesario convertirlos a objetos JavaScript para que funcionen. El nodo "MQTT in" puede actuar como suscriptor del Mosquitto Broker instalado en el servidor Ubuntu y suscribirse al tema "data/raspberry".

En la Figura 48 se muestra la vinculación del Nodo Mqtt y Nodo json

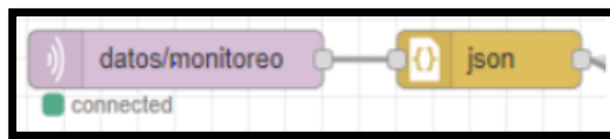


Figura 50. Vinculación del Nodo Mqtt y Nodo json

Las instancias de contenedor se crean para organizar la interfaz del programa, adaptadas a las necesidades específicas del usuario.

En la Figura 49 se muestran las tablas que están diseñadas para agrupar nodos y letras.

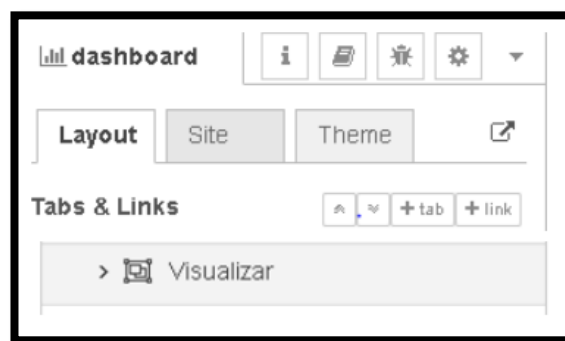


Figura 51. Creación de tablas y redes

Como se muestra en la Figura 50, se puede observar la distribución de las camaras en la tabla de la interfaz.

NO.	UID	IP	PORT
1	AAKK-004813-BFCEA	192.168.0.101	80
2	AAKK-005157-ABBDA	192.168.0.102	80

Figura 52. Tabla de la lista para las cámaras en la Interfaz

También se podrá observar en la Figura 51, El nombre de usuario, hora y fecha de ingreso a la interfaz.

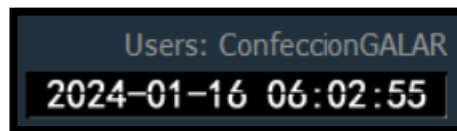


Figura 53. Visualización de usuario, hora y fecha

En la Figura 52 se muestra la tabla de conexión de las cámaras activadas en la interfaz.

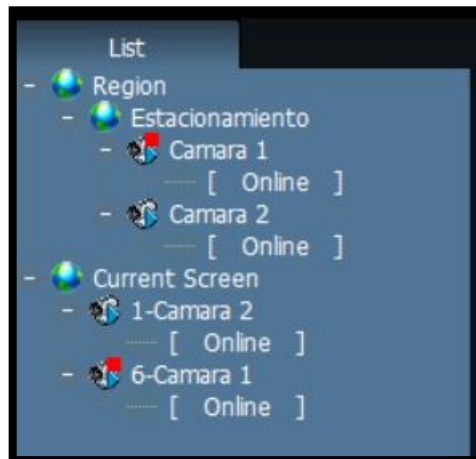


Figura 54. Tabla de la lista para las cámaras en la Interfaz

El proceso comienza encendiendo la placa Raspberry Pi 3 Modelo B+, que ejecuta un programa de autoregistro Python. En esta parte de la GUI, la conexión Wi-Fi de la placa Raspberry Pi 3 Modelo B+ está configurada para conectarse a una red que pueda acceder a Internet. Esto incluye ingresar el nombre de red y contraseña correspondiente a Confecciones GALAR. Si la conexión es exitosa, la comunicación comienza con el corredor MQTT ubicado en el servidor Ubuntu en la nube de Azure y se conecta al

corredor y al panel de Mosquitto de acuerdo con el cronograma especificado. Una conexión exitosa es esencial para transferir datos desde la placa Raspberry Pi4 al servidor. Los datos de monitorización y control del vehículo son parte del proceso de procesamiento de datos en el que la tarjeta Raspberry Pi 3 Modelo B+ procesa la información para su posterior transmisión al corredor mediante la conexión Wi-Fi establecida.

Los brókeres reciben informes que contienen datos del sector. Los datos recibidos luego se muestran en la GUI para mostrar la información de la placa como se muestra en la figura anterior. Todo el proceso se lleva a cabo mientras el dispositivo está encendido y funcionando.

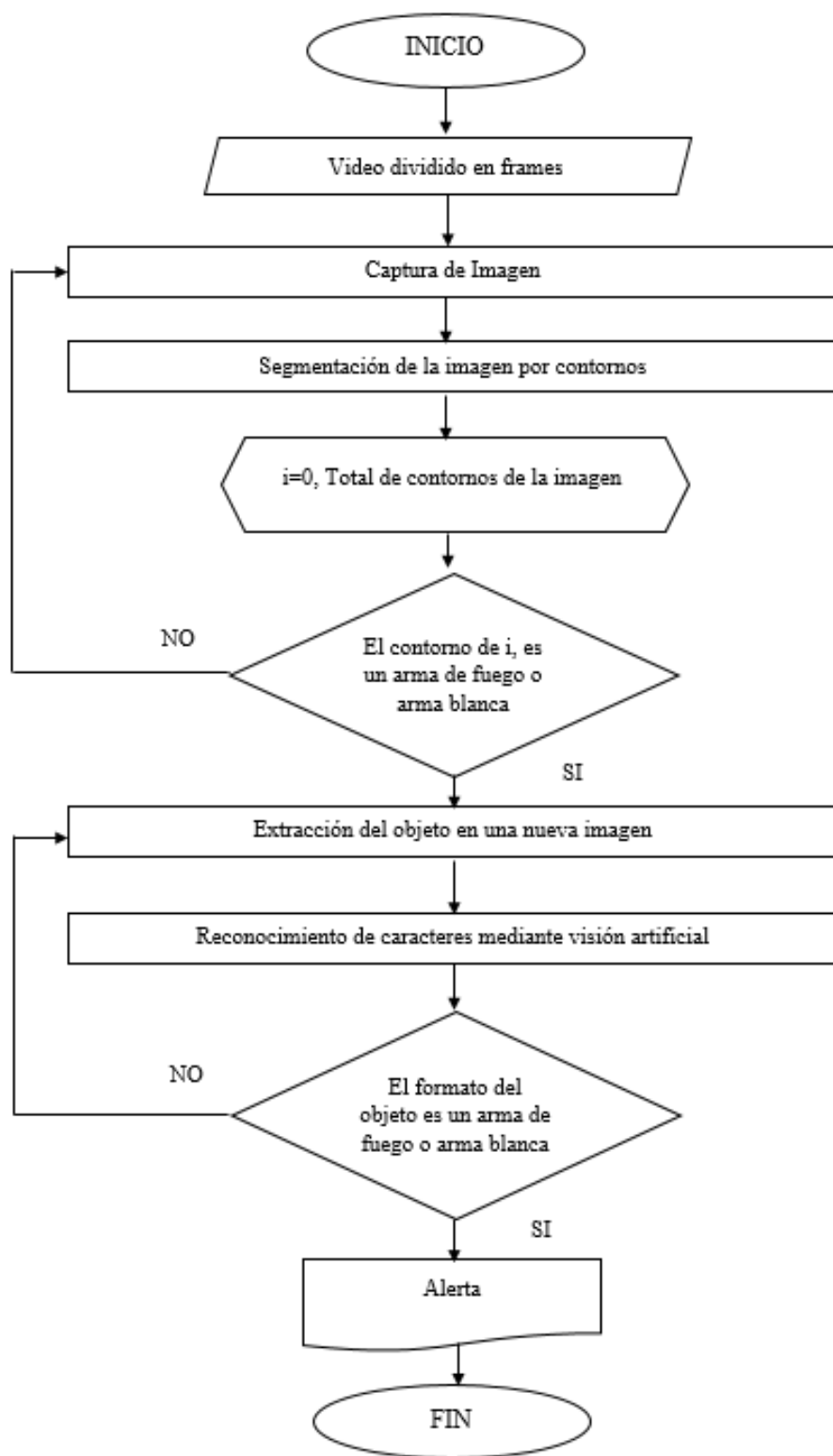


Figura 55. Diagrama de flujo del sistema de monitorización

El sistema comienza con la entrada de video de la cámara, detecta movimiento y captura las imágenes apropiadas. El programa está configurado para capturar un objeto de cierto tamaño, en este caso un vehículo, suponiendo que sea un vehículo, y procesar digitalmente la imagen. Luego, las imágenes se procesan digitalmente para ajustar la perspectiva, corregir y eliminar el ruido.

La imagen será en este caso un arma blanca. Este proceso digital identifica la posibilidad de robo al vehículo a partir de las imágenes capturadas.

### 3.10 Entrenamiento de la red

Para el desarrollo del sistema de detección, es necesario contar con una gran cantidad de imágenes con o con las que se podrá realizar el entrenamiento de la red. Las imágenes recolectadas se constituyeron en personas que portaban armas blancas o armas de fuego, con lo cual se realizó un video de simulación de robo portando dichas armas y también de personas que pasaban sin ningún tipo de arma frente al vehículo.

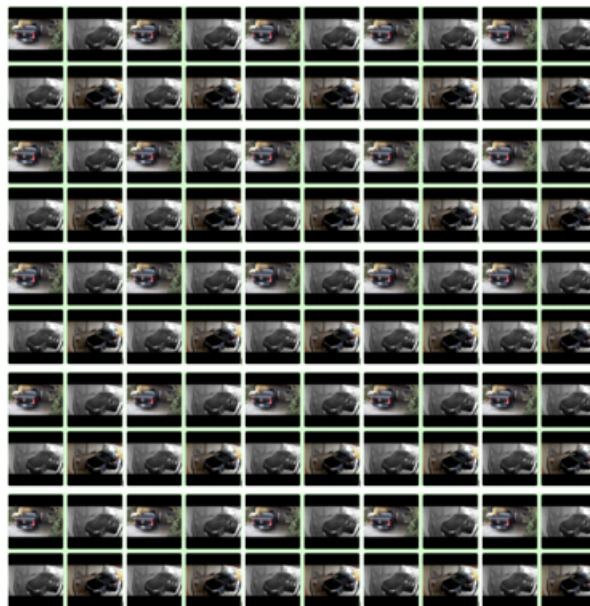


Figura 56. Imágenes recolectadas

Al video realizado se optó por convertirlo en imágenes o frames por lo cual se utilizó un código en python para la conversión del video con lo cual se obtuvo un total de 5400 imágenes o frames.

En la Tabla 20 se muestra las imágenes recolectadas y divididas para cada caso.

Tabla 20. Imágenes recolectadas

Imagenes con Armas	Imagenes sin Armas
2700 imagenes	2700 imagenes

En la Figura 55 se muestra la división de las imágenes para cada caso.

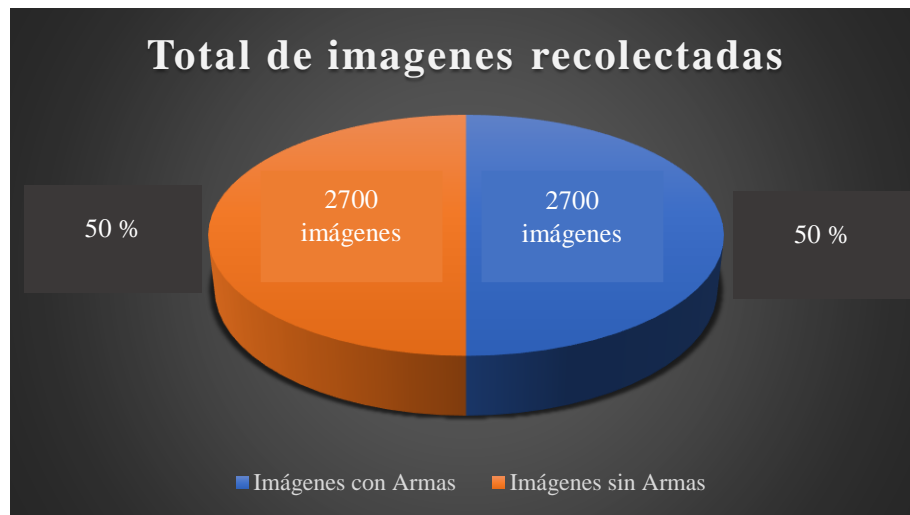


Figura 57. Imágenes recolectadas

Para el calculo del frames ocupadas para el entrenamiento de la neurona se uso un video de 3 minutos, repartidos entre el día y la noche:

La fórmula es:

Para el calculo del Número total de frames, se da a conocer en la Ecuacion 4:

$$\text{Velocidad de fotogramas} \times \text{Duración del video en segundos} \quad (4)$$

El video tiene una velocidad de fotogramas de 30 FPS, se da a conocer en la Ecuacion 5:

$$\text{Número total de frames} = 30 \text{ FPS} \times (3 \text{ minutos} \times 60 \text{ segundos/minuto}) \quad (5)$$

$$\mathbf{frames} = 30\text{FPS} \times \left(3\text{minutos} \times \frac{60\text{segundos}}{\text{minuto}}\right)$$

Número total de frames es:

$$\text{frames} = 30\text{FPS} \times 180\text{segundos} = 5400\text{frames}$$

En total se obtuvo 5400 imágenes.

Código para convertir el video en imágenes (frames)

En la Figura 56 se muestra el código de Python para convertir el video en frames.

```
import cv2

def dividir_video_a_frames(video_path, carpeta_destino, nombre_base='frame'):
    # Abre el video
    video = cv2.VideoCapture(video_path)

    # Inicializa variables
    success, frame = video.read()
    count = 0

    # Lee cada frame y lo guarda como una imagen
    while success:
        # Guarda el frame como una imagen
        nombre_frame = f'{nombre_base}_{count:04d}.png'
        ruta_frame = f'{carpeta_destino}/{nombre_frame}'
        cv2.imwrite(ruta_frame, frame)

        # Lee el siguiente frame
        success, frame = video.read()
        count += 1

    # Libera los recursos
    video.release()

if __name__ == "__main__":
    # Especifica la ruta del video y la carpeta de destino
    video_path = 'C:\Users\LENOVO\Documents\Tesis-Sistemas_de_Monitorizacion\Video-robo-a-vehiculo.mp4'
    carpeta_destino = 'C:\Users\LENOVO\Documents\Tesis-Sistemas_de_Monitorizacion\Imagenes-robo-a-vehiculo'

    # Llama a la función para dividir el video en frames
    dividir_video_a_frames(video_path, carpeta_destino)
```

Figura 58. Código de Python para convertir el video en frames

La base de datos comprende 5400 imágenes, en donde se dividió cada uno de los casos (imágenes con arma e imágenes sin arma) en 2700 imágenes y con lo cual se realiza el proceso de entrenamiento de la Convolutional Neural Network (CNN). Este enfoque se adoptó para analizar si las nuevas imágenes generadas contribuyen beneficiosamente al proceso de entrenamiento, proporcionando información complementaria a la inicial y evitando posibles casos de sobreentrenamiento del sistema. Asimismo, se evaluaron ambas arquitecturas (VGG y ResNet) considerando los resultados más favorables tanto en las fases de entrenamiento como en las evaluaciones preliminares. Posteriormente, se procedió a realizar el entrenamiento del modelo con la totalidad de la base de datos.

Los casos que se tomaron son los siguientes:

**a. Portando armas de fuego o arma blanca**

En este caso, se presenta en las imágenes de la base de datos un individuo o persona portando un arma blanca o arma de fuego, siendo estas imágenes recolectadas del video, dando a conocer que estan cometiendo el robo del automovil.

**b. Sin arma de fuego o arma blanca**

En esta base de datos se conforma por imágenes de personas transcurriendo cerca del vehiculo, dando a conocer que no esta portando un arma blanca o arma de fuego.

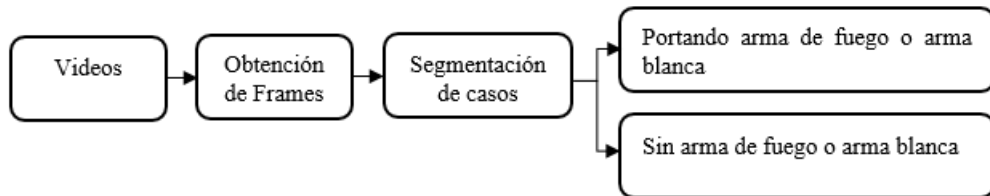


Figura 59. Diagrama de distribución para cada uno de los casos establecidos

En un robo real, la persona que esta cometiendo el hecho puede presentar el arma de blanca o arma de fuego en sus manos, apuntando hacia el vehiculo, por lo cual, esta posicion es captada por la cámara ya que se encuentra en una posicion visible del atacante.

**3.10.1 Redes Neuronales Convolucionales**

Para el entrenamiento de la red se utilizara la arquitectura de red neuronal convolucional, las cuales permitiran establecer la deteccion de objetos en imágenes, en este caso, se detectara la presencia de una persona en un acto de robo con un objeto cortopunzante o arma blanca y de un arma de fuego, por lo cual se establece las siguientes limitantes para el entrenamiento de la red.

**a. Arquitectura de Red**

Para el diseño de la red neuronal convolucional se utilizaron diferentes configuraciones en la estructura de la red. Estas configuraciones se establecieron



mediante dos tipos de arquitecturas, las cuales se han tomado porque obtienen un resultado muy favorable en la detección de imágenes.

- **Arquitectura VGG**

Esta arquitectura permite estructurar la red neuronal convolucional y tiene como idea establecer una red simple y profunda, además posee una estructura de red formada por filtros pequeños que poseen un tamaño de 3x3 en todas las capas y con lo cual se puede implementar una gran cantidad de capas y filtros.

En la Figura 58, se muestra la estructura de la arquitectura VGG, dando a conocer cada uno de los modelos de convolucion y MaxPooling empleados.

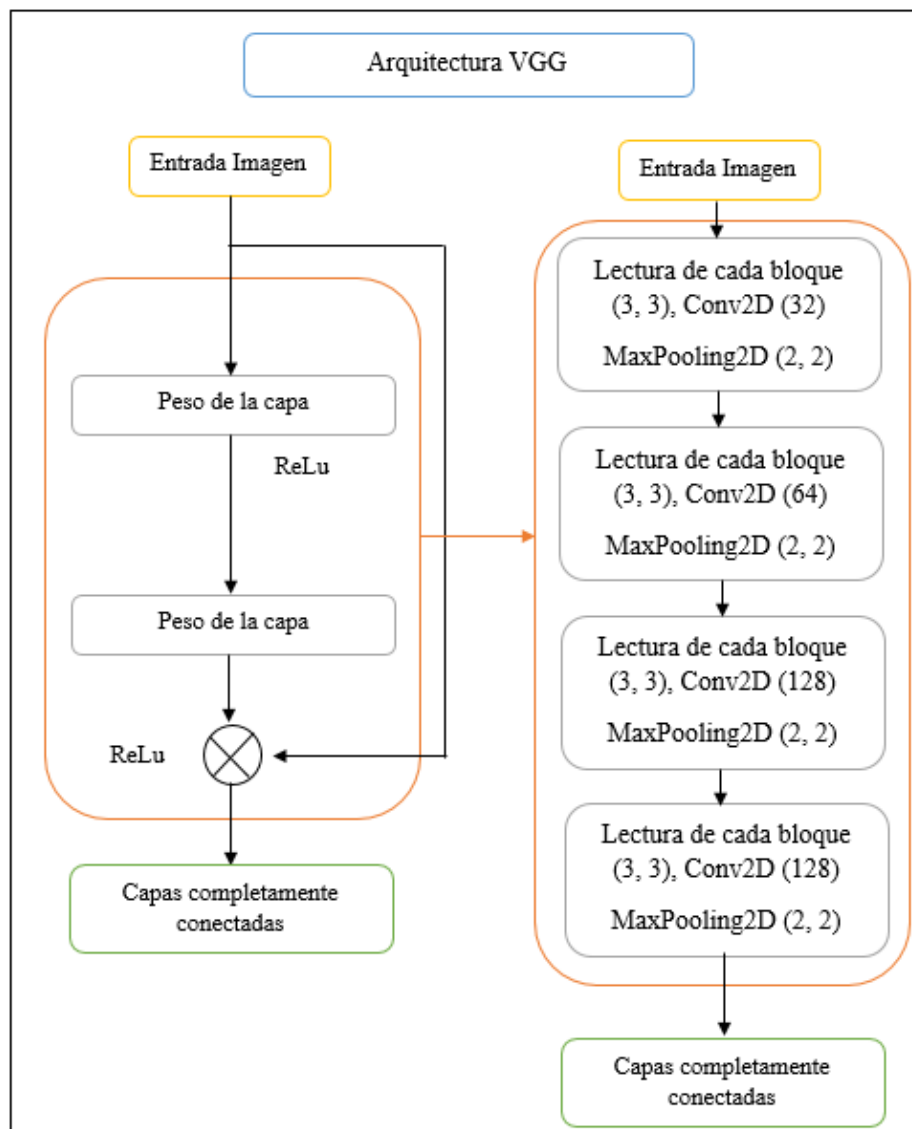


Figura 60. Arquitectura VGG

- **Capas del modelo**

Las capas del modelo comienzan desde la capa de entrada, capa oculta y las capas de salida (MaxPooling y Dense), en cada uno de los frames o imágenes detectadas por la cámara se da en la entrada y en la salida se toma las decisiones del robo.

Tabla 21. Capas del modelo de la Red Convolutiva

Capas	Tamaño de Filtro	Numero de Filtros
Convolucion 1	3x3	32
	MaxPooling 2x2	
Convolucion 2	3x3	64
	MaxPooling 2x2	
Convolucion 3	3x3	128
	MaxPooling 2x2	
Convolucion 4	3x3	128
	MaxPooling 2x2	

En la Figura 59, se muestra las capas del algoritmo

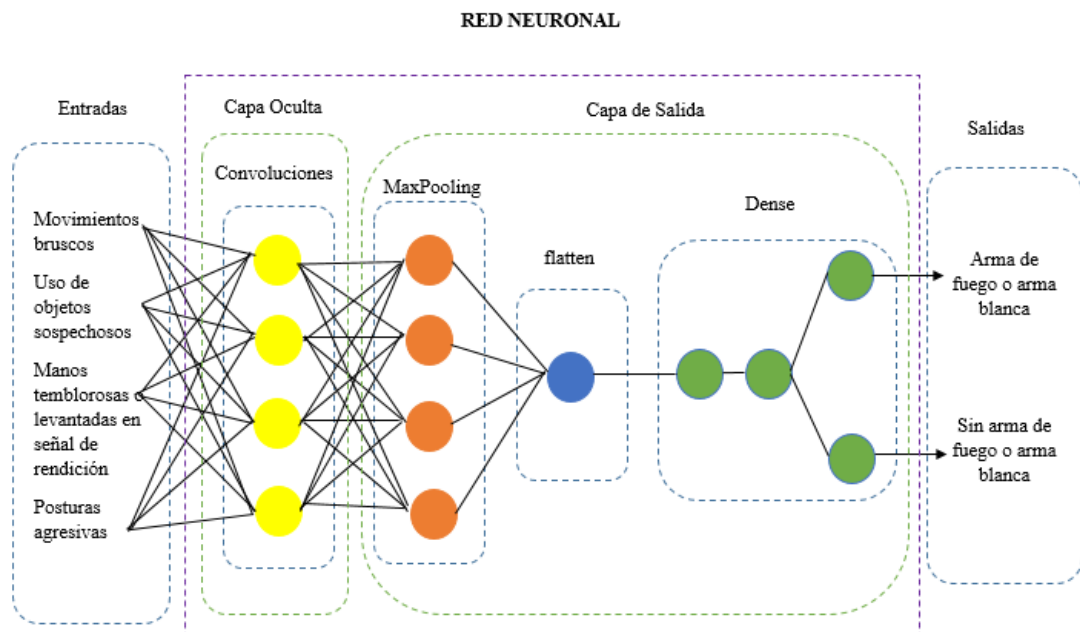


Figura 61. Capas del algoritmo

- **Capa de convolucion**

Se emplearon 4 capas de convolución (Conv2D) para analizar y extraer los patrones característicos o indicadores de robo de la imagen de entrada. Esto se logró aplicando filtros que se desplazan por cada núcleo o kernel, generando matrices de patrones y reduciendo la imagen a sus características más importantes. Luego de esta capa, se

aplicó una función de normalización por lotes para estandarizar las activaciones y el entrenamiento.

- ***Capa de Pooling***

Las capas de Pooling, en particular MaxPooling2D, se utilizaron para reducir las dimensiones espaciales de las imágenes después de la convolución, manteniendo los patrones más destacados.

- ***Capa Flatten***

Para convertir los datos de salida tridimensionales en un vector unidimensional, se emplearon las capas Flatten, simplificando así las operaciones matemáticas.

- ***Capa Dense***

Las capas densas, interconectadas, se usaron para la clasificación basada en los gestos corporales extraídos. Se implementaron funciones como ReLu para normalizar los datos de salida y mejorar el rendimiento del modelo. Finalmente, la activación softmax transformó los resultados de cada salida en probabilidades para la clasificación multicategoría, facilitando la identificación de aciertos y errores.

- ***Definición del modelo neuronal***

Las siguientes librerías o bibliotecas permite definir el modelo de red requerido.

```
from tensorflow.keras.models import load_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

Los siguientes comando permitiran establecer la extraccion de las características de las imágenes.

```
# Definición el modelo de red neuronal convolucional
model.add ([Conv2D (32, (3, 3), padding='same',
input_shape=(224, 224, 3))
model.add (MaxPooling2D((2, 2)))
```

```

model.add(BatchNormalization())
model.add(activation='relu')
model.add ([Conv2D(64, (3, 3), padding='same',
input_shape=(224, 224, 3))
model.add (MaxPooling2D((2, 2)))
model.add(BatchNormalization())
model.add(activation='relu')
model.add ([Conv2D(128, (3, 3), padding='same',
model.add (MaxPooling2D((2, 2)))
model.add(BatchNormalization())
model.add(activation='relu')
model.add ([Conv2D(128, (3, 3), padding='same',
model.add (MaxPooling2D((2, 2)))
model.add(BatchNormalization())
model.add(activation='relu')

```

A continuación, se categorizó en función de los indicadores de robo.

```

model.add(Flatten())
model.add(Dense(128))
model.add(BatchNormalization())
model.add(activation='relu')
model.add(Dense(256))
model.add(BatchNormalization())
model.add(activation='relu')
model.add(Dense(512))
model.add(BatchNormalization())
model.add(activation='relu')

```

- ***Código de entrenamiento***

El siguiente código permitirá entrenar el modelo.

```

model.fit( train_generator,

```

```

epochs=epochs,
validation_data=val_generator,
steps_per_epoch=train_generator.n//batch_size,
validation_steps=val_generator.n//batch_size,
callbacks=[tensorboard_callback]

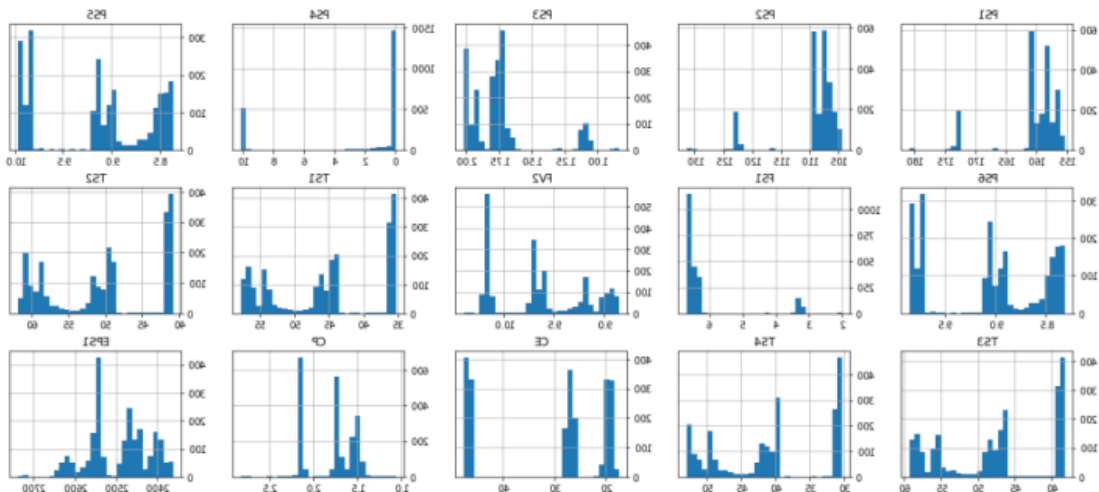
```

- ***Verificación de datos almacenados***

Los datos generados por OpenCV establecen el reconocimiento dependiendo del comportamiento de la persona o de sus indicadores de gestos corporales, sea estos casos, movimientos bruscos, posturas agresivas, objetos sospechosos y levantamiento de las manos.

- ***Distribucion normal***

Se crearon gráficos de distribución normal, revelando que ninguno sigue una distribución normal. Cada valor de la base de datos adopta parámetros que se ajustan a sus características específicas.



- ***Matriz de correlacion***

Basándose en los datos representados en la distribución normal, se procedió a crear una matriz de correlación, como se muestra en la Figura 41, que analiza las relaciones entre los conjuntos de datos de entrada. Esta matriz cubre todos los valores de entrenamiento y examina la relación lineal entre cada variable y sus valores anteriores.

fig, ax=plt.subplots(figsize=(8,6))

sns.heatmap(Data.corr(), annot=True, linewidths=5, linecolor="black")

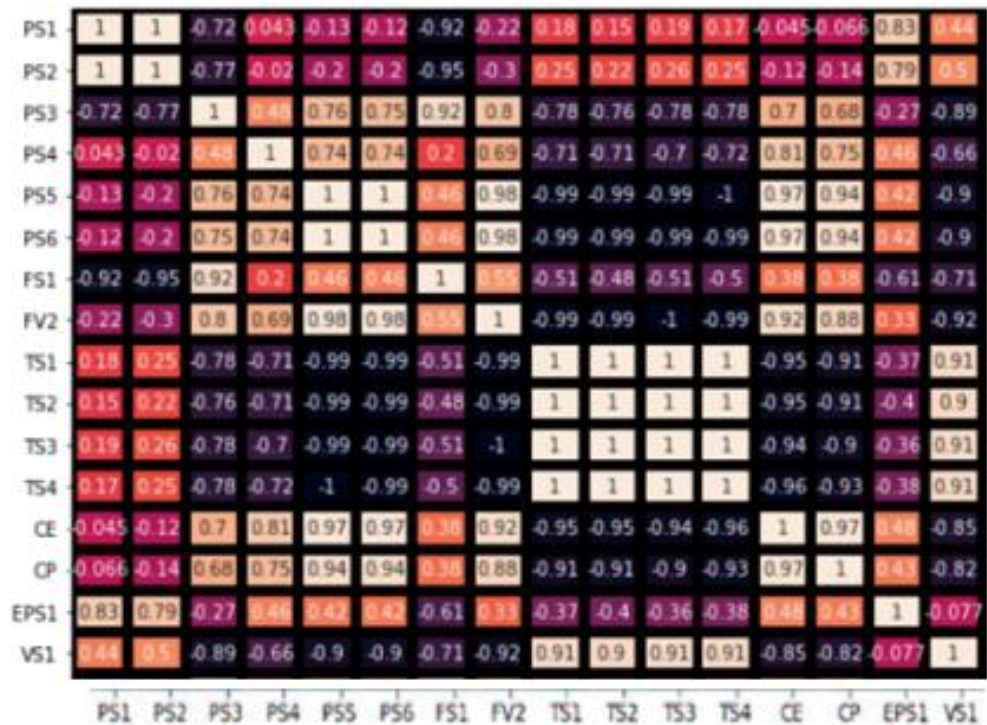


Figura 62. Matriz de correlación

- ***Precisión de las épocas***

Durante el análisis de la métrica de precisión a lo largo de varias épocas de entrenamiento, se observa un aumento gradual tanto en la precisión del conjunto de entrenamiento como en la de validación. Este fenómeno sugiere un proceso de aprendizaje efectivo en el que el modelo ha logrado capturar de manera adecuada los patrones presentes en los conjuntos de datos de entrenamiento y validación. La convergencia simultánea de ambas métricas indica un equilibrio sólido en el proceso de aprendizaje, lo que respalda la capacidad del modelo para generalizar de manera eficiente a nuevos datos.

- ***Evaluación de resultados de la arquitectura VGG a 100 épocas***

La arquitectura VGG proporcionó mejores resultados, lo cual permitió el entrenamiento del modelo con la base de datos. De las cuales se realizaron las pruebas

respectivas con las imágenes recolectadas. Así la curva corresponde a los valores obtenidos durante el entrenamiento y los valores resultantes de la evaluación.

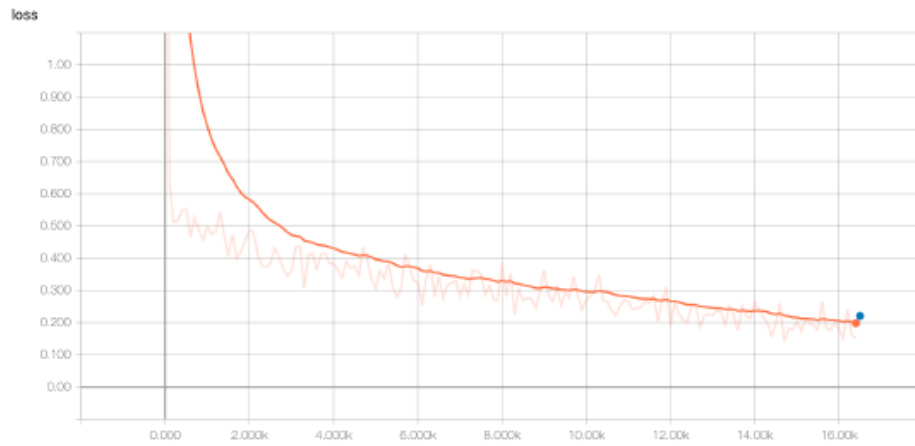


Figura 63. Curva de pérdidas del entrenamiento y evaluación de la imagen RGB

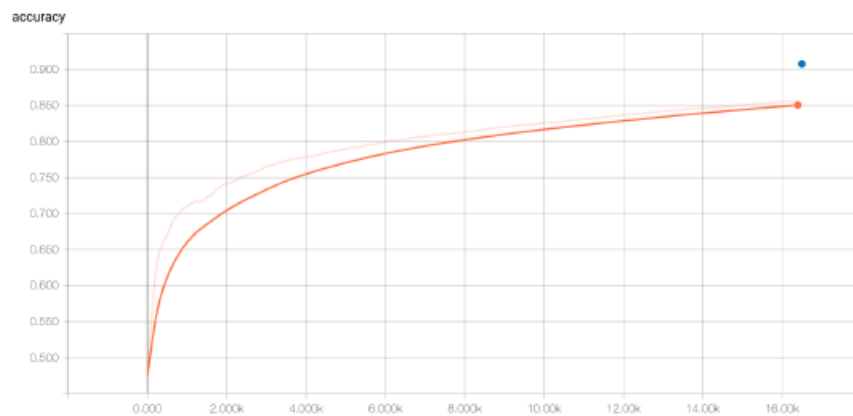


Figura 64. Curva de exactitud del entrenamiento y evaluación de la imagen RGB



Figura 65. Curva de pérdidas del entrenamiento y evaluación de la imagen en escala de grises.

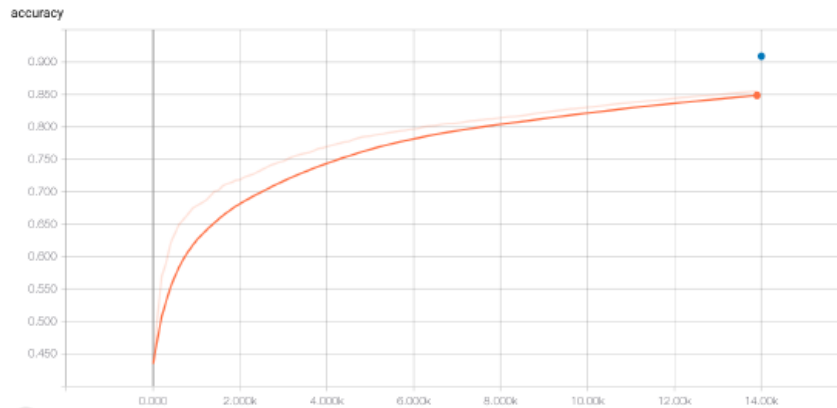


Figura 66. Curva de exactitud del entrenamiento y evaluación de la imagen en escala de grises.

En la Tabla 22, se muestra la comparación de los resultados de las pruebas de las imágenes, en escala a grises y RGB.

Tabla 22. Comparación de los resultados de las pruebas de las imágenes

Pruebas usando la Arquitectura VGG							
Configuración							
			Entrenamiento		Evaluación		
Pruebas	Imagen	Step	Loss	Accuracy	Loss	Accuracy	More
P1	RGB	2700	0.19	0.855	0.20	0.912	
P2	Gris	2700	0.17	0.857	0.19	0.924	

- **Matriz de confusión**

La matriz de confusión utilizada en la evaluación del rendimiento del algoritmo en técnicas de aprendizaje profundo se crea individualmente para cada conjunto de datos recopilados. Esta matriz muestra un conjunto de información expresado en porcentajes que reflejan todos los aciertos del modelo, considerando ciertas características y criterios de calificación. Representa las detecciones realizadas por el modelo para cada objeto durante el proceso de validación.

```
def init_cameras_and_models(self):
    # Ruta de la carpeta que contiene las imágenes para detección
    self.image_folder_path = "C:\Users\LENOVO\Documents\Tesis-Sistemas_de_Monitorizacion\Imágenes-escala-de-grises"

    # Inicia las cámaras simuladas
    # Aquí se integra la detección de objetos o reconocimiento facial
    # mediante el uso de algoritmos como YOLO
    self.captures = [cv2.VideoCapture(f"video_camara_{i}.mp4") for i in range(1, 5)]
```





Figura 67. Matriz de confusión de 100 épocas.

Se obtuvo el entrenamiento de 100 épocas, de esta manera se observa los siguientes resultados.

Tabla 23. Entrenamiento a 100 épocas.

Clases	Imágenes	Instancias	P	R	mAP50	mAP50-95
Movimientos bruscos	1350	640	0.947	0.799	0.836	0.702
Posturas agresivas	1350	177	0.934	0.635	0.702	0.468
Manos temblorosas o levantadas en señal de rendición	1350	170	0.971	0.906	0.937	0.614
Uso de objetos sospechosos	1350	293	0.936	0.857	0.867	0.459

Precisión: En el entrenamiento a 100 épocas se obtuvo el porcentaje a 0.947%, lo que supone una buena precisión.

Recall: En el entrenamiento a 100 épocas se obtuvo 0.799%.

mAP\_0.5: En el entrenamiento a 100 épocas se obtuvo un porcentaje de 0.836%,

mAP\_0.5:0.95: En el entrenamiento a 100 épocas se obtuvo 0.65%..

Estos resultados sugieren que a mayor número número de épocas al entrenamiento de la red, los valores mejoran significativamente.

### 3.10.2 Programación del algoritmo en Python

El siguiente código emplea el Keras para llevar a cabo una detección supervisada en un conjunto de características y etiquetas. Se procede a dividir el conjunto de datos en conjuntos de entrenamiento y prueba, se entrena el clasificador utilizando los datos de entrenamiento y se efectúan predicciones en los datos de prueba. Posteriormente, se calcula la precisión del clasificador mediante la comparación entre las etiquetas predichas y las etiquetas reales del conjunto de prueba.

En la Figura 66 se muestra las librerías utilizadas Keras y Tensorflow.

```
import tkinter as tk
from tkinter import tk
from PIL import Image, ImageTk
import cv2
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
import numpy as np

class VehicleSecurityApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Sistema de Control y Monitorización de Vehículos")

        # Inicializa la interfaz gráfica
        self.init_gui()

        # Inicializa las cámaras y modelos
        self.init_cameras_and_models()

        # Inicia la actualización continua de las cámaras
        self.show_cameras()

    def init_gui(self):
        # Crear frames para cada cámara
        self.camera_frames = [tk.Frame(self.root) for _ in range(4)]
        for i, camera_frame in enumerate(self.camera_frames):
            camera_frame.grid(row=0, column=i, padx=10, pady=10)

        # Se crea etiquetas para la visualización de las imágenes
        self.image_labels = [tk.Label(frame) for frame in self.camera_frames]
```

Figura 68. Librería Keras y Tensorflow

### **3.11 Funcionamiento del sistema**

El desarrollo de sistemas de detección, como es común en la mayoría de los sistemas que utilizan el aprendizaje automático como estrategia de análisis, requiere un gran repositorio para el entrenamiento de modelos. Como no existía una base de datos en línea sobre armas o personas que portan armas, se decidió crear una. Esta base de datos consta de dos categorías: la primera contiene imágenes de personas con armas y la segunda contiene imágenes de personas sin armas.

Hubo dos razones para la decisión de crear una base de datos utilizando imágenes de personas portando armas de fuego en lugar de imágenes únicamente de armas blancas. En primer lugar, intenta proporcionar a la red imágenes durante el entrenamiento que sean similares a las que encontraría en operaciones del mundo real donde las armas aparecen en entornos complejos con múltiples objetos. En segundo lugar, la industria de la seguridad incluye personas responsables de garantizar la seguridad en todas partes, como guardias de seguridad o agentes de policía. Las armas que portan están en la mayoría de los casos colocadas en lugares visibles de sus cuerpos. En tales casos, el sistema no es adecuado para la detección.

La primera categoría de la base de datos, que incluye imágenes donde se presenta un arma de fuego, se compone de imágenes recopiladas de diversos sitios web y de capturas de videos que muestran a personas con arma blanca en distintas situaciones, como robos reales y prácticas de tiro. Las imágenes obtenidas de videos se clasificaron en dos situaciones específicas: la primera corresponde a situaciones de robos reales en las que se evidencia la presencia de un arma de fuego.

En cuanto a la información extraída de videos, se realizaron selecciones en función de dos criterios principales. En primer lugar, se realizo un video simulado sobre un robo a un automovil en el cual se uso una arma blanca. Aunque existe una gran cantidad de videos de este tipo en la web, la mayoría carece de calidad adecuada debido a las condiciones de grabación y a la compresión utilizada para su publicación en línea. Por este motivo, se optó por realizar una simulacion en tiempo real, considerando aspectos como enfoque, resolución gráfica y luminosidad para asegurar que la calidad de la

imagen sea suficiente para el entrenamiento de la red y permita visualizar todas las características de manera efectiva.

En la Figura 66 se muestra la simulacion de robo a vehículo con arma blanca en el día.



Figura 69. Robo a vehículo con arma blanca (Día)

En la Figura 67 se muestra la simulación de robo a vehículo con arma blanca en la noche.



Figura 70. Robo a vehículo con arma blanca (Noche)

La segunda categoría de la base de datos contiene imágenes en las que no se encuentra alguna arma blanca. Para el primer tipo de imágenes de esta clase, se obtuvieron de videos que mostraban a personas caminando normalmente. El segundo tipo de imágenes para esta categoría se adquirió de la web y retrata a personas sin armas blancas en diversas posiciones y entornos. Estas imágenes fueron utilizadas para reemplazar las obtenidas de videos de robos reales en las que no se mostraba el arma blanca, ya que proporcionaban una calidad de imagen superior.

En la Figura 68 se muestra una persona transcurriendo cerca del automóvil.



Figura 71. Persona transcurriendo cerca del automóvil

### 3.12 Evaluación de las pruebas de detección de robo

Al momento de monitorizar las cámaras de vigilancia, se puede observar un robo simulado en la noche, en donde se puede visualizar a un hombre portando un arma blanca. La conductora sale del vehículo con los brazos levantados, dando a entender un acto de robo. En ese instante se detecta el caso de robo, como se puede observar en la Figura 69.



Figura 72. Detección de robo con arma blanca (noche) Cámara 1

Durante la supervisión de las cámaras de vigilancia, se puede presenciar una simulación de robo durante el día, en la que se observa a un individuo llevando consigo un arma blanca. La conductora del vehículo sale del mismo con los brazos en alto,

dando a entender la ocurrencia de un acto de robo. En ese momento, se detecta el incidente de robo, como se ilustra en la Figura 70.



Figura 73. Detección de robo con arma blanca (día) Cámara 1

En la Figura 71 se muestra en la cámara 2 el otro lado el tipo de robo, el cual establece las mismas características que la anterior detección.



Figura 74. Detección de robo con arma blanca (día) Cámara 2

Finalmente, se puede observar la interfaz grafica, donde se puede monitorizar las camaras activadas y a su vez controlar el movimiento de la camara con un joystick implementado dentro de la interfaz. Además se podra realizar grabaciones y subirlas a la nube, y tambien se puede observar la localizacion del vehiculo.

En la Figura 72 se muestra el monitoreo y control de las cámaras en tiempo real mediante la interfaz gráfica.



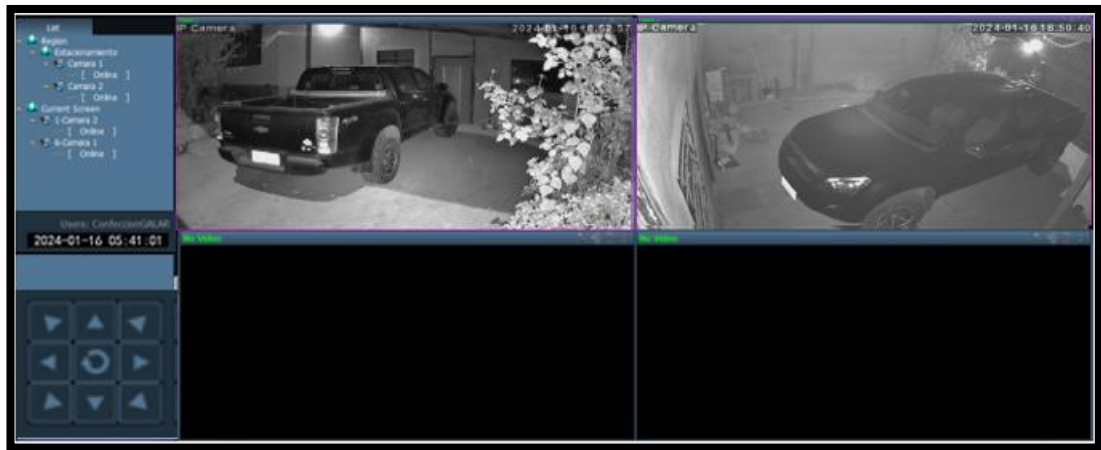


Figura 75. Monitoreo y control de las cámaras en tiempo real mediante la interfaz gráfica.

### 3.12.1 Resultados

#### Matriz de confusión

En el sistema electrónico de control y monitorización de vehículos en caso de robo para la empresa confecciones Galar mediante visión artificial, la matriz de confusión se ha realizado mediante un conjunto de pruebas que se pueden resumir en la Tabla 17:

Tabla 24. Matriz de confusión del sistema

Predicción	Predicción del Sistema	
	Predicción Positivos	Falsos Positivos
Valor Positivo	14 (VP)	1 (FN)
Valor Negativo	1 (FP)	4 (VN)

La matriz se calculo mediante las diferentes metricas como precision, exactitud, recall. Estas matrices brindan una informacion de eficiencia y eficacia del sistema electrónico de control y monitorización de vehículos en caso de robo para la empresa confecciones galar mediante visión artificial. Utilizando estos valores, se logro calcular la métrica de rendimiento.

Se tiene en la matriz de confusion:

- Verdaderos Positivos (TP): 14

- Falsos Positivos (FP): 1
- Verdaderos Negativos (TN): 4
- Falsos Negativos (FN): 1

Con los datos obtenidos en cada prueba se logro observar lo siguiente:

- Verdaderos Positivos (TP): El sistema ha trabajado correctamente en 10 casos positivos, lo cual, se logro detectar satisfactoriamente 10 casos de robo.
- Falsos Positivos (FP): El sistema ha detectado incorrectamente 2 casos de robo cuando las camaras estaban monitoreando el automovil. Es un tipo de error tipo I, que es conocido como “falsos positivos”.
- Verdaderos Negativos (TN): El sistema ha identificado satisfactoriamente 6 casos como negativos, lo cual, permite una deteccion adecuada a 6 casos que no eran de robo.
- Falsos Negativos (FN): El sistema ha detectado 2 casos que no eran de robo. Estos errores son de tipo II, o “falsos negativos”, esto representa que el sistema no detecto un caso de robo.

Para este caso el nivel de peligro por los fallos del sistema electrónico de control y monitorización de vehículos en caso de robo para la empresa confecciones Galar mediante visión artificial depende de dos falsos positivos (FP), esto ocurre cuando una persona se acerca demasiado al automovil con un objeto similar a un arma blanca, como es el caso de juguetes, materiales de trabajo o de oficina. Al igual que un falso negativo (FN), ocurre cuando no detecta los casos de robo cuando en realidad esta sucediendo. Este tipo de error es una advertencia, ya que no alertaria a la persona encargada de monitorizar las camaras y seria un riesgo de seguridad.

### **Validacion de resultados**

Con la Ecuacion 6, se calculo la precisión



$$\text{Precisión} = \frac{TP}{(TP + FP)} \quad (6)$$

$$\text{Precisión} = \frac{14}{(14 + 1)}$$

$$\text{Precisión} = 0,9333$$

Con la Ecuación 7, se calculó el Recall o tasa de verdaderos positivos

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (7)$$

$$\text{Recall} = \frac{14}{(14 + 1)}$$

$$\text{Recall} = 0,9333$$

Con la Ecuación 8, se calculó el F1-score, el cual permite calcular la métrica que combina la precisión y el recall.

$$F1score = 2 * \left( \frac{\text{Precisión} * \text{Recall}}{\text{Precisión} + \text{Recall}} \right) \quad (8)$$

$$F1score = 2 * \left( \frac{0,9333 * 0,9333}{0,9333 + 0,9333} \right)$$

$$F1score = 0,952$$

Con la Ecuación 9, se calculó la exactitud del sistema.

$$\text{Exactitud} = \frac{TP + TN}{(TP + FP + TN + FN)} \quad (9)$$

$$\text{Exactitud} = \frac{14 + 4}{(14 + 1 + 4 + 1)}$$

$$\text{Exactitud} = 0,95$$

Teniendo en cuenta la métrica usada, se puede decir que el sistema tiene una buena precisión y recall, lo que indicaría que es capaz de detectar la mayoría de casos positivos y negativos. El valor de F1-score es alto, lo que demuestra que el sistema tiene una buena precisión y recall. La exactitud del sistema es 0,95 lo que indica que detecta la mayoría de los casos.

Para la evaluación del sistema y su rendimiento se utilizará el F1-score, que combina la precisión como el recall de una métrica. El F1-score varía entre 0 y, donde el valor 1 es rendimiento perfecto y 0 es rendimiento nulo.

Para la obtención del porcentaje equivalente, se utilizará la ecuación 10.

$$\text{Porcentaje de rendimiento del sistema} = \text{F1score} * 100 \quad (10)$$

$$\text{Porcentaje de rendimiento del sistema} = 0,952 * 100$$

$$\text{Porcentaje de rendimiento del sistema} = 95,2\%$$

Este valor indica que el sistema tiene un buen rendimiento de detección de casos positivos y negativos, de manera proporcionada. Aunque es fundamental tener en cuenta que el rendimiento variara por cada prueba realizada.

### 3.12.2 Presupuesto

La asignación de recursos financieros para el proyecto “sistema electrónico de control y monitorización de vehículos en caso de robo para la empresa confecciones Galar mediante visión artificial” se divide en dos categorías: presupuesto de diseño y presupuesto de construcción. La estimación del presupuesto de diseño se basa en el tiempo dedicado al diseño y el salario de un ingeniero de telecomunicaciones, que según el Ministerio de Trabajo de Ecuador es de \$858. Para el cálculo del salario diario se tienen en cuenta 5 días hábiles por semana, para un total de 21 días hábiles por mes. Aplicando estos datos a la Ecuación 11, se obtiene:

$$\text{Salario}_{diario} = \frac{\text{Salario}_{mensual}}{\text{Dias}_{laborales}} \quad (11)$$

$$\text{Salario}_{diario} = \frac{858 \text{ [dólares]}}{21 \text{ [días]}}$$

$$\text{Salario}_{diario} = 40,86 \text{ [dólares]}$$

Se sabe que la jornada laboral es de 8 horas y la prima horaria se calcula mediante la Ecuación 12.

$$\text{Salario}_{hora} = \frac{\text{Salario}_{diario}}{\text{Horas}_{laborales}} \quad (12)$$

$$\text{Salario}_{hora} = \frac{40,86 \text{ [dólares]}}{8 \text{ [horas]}}$$

$$\text{Salario}_{hora} = 5,11 \text{ [dólares]}$$

El presupuesto completo de desarrollo del proyecto preveía 3 horas de trabajo por día durante 4 meses, un total de 252 horas. Este tiempo se divide en varias etapas de investigación, desarrollo, implementación y pruebas funcionales. Usando la Ecuación 13, obtenemos los siguientes resultados:

$$\text{Presupuesto}_{diseño} = \text{Horas}_{investigación} * \text{Salario}_{hora} \quad (13)$$

$$\text{Presupuesto}_{diseño} = 252 \text{ [horas]} * 5,11 \text{ [dólares]}$$

$$\text{Presupuesto}_{diseño} = 1287,72 \text{ [dólares]}$$

Cada componente y su correspondiente precio unitario se presentan de forma precisa y clara, facilitando una evaluación transparente del presupuesto asignado a cada componente. Muestra el costo de los microcontroladores, cámaras y otros equipos adquiridos para implementar los sistemas electrónicos de control y monitoreo del vehículo para evitar que los vehículos de la empresa sean robados por visión artificial.

En la Tabla 18 proporciona una descripción detallada de los costos de los materiales utilizados en la construcción de sistemas electrónicos.

Tabla 25. Descripción detallada de los costos de los materiales utilizados en la construcción de sistemas electrónicos

Ítem	Detalle	Cantidad	P. Unit. (USD)	P. Total (USD)
1	Raspberry Pi 3 Modelo B+	1	\$ 200,00	\$ 200,00
2	Cámara SV3C HX Series	2	\$ 70,00	\$ 140,00
3	Bateria Lipo Hoovo 6200 mAh	2	\$40,00	\$ 80,00
4	Modulo GPS	1	\$ 45,00	\$ 45,00
5	Case	1	\$ 20,00	\$ 20,00
6	Mouse y Teclado	1	\$ 15,00	\$ 15,00
7	Cable USB Tipo C	2	\$ 4,00	\$ 8,00
8	Fuente de 5V	1	\$ 40,00	\$ 40,00
9	Router	1	\$ 25,00	\$ 25,00
Subtotal				\$ 573
IVA (12%)				\$ 68,76
Total				\$ 641,76
Imprevisto (15%)				\$ 96,26
<b>TOTAL</b>				<b>\$ 738,02</b>

Finalmente, se utilizó la Ecuación 14 para calcular el costo total del proyecto de investigación.

$$Presupuesto_{total} = Presupuesto_{diseño} + Presupuesto_{construcción} \quad (14)$$

$$Presupuesto_{total} = 1287,72 \text{ [dólares]} + \$ 738,02 \text{ [dólares]}$$

$$Presupuesto_{total} = 2025,74 \text{ [dólares]}$$

## CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- Después de analizar los equipos de seguridad electrónica disponibles en el mercado automotriz para la prevención de robos, se puede afirmar que existe una amplia variedad de opciones diseñadas para proteger los vehículos de posibles intrusiones. Desde sistemas de alarma básicos hasta tecnologías más avanzadas como sistemas de seguimiento GPS y bloqueo remoto. Por otro lado, al tomar decisiones sobre la implementación de equipos de seguridad electrónica, es fundamental considerar cuidadosamente las características y capacidades de cada opción, así como también evaluar la compatibilidad con el vehículo y las necesidades de protección individuales. En última instancia, la combinación de tecnología de vanguardia, medidas de seguridad adicionales y prácticas de prevención de robos puede contribuir significativamente a mejorar la seguridad y tranquilidad de los propietarios de vehículos en un entorno automotriz cada vez más desafiante.
- El uso de una arquitectura de red que emplea filtros de convolución de mayor tamaño y prescinde de una gran cantidad de estos en sus capas convolucionales también arroja resultados positivos; sin embargo, sus valores de precisión en las predicciones no superan los obtenidos con la estructura de red mencionada anteriormente. Entrenar y evaluar la CNN con imágenes en escala de grises condujo a resultados superiores tanto en la precisión de las predicciones como en los valores de pérdida en comparación con el modelo entrenado con imágenes RGB.
- El diseño del sistema de monitoreo y control de vehículos debe considerar la integración de tecnologías como cámaras, sensores y sistemas de comunicación, asegurando una arquitectura robusta y una colaboración eficiente entre estos componentes para una respuesta efectiva ante eventos de robo. La integración de visión artificial para identificar las expresiones corporales cuando ocurre un robo del vehículo en base a los gestos del agresor y

la víctima son de importancia para que el sistema tenga una respuesta inmediata, almacenamos la información visual del siniestro para poderla usar como argumento legal.

## 4.2 Recomendaciones

- La complejidad asociada a la detección de armas de fuego se ve agravada por la diversidad de entornos presentes en las cámaras de seguridad. Por ende, la obtención de resultados efectivos en las detecciones requiere una extensa base de datos. Esto garantiza que el sistema esté entrenado con elementos comúnmente presentes en situaciones de asalto armado y, simultáneamente, evita el sobreentrenamiento de la red.
- Se aconseja emplear imágenes con un tamaño igual o superior a 224x224 píxeles para garantizar la efectividad en la detección de objetos, en este caso, armas de fuego. El uso de imágenes de menor tamaño tiende a reducir las dimensiones del objeto deseado, comprometiendo la preservación de sus características distintivas.
- Para el entrenamiento y el funcionamiento eficiente del sistema, se recomienda la utilización de una GPU. Dado que tanto el número como el tamaño de las imágenes son considerablemente grandes, un procesamiento óptimo mediante una GPU ayuda a reducir los tiempos de entrenamiento del modelo y asegura un rendimiento eficaz en las tareas de detección.
- Uno de los puntos más críticos del reconocimiento de expresiones corporales es que se debe tomar en cuenta una buena colocación de las cámaras con la finalidad de que capte el mayor área de la persona ya que al tener menos información, se procede a generar falsos negativos. El poder integrar un sistema sonoro en caso de la detección del robo podría ser adecuado en zonas residenciales con densidad poblacional elevada.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] E. Zambrano, «SISTEMA DE MONITORIZACIÓN Y CONTROL INTELIGENTE AUTÓNOMO PARA VEHÍCULOS COMBINANDO TECNOLOGÍA IOT Y REDES NEURONALES.» Univerisdad Israel , 2022.
- [2] M. Terra, «Informe de robos y recuperación vehicular.» Hunter Ecuador , 2023.
- [3] L. Hora, «Cerebros, depuradores y tableros de carros, entre los más robados en Tungurahua.» La Hora, Ambato, 2023.
- [4] L. Hora, «Suben las cifras de robos de vehículos en Tungurahua.» La Hora, 2022.
- [5] E. Onieva, «Técnicas Difusas y Evolutivas para el Control de Vehículos en Entornos Reales y Virtuales.» Universidad de Granada, 2022.
- [6] C. López, «Diseño e implementación de un prototipo de vehículo autónomo a escala con capacidades de visión artificial.» Universidad San Francisco de Quito USFQ, Quito, 2021.
- [7] M. X. Bonilla Jeréz, «Sistema autónomo de monitoreo de señales fisiológicas con gestión de emergencias para seguridad vial de ciclistas amateur.» Universidad Tecnica de Ambato, Ambato, 2019.
- [8] M. Fuentes, «Sistema de visión artificial para el control de calidad de hojas de cuero para la fabrica Gamacueros.» 2019.
- [9] J. Hernandez, «Sistema de monitoreo GPS para los vehículos de la escuela de capacitación para conductores profesionales de Píllaro.» Universidad Tecnica de Ambato, Ambato, 2019.
- [10] G. M. P. Aponte, «Sistema avanzado de asistencia al conductor empleando visión artificial en vehículos de transporte público.» Universidad Técnica de ambato, Ambato, 2019.
- [11] M. M. U. y. J. S. ] O. Reyes Ortiz, «Técnicas de Inteligencia artificial utilizadas en el procesamiento de imágenes y su aplicación en el análisis.» EIA, 2019.
- [12] J. E. Oñate Núñez, «El incremento de las tasas de criminalidad en la ciudad de Ambato, Ecuador durante el año 2022.» MS thesis, 2023.
- [13] K. Encarnación, «Formulación de indicadores para el análisis de la seguridad vial.» Universidad Politecnica Salesiana, Cuenca, 2015.
- [14] I. G. e. a. Torres, «"Sistema de alerta usando Módulo de Reconocimiento de voz para detectar problemas de robo de vehículos",» vol. Vol 4, n° 1, p. pag 20, 2020.
- [15] D. D. León, «Diseño de un sistema de tadquisición de imágenes basado en cámaras web USB y hardware reconfigurable.» RIELAC, 2017.

- [16] A. M. R. y. E. T. J. V. E. L. Becerril, «Sistema para la detección del estado de somnolencia en seres humanos, con reconocimiento de patrones,» Universidad Politécnica de Texcoco.
- [17] A. S. R. Álava, «Implementación de un algoritmo de detección de somnolencia humana, en tiempo real basado en visión artificial,» Escuela Politécnica Nacional, Quito, 2019.
- [18] F. M., «Evolucion de la internet el internet de las cosas,» Ecured , 2019.
- [19] controlf5it, «IOT a internet das coisa,» controlf5it, 2020.
- [20] L. Ambata, «Distance monitoring vision system for automobiles using image processing,» HNICEM, 2016.
- [21] R. R. V. y. K. F. Suarez, «Estudio de los procesadores Digitales de señales para el desarrollo de aplicaciones en tiempo real,» Universidad Tecnológica de Bolívar, Cartagena de Indias, 2006.
- [22] H. H. Rivera, «Sistema electrónico de adquisición y procesamiento de señales con comunicación USB,» Posgrado Interinstitucional en Ciencia y Tecnología, Santiago de Querétano, 2016.
- [23] M. M. U. y. J. S. O. Reyes Ortiz, «Técnicas de Inteligencia artificial utilizadas en el procesamiento de imágenes y su aplicación en el análisis de pavimentos,» 2019.
- [24] C. B. y. E. D. C. Jacobé de Naurois, «Detection and prediction of driver drowsiness using artificial neural network models,» Accident Analysis and Prevention, 2019.
- [25] D. A. A. Brítez, «Optimización multi-objetivo de arquitecturas de aprendizaje profundo para el procesamiento de señales EEG en plataformas de cómpu heterogéneas,» Universidad de Granada, Granada, 2022.
- [26] W. C. W. y. Y. T. Casañola, «Redes Neuronales Artificiales en la estimación del esfuerzo,» Revista Cubana de Ciencias Informáticas, 2021.
- [27] Proteccars, «Sistemas de Seguridad antirrobo efectivos para proteger tu vehículo,» Ecuador, 2022.
- [28] Pinsoft, «Cámara web digital D-Bugg M26,» Estados Unidos, 2021.
- [29] U. Electronics, «Raspberry Pi NoIR Cámara V2,» Estados Unidos, 2021.
- [30] Logitech, «WEBCAM C920 HD PRO,» Estados Unidos, 2022.
- [31] SV3C, «SV3C Camera HX Series,» China.
- [32] Arduino, «Arduino Uno,» Estados Unidos, 2023.
- [33] Arduino, «Arduino Nano,» Estados Unidos, 2023.
- [34] N. Mechatronics, «NODEMCU V2 ESP8266 WIFI,» Estados Unidos, 2023.



- [35] R. Pi, «Raspberry Pi 3 Model B+,» Estados Unidos, 2023.
- [36] INSHT, «Baterías de Ni-Cd. Uso y mantenimiento,» Ministerio de trabajo y asuntos Sociales España, España, 1984.
- [37] Jungheinrich, «Baterías de iones de litio,» China, 2023.
- [38] B. Equivalent, «HOOVO 7.4V 70C 6200mAh 2S Lipo Battery,» China, 2023.
- [39] G. F. Fernández, «Elementos de sistemas operativos, de representación de la información y de procesadores hardware y software,» Universidad Politécnica de Madrid, Madrid, 2015.
- [40] J. Salazar, «Redes Inalámbricas,» European Virtual Learning Platform for Electrical and Information Engineering, 2022.
- [41] E. M. Rojas, «Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo,» Revista Ibérica de Sistemas e Tecnologias de Informação, 2020.
- [42] O. Jativa, «Ejercicio completo en Python para clasificar más de 70mil imágenes en pocas líneas usando Keras,» Universidad Central del Ecuador, 2020.
- [43] S. S. Prado, «Cloud Computing: Fundamentos y despliegue de un servicio en la nube,» Universidad Autónoma de Madrid, Madrid, 2021.
- [44] P. P. Kumar, «Cloud Computing: Possibilities, Challenges and Opportunities with Special Reference to its Emerging Need in the Academic and Working Area of Information Science,» ElServier, 2012.
- [45] A. Follegatti, «Investigación de implementación de Cloud Computing en IT-EXPERT,» Universidad Peruana de Ciencias Aplicadas, Lima, 2018.
- [46] R. Pi, «Raspberry Pi 3 Modelo B+,» Estados Unidos, 2022.
- [47] SV3C, «SV3C HX Series,» China, 2023.
- [48] T. LINK, «Router Wi-Fi de doble banda AC1200,» Estados Unidos, 2023.
- [49] SIMCOM, «Modulo SIM808,» China, 2022.
- [50] securelist, «Nuevas tendencias en el mundo de las amenazas IoT,» securelist, 2019.
- [51] SCIELO, «Tecnología GPS, GPRS para el uso de las comunicaciones inalámbricas,» SCIELO, 2019.
- [52] Azimut, «Sistema de posicionamiento global GPS,» Azimut, 2022.
- [53] Palacios, «LA INTERNET DE LAS COSAS,» Internet Society, 2020.
- [54] P. A9G, «Pudding A9G,» China, 2020.
- [55] S. Electronica, «TTGO SIM 7000G,» Estados Unidos, 2021.

[56] U. Electronics, «SIM 808,» China, 2022.

## ANEXOS

### Anexo A. Raspberry Pi 3 Modelo B+

En la Figura A1 se muestra el datasheet del microcontrolador Raspberry pi 3 model B+.

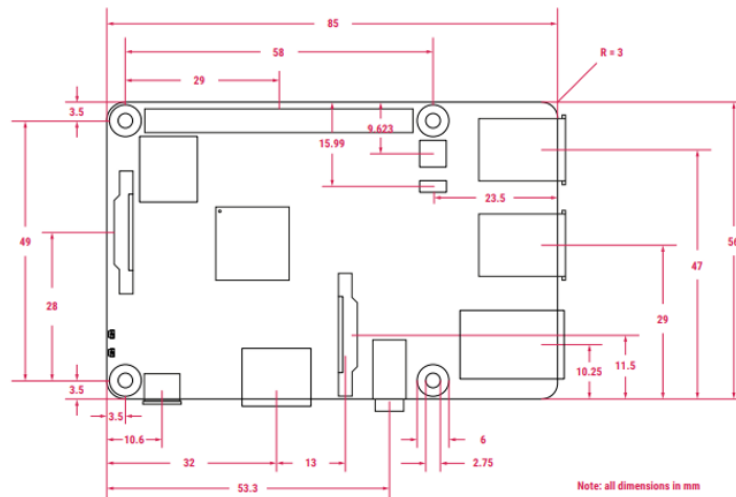


Figura A1. Datasheet del microcontrolador Raspberry pi 3 model B+.

En la Figura A2 se muestra las especificaciones Técnicas de Raspberry pi 3 model B+.

#### Specification

<b>Processor:</b>	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
<b>Memory:</b>	1GB
<b>Connectivity:</b>	<ul style="list-style-type: none"><li>• 2.4 GHz and 5 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 4.2, BLE</li><li>• Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)</li><li>• 4 × USB 2.0 interface</li></ul>
<b>Video and sound:</b>	<ul style="list-style-type: none"><li>• 1 x full size HDMI</li><li>• MIPI DSI display port</li><li>• MIPI CSI camera port</li><li>• 4 pole stereo output and composite video port</li></ul>
<b>Multimedia:</b>	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
<b>SD card support:</b>	Micro SD format for loading operating system and data storage
<b>Input Power:</b>	<ul style="list-style-type: none"><li>• 5V/2.5A DC via micro USB connector</li><li>• 5V DC via GPIO header</li><li>• Power over Ethernet (PoE)-enabled (requires separate PoE HAT)</li></ul>
<b>Operating temperature:</b>	0-50°C
<b>Production lifetime:</b>	Raspberry Pi 3 Model B+ will remain in production until at least January 2028
<b>Compliance:</b>	For a full list of local and regional product approvals, please visit <a href="http://pip.raspberrypi.com">pip.raspberrypi.com</a>

Figura 76. Especificaciones Técnicas de Raspberry pi 3 model B+ [46]

## Anexo B. Cámara SV3C HX Series

En la Figura B1 se muestra las especificaciones de la Cámara SV3C HX Series.



Figura B1. Especificaciones de la Cámara SV3C HX Series [47]

## Anexo C. Router Wi-Fi de doble banda AC1200

En la Figura C1 se muestra las dimensiones Router Wi-Fi de doble banda AC1200.



Figura C1. Dimensiones Router Wi-Fi de doble banda AC1200 [48]

En la Figura C2 se muestra las especificaciones Técnicas Router Wi-Fi de doble banda AC1200.

Especificaciones clave		
WIRELESS	Standards	Wi-Fi 5 IEEE 802.11ac/802.11n IEEE 802.11n/b/g 2.4 GHz
	WiFi Speeds	AC1200: 5 GHz: 867 Mbps (802.11ac) 2.4 GHz: 300 Mbps (802.11n)
	WiFi Range	3 Bedroom Houses 4x Fixed Antennas Cover more area with more antennas Beamforming Concentrates wireless signal strength towards clients to expand WiFi range
	WiFi Capacity	Medium Dual-Band Distribute devices to different bands for optimal performance 2x2 MU-MIMO Simultaneously communicates with multiple MU-MIMO clients
HARDWARE	Ethernet Ports	1x 10/100 Mbps WAN Port 4x 10/100 Mbps LAN Ports
SECURITY	WiFi Encryption	WEP WPA WPA2 WPA/WPA2-Enterprise (802.1x)
SOFTWARE	Network Security	SPI Firewall Access Control IP & MAC Binding
	Guest Network	1x 5 GHz Guest Network 1x 2.4 GHz Guest Network
	Parental Control	URL Filtering Time Controls
	Quality of Service	QoS by Device
PHYSICAL	Package Contents	Wi-Fi Router Archer C50 Power Adapter RJ45 Ethernet Cable Quick Installation Guide

Show More >

Figura C2. Especificaciones Técnicas Router Wi-Fi de doble banda AC1200 [48]

## Anexo D. Módulo SIM808

En la Figura D1 se muestra el Datasheet Modulo SIM808.

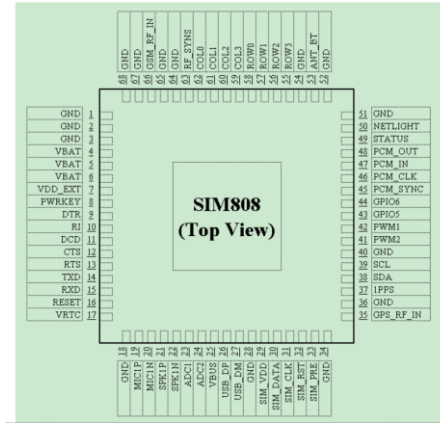


Figura D1. Datasheet Modulo SIM808 [49]

En la Figura D2 se muestra las especificaciones Técnicas Modulo SIM808.

Mode	Function	
Normal operation	GSM/GPRS SLEEP	Module will automatically go into sleep mode if the conditions of sleep mode are enabling and there is no on air and no hardware interrupt (such as GPIO interrupt or data on serial port). In this case, the current consumption of module will reduce to the minimal level. In sleep mode, the module can still receive paging message and SMS.
	GSM IDLE	Software is active. Module registered to the GSM network, and the module is ready to communicate.
	GSM TALK	Connection between two subscribers is in progress. In this case, the power consumption depends on network settings such as DTX off/on, FR/EFR/HR, hopping sequences, antenna.
	GPRS STANDBY	Module is ready for GPRS data transfer, but no data is currently sent or received. In this case, power consumption depends on network settings and GPRS configuration.
	GPRS DATA	There is GPRS data transfer (PPP or TCP or UDP) in progress. In this case, power consumption is related with network settings (e.g. power control level); uplink/downlink data rates and GPRS configuration (e.g. used multi-slot settings).
	Charge	The mode support charge function (Default is not support).
Power down	Normal power down by sending the AT command "AT+CPOWD=1" or using the PWRKEY. The power management unit shuts down the power supply for the baseband part of the module, and only the power supply for the RTC is remained. Software is not active. The serial port is not accessible. Power supply (connected to VBAT) remains applied.	
Minimum functionality mode	AT command "AT+CFUN" can be used to set the module to a minimum functionality mode without removing the power supply. In this mode, the RF part of the module will not work or the SIM card will not be accessible, or both RF part and SIM card will be closed, and the serial port is still accessible. The power consumption in this mode is lower than normal mode.	

Figura D2. Especificaciones Técnicas Modulo SIM808 [49]

## Anexo E. Batería Lipo HOOVO

En la Figura E1 se muestra las especificaciones técnicas de la Batería Lipo HOOVO.



Figura E1. Especificaciones Técnicas de la Batería Lipo HOOVO.

## Anexo F. Diagrama del dispositivo de Monitoreo

En la Figura F1 se muestra el diagrama del dispositivo de monitoreo.

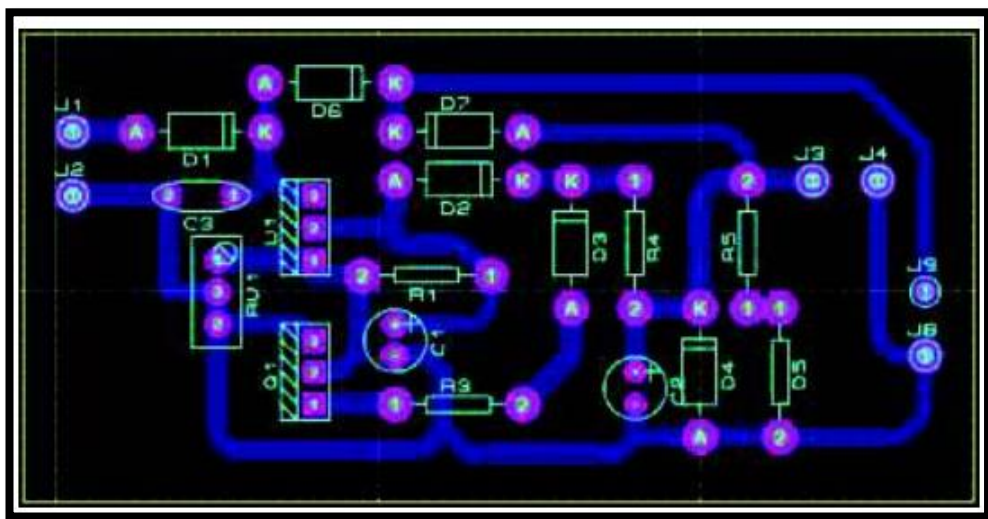


Figura F1. Diagrama del dispositivo de monitoreo

## Anexo G. Empresa Confecciones GALAR

En la Figura G1 se muestra la instalación de cámaras en la Empresa Confecciones GALAR.



Figura G1. Instalación de cámaras en la Empresa Confecciones GALAR



En la Figura G2 se muestra la conexión de las cámaras con el router TP Link.



Figura G2. Conexión de las cámaras con el router TP Link

En la Figura G3 se muestra la instalación de la Tarjeta Raspberry Pi 3 Modelo B+.



Figura G3. Instalación de la Tarjeta Raspberry Pi 3 Modelo B+

En la Figura G4 se muestra el estacionamiento Empresa Confecciones GALAR.



Figura G4. Estacionamiento Empresa Confecciones GALAR

## Anexo H. Frames de la simulación de robo

En la Figura H1 se muestra los frames de la simulación de robo en el estacionamiento.



Figura H1. Frames de la simulación de robo en el estacionamiento.

## Anexo I. Código en Python del sistema de control y monitorización

### Código conversión de Video a Imágenes (Frames)

```
import cv2

def dividir_video_a_frames(video_path, carpeta_destino,
nombre_base='frame'):
    # Abre el video
    video = cv2.VideoCapture(video_path)

    # Inicializa variables
    success, frame = video.read()
    count = 0

    # Lee cada frame y lo guarda como una imagen
    while success:
        # Guarda el frame como una imagen
        nombre_frame = f'{nombre_base}_{count:04d}.png'
        ruta_frame = f'{carpeta_destino}/{nombre_frame}'
        cv2.imwrite(ruta_frame, frame)

        # Lee el siguiente frame
        success, frame = video.read()
        count += 1

    # Libera los recursos
    video.release()

if __name__ == "__main__":
    # Especifica la ruta del video y la carpeta de destino
    video_path = 'C:\Users\LENOVO\Documents\Tesis-
Sistemas_de_Monitorizacion\Video-robo-a-vehiculo.mp4'
    carpeta_destino = 'C:\Users\LENOVO\Documents\Tesis-
Sistemas_de_Monitorizacion\Imagenes-robo-a-vehiculo'

    # Llama a la función para dividir el video en frames
    dividir_video_a_frames(video_path, carpeta_destino)
```

## Código de Detección de robo

```
import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk
import cv2
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense
import numpy as np

class VehicleSecurityApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Sistema de Control y Monitorización de
Vehículos")

        # Inicializa la interfaz gráfica
        self.init_gui()

        # Inicializa las cámaras y modelos
        self.init_cameras_and_models()

        # Inicia la actualización continua de las cámaras
        self.show_cameras()

    def init_gui(self):
        # Crear frames para cada cámara
        self.camera_frames = [ttk.Frame(self.root) for _ in range(4)]
        for i, camera_frame in enumerate(self.camera_frames):
            camera_frame.grid(row=0, column=i, padx=10, pady=10)

        # Se crea etiquetas para la visualización de las imágenes
        self.image_labels = [ttk.Label(frame) for frame in
self.camera_frames]

        # Se crea etiquetas para los resultados de detección de robo
        self.result_labels = [ttk.Label(frame, text="Sin robo
detectado", font=("Helvetica", 12)) for frame in self.camera_frames]

        # Colocacion las etiquetas en la interfaz
        for i in range(4):
            self.image_labels[i].grid(row=0, column=0)
            self.result_labels[i].grid(row=1, column=0)

    def init_cameras_and_models(self):
```

```

        # Ruta de la carpeta que contiene las imágenes para detección
        self.image_folder_path = "C:\Users\LENOVO\Documents\Tesis-
Sistemas_de_Monitorizacion\Imagenes-escala-de-grises"

        # Cargar modelos Keras para la detección de robo (uno para cada
cámara)
        self.models = [self.load_cnn_model(f'modelo_camara_{i}.h5') for
i in range(1, 5)]

        # Cargar modelos Keras para la detección de robo (uno para cada
cámara)
        self.models = [self.load_cnn_model(f'modelo_camara_{i}.h5') for
i in range(1, 5)]

def load_cnn_model(self, model_path):
    # Definir el modelo de red neuronal convolucional
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(224,
224, 3)),
        MaxPooling2D((2, 2)),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Flatten(),
        Dense(512, activation='relu'),
        Dense(1, activation='sigmoid')
    ])

    # Compilar el modelo
    model.compile(optimizer='adam',
                  loss='binary_crossentropy',
                  metrics=['accuracy'])

    # Cargar los pesos del modelo
    model.load_weights(model_path)

    return model

def show_cameras(self):
    # Inicializar variables para la combinación de predicciones
    combined_predictions = []

    # Obtener la lista de archivos de imágenes en la carpeta
    image_files = os.listdir(self.image_folder_path)

```



```

for i, file_name in enumerate(image_files):
    # Construir la ruta completa de la imagen
    image_path = os.path.join(self.image_folder_path, file_name)

    # Leer la imagen
    frame = cv2.imread(image_path)

    # Procesar la imagen con el modelo de detección de robo
    result = self.detect_robo(frame, i)

    # Mostrar la imagen en la interfaz
    img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    img = Image.fromarray(img)
    img = ImageTk.PhotoImage(img)

    self.image_labels[i].config(image=img)
    self.image_labels[i].img = img

    # Actualizar la etiqueta de resultado
    self.result_labels[i].config(text=result)

    # Guardar la predicción del modelo para la imagen actual
    combined_predictions.append(result)

# Tomar una decisión final combinando las predicciones
final_decision = self.combine_predictions(combined_predictions)

# Mostrar la decisión final en algún lugar de la interfaz
print("Decisión Final:", final_decision)

# Llamar a la función de forma continua después de un breve
intervalo
self.root.after(10, self.show_cameras)

def detect_robo(self, frame, camera_number):
    # Preprocesamiento de la imagen según las necesidades del
modelo
    frame = cv2.resize(frame, (224, 224)) # Se ajusto según los
requisitos del modelo
    frame = frame / 255.0 # Normaliza

    # Realización la predicción con el modelo de detección de robo
prediction = self.models[camera_number -
1].predict(np.expand_dims(frame, axis=0))

    # Determinación el resultado de la detección
    if prediction[0][0] > 0.5:
        return "Robo detectado"

```

```

        else:
            return "Sin robo detectado"

    def combine_predictions(self, predictions):
        # Implementacion de la lógica para combinar las predicciones de
        las 4 cámaras

        if "Robo detectado" in predictions:
            return "Robo detectado"
        else:
            return "Sin robo detectado"

if __name__ == "__main__":
    root = tk.Tk()
    app = VehicleSecurityApp(root)
    root.mainloop()

```

### Código de entrenamiento en Azure

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Definición de la arquitectura de la red neuronal
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224,
3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compilación del modelo
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Preprocesamiento los datos
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2,
zoom_range=0.2, horizontal_flip=True)

```



```

train_generator =
train_datagen.flow_from_directory('ruta_del_directorio_de_entrenamiento
', target_size=(224, 224), batch_size=32, class_mode='binary')

# Entrenamiento el modelo
model.fit(train_generator, epochs=10,
steps_per_epoch=len(train_generator))

# Elementos del modelo entrenado
model.save('modelo_entrenado.h5')

```

## Anexo J. Interfaz gráfica del sistema de control y monitorización

En la Figura J1 se muestra la interfaz gráfica del sistema de control y monitorización de vehículos en caso de robo en la Empresa Confecciones GALAR mediante visión artificial.

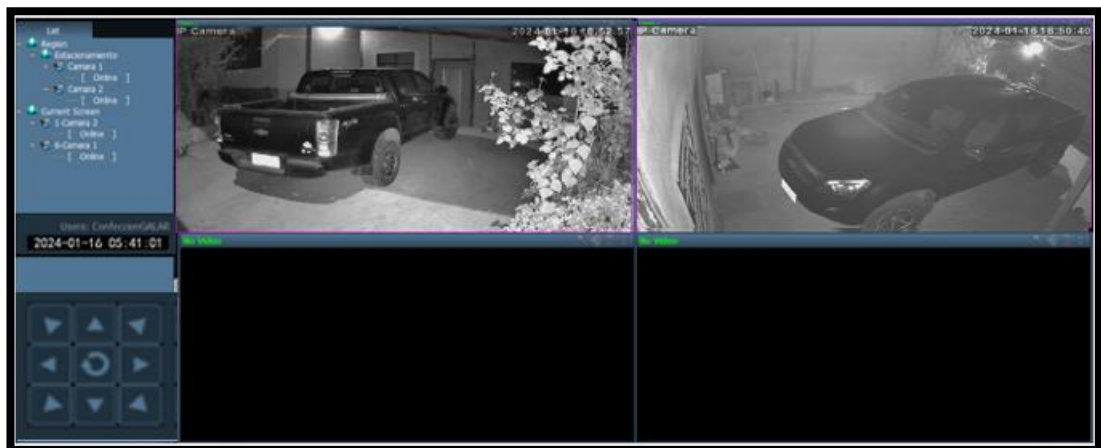


Figura J1. Interfaz gráfica del sistema de control y monitorización de vehículos en caso de robo en la Empresa Confecciones GALAR mediante visión artificial