



**UNIVERSIDAD TÉCNICA DE AMBATO**

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

**CARRERA DE TELECOMUNICACIONES**

**Tema:**

---

**SISTEMA AUTOMÁTICO DE REGISTRO DE AUTOMÓVILES Y  
MANEJO DEL HISTORIAL DE TRABAJOS PARA LA  
MICROEMPRESA “TECNICARS DAVID” UTILIZANDO MACHINE  
LEARNING**

---

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

**ÁREA:** Comunicaciones

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

**AUTOR:** Carlos Alfonso Pérez Valencia

**TUTOR:** Dr. Freddy Geovanny Benalcázar Palacios

**Ambato – Ecuador**

**agosto – 2023**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del trabajo de titulación con el tema: SISTEMA AUTOMÁTICO DE REGISTRO DE AUTOMÓVILES Y MANEJO DEL HISTORIAL DE TRABAJOS PARA LA MICROEMPRESA “TECNICARS DAVID” UTILIZANDO MACHINE LEARNING, desarrollado bajo la modalidad Proyecto de Investigación por el señor Carlos Alfonso Pérez Valencia, 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, agosto 2023.

-----  
Dr. Freddy Geovanny Benalcázar Palacios

**TUTOR**

## AUTORÍA

El presente trabajo de titulación titulado: SISTEMA AUTOMÁTICO DE REGISTRO DE AUTOMÓVILES Y MANEJO DEL HISTORIAL DE TRABAJOS PARA LA MICROEMPRESA “TECNICARS DAVID” UTILIZANDO MACHINE LEARNING 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, agosto 2023

---

Carlos Alfonso Pérez Valencia

C.C. 1804734687

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, agosto 2023

---

Carlos Alfonso Pérez Valencia

C.C. 1804734687

AUTOR

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del trabajo de titulación presentado por el señor Carlos Alfonso Pérez Valencia, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación titulado SISTEMA AUTOMÁTICO DE REGISTRO DE AUTOMÓVILES Y MANEJO DEL HISTORIAL DE TRABAJOS PARA LA MICROEMPRESA “TECNICARS DAVID” UTILIZANDO MACHINE LEARNING, 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, agosto 2023.

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

-----  
Ing. Pamela Ana Castro Martin  
PROFESOR CALIFICADOR

-----  
Dr. Jaime Guilcapi Mosquera  
PROFESOR CALIFICADOR

## **DEDICATORIA**

El presente proyecto de titulación se lo dedico a mis padres, Miriam y Hector, por apoyarme en toda la carrera universitaria, a familiares y amigos por el apoyo moral durante toda mi vida.

**Carlos**

## **AGRADECIMIENTO**

El presente proyecto de titulación se lo dedico a mis padres, Miriam Valencia y Héctor Pérez, por ser el pilar fundamental durante toda mi vida, por todo el esfuerzo y sacrificio realizado con el fin de posibilitar todo lo necesario para realizar mis estudios. A mis hermanos por su apoyo incondicional, por sus consejos para afrontar los retos que se me han presentado a lo largo de mi vida. A mis, amigos y compañeros, porque me han brindado su ayuda durante mi etapa universitaria.

**Carlos**

## ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR .....	ii
APROBACIÓN DEL TRIBUNAL DE GRADO .....	v
Dedicatoria.....	vi
Agradecimiento.....	vii
RESUMEN EJECUTIVO.....	xvi
ABSTRACT.....	xvii
CAPÍTULO I .....	1
MARCO TEÓRICO .....	1
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 Reconocimiento de placas.....	7
1.3.2 Machine Learning .....	7
1.3.3 Tipos de Machine Learning .....	7
1.3.4 Aprendizaje Supervisado .....	8
1.3.5 Aprendizaje semi-supervisado .....	8
1.3.6 Aprendizaje sin supervisión .....	8
1.3.7 Lógica Difusa.....	9
1.3.8 Principio de incompatibilidad .....	10
1.3.9 Representación de la información imprecisa: .....	10
1.3.10 Inferencia sobre información imprecisa:.....	10
1.3.11 Regla Composicional de Inferencia .....	10
1.3.12 Conjuntos difusos.....	11
1.3.13 Función de Pertenencia .....	12
1.3.14 Razonamiento aproximado .....	12
1.3.15 Reconocimiento y procesado digital de imágenes .....	13
1.3.16 Reconocimiento óptico de caracteres .....	13
1.3.17 Binarización .....	13
1.3.18 Fragmentación o segmentación de la imagen .....	14
1.3.19 Proceso de Adquisición de Imágenes.....	14

1.3.20 Elementos para captar la imagen.....	15
1.3.21 Proceso de Adquisición de la Imagen: .....	15
1.3.22 Procesamiento de imágenes y análisis de placas vehiculares .....	16
1.3.23 Técnicas de Procesamiento y Corrección de Imágenes .....	16
1.3.24 Revisión y selección de librerías para realizar el procesamiento de imágenes ...	19
1.3.26 Comparativas y elección de la herramienta .....	22
1.3.30 Técnicas ROC .....	32
1.3.32 Comunicaciones .....	36
1.3.33 Tipos de protocolos de red .....	37
1.3.34 Almacenamiento en la nube .....	37
1.3.35 Base de Datos.....	38
1.3.36 Cámaras.....	42
1.3.37 Medios de Transmisión.....	42
1.4 Objetivos .....	45
1.4.1. Objetivo General.....	45
1.4.2. Objetivos Específicos .....	45
CAPÍTULO II.....	46
METODOLOGÍA.....	46
2.1. Materiales.....	46
2.2. Métodos.....	46
2.2.1. Modalidad de investigación.....	46
2.2.2. Recolección de información .....	47
2.2.3. Procesamiento y análisis de datos .....	47
CAPÍTULO III.....	48
RESULTADOS Y DISCUSIÓN .....	48
3.1. Análisis y discusión de los resultados.....	48
3.1.1 Desarrollo de la propuesta.....	48
3.1.2 ETAPA 1.- Adquisición.....	49
3.1.3 ETAPA 2.- Procesamiento y transmisión .....	52
3.1.4 ETAPA 3.- Servicio en la nube y visualización.....	56
3.1.5 Estancia en la Nube.....	61

3.1.6	Instalación del Bróker Mosquitto.....	66
3.1.7	Instalación de MariaDB .....	67
3.1.8	Instalación del servidor Apache.....	68
3.1.10	Instalación de Node-Red para creación y acceso a la interfaz gráfica .....	70
3.1.11	Conexión con MySql .....	70
3.2	Configuración de Node-Red .....	72
3.2.1	Configuración de Node-Red para el ingreso de datos de los usuarios-clientes.....	73
3.2.2	Configuración de Node-Red para almacenar los datos que arroja la Raspberry Pi4 .....	74
3.3	Flujograma de procesos del Sistema de Detección.....	80
3.4	Reconocimiento de caracteres de placa vehicular con Lógica Difusa.....	83
3.4.1	La Transformación de Características Invariantes de Escala.....	84
3.4.2	Resultados de Segmentación de Imágenes.....	88
3.4.3	Resultados de Extracción de Características.....	89
3.5	Machine Learning aplicado al historial de la base de datos del sistema de registro .....	94
3.5.1	Programación del algoritmo en Python.....	95
3.6	Funcionamiento del Sistema.....	96
3.6.1	Procedimiento para la obtención de la placa del vehículo .....	96
3.7	Pruebas de Funcionamiento .....	100
3.7.1	Verificación del funcionamiento del algoritmo programado .....	101
3.7.2	Análisis de resultados.....	103
3.7.3	Funcionamiento del sistema con letras y números borrosos e incompletos.....	116
3.7.3	Realización de pruebas con placas reales borrosas .....	120
3.7.4	Verificación de la precisión y eficiencia del sistema .....	121
3.7.5	Probabilidad de Detección de placas.....	122
3.7.6	Probabilidad de Aciertos de consulta de Datos con la base de datos .....	123
3.7.8	Problemas encontrados .....	123
3.8	Costos.....	124
CAPÍTULO IV .....		126
CONCLUSIONES Y RECOMENDACIONES .....		126
4.1.	Conclusiones .....	126

4.2. Recomendaciones.....	128
Bibliografía .....	130
ANEXOS .....	134
Anexo 1. Datasheed Tarjeta Raspberry Pi4	135
Anexo 2. Datasheed Cámara .....	138
Anexo 3. Taller de servicio Automotriz "Técnicars David" .....	140
Anexo 4. Instalación cámara .....	141
Anexo 5. Instalación Raspberry .....	142
Anexo 6. Código en Python Sistema Automático de Registro .....	143
Anexo 7. Código Diseño Interfaz gráfica.....	152

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Características principales de la Lógica Difusa. [14].....	11
<b>Tabla 2.</b> Tipos de Operadores Threshold. [22].....	19
<b>Tabla 3.</b> Modelos de color de placa de Instituciones Públicas del Estado .....	26
<b>Tabla 4.</b> Servicio que posee el automóvil en Ecuador. [28].....	27
<b>Tabla 5.</b> Identificación de letra por provincia. [28].....	28
<b>Tabla 6.</b> Tipos de protocolos de red. [31].....	37
<b>Tabla 7.</b> Comparación de las cámaras. [38] [39] [40].....	51
<b>Tabla 8.</b> Comparación de Microordenadores. [41] [42].....	53
<b>Tabla 9.</b> Tecnologías Inalámbricas. [43] .....	55
<b>Tabla 10.</b> Comparación de las plataformas de la Nube. [44] .....	56
<b>Tabla 11.</b> Protocolos de comunicación comparación. [45] .....	58
<b>Tabla 12.</b> Comparación de la Base de datos. [46] [47] .....	59
<b>Tabla 13.</b> Coeficientes de área. [48].....	88
<b>Tabla 14.</b> Tabla de Resultados de placas.....	103
<b>Tabla 15.</b> Tabla de Resultados erróneos.....	108
<b>Tabla 16.</b> Tabla de costos de implementación del sistema.....	124

## ÍNDICE DE FIGURAS

<b>Figura1.</b> Esquema de modelado de la Lógica Difusa [13] .....	9
<b>Figura2.</b> Imagen Normal [22] .....	19
<b>Figura3.</b> Imagen con Threshold . [22].....	19
<b>Figura4.</b> Resultados de tiempo de procesamiento de las librerías. [26].....	22
<b>Figura5.</b> Librería a utilizar OPEN CV. [26].....	23
<b>Figura6.</b> Diseño de una placa en Ecuador. [27] .....	24
<b>Figura7.</b> Formato de Placa actual y placa antigua. [27] .....	25
<b>Figura8.</b> Formato de placa de acuerdo con el nuevo reglamento 2012. [27] .....	25
<b>Figura 9.</b> Entorno de Trabajo utilizado. [30].....	36
<b>Figura 10.</b> Base de Datos Mysql [33].....	39
<b>Figura 11.</b> Base de Datos Oracle [34] .....	40
<b>Figura 12.</b> Base de datos PostgreSQL. [35] .....	41
<b>Figura 13.</b> Modelos de cámaras IP. [36] .....	42
<b>Figura 14.</b> Par trenzado cable UTP categoría 6. [37].....	44
<b>Figura 15.</b> Arquitectura adquisición de datos del sistema de registro automático....	48
<b>Figura 16.</b> Arquitectura procesamiento de datos del sistema.....	49
<b>Figura 17.</b> Arquitectura almacenamiento de datos del sistema.....	49
<b>Figura 18.</b> Instalación de cámara .....	52
<b>Figura 19.</b> Tarjeta Rasberry pi4 .....	54
<b>Figura 20.</b> Instalación de la Tarjeta Raspberry Pi4 .....	54
<b>Figura 21</b> Suscripción de Azure Education.....	61
<b>Figura 22.</b> Creación del grupo de Recursos Tecnicars.....	62
<b>Figura 23.</b> Validación del grupo de recursos .....	62
<b>Figura 24.</b> Creación del Servidor Ubuntu Server.....	63
<b>Figura 25.</b> Configuración de la Autenticación SSH.....	64
<b>Figura 26.</b> Servidor web creado .....	65
<b>Figura 27.</b> Habilitación de puertos en Microsoft Azure.....	65
<b>Figura 28.</b> Bróker de Mosquitto funcionando correctamente .....	66
<b>Figura 29.</b> Servicio de MariaDB funcionando correctamente .....	67
<b>Figura 30.</b> Servidor Apache funcionando correctamente.....	68
<b>Figura 31.</b> phpMyAdmin en el Navegador .....	69
<b>Figura 32.</b> pm2 instalado y ejecución en Node-Red .....	70

<b>Figura 33.</b> Base de datos .....	71
<b>Figura 34.</b> Almacenamiento de los datos del cliente o usuario.....	71
<b>Figura 35.</b> Almacenamiento de los datos de la Raspberry Pi4.....	72
<b>Figura 36.</b> Relación de tablas por claves foráneas .....	72
<b>Figura 37.</b> Interfaz web de Node-Red .....	73
<b>Figura 38.</b> Nodo Form.....	74
<b>Figura 39.</b> Ingreso de datos a la base de datos .....	74
<b>Figura 40.</b> Nodo Función .....	75
<b>Figura 41.</b> Nodo mysql.....	75
<b>Figura 42.</b> Nodo mqtt in y Nodo json .....	75
<b>Figura 43.</b> Creación de tablas y grupos .....	76
<b>Figura 44.</b> Interfaz del Dashboard para el ingreso de Usuarios .....	76
<b>Figura 45.</b> Interfaz del Dashboard /Control de ingreso de vehículos.....	77
<b>Figura 46.</b> Interfaz del Dashboard /Proforma requerimientos .....	78
<b>Figura 47.</b> Interfaz del Dashboard /Check list de calidad .....	79
<b>Figura 48.</b> Interfaz del Dashboard /Cliente notas.....	80
<b>Figura 49.</b> Dato de la placa enviado desde la Raspberry Pi4 al servidor .....	81
<b>Figura 50.</b> Diagrama de flujo de Sistema de Registro .....	82
<b>Figura 51.</b> Dimensión de la placa de matrícula. [48] .....	88
<b>Figura 52.</b> Ejemplo de una placa. [48] .....	89
<b>Figura 53.</b> Caracteres Segmentados de la placa. [48] .....	89
<b>Figura 54.</b> Imagen en escala de pirámide gaussiana para "A" [48].....	90
<b>Figura 55.</b> Puntos máximos extraídos para "A" [48]. .....	91
<b>Figura 56.</b> Características determinadas en los caracteres de la placa [32]. .....	92
<b>Figura 57.</b> Diagrama de bloques de Lógica Difusa [32]. .....	93
<b>Figura 58.</b> Membresía Función del sistema OCR [32]. .....	94
<b>Figura 59.</b> Gráfico de un algoritmo de Clasificación. [49] .....	95
<b>Figura 60.</b> Librería Sklearn para clasificación Supervisada.....	95
<b>Figura 61.</b> Arquitectura del Sistema Automático de Registro .....	96
<b>Figura 62.</b> Detección de Imagen en Movimiento .....	97
<b>Figura 63.</b> Captura de imagen a procesar .....	97
<b>Figura 64.</b> Parte de la imagen a procesar .....	98
<b>Figura 65.</b> Imagen en movimiento capturada por la Cámara .....	98

<b>Figura 66.</b> Imagen transformada a escala de grises y detección de bordes.....	99
<b>Figura 67.</b> Imagen mejorada con eliminación de Ruido .....	99
<b>Figura 68.</b> Área para procesar con algoritmos de Lógica Difusa.....	100
<b>Figura 69.</b> obtención de datos de placa de la imagen.....	100
<b>Figura 70.</b> Prueba de funcionamiento con placa borrosa .....	101
<b>Figura 71.</b> Captura de imagen procesada para los indicadores .....	102
<b>Figura 72.</b> Pruebas de Funcionamiento de la detección de placas .....	102
<b>Figura 73.</b> Base de datos para entrenar el algoritmo de clasificación.....	109
<b>Figura 74.</b> Ingreso de Vehículo detectado por movimiento.....	110
<b>Figura 75.</b> Reconocimiento de placa por el sistema.....	110
<b>Figura 76.</b> Envío de dato del sistema al Servidor.....	111
<b>Figura 77.</b> Dato de la placa recibido desde el sistema al servidor .....	111
<b>Figura 78.</b> Historial de trabajos previos realizados .....	112
<b>Figura 79.</b> Historial de trabajos previos realizados .....	112
<b>Figura 80.</b> Captura de imagen por movimiento .....	113
<b>Figura 81.</b> Datos de la placa detectados por el sistema.....	113
<b>Figura 82.</b> Envío de dato de la placa al servidor .....	114
<b>Figura 83.</b> Datos correctos enviados al servidor .....	114
<b>Figura 84.</b> Datos del vehículo mostrados por el algoritmo clasificador .....	115
<b>Figura 85.</b> Historial del vehículo mostrado automáticamente por el sistema de base de datos .....	115
<b>Figura 86.</b> Reconstrucción de letra V.....	116
<b>Figura 87.</b> Reconstrucción de número 2 .....	117
<b>Figura 88.</b> Reconstrucción de número 5 .....	117
<b>Figura 89.</b> Reconstrucción de letra C.....	118
<b>Figura 90.</b> Reconstrucción de letra P .....	118
<b>Figura 91.</b> Reconstrucción de número 8 .....	119
<b>Figura 92.</b> Reconstrucción de letra B.....	119
<b>Figura 93.</b> Reconstrucción de número 6 .....	120
<b>Figura 94.</b> Reconstrucción con imagen real letra T .....	120
<b>Figura 95.</b> Reconstrucción imagen real letra P, números 5,6.....	121
<b>Figura 96.</b> Representación de las pruebas correctas e incorrectas realizadas .....	122

## RESUMEN EJECUTIVO

El presente proyecto de investigación propone la implementación de un sistema de registro automático para optimizar y agilizar los procesos de administración en el Taller “Tecnics David” demostrando la relevancia de este trabajo en el campo de la automatización de servicios automotrices. La importancia del proyecto radica en la implementación de la Lógica Difusa como técnica de reconocimiento de caracteres y en la integración de diferentes tecnologías y herramientas para lograr un sistema completo y eficiente. El Sistema utiliza la Lógica Difusa como técnica de reconocimiento de caracteres y el aprendizaje automático para el manejo del historial de trabajos realizados. La técnica de la Lógica Difusa muestra la capacidad del sistema para manejar situaciones de incertidumbre o borrosidad en las imágenes de las placas. Esta técnica permite reconstruir y completar los caracteres de las placas en caso de que estén ilegibles o borrosos.

El sistema aprovecha las diferentes tecnologías y herramientas de código abierto. Se utilizó el lenguaje de programación Python, la biblioteca OpenCV para la detección de movimiento y bordes, y la Lógica Difusa para la extracción de datos de las imágenes de las placas. La Interfaz del sistema cuenta con un Dashboard realizado en Node-Red, con una visualización intuitiva y eficiente para la gestión de vehículos en el taller. Esta interfaz gráfica cuenta con los requerimientos específicos del taller, por lo que brinda una experiencia amigable al usuario y un registro completo de la información necesaria. El sistema de registro automático ha sido sometido a pruebas exhaustivas, demostrando su funcionamiento eficiente y confiable. Se ha evaluado la confiabilidad del sistema, logrando un alto porcentaje de reconocimiento de placas vehiculares mediante la Lógica Difusa con un 90.90% de efectividad, obteniendo un bajo margen de error, lo que indica una precisión satisfactoria en la detección y registro de vehículos. agilizando y mejorando el proceso de gestión de vehículos en el taller automotriz "Tecnics David".

**Palabras clave:** Lógica difusa, reconocimiento, binarización, fuzziificación

## ABSTRACT

The present research project proposes the implementation of an automatic registration system to optimize and streamline administrative processes in "Técnicars David" Workshop, demonstrating the relevance of this work in the field of automotive service automation. The importance of the project lies in the implementation of fuzzy logic as a character recognition technique and the integration of different technologies and tools to achieve a complete and efficient system. The system utilizes fuzzy logic as a character recognition technique and machine learning for managing the history of jobs performed. The fuzzy logic technique demonstrates the system's ability to handle situations of uncertainty or blurriness in plate images, allowing for the reconstruction and completion of characters in case they are illegible or blurry.

The system leverages various open-source technologies and tools. It uses the Python programming language, the OpenCV library for motion and edge detection, and fuzzy logic for data extraction from plate images. The system interface features a dashboard built on Node-Red, providing an intuitive and efficient visualization for vehicle management in the workshop.

This graphical interface is tailored to the specific requirements of the workshop, offering a user-friendly experience and comprehensive registration of necessary information. The automatic registration system has undergone rigorous testing, demonstrating efficient and reliable operation. The system's reliability has been evaluated, achieving a high percentage of vehicle plate recognition using fuzzy logic with 90.90% effectiveness. The system has also achieved a low error rate, indicating satisfactory precision in vehicle detection and registration. This has resulted in improved efficiency and enhanced vehicle management processes at "Técnicars David" workshop.

**Keywords:** Fuzzy logic, recognition, binarization, fuzzification.

# CAPÍTULO I

## MARCO TEÓRICO

### 1.1. Tema de Investigación

SISTEMA AUTOMÁTICO DE REGISTRO DE AUTOMÓVILES Y MANEJO DEL HISTORIAL DE TRABAJOS PARA LA MICROEMPRESA “TECNICARS DAVID” UTILIZANDO MACHINE LEARNING

#### 1.1.1 Planteamiento del problema

La tecnología se ha mantenido en constante evolución por lo que facilita y optimiza las actividades humanas diarias. El mundo está interconectado y la tecnología digital puede ser tan beneficiosa para el desarrollo humano. Vivimos en una era tecnológica que está transformando las sociedades a una velocidad insospechada y, al mismo tiempo, logra avances sin precedentes en diferentes sectores. En el ámbito empresarial nace la necesidad de contar con soluciones para mejorar la producción. El número de empresas a nivel mundial que utiliza tecnologías de automatización inteligente en su modelo de negocio sigue creciendo constantemente. Hasta el presente año se ha aumentado al 73% frente al 58% del 2022, según el estudio anual de Deloitte Automation with intelligence, en el que se ha entrevistado a 441 empresas de todo el mundo. En el informe consta que tres de cada cuatro empresas ya han incorporado tecnologías como la RPA (automatización de procesos mediante robots), el Machine Learning o aprendizaje automático, el procesamiento de lenguaje natural o el BPM (Gestión de Procesos de Negocio) [1].

Los países latinoamericanos están incorporando la automatización en las empresas para mejorar su producción. Brasil es uno de los países con mayor crecimiento en la implementación de la automatización, superando el 20%, lo que lo ubica en la cima de la lista de países de esta región. Seguidamente, con un 18% y 19% se encuentran Chile y Argentina, respectivamente. Por su parte Bolivia, Venezuela, Ecuador, Perú y

Uruguay han mostrado un aumento entre el 10% y 15% con un avance tecnológico significativo hasta el presente año. La automatización ha revolucionado el negocio de las empresas en Sudamérica, ayudando a reducir los costos generales y aumentar la productividad con consistencia y precisión. La Inteligencia Artificial juega un papel importante, actualmente impulsa un crecimiento continuo en el sector empresarial sudamericano. Siendo uno de estos, los sistemas automáticos de registro vehicular con Inteligencia Artificial mediante detección de placas que facilitan la logística de empresas y han venido tomando mayor importancia con el pasar de los años, debido al incremento de eficiencia en la productividad de las empresas en Latinoamérica [2].

En Ecuador se está estimando la factibilidad técnica de automatización en diferentes sectores empresariales, así por ejemplo, en el sector automotriz no existen empresas que tengan procesos de fabricación, ensamble o provean servicios altamente especializados, todas las actividades que se realizan aún requieren de un alto grado de intervención humana, lo cual aumenta el riesgo de cometer errores en las mismas. Por lo que existe la posibilidad de desarrollar localmente metodologías que puedan realizar controles para disminuir los errores humanos y de esta manera brindar servicios de calidad a los usuarios finales. Las microempresas del Ecuador dedicadas al mantenimiento automotriz no poseen sistemas de registro automáticos y dependen totalmente del personal encargado que permite el acceso a sus instalaciones, lo cual en muchos casos se realiza sin el control efectivo de los vehículos que ingresan. Además, si se produce la ausencia del personal encargado, el sistema se vuelve ineficiente causando incomodidad en los clientes. En la actualidad las pequeñas empresas necesitan sistematizar procesos comunes como el control de registros, control de inventarios, proformas, facturas, control de actividades de una empresa, entre otras, con el fin de reducir el tiempo que implica realizarlo de forma manual; lo que permitiría agilizar la entrega de información tanto para los clientes como para la propia administración de las empresas [3]. Las microempresas que están en constante crecimiento, tal es el caso del taller de servicio automotriz “Técnicars David” que posee convenios de trabajo con instituciones públicas del estado como el MIES y el IESS, recientemente obtuvo un convenio de trabajo con las Fuerzas Armadas para realizar trabajos a el parque automotor de dicha institución, es así como nace la

necesidad de mejorar los procesos administrativos del taller automotriz implementando la tecnología como lo es la automatización.

## **1.2 Antecedentes investigativos**

Se dispone de investigaciones previas realizadas en los últimos años como base inicial para el desarrollo de este proyecto. Estos estudios están estrechamente vinculados con el tema y se detallan a continuación:

Amendaño Aucatoma Santiago Vladimir y Condolo Ortiz Stalin Santiago en el año 2022 de la Universidad de las Fuerzas Armadas, presentaron el trabajo titulado “Prototipo de un sistema recomendador basado en Lógica Difusa para evaluar Obras de Relevancia de la Universidad de las Fuerzas Armadas ESPE”, el sistema está enfocado en automatizar la gestión administrativa con el fin de ofrecer atención efectiva y eficiente al personal académico que solicita evaluación de sus obras. El trabajo contribuye con el desarrollo de una aplicación web para agilizar el proceso de evaluación de obras basado en la normativa, en los procesos definidos, y en los informes de evaluadores. Su implementación se basó en la metodología Scrum y herramientas como Axure RP, .NET, SQL Server, React, Yarn y NodeJS. El software permite registrar en línea la solicitud y enviar la documentación de respaldo en forma digital para su análisis; asigna evaluadores internos y externos, envía y recibe informes, genera reportes, lleva un control automático y el solicitante puede verificar el estado de su trámite. Se creó un módulo recomendador, que mediante inferencia difusa sugiere la aprobación o no de la obra evaluada en base a las reglas definidas. Con el apoyo de miembros de la comisión se probó el aplicativo registrando expedientes seleccionados. Los resultados mostraron una reducción en los errores humanos o de comunicación, reducción de tiempo en la resolución, obtención de informes en forma ágil, permitir el seguimiento del trámite y disminuir uso de documentos en físico [4]

Navas Núñez Lourdes Gianella y Obando Reinoso Christopher Sebastián en el año 2021 de la Universidad de las Fuerzas Armadas, desarrollaron el proyecto titulado “Diseño e implementación de un prototipo de sistema de seguridad integral con el fin monitorear el acceso de automóviles, utilizando visión artificial y chatbot para el

ingreso al conjunto residencial”, el dispositivo está enfocado en monitorear el acceso de automóviles, utilizando Visión Artificial junto con la Red Neuronal DetectNet; de esta manera se enmarca la zona de interés obteniendo el par de coordenadas (x, y), con las cuales se compara la posición para implementar un “sensor” por medio de Visión Artificial, permitiendo capturar la imagen al pasar por una zona delimitada. El sistema utiliza una luminaria led de luz blanca LEDVANCE OSRAM Floodlight 10W, seleccionada según la norma UNE EN-13201 junto con una cámara de seguridad IP Hikvision. Los algoritmos se implementaron y compilaron con OpenCV en Python3, lo cual hizo posible el diseño de una interfaz gráfica bajo la norma ISO 9241. En este trabajo, se realizó la validación de la hipótesis por el método de T-Student, afirmando que el diseño e implementación de un prototipo de sistema de seguridad integral permite controlar y monitorear el ingreso de automóviles al conjunto residencial, utilizando Visión Artificial y Chatbot con un porcentaje de transformaciones OCR afirmativas mayor o igual al 95% [5]

Luis Manuel Márquez Rodríguez en el año 2019 de la Universidad Nacional Abierta y a Distancia, UNAD, desarrolló un trabajo de “Diseño e implementación de un software de reconocimiento de placas vehiculares en tiempo real”. El sistema está enfocado en el reconocimiento de placas vehiculares a partir de imágenes en tiempo real desarrollando una interfaz gráfica bajo Java y una librería de reconocimiento de caracteres, como apoyo a labores de vigilancia, búsqueda y ubicación del vehículo. Para el sistema detector y lector de matrículas vehiculares se utilizó la Visión Artificial, además de un algoritmo de reconocimiento óptico de caracteres, con la librería de código abierto OpenCV basada en C++ y para la extracción y levantamiento de las imágenes se utilizó el detector SURF (Speeded Up Robust Features), basado en un sistema transformador de la información SIFT (Scale Invariant Feature Transform). Los resultados obtenidos mediante pruebas de funcionalidad arrojaron datos favorables con el 88,5% de efectividad en el reconocimiento de la placa, siendo este un sistema compuesto con una cámara de resolución media; sin embargo, para una imagen de mayor claridad se recomienda la adopción de cámaras de altas resoluciones, captadoras de imagen aun en movimiento. En resumen, la calidad utilizada es aceptable para pequeños negocios y empresas [6].

Darwin Darío Espinoza Saquicela y Christian Antonio Salinas Escobar en el año 2019 de la Universidad del Azuay realizaron el trabajo titulado “Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares”. Este proyecto trata sobre la implementación de un Sistema de Reconocimiento de Placas Vehiculares – LPR basado en el análisis de imágenes digitales. Adicionalmente desarrollo de una interfaz gráfica WEB para la administración de usuarios. El proyecto utilizó Tesseract, más conocido como OCR (Optical Character Recognition), como método principal para el reconocimiento de caracteres, con redes neuronales (ANN) y algoritmos (KNN).

Y para el análisis requerido en la detección de placas, el sistema utilizó librerías provistas en el paquete de visión por computador OpenCV. Este sistema fue evaluado en un conjunto de más de 200 imágenes reales, obtenidas en diferentes horarios del día y bajo distintas condiciones de iluminación. Obteniendo así una tasa de reconocimiento de 91 % aproximadamente, dentro del cual se considera aquellas placas que incluso serían difíciles de leer por un ser humano, sea por deterioro, mala ubicación, pintura discordante a la normativa vigente, etc. [7]

Milton Daniel Torres Carrera en el año 2020 de la Universidad Politécnica Salesiana de Quito realizó el trabajo cuyo tema es “Reconocimiento Automático de la placa de un vehículo de Ecuador”. En el trabajo desarrolla un sistema que tiene como objetivo el reconocimiento de placas de vehículos de Ecuador, usando técnicas de Visión por Computador. Como parte del pre-procesamiento de las imágenes se ha realizado redimensionamiento y binarización sobre las imágenes originales. El sistema identifica mediante tecnología RFID, OCR y LPR el paso de los vehículos utilizando el software Intellisoft Parking que permite el reconocimiento de placas vehiculares. Para la segmentación de la imagen se utilizó detección de bordes, componentes conectados y un algoritmo preliminar para la identificar la ubicación de las letras y los números de la placa. El reconocimiento automático se realiza mediante dos redes neuronales sobre las secciones consideradas como “caracteres”, teniendo en cuenta las particularidades de las placas ecuatorianas. Se realizaron pruebas de funcionamiento arrojando como resultado que los errores obtenidos bordean el 3% [8].

### **1.3 Fundamentación teórica**

En Ecuador el parque automotor tuvo un incremento de 1.4 millones de vehículos en la última década, según datos del INEC, lo que ha generado una alta demanda de talleres automotrices de mantenimiento y reparación, los cuales han implementado nuevos procesos de mantenimiento, especializándose en diferentes áreas de servicio, siendo poco lo que se conoce de la automatización en los servicios automotrices frente a los requerimientos y exigencias del cliente. El desarrollo de la tecnología automotriz en las últimas décadas ha dado un impulso notable para el crecimiento del parque automotor. Así, la incorporación de dispositivos electrónicos y mecánicos en los vehículos responde a un patrón de exigencias del consumidor actual, quienes apegados a leyes vigentes de seguridad y cuidado ambiental procuran automóviles cada vez más sofisticados produciendo así una obligatoria necesidad de innovación tecnológica por parte de los centros de servicio automotriz, para mantenerse competitivos dentro del mercado, para que estos talleres logren un desarrollo continuo ofreciendo servicios de calidad y estandarizando procesos [9]

La microempresa Tecnicars David, ubicada en la ciudad de Ambato, en el sector Huachi Chico, entre las calles Carlos Rubira Infante y Miguel Ángel Cáceres es un centro de servicio automotriz multimarca. Su trabajo consiste en el mantenimiento, reparación y revisión de motores, frenos, cajas automáticas, así también como de sistemas de suspensión, inyección electrónica y electricidad automotriz. Posee convenios con instituciones públicas del estado, por lo que realizan mantenimiento a los vehículos de dichas instituciones. Actualmente el control de acceso de entrada de los vehículos es deficiente, además de no poseer un sistema de registro digital automático y un programa que facilite los procesos administrativos de trabajo de la empresa. El problema detectado es que se lleva de manera manual los registros de los trabajos realizados de cada vehículo. El centro de servicio automotriz está creciendo y necesita un sistema automático de registro, un software que facilite el manejo del historial y los trabajos a realizarse en cada vehículo para la correcta organización y funcionamiento de la microempresa.

### **1.3.1 Reconocimiento de placas**

El reconocimiento de caracteres e imágenes se lo realiza utilizando diferentes métodos tales como Visión Artificial, Redes Neuronales, métodos estructurales basados en la apariencia y otros. El problema es el reconocimiento automático de caracteres en las placas o matrículas de los vehículos. El algoritmo que se propone en el trabajo de investigación adaptará la cantidad y forma de caracteres utilizando la Lógica Difusa y la librería OPEN CV para el reconocimiento de las imágenes. El proceso para el reconocimiento automático de caracteres de las matrículas de un vehículo es el siguiente: captura o adquisición de la imagen, mejoramiento de la imagen, extracción de la región donde se encuentra la placa. Luego se realiza la aplicación del reconocimiento de caracteres [10].

### **1.3.2 Machine Learning**

El Machine Learning o aprendizaje automático es un campo de la Informática que tiene como objetivo enseñar a las computadoras cómo aprender y actuar sin estar programado explícitamente. Más específicamente, el aprendizaje automático es un enfoque para el análisis de datos que implica la creación y adaptación de modelos, que permiten a los programas aprender a través de la experiencia. El aprendizaje automático implica la construcción de algoritmos que adaptan sus modelos para mejorar su capacidad de realizar predicciones [11].

### **1.3.3 Tipos de Machine Learning**

Todos los algoritmos de aprendizaje automático tienen como objetivo aprender y mejorar su precisión a medida que procesan más conjuntos de datos. La forma en que se pueden clasificar las tareas que resuelven los algoritmos de aprendizaje automático es por la cantidad de retroalimentación que presentan al sistema. En algunos escenarios, se proporciona a la computadora una cantidad significativa de datos de entrenamiento etiquetados, lo que se denomina aprendizaje supervisado. En otros casos, no se proporcionan datos etiquetados y esto se conoce como aprendizaje no supervisado. Por último, en el aprendizaje semi-supervisado, se proporcionan algunos

datos de entrenamiento etiquetados, pero la mayoría de los datos de entrenamiento no están etiquetados [11].

#### **1.3.4 Aprendizaje Supervisado**

El aprendizaje supervisado es la forma de aprendizaje automático más práctica y ampliamente adoptada. Implica la creación de una función matemática que relaciona las variables de entrada con las variables de salida preferidas. Se proporciona una gran cantidad de conjuntos de datos de entrenamiento etiquetados que proporcionan ejemplos de los datos que la computadora procesará [12]. Las tareas de aprendizaje supervisado se pueden presentar además como problemas de clasificación o regresión. Los problemas de clasificación utilizan métodos de agrupación estadística para generar una categorización, mientras que los problemas de regresión utilizan el análisis de regresión estadística para proporcionar resultados numéricos [12].

#### **1.3.5 Aprendizaje semi-supervisado**

El aprendizaje semi-supervisado es en realidad lo mismo que el aprendizaje supervisado, excepto que, de los datos de capacitación proporcionados, solo se etiqueta una cantidad limitada. El reconocimiento de imágenes es un buen ejemplo de aprendizaje semi-supervisado. En este ejemplo, podríamos proporcionar al sistema varias imágenes etiquetadas que contienen objetos que deseamos identificar y luego procesar muchas más imágenes sin etiquetar en el proceso de entrenamiento [12].

#### **1.3.6 Aprendizaje sin supervisión**

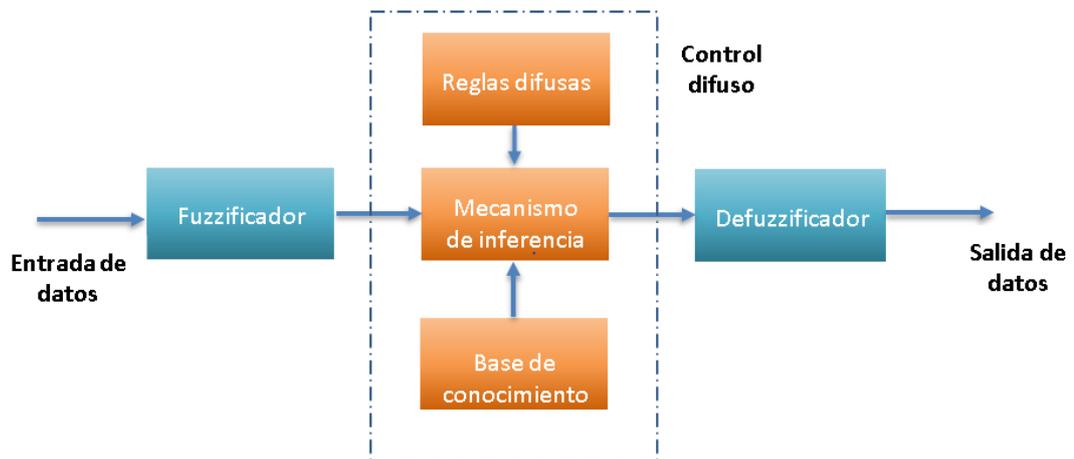
En el aprendizaje no supervisado, todas las entradas no están etiquetadas y el algoritmo debe crear una estructura a partir de las entradas por sí solo. Los problemas de agrupación o problemas de análisis de agrupación son tareas de aprendizaje no supervisadas que buscan descubrir agrupaciones dentro de los conjuntos de datos de entrada [12].

### 1.3.7 Lógica Difusa

La Lógica Difusa permite representar el conocimiento común, que es mayoritariamente del tipo lingüístico cualitativo y no necesariamente cuantitativo, en un lenguaje matemático a través de la teoría de conjuntos difusos y funciones características asociadas a ellos. Permite trabajar a la vez con datos numéricos y términos lingüísticos; los términos lingüísticos son inherentemente menos precisos que los datos numéricos, pero en muchas ocasiones aportan una información más útil para el razonamiento humano. El aspecto de los sistemas basados en la teoría de la Lógica Difusa es que tienen la capacidad de reproducir aceptablemente los modos usuales del razonamiento, considerando que la certeza de una proposición es una cuestión de grado.

En la figura 1 se puede apreciar el esquema del funcionamiento general de la Lógica Difusa, empezando desde la parte derecha de la figura con la entrada de datos avanzando hacia todo el control difuso y finalizando con la salida de datos [13].

Esquema del funcionamiento de la Lógica Difusa



**Figura1.** Esquema de modelado de la Lógica Difusa [13]

Las particularidades más importantes de la Lógica Difusa son:

La flexibilidad, la tolerancia con la imprecisión, la capacidad para moldear problemas no-lineales y su fundamento en el lenguaje de sentido común [14].

Con la utilización de la Lógica Difusa se podrá resolver procesos muy complejos para el ser humano; es decir, cuando hace falta un modelo matemático simple o también para procesos altamente no lineales [14].

### **1.3.8 Principio de incompatibilidad**

Este principio nos indica que el comportamiento de un sistema complejo no se realiza de forma exacta, es decir, conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes. Por lo que nace la necesidad de obtener herramientas mediante el cual se pueda controlar de manera rigurosa y confiable la información que no esté clara [14].

### **1.3.9 Representación de la información imprecisa:**

Esto conlleva la utilización de la Teoría de Conjuntos Difusos. De esta manera se puede describir mediante la práctica a los sistemas complejos en las relaciones de entrada y salida de información, a través de proposiciones condicionales que son: Si-Entonces, por ejemplo, Si la Obra evaluada cumple con todos los requisitos Entonces se considera como una Obra de Relevancia. De esta forma las variables que se tiene de entrada y las de salida tiene una misma correspondencia y sentido común [14].

### **1.3.10 Inferencia sobre información imprecisa:**

La Regla Composicional de Inferencia es el método de inferencia generalizado que surge de la necesidad de combinar información para generar nuevos hechos [14].

### **1.3.11 Regla Composicional de Inferencia**

Este principio permite describir las características más fundamentales de la Lógica y los Sistemas Difusos [14].

**Tabla 1.** Características principales de la Lógica Difusa. [14]

<b>Características</b>	<b>Descripción</b>
<b>Razonamiento Exacto:</b>	¿La aplicación de la Lógica Difusa permite fuzzificar cualquier sistema lógico, lo cual significa que se puede incorporar el conocimiento humano de manera sistemática, incluso dentro de sistemas de ingeniería [15].
<b>Restricciones difusas:</b>	Los sistemas difusos resultan especialmente útiles y de gran interés cuando nos enfrentamos a situaciones en las que obtener un modelo exacto del sistema resulta difícil [15].
<b>La inferencia:</b>	El proceso de difusión de las restricciones difusas se percibe como una forma de difundir o propagar estas restricciones en un sistema [15].
<b>Ayuda a la decisión:</b>	La Lógica Difusa es capaz de facilitar la toma de decisiones incluso cuando se cuenta con valores incompletos o información imprecisa [15].

**Elaborado por:** El investigador

### 1.3.12 Conjuntos difusos

Esta teoría está estrechamente relacionada a los valores de pertenencia; es decir, definido entre pertenencia total (valor uno) o no pertenencia (valor cero). El proceso que identifica el grado de pertenencia (uno o cero) es denominado Fuzzificación. Al tener los valores fuzzificados se da paso a implementar reglas lingüísticas y así tener una salida, la cual puede ser Defuzzificada obteniendo un valor discreto crisp [15].

### 1.3.13 Función de Pertenencia

Los conjuntos difusos se los cataloga de forma general como conjunto con límites difusos. Siendo X el Universo, y el resto de los elementos como  $x$ . Así se define sobre el conjunto C sobre X con una función característica de C como  $f_C(x)$  [15].

$$f_C(x) = \begin{cases} 1 & \text{cuando } x \in C \\ 0 & \text{cuando } x \notin C \end{cases}$$

Esta función es un mapeo del conjunto de dos elementos, donde  $f_C(x)$  es 1 si  $x$  está en el conjunto C y 0 si  $x$  no está en el conjunto C. Por ende, para encontrar el grado de pertenencia de un determinado conjunto difuso se aplica la siguiente función:

$$\mu_A = X \rightarrow [0,1].$$

$$\mu_A(x) = 1, \quad x \text{ está totalmente en } A$$

$$\mu_A(x) = 0, \quad x \text{ no está en } A$$

$$0 < \mu_A(x) < 1, \quad \text{si } x \text{ está parcialmente en } A.$$

### 1.3.14 Razonamiento aproximado

Con la utilización de conjuntos difusos se puede dar un significado matemático a proposiciones como “este auto es grande” o “Juan es un gordo”, se utiliza modificadores lingüísticos (muy, demasiado, algo, extremadamente etc.) adaptando calificativos que ayuden a comprender lo que se quiere decir. Cuando hay un gran número de hechos y reglas difusas, el sistema verifica toda esa información en una etapa llamada razonamiento [15].

### **1.3.15 Reconocimiento y procesado digital de imágenes**

Es el conjunto de técnicas englobadas dentro del preprocesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de esta, que posibilite efectuar operaciones para el posterior análisis y extracción de información. Se usan con distintas finalidades, destacando entre otras la detección de bordes, la segmentación de imágenes mediante el reconocimiento de líneas rectas y curvas, análisis del histograma, mejora del contraste, eliminación del ruido o tratamiento de regiones mediante el uso de operadores morfológicos [16]

### **1.3.16 Reconocimiento óptico de caracteres**

El reconocimiento óptico de caracteres comúnmente conocido por el acrónimo OCR, es el proceso dirigido a la digitalización de textos, los cuales se identifican automáticamente a partir de imágenes patrón de símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenar sus propiedades de identificación en forma de datos [17].

Todos los algoritmos de ROC tienen la finalidad de poder diferenciar un texto de una imagen cualquiera [17].

Para hacerlo se basan en cuatro etapas:

1. Binarización o caracterización.
2. Fragmentación o segmentación de la imagen.
3. Adelgazamiento de los componentes.
4. Comparación con patrones.

### **1.3.17 Binarización**

Los algoritmos de ROC parten como base de una imagen binaria (dos colores). Por lo tanto, es conveniente convertir una imagen de escala de grises, o una de color, en una imagen en blanco y negro, de tal forma que se preserven las propiedades esenciales de

la imagen. Una forma de hacerlo es mediante el histograma de la imagen, donde se muestra el número de píxeles para cada nivel de grises que posee la imagen. Para binarizarla tenemos que escoger un umbral adecuado, a partir del cual todos los píxeles que no lo superen se convertirán en negro y el resto en blanco. Mediante este proceso obtenemos una imagen en blanco y negro donde quedan claramente marcados los contornos de los caracteres y símbolos que contiene la imagen. A partir de aquí podemos aislar las partes de la imagen que contienen texto (más transiciones entre blanco y negro) [17].

### **1.3.18 Fragmentación o segmentación de la imagen**

Este es el proceso más costoso y necesario para el posterior reconocimiento de caracteres. La segmentación de una imagen implica la detección mediante procedimientos de “etiquetado determinista” o estocástico de los contornos o regiones de la imagen, basándose en la información de intensidad o información espacial. Permite la descomposición de un texto en diferentes entidades lógicas, que han de ser suficientemente invariables, para ser independientes del escritor, y suficientemente significativas para su reconocimiento [18].

No existe un método genérico para llevar a cabo esta segmentación de la imagen que sea lo suficientemente eficaz para el análisis de un texto. Aunque las técnicas más utilizadas son variaciones de los métodos basados en proyecciones lineales. Una de las técnicas más clásicas y simples para imágenes de niveles de grises consiste en la determinación de los modos o agrupamientos (clústeres) a partir del histograma, de tal forma que permitan una clasificación o umbralización de los píxeles en regiones homogéneas [18].

### **1.3.19 Proceso de Adquisición de Imágenes**

Consiste en la captación de imágenes en línea mediante cámaras basadas en matrices de sensores sensibles a la luz (CCD o CMOS), el posterior tratamiento de estas mediante técnicas de análisis de imagen) [19].

### 1.3.20 Elementos para captar la imagen

Los elementos básicos son los siguientes:

El sistema de iluminación (fluorescente, LED, polarizada, backligh, láser, luz natural)

El sensor o cámara de captura de imagen (CMOS), CCD, InGaAs).

Procesamiento de imagen (Procesador y algoritmos de pretratamiento y filtrado de la imagen, de segmentación y reconocimiento de formas, de extracción de descriptores y de clasificación)

Sincronía con el proceso, para adquirir la imagen en el momento adecuado y para actuar con el proceso o separar unidades defectuosas.

### 1.3.21 Proceso de Adquisición de la Imagen:

**Captación:** Proceso de obtener una imagen. El proceso de captura está centrado en tres aspectos importantes, necesarios para obtener una imagen de calidad que no altere las condiciones reales agregando errores considerables al sistema [19].

**Pre procesamiento:** Métodos para quitar o reducir características no deseadas en la imagen como el ruido. El preprocesamiento consiste en la aplicación de técnicas que permitan el realce o mejoramiento de algunas características importantes en las imágenes originales para facilitar el proceso de segmentación [19].

**Segmentación:** Dividir una imagen en objetos que sean de nuestro interés. La técnica de segmentación más conocida dentro del procesamiento de imágenes es quizás la umbralización. Este método permite separar dos o más regiones de una imagen a partir de un análisis del histograma [19].

**Descripción:** Obtención de características de un objeto como su forma, tamaño, color con el fin de diferenciarlo de otros objetos [19].

**Reconocimiento:** Proceso a través del cual se identifica un objeto dentro de una escena [19].

**Interpretación:** Asociar un significado a un conjunto de objeto reconocidos [19].

### **1.3.22 Procesamiento de imágenes y análisis de placas vehiculares**

El procesamiento de imágenes desempeña un papel fundamental al mejorar diversos aspectos y resaltar detalles importantes que se desean analizar con mayor eficiencia. Las imágenes pueden generarse a través de fotografías, monitores de televisión, cámaras de video, entre otros. Este procesamiento se lleva a cabo mediante algoritmos y operadores avanzados, los cuales gestionan la estructura de la imagen con el objetivo de mejorar su calidad y facilitar la extracción de información en el contexto de imágenes gráficas visuales.

### **1.3.23 Técnicas de Procesamiento y Corrección de Imágenes**

En el ámbito del procesamiento y corrección de imágenes, existen conjuntos de algoritmos que se dedican a la segmentación, es decir, a la tarea de separar los objetos específicos de interés del resto de la imagen [20].

Utilizando estos métodos, es posible identificar los píxeles que componen los objetos de interés y distinguir aquellos que no se desean en la imagen. Además, se puede mejorar la calidad de la imagen con el fin de obtener fotografías que brinden mejores resultados en el proceso de reconocimiento. A continuación, se detallarán las técnicas que se emplearán en la tesis: Suavizado (Smoothing), Relleno de área (Flood Fill), Redimensionamiento (Resize) y Umbralización (Threshold).

#### **Smoothing**

El término "Suavizado" se utiliza comúnmente para referirse al desenfoque de una imagen y es una técnica de procesamiento de imágenes de fácil aplicación y uso frecuente. Su objetivo principal es reducir el ruido presente en una fotografía tomada con una cámara. El Suavizado también resulta útil cuando se desea disminuir la resolución de una imagen, buscando obtener resultados óptimos para la visualización de esta [21].

## **CV\_BLUR**

Este operador, conocido como "Desenfoque simple" o "Simple blur", es compatible con imágenes que contengan de 1 a 4 canales y se aplica a imágenes de 8 bits o imágenes de punto flotante de 32 bits. Su funcionamiento consiste en tomar un píxel y calcular el promedio simple de todos los píxeles vecinos, asignando dicho valor al píxel de resultado [21].

## **CV\_BLUR\_NO\_SCALE**

Existen diferentes operadores de suavizado que pueden variar en la forma en que trabajan con diferentes clases de imágenes. Uno de ellos es el operador CV\_BLUR\_NO\_SCALE, el cual es similar al anteriormente mencionado, excepto que no se realiza una división al calcular el promedio. Debido a esto, las imágenes de origen y destino deben tener diferentes precisiones numéricas para evitar desbordamientos durante la operación de suavizado [21].

## **CV\_MEDIAN**

El filtro en cuestión sustituye cada píxel por el valor mediano de un área cuadrada que lo rodea. Este filtro puede ser aplicado a imágenes de un solo canal, tres canales o cuatro canales, siempre y cuando sean imágenes de 8 bits. El resultado es una imagen suavizada de forma simple promediada, pero es importante tener en cuenta que puede ser sensible a imágenes con mucho ruido, especialmente si presentan puntos atípicos aislados, a veces conocidos como "ruido de disparo" [21].

## **CV-GAUSSIAN**

El filtro gaussiano es considerado como uno de los más útiles, aunque no el más rápido. El proceso de filtrado gaussiano se lleva a cabo mediante la convolución de cada punto en la matriz de entrada con un núcleo gaussiano y luego sumando los resultados para generar la matriz de salida [21].

## **CV\_BILATERAL**

Este filtro se utiliza para suavizar una imagen y no se trata de una convolución en el sentido tradicional. Su objetivo principal es reducir la cantidad de colores presentes en una imagen, pero sin alterar los bordes abruptos que existen en la misma [21].

## **FUNCIÓN FLOOD FILL**

Esta función es altamente beneficiosa y se emplea con frecuencia para resaltar o aislar determinadas partes de una imagen con el fin de su posterior procesamiento o análisis. Además, se utiliza para generar máscaras a partir de una imagen de entrada, las cuales pueden ser empleadas en rutinas posteriores para agilizar o limitar el procesamiento únicamente a los píxeles indicados por la máscara [21]. El resultado de aplicar la operación de Relleno de Área (Flood Fill) siempre será una región continua y única. La función `cvFloodFill()` asigna un color a un píxel vecino si se encuentra dentro de un rango especificado, ya sea basado en el valor del píxel actual o si el píxel vecino se encuentra dentro de un rango específico del valor original [21].

En muchas ocasiones nos encontramos con la necesidad de convertir una imagen de un tamaño a otro. Es posible que se desee aumentar el tamaño de la imagen (zoom in) o reducirlo (alejarse), y para lograrlo se puede utilizar la función `cvResize()`. En general, el objetivo es que el mapeo de la imagen original a la imagen de destino redimensionada sea lo más suave posible y que los valores de los píxeles no se alteren. El argumento de interpolación controla específicamente cómo se manejará este proceso. La interpolación se presenta cuando se reduce una imagen y un píxel de la imagen de destino cae en medio de los píxeles de la imagen original [21].

## **THRESHOLD**

La función `cvThreshold()` opera utilizando una matriz y un umbral específico para llevar a cabo el proceso de binarización de una imagen. Puede aplicarse a imágenes en escala de grises, con inversión u otros enfoques, con el objetivo de obtener resultados óptimos. Durante este proceso, se verifica el estado de cada elemento de la matriz para determinar si está por encima o por debajo del umbral establecido [22].



**Figura3.** Imagen con Threshold . [22]



**Figura2.** Imagen Normal [22]

A continuación, se presentan tipos de operadores con los que trabaja Threshold

**Tabla 2.** Tipos de Operadores Threshold. [22]

Threshold type	Operation
CV_THRESH_BINARY	$Dst, = (src, > T) ? M : 0$
CV_THRESH_BINARY_INV	$Dst, = (src, > T) ? 0 : M$
CV_THRESH_TRUNC	$Dst, = (src, > T) ? 0 : M : src$
CV_THRESH_TOZERO_INV	$Dst, = (src, > T) ? 0 : src$
CV_THRESH_TOZERO	$Dst, = (src, > T) ? src : 0$

**Elaborado por:** El Investigador

### 1.3.24 Revisión y selección de librerías para realizar el procesamiento de imágenes

Hay diferentes bibliotecas disponibles que han sido desarrolladas para realizar el procesamiento de imágenes a través del reconocimiento óptico de caracteres. Algunas de estas bibliotecas incluyen Torch3vision, Vlx, LTI-Lib, Opencv, entre otras. A continuación, se detallará cada una de ellas para evaluar y seleccionar la herramienta más adecuada para utilizar en el contexto específico.

## Librerías

### TORCH3VISIÓN

Esta biblioteca, implementada en C++, está destinada a realizar el procesamiento de imágenes y cuenta con algoritmos para la extracción de características básicas de imágenes. Además, ofrece funcionalidades como rotación, volteo, detección de bordes y decodificación/codificación de archivos de video. También permite la captura de imágenes desde tarjetas o dispositivos USB. Otro aspecto destacado es su capacidad para leer y escribir en diversos formatos de imágenes gráficas, como pgm, ppm, gif, tif y jpeg [23].

### VLX

VLX es un conjunto de bibliotecas desarrollado en C++. A diferencia de ser una única biblioteca, VLX ofrece una funcionalidad completa al integrar varias bibliotecas. Fue creado con el objetivo de proporcionar un sistema rápido y estable. Una de las características más destacadas de VLX es su capacidad de utilizar solo las bibliotecas que sean necesarias, sin tener dependencias obligatorias entre ellas. Esto permite utilizar únicamente las bibliotecas que sean relevantes y de utilidad para un proyecto en particular [24].

Las bibliotecas del núcleo en VXL son:

**VNL (numéricos):** contenedores numéricos y algoritmos, por ejemplo matrices, vectores, descomposiciones, optimizadores.

**VIL (imágenes):** Cargar, guardar y manipular imágenes en muchos formatos de archivo comunes, incluyendo imágenes muy grandes.

**VGL (geometría):** Geometría de puntos, curvas y otros objetos elementales en 1, 2 o 3 dimensiones.

## **LTI-LIB**

El LTI-Lib es una biblioteca enfocada en objetos que contiene algoritmos y estructuras de datos comúnmente utilizados en el procesamiento de imágenes. Fue desarrollada en la Cátedra de Informática Técnica (Lehrstuhl fuer Technische Informatik) LTI en la Universidad de Tecnología de Aachen. Esta biblioteca forma parte de varios proyectos de investigación relacionados con la visión por computadora y la robótica [25].

Entre las diversas funcionalidades que ofrece, se encuentran el reconocimiento de objetos, el reconocimiento de lenguaje y gestos. Con más de 500 clases disponibles, incluye opciones para álgebra lineal, clasificación, procesamiento de imágenes y herramientas de visualización de dibujo. Gracias a su arquitectura orientada a objetos, simplemente es necesario crear un objeto visor y proporcionarle la imagen que se desea mostrar [25].

## **OPENCV**

OpenCV es una biblioteca de procesamiento de imágenes de código abierto desarrollada por Intel. Es una de las librerías más reconocidas, completas y activas en la actualidad. Con más de 500 algoritmos disponibles, ofrece una amplia gama de funciones para el procesamiento de imágenes, descripciones geométricas, segmentación, reconocimiento facial, calibración de cámaras, visión estéreo, visión robótica y seguimiento. Además, proporciona la capacidad de aprovechar las capacidades de cómputo de manera adicional [26].

OpenCV ha sido creado para ser utilizado en diversas aplicaciones, que incluyen el procesamiento de imágenes y análisis, análisis estructural, análisis de movimiento, reconocimiento de modelos, reconstrucción 3D, así como también para el desarrollo de interfaces gráficas y la adquisición de datos.

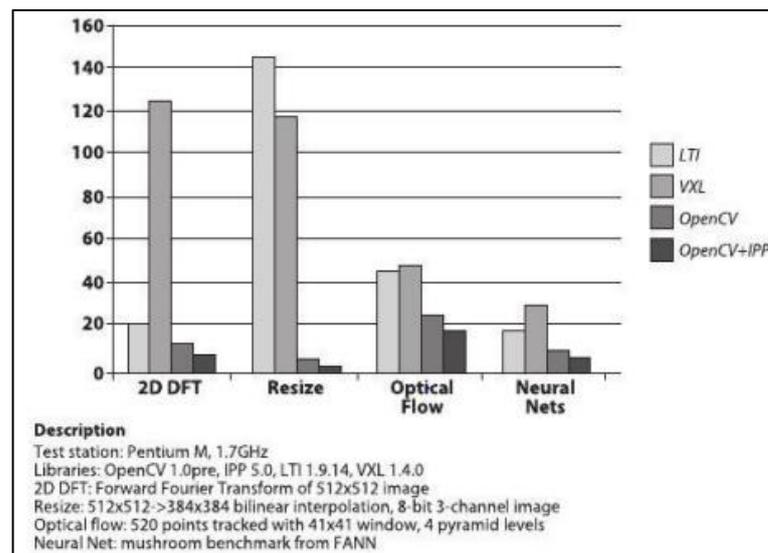
OpenCV cuenta con una amplia cantidad de documentación, que incluye libros y una gran cantidad de ejemplos que son fácilmente accesibles y sirven como guía para el desarrollo de proyectos. Además, esta biblioteca es multiplataforma, lo que significa que está disponible para sistemas operativos como Linux, Windows y Android.

Además, proporciona soporte para varios lenguajes de programación, como C++, C, Python y Java, lo que brinda flexibilidad a los desarrolladores para elegir el lenguaje que mejor se adapte a sus necesidades [26].

### 1.3.26 Comparativas y elección de la herramienta

Con el fin de realizar una comparación más objetiva entre las distintas bibliotecas, se llevó a cabo una evaluación de la velocidad de cálculo utilizando algoritmos comunes. Estos incluyeron tareas como cambiar el tamaño de una imagen y realizar un Optical Flow para el seguimiento de 520 puntos [26].

El gráfico a continuación representa los resultados, donde los valores más altos indican un mayor tiempo de procesamiento.



**Figura4.** Resultados de tiempo de procesamiento de las librerías. [26]

La biblioteca OpenCV tiene un mayor rendimiento en operaciones como la DFT, redimensionamiento, flujo óptico y redes neuronales debido a su implementación optimizada de algoritmos, soporte de hardware y su enfoque en el procesamiento de imágenes en tiempo real. Esto la convierte en una opción popular para el procesamiento de imágenes en diversas aplicaciones. En la figura 4 se observa los resultados que son claros y contundentes: OpenCV y OpenCV+IPP son los ganadores indiscutibles. En base a la investigación y exposición realizada, se concluye de manera

inequívoca que para el desarrollo de este proyecto de tesis, se utilizará OpenCV como la librería principal.



**Figura5.** Librería a utilizar OPEN CV. [26]

### **1.3.26 Placas Vehiculares en el Ecuador**

En Ecuador, al igual que en cualquier otra parte del mundo, las placas de los vehículos son utilizadas como un medio de registro para identificarlos y permitir su circulación legal en todo el país. En esta sección se explorarán las características de las placas, las cuales deben ser tomadas en cuenta al desarrollar un sistema que permita la detección y reconocimiento efectivo de los caracteres en ellas.

### **1.3.27 Características de las placas**

En Ecuador, al momento de registrar un vehículo, la Agencia Nacional de Tránsito (ANT) es la encargada de proporcionar las placas de identificación. Esta entidad otorga dos placas metálicas que deben colocarse en la parte frontal y posterior del automóvil. Estas placas están compuestas por tres letras y tres o cuatro dígitos, con rangos que van desde 000 hasta 9999, siguiendo los formatos ABC-123 y ABC-1234. El color de las letras en la placa debe ser negro mate, y en la parte superior izquierda se imprimirá en color el logotipo de la ANT. Además, las placas contarán con un relieve de 2mm para mejorar su visibilidad, cumpliendo así con las normas de seguridad. Asimismo, se aplicará una capa de pintura de laca anticorrosiva para proteger las placas [27].

El diseño de las placas es exclusivo en todo el territorio nacional. Las letras impresas en ellas deben tener un ancho de 38 mm y una altura de 70 mm. En la parte superior de la placa, en un campo de 146 mm de ancho por 27 mm de alto, se encontrará impresa la palabra "ECUADOR" en letras mayúsculas. El guión que separa la serie alfanumérica tendrá un ancho de 20 mm [27].



**Figura6.** Diseño de una placa en Ecuador. [27]

### **Dimensiones de las placas**

Tanto el reglamento de la Agencia Nacional de Tránsito (ANT) como el artículo de la ley orgánica de transporte terrestre establecen que todas las placas de vehículos deben ser de material metálico y cumplir con ciertas dimensiones. Estas placas deben tener una longitud de 404 mm y una altura de 154 mm, y además deben ser completamente reflectantes [20].

Sin embargo, en la provincia de Azuay aún existen placas en un tamaño distinto, ya que no han sido actualizadas, y tienen dimensiones de 305 mm de largo y 155 mm de alto en los vehículos, y no incluyen el logotipo de la ANT [27].



**Figura7.** Formato de Placa actual y placa antigua. [27]

### 1.3.28 Nomenclatura de Placas en Ecuador

Dentro de la nomenclatura de las placas, según la Agencia Nacional de Tránsito (ANT) en Ecuador, se establece que la primera letra de la placa indica la provincia a la que pertenece el vehículo, es decir, la provincia en la que se registró inicialmente. La segunda letra de la placa identifica el tipo de matrícula autorizada, de acuerdo con la función específica del vehículo. Por su parte, la tercera letra es correlativa. Un ejemplo de esta nomenclatura sería ABC-123 o PBC-1234 [27].

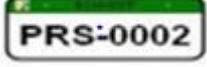
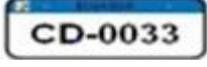
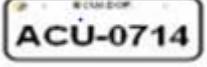
Además, se observa que, de acuerdo con el tipo de vehículo, las placas presentan diferentes colores. Con la modificación del reglamento de tránsito en Ecuador en junio de 2012, también se realizaron cambios en la forma de las placas. Específicamente, para los vehículos no particulares, las nuevas placas mantendrán el color distintivo, pero ya no se aplicará en toda la superficie de la placa, sino únicamente en el borde superior. El resto de la placa será de color blanco [27].



**Figura8.** Formato de placa de acuerdo con el nuevo reglamento 2012. [27]

Se implementó este cambio con el objetivo de mejorar la visibilidad de las placas, especialmente frente a cámaras y radares. Se espera que, en un lapso de 5 años, todas las placas sean reemplazadas por las nuevas. No obstante, las placas antiguas seguirán siendo válidas. Como resultado, se establece que el color de fondo de la placa variará según el tipo de servicio que el vehículo brinde [27].

**Tabla 3.** Modelos de color de placa de Instituciones Públicas del Estado

SERVICIO	COLOR	PLACA
PUBLICO O COMERCIAL	FRANJA NARANJA	
ORGANISMOS DEL ESTADO	FRANJA ORO	
GAD'S REGIONALES, PROVINCIALES, MUNICIPALES Y PARROQUIALES	FRANJA VERDE LIMÓN	
DIPLOMÁTICOS Y ORGANISMOS INTERNACIONALES	FRANJA AZUL	
INTERNACIÓN TEMPORAL	FRANJA ROJA	
PARTICULAR	COLOR BLANCO	
MOTOS	COLOR BLANCO	

**Elaborado por:** El investigador

De la misma forma se describe mediante la siguiente tabla que de acuerdo con el servicio que posee el automóvil [28].

**Tabla 4.** Servicio que posee el automóvil en Ecuador. [28]

<b>NUMERO</b>	<b>TIPO</b>	<b>LETRA</b>
<b>1</b>	Estado	E
<b>2</b>	Oficial	X
<b>3</b>	Consejo Provincial	S
<b>4</b>	Municipio	M
<b>5</b>	Cuerpo consular	CC
<b>6</b>	Cuerpo Diplomático	CD
<b>7</b>	Organismos Internacionales	OI
<b>8</b>	Asistencia Técnica	AT
<b>9</b>	Internacional Temporal	IT
<b>10</b>	Vehículos Comerciales	A,U,Z
<b>11</b>	Particulares	Cualquiera menos los anteriores

**Elaborado por:** El investigador

### **1.3.29 Tabla de Identificación por Provincias en la primera letra de la Placa Automovilística.**

Para que los vehículos puedan circular legalmente en el país, es necesario que se sometan a un proceso de matriculación previo para obtener una placa. Estas placas están categorizadas mediante las letras del abecedario, distribuidas por provincias donde se realiza la matriculación del vehículo [28].

**Tabla 5.** Identificación de letra por provincia. [28]

Azuay	A	Galápagos	W	Pastaza	S
Bolívar	B	Guayas	G	Pichincha	P
Cañar	U	Imbabura	I	Orellana	Q
Carchi	C	Loja	L	Sucumbíos	K
Cotopaxi	X	Los Ríos	R	Tungurahua	T
Chimborazo	H	Manabí	M	Zamora Chinchipe	Z
El Oro	O	Morona Santiago	V	Santa Elena	Y
Esmeraldas	E	Napo	N	Santo Domingo de los Tsáchilas	J

**Elaborado por:** El investigador

Las placas utilizadas en los vehículos pertenecientes a las Fuerzas Armadas del Ecuador comienzan con la letra "F" y suelen ir acompañadas de la letra "T" (para la Fuerza Terrestre), "N" (Fuerza Naval) o "AE" (Fuerza Aérea). Estas placas suelen tener el mismo color que las placas estatales [28].

En el pasado, los vehículos de la Policía Nacional del Ecuador utilizaban matrículas en las que la segunda letra era una "W". Estas matrículas tenían el mismo color que las placas de los vehículos particulares, es decir, blanco-plateado. Sin embargo, actualmente los nuevos vehículos de la Policía Nacional están siendo entregados con matrículas gubernamentales [28].

#### **Estandarización la numeración de las placas de transporte.**

En Ecuador, las placas de los automóviles están distribuidas de manera específica para evitar repeticiones. El sistema de placas vehiculares en Ecuador consta de tres componentes principales: letras, números y caracteres especiales. A continuación, se muestra cómo está organizada esta distribución:

**Prefijo Geográfico:** Las placas en Ecuador comienzan con un prefijo geográfico que identifica la provincia donde el vehículo está registrado. Cada provincia tiene asignada una combinación única de letras.

**Letras y Números:** Luego del prefijo geográfico, sigue una combinación de letras y números que sigue una secuencia específica. En este componente, se evita la repetición de secuencias ya utilizadas para garantizar que no se repitan placas dentro de una misma provincia.

**Caracteres Especiales:** Al final de la placa, se agrega un carácter especial como un número, una letra o un símbolo. Este componente adicional ayuda a aumentar la cantidad de combinaciones posibles, reduciendo aún más la probabilidad de repeticiones.

El sistema se basa en un algoritmo que asigna las secuencias de letras y números de manera que sean únicas y no se repitan dentro de una misma provincia. Además, este sistema está sujeto a regulaciones y cambios que pueden introducirse con el tiempo para adaptarse a las necesidades cambiantes del parque vehicular.

En resumen, la combinación de prefijos geográficos, secuencias de letras y números, y caracteres especiales permite distribuir de manera única las placas de los automóviles en Ecuador, minimizando la posibilidad de repeticiones y garantizando la identificación individual de cada vehículo.

### **ALGORITMO PARA ASIGNACION DE PLACAS**

El algoritmo que asigna las secuencias de letras y números de manera única y no repetitiva dentro de una misma provincia se basa en el concepto de permutaciones y combinaciones.

Supongamos que en una provincia determinada, se tiene un conjunto de letras {A, B, C, ..., Z} y un conjunto de números {0, 1, 2, ..., 9}. Además, se define una longitud fija para la placa, por ejemplo, de 7 caracteres.

La cantidad de combinaciones posibles de 7 caracteres tomados de los conjuntos de letras y números, sin repetición, se calcula mediante la fórmula de la combinatoria:

$$\text{Combinaciones} = \frac{n!}{r!(n-r)!}$$

Donde  $n$  es el número total de elementos en el conjunto (en este caso, letras y números), y  $r$  es el número de elementos seleccionados para la placa (en este caso, 7 caracteres).

Este valor de combinaciones representa la cantidad de placas únicas posibles dentro de una provincia. Sin embargo, para asegurar aún más la unicidad y evitar repeticiones, se aplican restricciones adicionales, como no permitir combinaciones que puedan confundirse fácilmente (por ejemplo, no usar combinaciones como "O" y "0" juntas) y utilizar caracteres especiales en posiciones específicas.

El algoritmo de asignación de placas también puede tener en cuenta la historia de placas emitidas previamente para evitar repeticiones exactas y minimizar la probabilidad de duplicados.

En resumen, el algoritmo de asignación de placas se basa en cálculos de combinaciones y permutaciones, con la aplicación de restricciones específicas y la consideración de la historia de emisión de placas, para garantizar que las secuencias de letras y números sean únicas y no se repitan dentro de una misma provincia.

### **Modelo matemático para asignación de placas:**

Símbolos y Variables:

$N$  = Número total de letras y números disponibles en los conjuntos.

$L$  = Conjunto de letras disponibles:  $L=\{A,B,C,\dots,Z\}$ .

$D$  = Conjunto de números disponibles:  $D=\{0,1,2,\dots,9\}$ .

$C$  = Longitud de la placa en caracteres.

$P$  = Número total de placas posibles en una provincia.

$U$  = Número total de placas emitidas hasta el momento en la provincia.

### **Restricciones:**

No repetir caracteres en la misma posición de la placa.

Evitar combinaciones confusas (como "O" y "0" juntas).

Considerar caracteres especiales en posiciones específicas (si aplicable).

**Modelo Matemático:**

$$P=N^C$$

El número total de placas posibles en una provincia (P) se calcula elevando el número de letras y números disponibles (N) a la longitud de la placa (C).

$$U \leq P$$

El número total de placas emitidas (U) debe ser menor o igual al número total de placas posibles (P).

**Objetivo:**

El objetivo del algoritmo es generar placas únicas (U) que cumplan con las restricciones mencionadas anteriormente.

**Proceso de Generación:**

1.-Inicializar  $U=0$ .

2.-Para cada combinación posible de caracteres en la placa:

Aplicar las restricciones (evitar repeticiones, combinaciones confusas, caracteres especiales).

Si la placa cumple con las restricciones, U aumenta en 1.

Si U alcanza P, detener el proceso.

Este modelo matemático simplificado describe la esencia del algoritmo de asignación de placas, pero en la práctica, se requerirían detalles adicionales y ajustes para abordar todas las consideraciones del proceso de asignación, incluyendo la implementación de restricciones específicas y la administración de placas emitidas previamente.

### **1.3.30 Técnicas ROC**

En los últimos años, la digitalización de información como textos, imágenes y sonido, ha captado mucho interés en la sociedad. En el caso de los textos, se genera constantemente una gran cantidad de información escrita, ya sea en formato tipográfico o manuscrito, en diversos tipos de medios. En este contexto, en el ámbito del reconocimiento de caracteres en las placas vehiculares, automatizar la introducción de los caracteres, evitando el ingreso manual a través del teclado, conlleva a un significativo ahorro de recursos humanos y un aumento en la productividad. Además, se logra mantener e incluso mejorar la calidad de numerosos servicios.

Por lo tanto, el reconocimiento óptico de caracteres tiene como objetivo principal la capacidad de identificar de manera automática símbolos o caracteres que forman parte de un alfabeto específico a partir de una imagen, con el fin de almacenarlos como datos con los que se pueda interactuar mediante un programa de edición de texto u otro similar.

#### **Definición de ROC**

El Reconocimiento Óptico de Caracteres (ROC), también conocido como OCR (Optical Character Recognition) en inglés, es una tecnología que permite convertir diversos tipos de documentos, como escaneos de papel, archivos PDF o imágenes capturadas por una cámara digital, en datos que pueden ser buscados y editados.

El software ROC examina minuciosamente la imagen, píxel por píxel, en busca de formas que coincidan con los caracteres deseados. Dependiendo de la complejidad, buscará diferentes coincidencias utilizando varias fuentes predefinidas. El ROC tiene la capacidad de analizar distintos elementos, como bloques de texto, imágenes o tablas, y verifica los espacios en blanco; descompone el texto en líneas, palabras y caracteres para que el programa pueda realizar el reconocimiento y encontrar el valor exacto o más cercano a lo buscado.

## Sistemas ROC Funcionalidad

En cualquier sistema de reconocimiento óptico de caracteres se distinguen al menos cuatro etapas, que reciben nombres similares o idénticos en distintos libros o por parte de diferentes autores. Es importante destacar que en cada una de estas etapas se deben emplear diversas técnicas existentes para poder procesar de manera óptima las imágenes de entrada [29].

- Adecuación de la imagen (preproceso).
- Selección de la zona de interés (segmentación).
- Representación digital de la imagen (extracción de características).
- Identificación del contenido presente en la imagen (reconocimiento).

### 1.3.31 Adecuación de la imagen (preproceso)

En la etapa de preprocesamiento, el objetivo es eliminar cualquier tipo de ruido o imperfección de la imagen que no pertenezca al carácter, además de normalizar su tamaño. En esta etapa puede ser necesario aplicar una técnica de binarización a la imagen [29].

Se emplean diversos algoritmos para eliminar el ruido que suele aparecer en una imagen digital debido a manchas reales, gráficas imperfectas o defectos técnicos en la adquisición o binarización de la imagen, como:

**Etiquetado:** Se realiza la división de la imagen en regiones compuestas por componentes conectados.

**Erosión o expansión:** Se utiliza para eliminar pequeños conjuntos de píxeles.

**Umbralizado de histograma:** Se utiliza para identificar y eliminar los objetos más brillantes u oscuros de la imagen.

### **Selección de la zona de Interés (Segmentación):**

Cuando se menciona la segmentación, se refiere a la selección de la región de interés basada en los valores o parámetros específicos que se desean extraer de una imagen en la que los caracteres estén claramente definidos. A través de la segmentación es posible

separar estos caracteres de manera relativamente sencilla, siempre y cuando se hayan propuesto previamente en el proceso; la segmentación de los caracteres puede variar en diferentes situaciones, desde condiciones favorables hasta menos favorables. En este contexto, la segmentación de los caracteres se basa en la delimitación de los límites de los campos en los que se espera que se completen los formularios [29].

### **Representación digital de la imagen (extracción de características):**

Una vez que se haya llevado a cabo la segmentación, se obtendrá una imagen normalizada que contiene la información más precisa para ser "reconocida". Sin embargo, es importante destacar que esta representación de la información en forma de matriz bidimensional de valores binarios, niveles de gris o colores RGB no garantiza que las características de la fotografía sean óptimas para los resultados deseados [29].

Por ende, el proceso de extracción de características mediante técnicas específicas tiene como objetivo eliminar diferentes componentes o contornos que no forman parte de la imagen, al mismo tiempo que se asegura de que la tipología y proporción original se mantengan sin deformarse.

Como parte del proceso, se debe realizar un escaneo paralelo en el cual se identifiquen los píxeles que se pueden eliminar simultáneamente. Esto permitirá facilitar la clasificación y el reconocimiento al simplificar la forma de los componentes [29].

### **Distinción del carácter contenido en la imagen (reconocimiento)**

En la etapa final, se lleva a cabo el reconocimiento de los objetos, en este caso, imágenes de caracteres. El desafío en esta etapa es desarrollar un método de reconocimiento mediante Lógica Difusa que pueda distinguir la clase a la que pertenece un objeto dentro de un conjunto limitado de clases posibles.

Este enfoque general del reconocimiento de formas se aplica al ROC al asignar un carácter hipotético a una imagen de entrada. En el contexto de las placas vehiculares, por ejemplo, nuestro objetivo de búsqueda podría ser la letra "A", y mediante este proceso de reconocimiento, esperamos obtener como resultado la letra "A" [29].

### **1.3.31 Software a utilizar en el desarrollo**

El entorno de trabajo de programación para el procesamiento de imágenes digitales mediante el algoritmo con Lógica Difusa se realizó a cabo utilizando Python y haciendo uso de librerías externas.

#### **Python**

Python ha experimentado un crecimiento constante pero gradual, convirtiéndose en el quinto lenguaje de programación más utilizado en 2021. Su popularidad se debe a varios factores, entre ellos su eficiencia en comparación con otros lenguajes de programación, así como la comunidad activa y la abundancia de recursos disponibles. Una de las ventajas destacadas de Python es la amplia gama de bibliotecas de código abierto disponibles. Estas bibliotecas son utilizadas en diversas disciplinas, pero han adquirido una mayor relevancia en los últimos años en los campos del Deep Learning y el Machine Learning [30].

Después de considerarlo cuidadosamente, se ha seleccionado Python como el lenguaje de programación preferido para el desarrollo de este proyecto debido a su capacidad para trabajar de manera eficiente con Machine Learning. Python se destaca por ser extremadamente versátil y flexible, lo que lo hace compatible con casi todos los sistemas operativos y plataformas disponibles en la actualidad. Además, la abundancia de bibliotecas de Python específicas para el Machine Learning también ha influido en esta elección.



**Figura 9.** Entorno de Trabajo utilizado. [30]

### **1.3.32 Comunicaciones**

Los sistemas o redes de computadoras están interconectados y forman la base de las comunicaciones. Estos sistemas se diseñan utilizando diversos protocolos de comunicación. Al establecer una conexión a internet, se utilizan diferentes protocolos dependiendo del tipo de conexión requerida. Sin embargo, es importante tener en cuenta que la comunicación en redes no se limita únicamente a internet cuando se trata de transmitir datos e intercambiar mensajes.

Los protocolos de red definen las características de la conexión. Estos protocolos son conjuntos de reglas formales, estándares y políticas que establecen restricciones, procedimientos y formatos para el intercambio de paquetes de información, permitiendo la comunicación entre dos servidores o dispositivos a través de una red.

Los protocolos de red incluyen mecanismos que permiten a los dispositivos identificarse y establecer conexiones entre sí. También definen las reglas de formato para la creación de paquetes y la estructura de los datos en los mensajes enviados y recibidos. Algunos protocolos admiten el reconocimiento de mensajes y la compresión de datos, lo cual está diseñado para lograr una comunicación confiable y de alto rendimiento en la red [31].

### 1.3.33 Tipos de protocolos de red

Los protocolos para la transmisión de datos en internet más importantes son TCP (Protocolo de Control de Transmisión) e IP (Protocolo de Internet). De manera conjunta (TCP/IP) podemos enlazar los dispositivos que acceden a la red, algunos otros protocolos de comunicación asociados a internet son POP, SMTP y HTTP [31]. Existen varios tipos de protocolos de red:

**Tabla 6.** Tipos de protocolos de red. [31]

<b>Tipos</b>	<b>Descripción</b>
<b>Protocolos de comunicación de red:</b>	Protocolos de comunicación de paquetes básicos como TCP / IP y HTTP [31].
<b>Protocolos de seguridad de red:</b>	Implementan la seguridad en las comunicaciones de red entre servidores, incluye HTTPS, SSL y SFTP [31].
<b>Protocolos de gestión de red:</b>	Proporcionan mantenimiento y gobierno de red, incluyen SNMP e ICMP [31].

**Elaborado por:** El Investigador

### 1.3.34 Almacenamiento en la nube

Este servicio proporciona accesibilidad sin restricciones, permitiendo a los usuarios acceder a sus datos desde cualquier lugar y en cualquier momento. Esto es posible debido a que los datos se almacenan en servidores remotos administrados por proveedores de almacenamiento en la nube. Además, el almacenamiento en la nube

generalmente se considera más seguro y económico que el almacenamiento local, ya que los proveedores implementan medidas avanzadas de seguridad y copias de seguridad. Debido a estas ventajas, se ha convertido en una solución popular para el almacenamiento de datos y documentos importantes. La nube ofrece diferentes opciones de almacenamiento, como el almacenamiento personal para usuarios individuales que desean realizar copias de seguridad de su correo electrónico, fotos, videos y otros archivos personales, así como el almacenamiento empresarial que permite a las empresas transferir y almacenar sus archivos de datos de manera segura, así como compartirlos entre diferentes ubicaciones [32].

### **1.3.35 Base de Datos**

#### **Mysql**

MySQL es actualmente el sistema de gestión de bases de datos relacionales más utilizado en el mundo, gracias a su carácter de código abierto. Inicialmente desarrollado por MySQL AB, la empresa fue adquirida por Sun Microsystems en 2008 y posteriormente por Oracle Corporation en 2010. Cabe mencionar que Oracle ya era propietaria del motor InnoDB para MySQL [33].

MySQL se destaca por ser un sistema de gestión de bases de datos que cuenta con una doble licencia. Por un lado, ofrece una licencia de código abierto que permite a los usuarios acceder, modificar y distribuir el software de acuerdo con sus necesidades. Por otro lado, existe una versión comercial, gestionada por Oracle, que brinda servicios y soporte adicionales para aquellos que requieren funcionalidades avanzadas y asistencia técnica especializada. Esta opción comercial proporciona una mayor flexibilidad y soporte para implementaciones empresariales de MySQL [33].

MySQL ofrece diversas ventajas que lo convierten en una opción atractiva para los desarrolladores. La principal ventaja radica en su capacidad para trabajar con bases de datos relacionales, lo que implica el uso de múltiples tablas interconectadas para

almacenar y organizar la información de manera eficiente. Al ser un sistema de código abierto, MySQL es accesible y ampliamente utilizado en el desarrollo web [33].



**Figura 10.** Base de Datos Mysql [33]

## **Oracle**

Oracle Database es un potente sistema de gestión de bases de datos relacionales (RDBMS) desarrollado por Oracle, una reconocida empresa estadounidense de software y hardware. Esta solución de bases de datos optimiza la gestión y seguridad de los conjuntos de datos al crear esquemas estructurados que solo pueden ser accedidos por administradores autorizados. Oracle Database se basa en un modelo relacional de base de datos, lo que permite almacenar y representar los datos empresariales y de los clientes en forma de conjuntos de datos organizados. Los datos se estructuran en columnas, tablas y filas, y se establecen relaciones entre ellos mediante atributos. Una de las principales ventajas de Oracle Database es su capacidad para organizar y presentar grandes volúmenes de datos de manera intuitiva y eficiente [34].

La arquitectura de los sistemas Oracle Database se compone de varios componentes interconectados que trabajan en conjunto para garantizar el almacenamiento y la gestión eficiente de los datos [34]. Estos componentes incluyen:

- **Base de datos:** Es el contenedor principal que almacena los archivos de la base de datos. Aquí se guardan todos los datos, como tablas, índices y otros objetos relacionados [34].

- Instancias de la base de datos: Son los procesos que gestionan y manipulan los datos en la base de datos. Cada instancia se encarga de administrar un conjunto específico de datos y proporciona acceso a esos datos a través de las conexiones de los clientes [34].
- Procesos de escucha: Son responsables de establecer y gestionar las conexiones entre los clientes de la base de datos y las instancias de la base de datos. Estos procesos aseguran una comunicación adecuada y segura entre los usuarios y la base de datos [34].
- La arquitectura de Oracle Database separa las estructuras de datos lógicas y físicas. Las estructuras de datos lógicas se refieren a la organización lógica de los datos, como tablas, vistas y procedimientos almacenados. Por otro lado, las estructuras de datos físicas se refieren a cómo se almacenan los datos en los discos y cómo se accede a ellos de manera eficiente [34].

En resumen, la arquitectura de Oracle Database está diseñada para garantizar la integridad, seguridad y eficiencia en la gestión de los datos, mediante la separación de las estructuras lógicas y físicas, y la interacción coordinada de la base de datos, las instancias y los procesos de escucha [34].



**Figura 11.** Base de Datos Oracle [34]

## **PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos de código abierto y gratuito. Una de las ventajas destacadas de PostgreSQL es su amplio conjunto de tipos de datos avanzados, lo que permite una mayor flexibilidad al almacenar y manipular la información. Además, ofrece capacidades de optimización de rendimiento avanzadas,

que generalmente se encuentran en sistemas de bases de datos comerciales, como Microsoft SQL Server u Oracle [35].

Estos detalles resaltan la capacidad de PostgreSQL para ofrecer características avanzadas que normalmente solo están disponibles en sistemas de bases de datos comerciales, a la vez que mantiene su enfoque en el código abierto y su gratuidad [35].

PostgreSQL destaca como uno de los sistemas de bases de datos más avanzados y ampliamente utilizados en el mundo. PostgreSQL posee una serie de características notables que lo hacen destacar entre otros sistemas [35].

PostgreSQL se destaca por su facilidad de uso, lo que lo convierte en una opción popular entre los usuarios. La administración de PostgreSQL se simplifica gracias a herramientas como PgAdmin, que ofrece una interfaz gráfica similar a phpMyAdmin, pero diseñada específicamente para PostgreSQL. Con PgAdmin, tanto los principiantes como los usuarios experimentados pueden llevar a cabo diversas tareas de manera sencilla. Además de ser accesible para los principiantes, también ofrece funcionalidades avanzadas que permiten a los usuarios expertos realizar tareas más complejas. En resumen, la usabilidad intuitiva de PostgreSQL y su herramienta PgAdmin facilitan la administración y el manejo de la base de datos, sin importar el nivel de experiencia del usuario [35].



**Figura 12.** Base de datos PostgreSQL. [35]

### 1.3.36 Cámaras

La cámara de red, también conocida como cámara IP, es aquella que transmite video a través de una red IP utilizando conmutadores de red, y registra la información en un servidor de PC con un software de gestión de video instalado. Este sistema es completamente digital, ya que no utiliza componentes analógicos [36].

Una ventaja significativa es que las imágenes capturadas se digitalizan en la misma cámara y se mantienen intactas a lo largo del sistema. Esto asegura una calidad de imagen óptima y constante, a diferencia de las cámaras analógicas. Además, la red IP puede transmitir energía eléctrica a las cámaras de red, así como audio bidireccional. Por otro lado, una red IP permite la configuración remota de las cámaras, lo que facilita el envío de video y otros datos a cualquier ubicación sin degradación de la calidad [36].



**Figura 13.** Modelos de cámaras IP. [36]

### 1.3.37 Medios de Transmisión

La infraestructura de transmisión desempeña un papel vital en las redes de computadoras, ya que se compone de los enlaces que conectan los distintos dispositivos de red y facilitan el transporte de información dentro de la propia red. Estos medios de transmisión se clasifican en alámbricos, ópticos y electromagnéticos según su estructura física [37].

## **Cable UTP**

El cable de par trenzado no apantallado (UTP) es ampliamente reconocido como el tipo de cable de par trenzado más común y está ganando rápidamente popularidad como la opción principal para el cableado de áreas locales. En cada segmento de cable UTP, la longitud máxima permitida es de 100 metros [37].

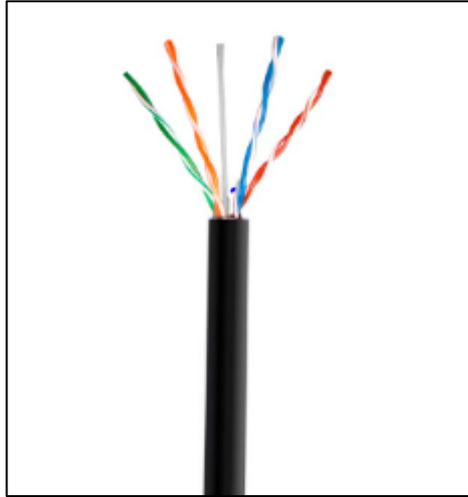
**Categorías 1 y 2:** Estos cables se han utilizado principalmente para transmitir señales de voz y datos a velocidades muy bajas. Sin embargo, no son adecuados para la transmisión de datos de alta velocidad [37].

**Categoría 3:** Los cables, conectores y accesorios están diseñados y especificados para soportar frecuencias de hasta 16 MHz. Este tipo de cableado, generalmente conocido como categoría de cableado específica, se utiliza comúnmente para transmisiones de voz y datos con velocidades de hasta 10 Mbps [37].

**Categoría 4:** La especificación de los cables, conectores y accesorios establece que su capacidad llega hasta los 20 MHz, siendo habitual su uso para la transmisión de voz y datos a velocidades de hasta 16 Mbps [37].

**Categoría 5:** Los cables, conectores y accesorios están diseñados para soportar hasta 100 MHz de frecuencia y se utilizan comúnmente en nuevas instalaciones donde se requieren velocidades de transmisión que pueden alcanzar e incluso superar los 100 Mbps [37].

**Categoría 6:** El estándar de cables utilizado para Gigabit Ethernet y protocolos de redes es compatible con los estándares de categoría 5/5e y categoría 3, lo que significa que puede utilizarse en redes que utilicen esos estándares anteriores. Este estándar se ha desarrollado con características y especificaciones específicas para minimizar problemas como la diafonía y el ruido. Es capaz de soportar frecuencias de hasta 250 MHz en cada par de cables y alcanza una velocidad de transmisión de 1 Gbps [37].



**Figura 14.** Par trenzado cable UTP categoría 6. [37]

## **1.4 Objetivos**

### **1.4.1. Objetivo General**

Implementar un Sistema automático de registro de automóviles y manejo del historial de trabajos para la microempresa “Tecninars David” utilizando Machine Learning.

### **1.4.2. Objetivos Específicos**

- Determinar las técnicas y herramientas para la detección de placas de automóviles y registro de su información en el taller de servicio automotriz “Tecninars David”.
- Comparar la arquitectura lógica y física a emplear en el sistema automático de registro de automóviles.
- Diseñar una interfaz gráfica para la visualización de la gestión de automóviles que ingresan al taller.
- Realizar pruebas de funcionamiento y determinar el nivel de precisión del sistema desarrollado.

## **CAPÍTULO II METODOLOGÍA**

### **2.1. Materiales**

Para la implementación de este proyecto, se emplearon diversos componentes y tecnologías para su funcionamiento. La adquisición de imágenes de las placas vehiculares se llevó a cabo mediante el uso de una cámara, mientras que la tarjeta Raspberry Pi 4 se encargó del procesamiento de los datos obtenidos. Para establecer la conexión entre la cámara y la Raspberry Pi, se utilizó un cable UTP.

En cuanto al desarrollo del Dashboard, se utilizó el software Node-Red para su creación. Este software se encarga de recopilar los datos enviados desde la tarjeta Raspberry Pi4 hacia el bróker de Mosquitto. Estos datos son almacenados en una base de datos gestionada por MariaDB, que a su vez trabaja en conjunto con técnicas de Machine Learning aplicadas a la base de datos del sistema.

Es importante destacar que todas estas aplicaciones y tecnologías se encuentran alojadas en un servidor Ubuntu, el cual se encuentra hospedado en Microsoft Azure. Este entorno de servidor brinda un entorno confiable y escalable para garantizar el correcto funcionamiento del sistema.

### **2.2. Métodos**

#### **2.2.1. Modalidad de investigación**

El presente proyecto se basó en una Investigación Aplicada, en la cual se utilizaron los conocimientos adquiridos durante la carrera para crear una solución efectiva y se implementó un prototipo en un centro de servicio automotriz.

Se llevó a cabo una investigación bibliográfica que implicó recopilar información de diversas fuentes, como tesis, artículos científicos, publicaciones en revistas y libros, relacionados con temas de Machine Learning y Lógica Difusa.

Adicionalmente, se realizó una Investigación de Campo, que implicó el estudio y la implementación del prototipo dentro del centro automotriz "Técnicars David".

### **2.2.2. Recolección de información**

Para recopilar la información necesaria, se utilizaron diversas fuentes confiables, como libros, revistas y fuentes en línea, además de proyectos ya desarrollados y datos de incidencia y prevalencia relacionados con Lógica Difusa. Se tuvo especial cuidado en seleccionar únicamente fuentes confiables que permitieran un desarrollo adecuado del proyecto.

Además, se llevaron a cabo visitas técnicas al taller de servicio automotriz para obtener información adicional, como fichas técnicas y entrevistas con los trabajadores, lo que ayudó a complementar la investigación y obtener datos precisos y relevantes.

### **2.2.3. Procesamiento y análisis de datos**

Para procesar y analizar los datos, se llevaron a cabo los siguientes pasos:

- Se realizó un análisis exhaustivo de todas las fuentes de información documental relacionadas con el diseño de un sistema de registro automático de automóviles.
- Se estudiaron las diferentes alternativas de solución disponibles para el control de placas, con el objetivo de determinar cuál sería la mejor opción para implementar.
- Se planteó una propuesta de solución basada en los resultados obtenidos en la investigación y en el análisis de las alternativas disponibles.
- Finalmente, se realizó un control y verificación de los datos obtenidos a través del testeo del dispositivo, lo que permitió asegurar la efectividad de la solución propuesta.

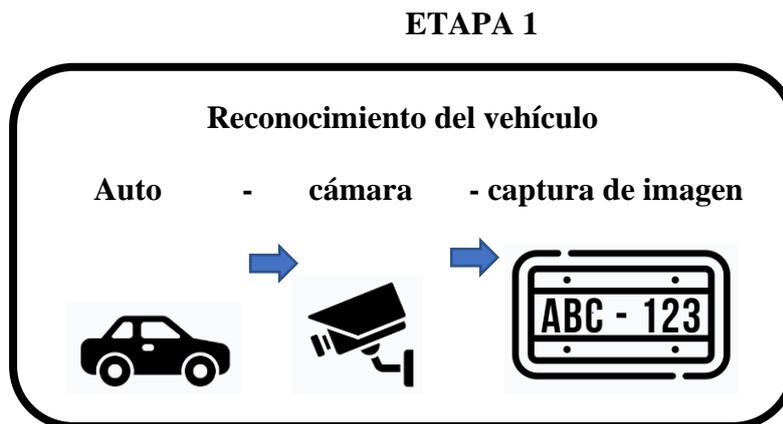
## CAPÍTULO III RESULTADOS Y DISCUSIÓN

### 3.1. Análisis y discusión de los resultados

#### 3.1.1 Desarrollo de la propuesta

Para dar cumplimiento con los objetivos planteados del sistema automático de registro de automóviles y manejo del historial de trabajos para la microempresa “Técnicars David ” utilizando Machine Learning se implementó un sistema que adquiere la información del vehículo a través del reconocimiento de la placa mediante el uso de una cámara instalada en el ingreso del centro automotriz. El sistema captura la imagen de la placa para después ser procesada mediante un algoritmo con Lógico Difusa y así detectar los datos alfanuméricos de la placa vehicular. Los datos obtenidos del sistema serán enviados a un servidor web mediante un módulo inalámbrico.

Se desarrolló un sistema de registro automatizado con el propósito de recopilar todos los datos mencionados previamente. La estructura de este dispositivo se encuentra representada en la figura 2 y consta de tres etapas: adquisición, procesamiento y servicio en la nube y visualización.



**Figura 15.** Arquitectura adquisición de datos del sistema de registro automático

**Elaborado por:** El investigador

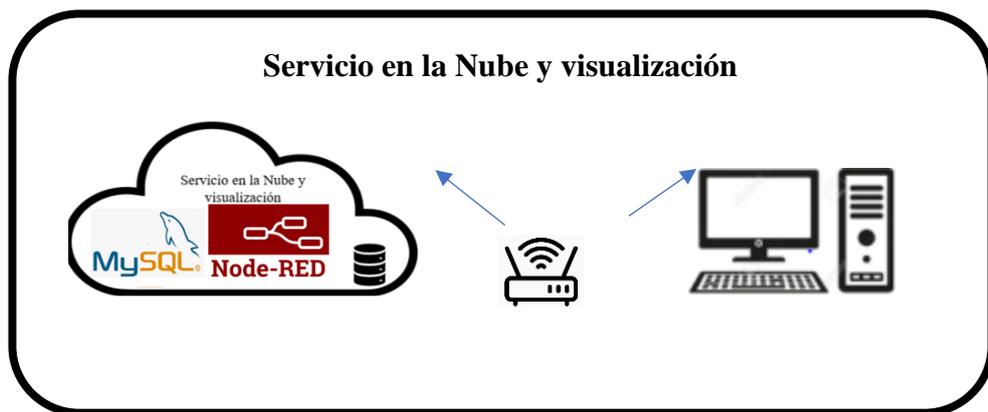
## EPATA 2



**Figura 16.** Arquitectura procesamiento de datos del sistema

**Elaborado por:** El investigador

## ETAPA 3



**Figura 17.** Arquitectura almacenamiento de datos del sistema

**Elaborado por:** El investigador

### 3.1.2 ETAPA 1.- Adquisición

Se han escogido cuidadosamente todos los componentes electrónicos necesarios para crear el Sistema, de manera que cumplan con los requisitos indispensables para lograr los resultados deseados.

### 3.1.2.1 Para la adquisición de imágenes

Al capturar imágenes con una cámara, es esencial considerar diversos aspectos como la configuración de iluminación, la calidad de la resolución de la cámara, la distancia al objeto a fotografiar y, especialmente, el propósito principal, que consiste en obtener datos valiosos a partir de la imagen.

Hay que tomar en cuenta los siguientes factores para el sistema:

**Sistema de iluminación.** - Se trata de una componente sumamente crucial del sistema, dado que una imagen de entrada adecuada optimiza las posibilidades de procesamiento subsiguiente. Cuanto más óptima sea la solución de iluminación, mayor será el rendimiento de su sistema de reconocimiento basado en Lógica Difusa.

**Cámara y lentes.** - Su función consiste en capturar y enfocar la imagen de la escena utilizando un sensor electrónico que convierte las variaciones de luz en señales eléctricas. Existe una amplia variedad de cámaras disponibles, que se seleccionan en función de las necesidades específicas de la aplicación. Estas cámaras pueden ser tanto analógicas como digitales.

No convendría usar una cámara de baja resolución para estudiar imágenes del espacio exterior.

### 3.1.2.2 Selección de Cámara

En la siguiente tabla se presenta una comparativa entre varias cámaras disponibles en el mercado, teniendo en cuenta sus características más relevantes en relación con el propósito del proyecto. Se ha elegido la cámara de seguridad Cámara 4 en 1 Bullet HD 1080p 2 Megapíxeles Negra para exteriores debido a sus excelentes características, que incluyen una resolución de vigilancia exterior de 1080p, capacidad de conexión cableada, resistencia al agua, visión nocturna. Además, esta cámara ofrece un precio comparativamente económico con relación a otras opciones disponibles. La utilización de esta cámara permitirá capturar imágenes de la placa de vehículo, estas imágenes

serán procesadas mediante un algoritmo de Lógica Difusa para extraer los datos alfanuméricos de la placa.

**Tabla 7.** Comparación de las cámaras. [38] [39] [40]

<b>Características</b>	<b>Cámara 4 en 1 Bullet HD 1080p 2 Megapixeles Negra</b>	<b>Camara Hikvision Ds-2ce16c0t-irpf Bullet Tubo 720p</b>	<b>Hanwha HCO-6070R Serie Wisenet HD Plus</b>
<b>Figura</b>			
<b>Lente</b>	3.6 mm	2.8 mm	3.2 mm
<b>Sensor</b>	CMOS 1/4''	1 MP CMOS	2 MP CMOS
<b>Angulo de Visión</b>	70 grados	70.9 grados	75 grados
<b>Cuerpo de la cámara</b>	Cuerpo Metálico	Cuerpo metálico	Cuerpo Metálico en conjunto con aluminio
<b>Dimensiones</b>	12 cm largo, 5 cm ancho y 5 cm de espesor	70 mm × 154.5 mm ( 2.76" × 6.08")	8.7 cm de alto, 8.32 cm de ancho y 25.98 cm de espesor
<b>Peso</b>	220 gramos		914 gramos
<b>Voltaje de Operación</b>	12 volts de corriente directa y/o 24 volts de corriente alterna	12 V DC ± 15%, max. 4 W	12 volts de corriente directa y/o 24 volts de corriente alterna

**Elaborado por:** El investigador



**Figura 18.** Instalación de cámara

**Elaborado por:** El investigador

### **3.1.2.3 Alimentación del dispositivo**

Tomando en cuenta que se planteó desarrollar un sistema de registro automático que se encuentre en funcionamiento durante el día, se identificó y se propuso dos métodos de alimentación necesarios para garantizar el suministro y el funcionamiento continuo del sistema. La primera es el método clásico, que es estar conectado de forma permanente a la fuente de alimentación de luz eléctrica. Mientras que la otra forma será la de conectarse a una fuente de alimentación externa.

## **3.1.3 ETAPA 2.- Procesamiento y transmisión**

### **3.1.3.1 Selección del microcontrolador**

La obtención y procesamiento de datos desempeñan un papel fundamental en el funcionamiento del prototipo, ya que es necesario capturar imágenes que serán procesadas por un microcontrolador. Este microcontrolador es el responsable de extraer y gestionar los datos alfanuméricos obtenidos de las imágenes, con el fin de

interpretarlos y visualizarlos posteriormente. A continuación, se muestra una tabla con las características de las dos tarjetas más adecuadas, las cuales cumplen con los requisitos establecidos para alcanzar los objetivos del proyecto.

**Tabla 8.** Comparación de Microordenadores. [41] [42]

Parámetros	RASPBERRY PI4	JETSON NANO
Figura		
Frecuencia de Operación	2 GHz de frecuencia de funcionamiento	2.3 GHz de frecuencia de funcionamiento
Número de entradas digitales	26 pines para entradas y salidas digitales y dos de ellos pueden usarse para control PWM.	pines GPIO y conectores UART
Voltaje de entrada	Salida de voltaje de 5.1VDC/3	Salida de voltaje de 5V
CPU	1.5-GHz, 4-core Broadcom BCM2711 (Cortex-A72)	Quad-Core ARM A57 @ 1.43 GHz
Procesador	ARM Cortex-A72	ARM de 64 bits de cuatro núcleos
Conectividad	Bluetooth 5.0, Wi-Fi 802.11 Puerto Gigabit Ethernet.	Puerto Gigabit Ethernet M.2 Key E
Puertos	GPIO 40 pines GPIO 40 pines 2 x micro HDMI 2 x USB 2.0 2 x USB 3.0 CSI (cámara Raspberry Pi)	puertos USB-A 3.0, salidas de vídeo HDMI y DisplayPort, así como un puerto Ethernet LAN

**Elaborado por:** El investigador

En esta sección, disponemos de dos tarjetas que funcionan como minicomputadoras y ofrecen un excelente rendimiento de procesamiento. Como parte del proyecto se desea implementar estas tarjetas microcontroladoras para realizar el procesamiento digital de imágenes.

Aunque se han mencionado todas las características de ambas tarjetas, en general comparten atributos similares. Ambas son computadoras de placa única con procesadores ARM y 4 GB de RAM, y brindan una amplia conectividad para periféricos. No obstante, Raspberry Pi 4 también ofrece una opción de memoria de 8 GB. En términos de GPU, destaca la Jetson Nano con su GPU Maxwell de 128 núcleos a 921 MHz, mientras que la GPU de Raspberry Pi 4 es menos potente.

En cuanto a la CPU, Raspberry Pi utiliza la ARM Cortex-A72 de cuatro núcleos y 64 bits a 1,5 GHz, que es la versión más reciente y de mayor rendimiento. Esta CPU ofrece un rendimiento más alto y una velocidad de reloj más rápida. Por lo que es la elección ideal para el desarrollo de este proyecto, debido a su rendimiento más rápido. El algoritmo de reconocimiento de placas con Lógica Difusa prácticamente realiza todo el trabajo y requiere un procesamiento más rápido, que es proporcionado por la tarjeta Raspberry Pi 4.



**Figura 19.** Tarjeta Raspberry pi4

**Fuente:** [41]



**Figura 20.** Instalación de la Tarjeta Raspberry Pi4

**Elaborado por:** El Investigador

### 3.1.3.2 Selección de la tecnología inalámbrica

En la actualidad, hay diversas tecnologías inalámbricas disponibles que son adecuadas para llevar a cabo este proyecto. La opción del Bluetooth se descarta debido a su alcance limitado de 10 metros. Así mismo, el uso de la tecnología ZigBee se desestima debido a los altos costos asociados.

A continuación, se presenta una tabla comparativa de las tecnologías inalámbricas

**Tabla 9.** Tecnologías Inalámbricas. [43]

<b>Parámetro</b>	<b>Wi-Fi</b>	<b>ZigBee</b>	<b>Bluetooth</b>
<b>Estándar</b>	802.11	802.15.4	802.15.1
<b>Frecuencia de la señal</b>	2.4 – 5 GHz	2.4 GHz, 915 MHz 868 MHz	2.4 GHz
<b>Ancho de banda</b>	2 – 100 Mbps	20 – 250 kbps	1 Mbps
<b>Número de nodos por red</b>	255	65000	7
<b>Distancia de transmisión sin interferencias</b>	50 - 100 m	1 – 75 m	10 – 100 m
<b>Memoria</b>	100 kB	32 - 60 kB	100 kB
<b>Topología</b>	Bus, anillo, estrella y árbol	Punto a punto, estrella, malla, árbol y clúster	Punto a punto, estrella y árbol

**Elaborado por:** El investigador

Por consiguiente, se decidió utilizar la tecnología Wi-Fi debido a su facilidad de conexión a internet y a la amplia disponibilidad de equipos compatibles.

### 3.1.4 ETAPA 3.- Servicio en la nube y visualización

#### 3.1.4.1 Selección del servidor

Antes de seleccionar e instalar el servidor destinado a recopilar los datos enviados desde el controlador, se llevó a cabo un estudio exhaustivo de las plataformas más comunes de creación de servidores web en la nube. A continuación, se presenta una tabla que resume las características principales de cada una de estas plataformas.

**Tabla 10.** Comparación de las plataformas de la Nube. [44]

<b>Parámetros</b>	<b>Google Cloud Platform</b>	<b>Microsoft Azure</b>	<b>Amazon EC2-AWS</b>
<b>Soporte disponible</b>	Foros, documentación, preparación, comunicaciones telefónicas	Foros, comunicaciones telefónicas, chats en vivo, documentación	Herramientas de aprendizaje, Foros, documentación.
<b>Bases de datos</b>	Cloud SQL Cloud Spanner Cloud Bigtable Cloud Firestore Cloud Momorystore Firebase Realtime Database	Azure SQL	PostgreSQL MySql MariaDB Memcached
<b>Sistema Operativo del servidor</b>	Linux, Windows	Linux, Windows	Windows, Linux
<b>Preconfiguración del sistema operativo</b>	Ubunto Centos os OpenSUSE Linux FreeBSD	Azure SQL	MySql Redis MariaDB MongoDB Memcached

**Elaborado por:** El investigador

Después de realizar una exhaustiva evaluación de las opciones disponibles, se decidió seleccionar Microsoft Azure como plataforma de servidor para este proyecto. Azure ha estado en operación desde 2010 y cuenta con una nube más amplia, mientras que la plataforma de Google Cloud comenzó en 2011 y ha acumulado importantes clientes como Airbus, Coca-Cola, HTC y Spotify, entre otros.

En lo que respecta a la información sobre el manejo y el correcto uso de las plataformas de Google Cloud, Microsoft Azure y Amazon EC2, cada una de ellas dispone de una amplia variedad de foros y documentación disponibles. En cuanto a las especificaciones, es importante destacar que AWS ofrece una mayor cantidad de sistemas operativos preconfigurados, mientras que Microsoft Azure tiene un alcance más amplio en términos de funcionalidad.

En cuanto a los precios, el costo varía en función del uso que se le dé a la plataforma. Sin embargo, Microsoft Azure ofrece una ventaja adicional, mediante el registro con el correo institucional se puede contar con un crédito de \$100 durante un año, además de poder contar con servicios gratuitos para incentivar el aprendizaje, la investigación sin fines comerciales, el diseño, desarrollo, pruebas y demostración de aplicaciones. Esto es especialmente importante para la realización de este proyecto. En consecuencia, se seleccionó la plataforma en la nube de Microsoft Azure debido a que cumple con las necesidades específicas del proyecto en términos de flexibilidad y versatilidad.

#### **3.1.4.2 Selección del protocolo de comunicación**

Se realizó un estudio para seleccionar el protocolo de comunicación adecuado en el ámbito del Cloud Computing e Internet de las cosas. La siguiente tabla muestra las características más relevantes de los protocolos analizados. MQTT es un protocolo de comunicación que permite a múltiples clientes intercambiar datos a través de un punto central, siendo eficiente para la transmisión de datos en tiempo real. Por otro lado, CoAP permite la transferencia de información de estado de forma uno a uno entre el cliente y el servidor. En comparación con MQTT, HTTP tiene un mayor costo en términos de establecimiento de conexión, debido a que en MQTT se pueden enviar y

recibir múltiples mensajes en la misma sesión TCP, aprovechando así las ventajas que ofrece este protocolo.

**Tabla 11.** Protocolos de comunicación comparación. [45]

<b>Parámetros</b>	<b>HTTP</b>	<b>MQTT</b>	<b>CoAP</b>
<b>Arquitectura</b>	Cliente/servidor	Cliente/Broker	Cliente/ Servidor o Cliente/Broker
<b>Calidad de servicio (QoS)/Confiabilidad</b>	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 confirmable (similar a por lo menos una vez). Mensaje confirmable (similar a como máximo una vez)
<b>Conectividad</b>	Uno-a-uno	Uno-a-uno, uno a muchos y muchos-a-muchos	Uno a uno y muchos a muchos.
<b>Consumo de banda ancha</b>	Implica un mayor ancho de banda	Consumo más banda ancha	Implica el ancho de banda más bajo.
<b>Puerto predeterminado</b>	80/ 443 (TLS/SSL)	1883/8883 (TLS/SSL)	5683(UDP Port)/ 5684(DLTS)
<b>Consumo de energía</b>	Requiere una mayor potencia/energía consumida, fue mucho mayor que con MQTT	MQTT es más eficiente	Es más eficiente en términos de energía
<b>Aplicaciones</b>	Web	Automatizaciones a nivel del hogar y empresarial	Redes inteligentes, automatización de edificios y casas inteligentes

**Elaborado por:** El Investigador

Se realizó un estudio para seleccionar el protocolo de comunicación adecuado en el ámbito del Cloud Computing. La siguiente tabla muestra las características más relevantes de los protocolos analizados. MQTT es un protocolo de comunicación que permite a múltiples clientes intercambiar datos a través de un punto central, siendo eficiente para la transmisión de datos en tiempo real. Por otro lado, CoAP permite la transferencia de información de estado de forma uno a uno entre el cliente y el servidor. En comparación con MQTT, HTTP tiene un mayor costo en términos de establecimiento de conexión, debido a que en MQTT se pueden enviar y recibir múltiples mensajes en la misma sesión TCP, aprovechando así las ventajas que ofrece este protocolo. En este contexto, el protocolo de comunicación más idóneo para este proyecto es MQTT, debido a su flexibilidad y capacidad para satisfacer las necesidades de transmisión de grandes volúmenes de datos en tiempo real.

### 3.1.4.3 Selección de la base de datos

**Tabla 12.** Comparación de la Base de datos. [46] [47]

<b>Parámetros</b>	<b>PostgreSQL</b>	<b>Oracle</b>	<b>MySQL</b>
<b>Descripción</b>	Código abierto, Sistema de gestión de base de datos objeto.	Es un sistema de gestión de bases de datos relacionales	Es un sistema de manejo de bases de datos con código abierto.
<b>S.O. Compatible</b>	Windows, Linux y Mac OS.	Linux, Windows, Unix y Mac OS X y Unix.	Windows, Linux OSX, Solaris, FreeBSD.
<b>Licencia</b>	Es libre	Es libre a nivel de usuario	Es privada
<b>Lenguaje de Programación</b>	C/C++	Visual C++	C, C++
<b>Arquitectura de implementación</b>	Cliente/Servidor	Cliente/Servidor	Cliente/Servidor
<b>Compatibilidad con Node-RED</b>	Si es compatible	Si es compatible	Si es compatible

**Elaborado por:** El Investigador

Se ha optado por implementar un sistema que permita clasificar y organizar los datos de acuerdo con la placa de los vehículos, es decir, los datos se ordenarán utilizando un identificador para cada vehículo. En el mercado, PostgreSQL es reconocido como el sistema de gestión de bases de datos más robusto, aunque no es tan ampliamente utilizado como MySQL y cuenta con menos recursos de aprendizaje, como tutoriales y cursos disponibles en comparación con MySQL.

Por otro lado, Oracle es una de las bases de datos más utilizadas por las empresas, pero su costo es alto, incluyendo el costo de la información. En contraste, MySQL es un sistema de base de datos muy popular y ampliamente utilizado en todo el mundo. Existe una gran cantidad de información y tutoriales disponibles para MySQL, lo que lo hace adecuado para proyectos de pequeña y mediana escala.

Se ha elegido MySQL debido a su facilidad de uso para adquirir datos de dispositivos electrónicos. Además, puede conectarse directamente con Node-RED, lo que permite representar de manera dinámica los parámetros del sistema de monitoreo.

#### **3.1.4.4 Selección de la herramienta de visualización**

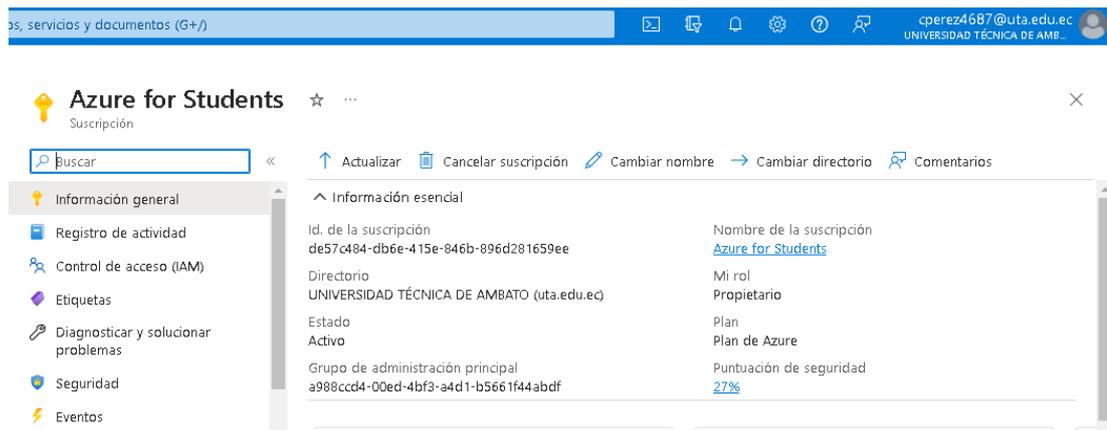
Node-RED fue elegido como la herramienta de visualización de datos recopilados por el sistema de registro automatizado.

Primero, Node-RED es altamente compatible con la base de datos MySQL elegida para almacenar la información del vehículo. Esta compatibilidad facilita la integración de los dos sistemas y asegura una gestión eficiente de los datos. Además, Node-RED es una plataforma de programación visual completamente gratuita, lo que significa que no tiene ningún costo adicional para usarla y está disponible para cualquiera que quiera usarla. También es muy intuitivo y fácil de usar, que es la opción ideal para atraer el flujo de información de los dispositivos electrónicos. Los datos se pueden visualizar en tiempo real y dinámica que ayudan a explicar los resultados.

En resumen, Node-RED es una excelente opción para visualizar los datos recopilados por el sistema de registro, razón por la cual fue utilizado para la realización de este proyecto de titulación.

### 3.1.5 Estancia en la Nube

Se utilizó Azure Virtual Machines, un servicio que brinda facilidad en la virtualización sin requerir hardware físico. Para acceder a los servicios en la nube de Microsoft Azure, se utilizó una suscripción de Azure Education obtenida a través de un correo electrónico de una institución educativa. Se procedió a implementar una máquina virtual de Ubuntu Server, la cual alberga los servicios de Node-RED, Mosquitto y MariaDB.



**Figura 21** Suscripción de Azure Education

**Elaborado por:** El Investigador

Fue necesario generar un conjunto de recursos para poder hacer uso de los servicios de Azure, tal y como se puede apreciar en la siguiente figura 21.

## Crear una máquina virtual ...

**Detalles del proyecto**

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* ⓘ Azure for Students

Grupo de recursos \* ⓘ (Nuevo) Tecnicars  
[Crear nuevo](#)

**Detalles de instancia**

Nombre de máquina virtual \* ⓘ Tecnicars ✓

Región \* ⓘ (US) West Central US

Opciones de disponibilidad ⓘ No se requiere redundancia de la infraestructura

Tipo de seguridad ⓘ Máquinas virtuales de inicio seguro  
[Configurar características de seguridad](#)

[Revisar y crear](#) < Anterior Siguiente: Discos >

**Figura 22.** Creación del grupo de Recursos Tecnicars

**Elaborado por:** El Investigador

Después de asignar un nombre al grupo de recursos y seleccionar una región, se procedió a verificar la validez de los datos ingresados.

## Crear una máquina virtual ...

✓ Validación superada

**Datos básicos**

Suscripción	Azure for Students
Grupo de recursos	(nuevo) Tecnicars
Nombre de máquina virtual	Tecnicars
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 B2s (2 vcpu, 4 GiB de memoria)
Tipo de autenticación	Contraseña
Nombre de usuario	Tecnicars

**Figura 23.** Validación del grupo de recursos

**Elaborado por:** El Investigador

Una vez completada la creación del grupo de recursos, los servicios de Azure están listos para ser utilizados.

## Servidor en la nube de Microsoft Azure

El servidor está instalado en Microsoft Azure como una máquina virtual de Ubuntu Server. Este servidor ya tiene instalado Node-RED, una herramienta que facilita y agiliza la comunicación con el hardware. Node-RED se encarga de recolectar las métricas enviadas al Mosquito Broker, también instalado en el mismo servidor, desde la tarjeta Raspberry Pi a través de una conexión WiFi. Los datos recopilados se almacenan en la base de datos MariaDB.

Luego, estos datos se muestran en el nodo dashboard propio de Node-RED. El servidor utiliza Ubuntu Server 20.04 LTS - Gen2, el cual es apto para el desarrollo de este proyecto.

### Crear una máquina virtual ...

Configurar características de seguridad

Imagen \* ⓘ

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

Arquitectura de VM ⓘ  Arm64  x64

Ejecución de Azure Spot con descuento ⓘ

Tamaño \* ⓘ

[Ver todos los tamaños](#)

**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.

**Figura 24.** Creación del Servidor Ubuntu Server

**Elaborado por:** El Investigador

Para implementar el servidor web se requiere especificar la suscripción, el grupo de recursos, la región y la zona en la que se desea crear el servidor, así como la cantidad de memoria RAM y el número de núcleos de CPU disponibles. En este caso, se decidió utilizar 1 GB de memoria RAM y un núcleo de CPU, ya que es suficiente para ejecutar los servicios necesarios para el desarrollo del proyecto.

La autenticación para acceder al servidor web se realiza mediante el uso de SSH con clave pública.

**Crear una máquina virtual** ...

Subred \* ⓘ (nuevo) default (10.0.0.0/24) ▼

IP pública ⓘ (nuevo) Tecnicars-ip ▼  
Crear

Grupo de seguridad de red de NIC ⓘ  Ninguno  Básico  Opciones avanzadas

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

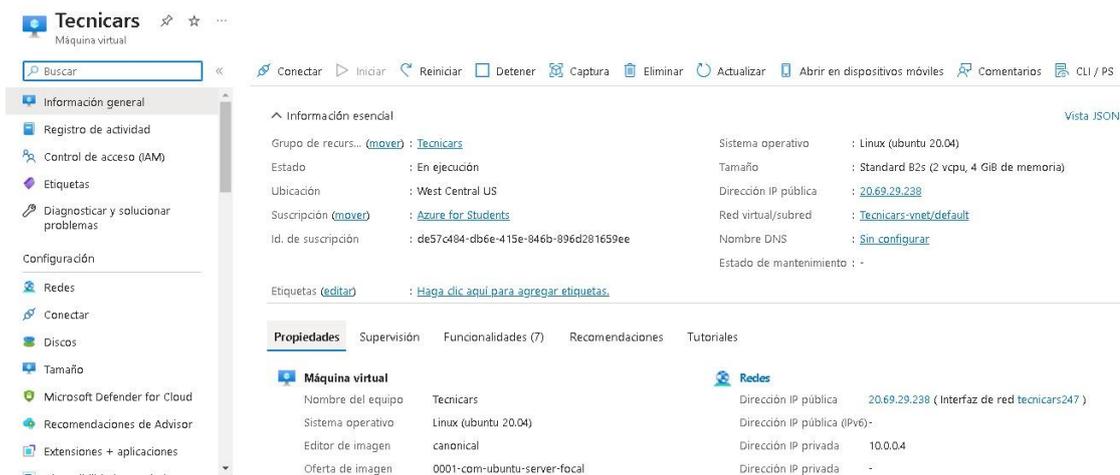
Seleccionar puertos de entrada \* (SSH (22)) ▼

**⚠ Esto permitirá que todas las direcciones IP accedan a la máquina virtual.** Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

**Figura 25.** Configuración de la Autenticación SSH

**Elaborado por:** El Investigador

Una vez que la máquina virtual ha sido desplegada, es posible visualizar las características del servidor web, así como su dirección IP pública. Esta información es crucial para evaluar el rendimiento del servidor y poder acceder a este desde cualquier ubicación con conexión a Internet. Además, esto facilita el monitoreo y la realización de modificaciones en el servidor de forma remota, lo cual representa una gran ventaja para el desarrollo y mantenimiento del proyecto.



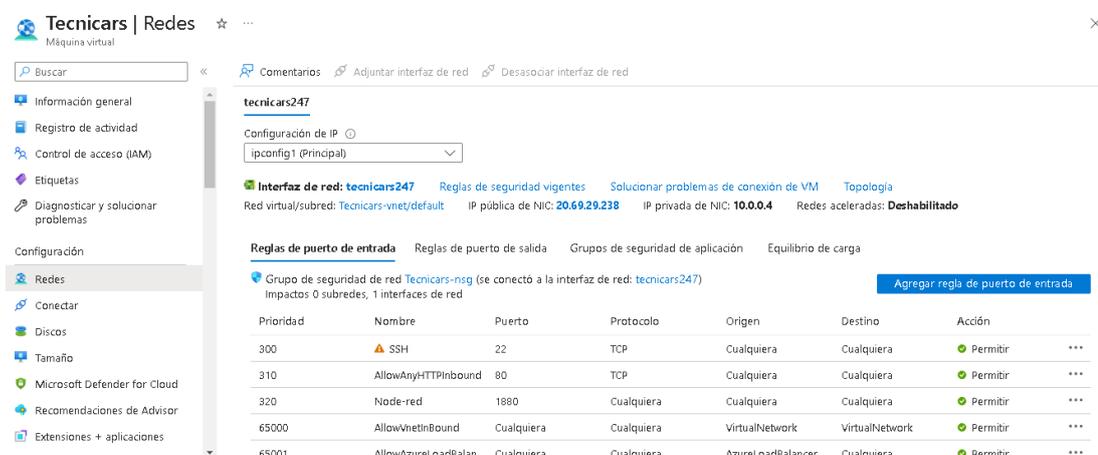
**Figura 26.** Servidor web creado

**Elaborado por:** El Investigador

Después de haber descargado la clave de acceso SSH, se estableció una conexión segura con el servidor virtual utilizando PUTTY. Una vez conectado, se tuvo acceso a la máquina virtual para llevar a cabo todas las configuraciones necesarias a fin de observar el funcionamiento adecuado de los servicios de Node-RED, Mosquitto y MariaDB. Esto permite la seguridad y confiabilidad del proyecto, lo que permitió un monitoreo eficiente de los datos provenientes del sistema a través de la cámara del sistema.

### Habilitación de los puertos en Azure

Con el fin de utilizar los servicios de Node-RED, Mosquitto, MariaDB y Apache, se procedió a habilitar los puertos correspondientes en la sección de reglas de seguridad del servidor web en Azure.



**Figura 27.** Habilitación de puertos en Microsoft Azure

**Elaborado por:** El Investigador

### 3.1.6 Instalación del Bróker Mosquitto

El Broker de Mosquitto es un intermediario de mensajes de código abierto que forma parte de la Fundación Eclipse. Funciona con el protocolo MQTT (Message Queuing Telemetry Transport), el cual permite a los dispositivos publicar mensajes o suscribirse a un tópico. Mosquitto es un software ligero y es ideal tanto para computadoras de baja potencia como para servidores de alto rendimiento. Para su instalación, se utilizaron los siguientes partes:

Primero se realizó una actualización de los paquetes de Ubuntu utilizando los siguientes comandos:

1. Se procedió a actualizar los paquetes de Ubuntu utilizando los siguientes comandos:

```
sudo apt-get update
```

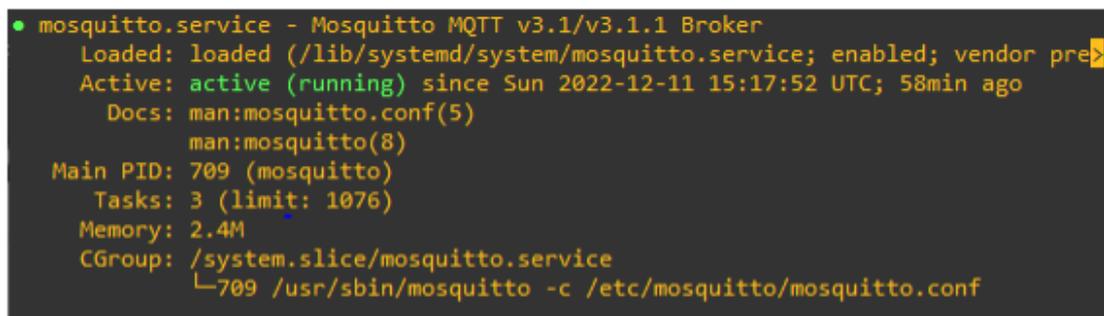
```
sudo apt-get upgrade
```

2. Luego, se llevó a cabo la instalación del Broker de Mosquitto a partir de los repositorios previamente actualizados, utilizando el siguiente comando:

```
sudo apt-get install mosquitto mosquitto-clients
```

3. Se utilizó el siguiente comando para verificar que la instalación se completó correctamente y que el servicio se está ejecutando adecuadamente:

```
service mosquitto status
```



```
● mosquitto.service - Mosquitto MQTT v3.1/v3.1.1 Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre
   Active: active (running) since Sun 2022-12-11 15:17:52 UTC; 58min ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
  Main PID: 709 (mosquitto)
    Tasks: 3 (limit: 1076)
   Memory: 2.4M
   CGroup: /system.slice/mosquitto.service
           └─709 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
```

**Figura 28.** Bróker de Mosquitto funcionando correctamente

**Elaborado por:** El Investigador

### 3.1.7 Instalación de MariaDB

A través de la terminal de PUTTY, el cual se encuentra conectado al servidor web de Ubuntu Server, se procedió a instalar el servicio de MariaDB utilizando los siguientes comandos:

1.- MariaDB fue instalado utilizando el siguiente comando:

```
sudo apt-get install mariadb-server
```

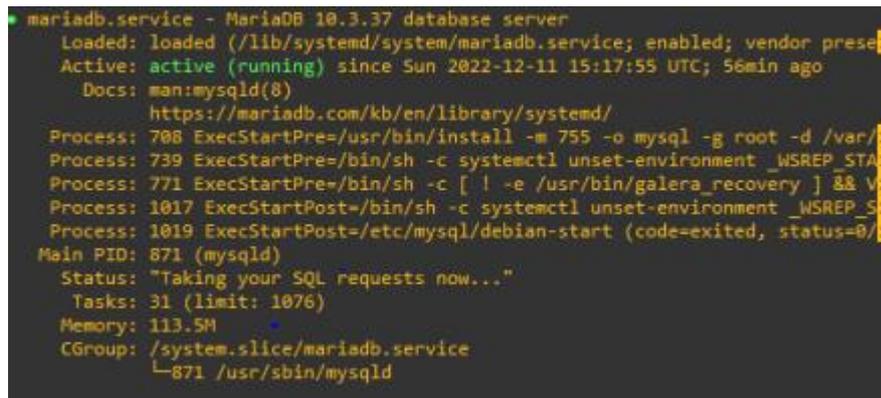
2.- Después de completar la instalación, se habilitó y se inició el servicio de MariaDB utilizando el siguiente comando:

```
sudo systemctl enable mariadb
```

```
sudo systemctl restart mariadb
```

3.- Se utilizó el siguiente comando para verificar el correcto funcionamiento del servicio de MariaDB.

```
sudo systemctl status mariadb
```



```
● 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:mysql(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 _WSREP_STA
   Process: 771 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && V
   Process: 1017 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_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 29.** Servicio de MariaDB funcionando correctamente

**Elaborado por:** El Investigador

Finalmente, se agregó una contraseña y se realizó una pequeña configuración con el siguiente comando:

```
sudo mysql_secure_installation
```

al cual se asignó una contraseña y se realizó una breve configuración.

### 3.1.8 Instalación del servidor Apache

Apache es un servidor web ampliamente utilizado que se emplea para alojar todos los archivos necesarios para visualizar un sitio web. La instalación de Apache resulta esencial para la gestión de la base de datos MariaDB y se llevó a cabo siguiendo los siguientes pasos:

1.- Se procedió a instalar el software Apache utilizando el siguiente comando:

```
sudo apt-get install apache2.
```

2.- Después, se habilitó y reinició el servicio de Apache mediante los comandos que se detallan a continuación:

```
sudo systemctl enable apache2
```

```
sudo systemctl restart apache2
```

Para confirmar que el servicio de Apache está funcionando adecuadamente, se ingresó la dirección IP del servidor (20.69.29.238) en cualquier navegador. Si la configuración del servidor es correcta, se mostrará un mensaje en la página, como se ilustra en la figura.



**Figura 30.** Servidor Apache funcionando correctamente

**Elaborado por:** El Investigador

### 3.1.9 Instalación de PHP MyAdmin para gestión de la base de datos

Este software proporciona una interfaz gráfica de usuario fácil de usar para administrar MariaDB. La instalación se llevó a cabo utilizando el siguiente comando:

```
sudo apt install phpMyAdmin
```

Con el fin de acceder a phpMyAdmin en Ubuntu Server, fue necesario realizar una configuración adicional en el servidor Apache. El archivo de configuración fue editado mediante el siguiente comando:

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

Se procedió a abrir un archivo en el que se añadió la siguiente línea de código:

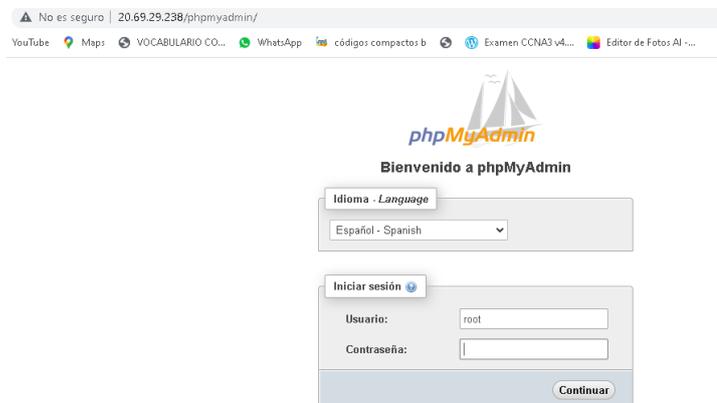
```
Include /etc/phpmyadmin/apache.conf
```

Las configuraciones fueron guardadas utilizando la combinación de teclas "Ctrl + x" y luego presionando la tecla "y" para confirmar. Después de guardar el archivo, se reinició el servicio de Apache mediante el siguiente comando:

```
sudo service apache2 reload
```

Con el fin de confirmar que el servicio de phpMyAdmin está funcionando adecuadamente, se accedió a la dirección correspondiente en el navegador.

<http://20.69.29.238/phpmyadmin>.



**Figura 31.** phpMyAdmin en el Navegador

**Elaborado por:** El Investigador

### 3.1.10 Instalación de Node-Red para creación y acceso a la interfaz gráfica

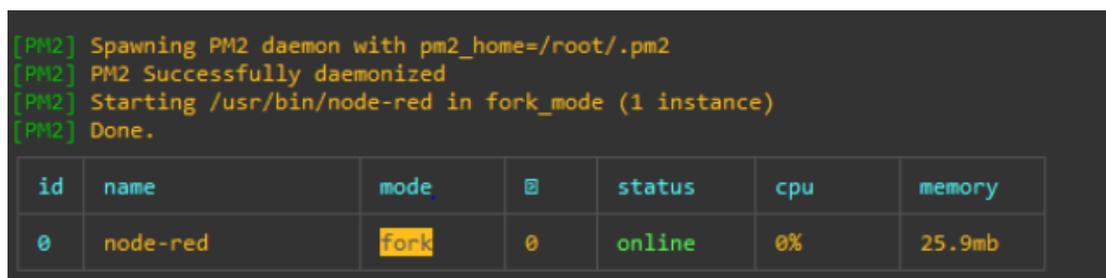
Se empleó la terminal de PUTTY para llevar a cabo la instalación de Node-RED en Ubuntu Server utilizando los comandos que se muestran a continuación:

Se procedió a instalar node.js y Node-RED utilizando los siguientes comandos:

```
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
```

lograr que Node-RED se inicie automáticamente cada vez que el servidor se inicie se utilizó PM2.

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



```
[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.
```

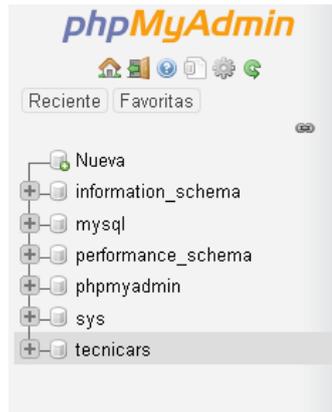
id	name	mode	Ⓜ	status	cpu	memory
0	node-red	fork	0	online	0%	25.9mb

**Figura 32.** pm2 instalado y ejecución en Node-Red

**Elaborado por:** El Investigador

### 3.1.11 Conexión con MySql

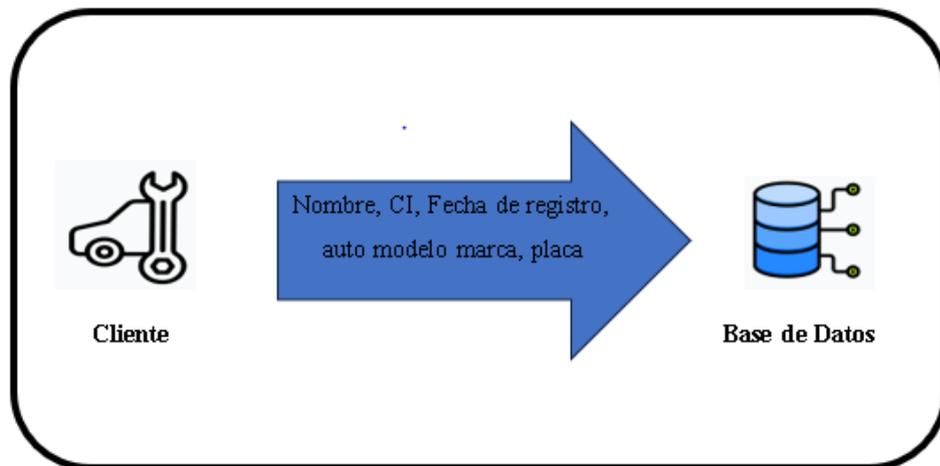
Para almacenar los datos que adquiere el sistema se necesita una base de datos que permita registrar todos los valores según la fecha. El administrador de base de datos phpMyAdmin se utiliza para este propósito. El primer paso es crear una base de datos llamada "Tecnicars David". Luego se crearon las tablas necesarias en la base de datos para almacenar estos valores.



**Figura 33.** Base de datos

**Elaborado por:** El Investigador

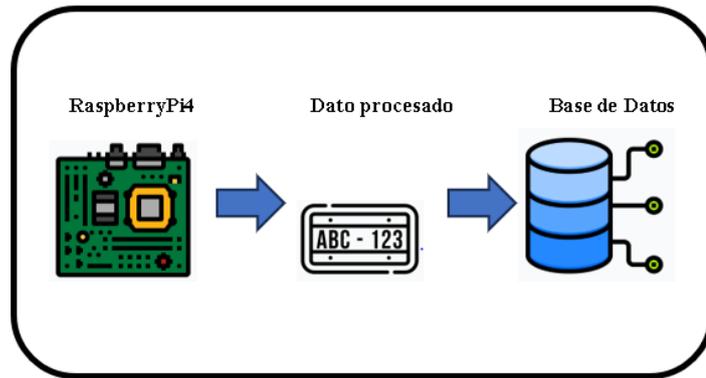
La primera tabla utilizada para almacenar los datos de los clientes se denomina "usuarios". En esta tabla se registran los siguientes datos de los usuarios que ingresan al taller de servicio automotriz: nombre, apellido, edad, documento de identificación, fecha de registro, vehículo, modelo.



**Figura 34.** Almacenamiento de los datos del cliente o usuario

**Elaborado por:** El Investigador

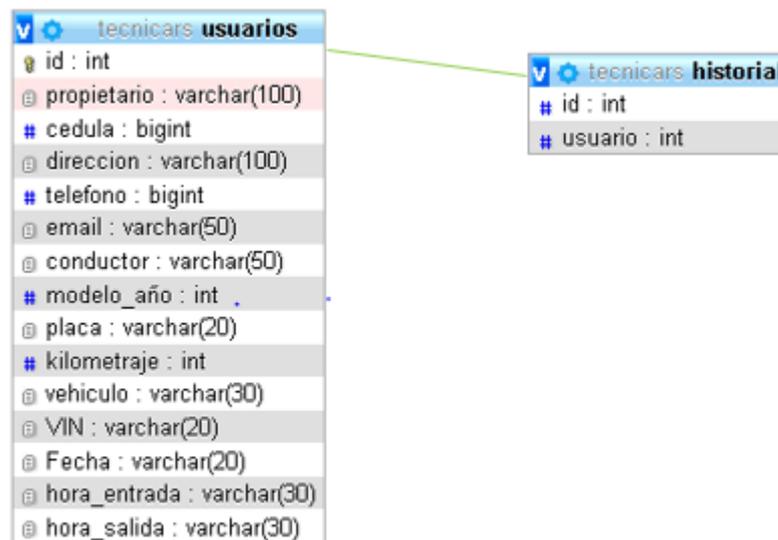
Además, existe otra tabla que se utilizó para almacenar el historial de trabajos realizados y también datos provenientes de la Raspberry Pi. Esta tabla registra datos de la placa del vehículo que es extraída por el sistema de registro automático.



**Figura 35.** Almacenamiento de los datos de la Raspberry Pi4

**Elaborado por:** El Investigador

Para establecer la relación entre las tablas, se utilizó la función de claves foráneas en la base de datos de MariaDB. En este caso, la tabla "usuarios" tiene un identificador único llamado "id", que se utiliza como clave primaria. Esta clave primaria se relaciona con el campo "\_id" en la tabla "historial", que actúa como clave foránea. Se puede observar en la siguiente figura cómo están relacionadas las tablas.



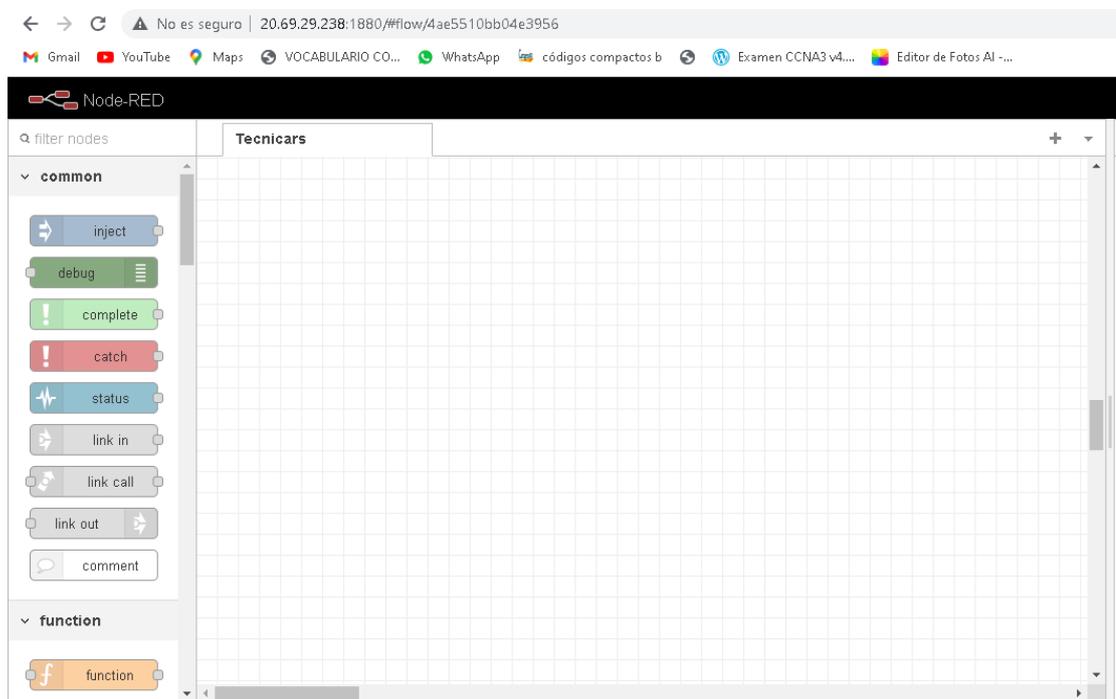
**Figura 36.** Relación de tablas por claves foráneas

**Elaborado por:** El Investigador

### 3.2 Configuración de Node-Red

Node-RED es un software de código abierto versátil que se puede ejecutar en una variedad de sistemas operativos, incluyendo 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 convierte en una elección óptima para aplicaciones de IoT. Ofrece una interfaz visual que permite la visualización y control en tiempo real, lo cual simplifica su manejo. Node-RED se basa en JavaScript, lo que le otorga flexibilidad para manejar una amplia gama de aplicaciones y crear funciones personalizadas según las necesidades de cada usuario. Para acceder a la interfaz de Node-RED, simplemente se utiliza la dirección IP del servidor junto con el puerto predeterminado de Node-RED, de la siguiente manera: <http://20.69.29.238/:1880>.



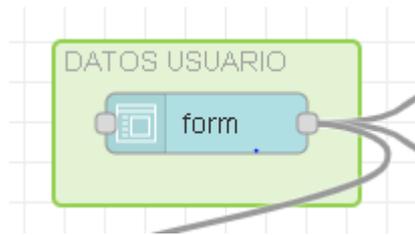
**Figura 37.** Interfaz web de Node-Red

**Elaborado por:** El Investigador

### 3.2.1 Configuración de Node-Red para el ingreso de datos de los usuarios-clientes

Para almacenar las identificaciones de los usuarios-clientes, se utiliza un sistema de base de datos donde los valores se ingresan manualmente. Sin embargo, el sistema de registro automático de autos proporciona automáticamente los datos de la placa del vehículo. Al tratarse de un nuevo usuario, es necesario completar manualmente los campos de nombre, propietario, dirección, correo electrónico, modelo/año del

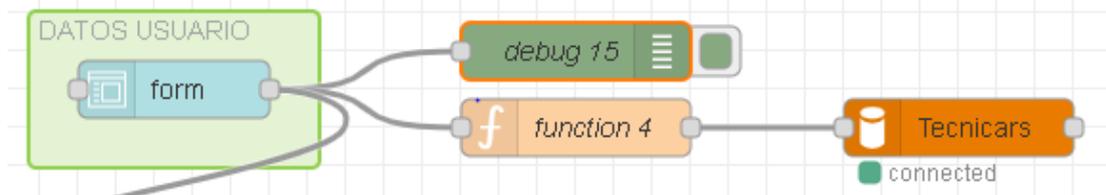
vehículo, edad y documento de identificación. Para llevar a cabo esta operación, se utiliza el nodo "form" en Node-RED.



**Figura 38.** Nodo Form

**Elaborado por:** El Investigador

En la figura 39 muestra un nodo que facilita la introducción de los datos de los usuarios. Para almacenar estos valores en la base de datos, se emplearon los nodos "function" y "mysql". La inserción de los datos se realizó mediante la sentencia SQL . “INSERT INTO usuarios (fecha, horario\_entrada, horario\_salida, propietario, cédula, dirección, teléfono, email, conductor, modelo, placa, kilometraje, vehículo, vin). La conexión de estos nodos se muestra en la figura 40.



**Figura 39.** Ingreso de datos a la base de datos

**Elaborado por:** El Investigador

En la figura 40, se puede observar la verificación (connected), esto indica que la conexión con la base de datos de MariaDB se ha establecido correctamente.

### 3.2.2 Configuración de Node-Red para almacenar los datos que arroja la Raspberry Pi4

Se utilizó un nodo de función para establecer la conexión con la base de datos, y este nodo está configurado con una sentencia SQL. Esta sentencia SQL recopila todos los valores necesarios y los guarda en la base de datos.



**Figura 40.** Nodo Función

**Elaborado por:** El Investigador

Para lograr la inserción de los valores desde Node-RED hacia la base de datos, se empleó el nodo "mysql", el cual facilita esta conexión una vez que se ha realizado una configuración previa. Esta configuración incluye proporcionar el nombre de la base de datos, y los datos referentes al usuario para establecer la conexión.



**Figura 41.** Nodo mysql

**Elaborado por:** El Investigador

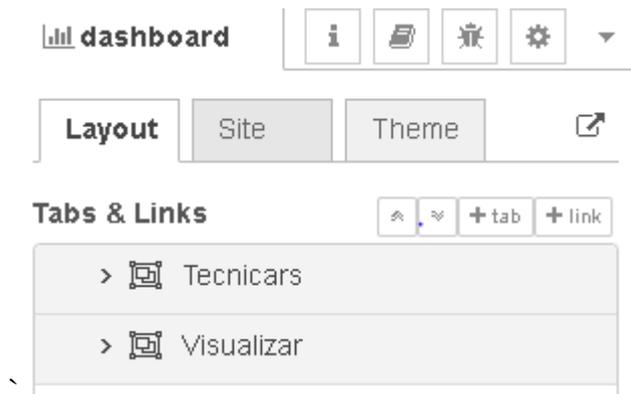
Para adquirir los datos que serán almacenados posteriormente, se empleó el nodo "MQTT in" en Node-RED. Este nodo posibilita obtener todos los datos provenientes de la tarjeta Raspberry Pi4. Los datos provenientes de la tarjeta están en formato JSON, por lo tanto, se requiere convertirlos en objetos JavaScript para poder manipularlos. El nodo "MQTT in" actúa como suscriptor al Broker de Mosquitto instalado en el servidor Ubuntu Server, y se suscribe al tópico "datos/raspberry".



**Figura 42.** Nodo mqtt in y Nodo json

**Elaborado por:** El Investigador

Con el fin de estructurar la interfaz del programa, se generaron instancias contenedoras adaptadas a las necesidades particulares del usuario. Estas instancias se presentan en la figura 43, donde se crearon tablas que agrupan nodos y texto.



**Figura 43.** Creación de tablas y grupos

**Elaborado por:** El Investigador

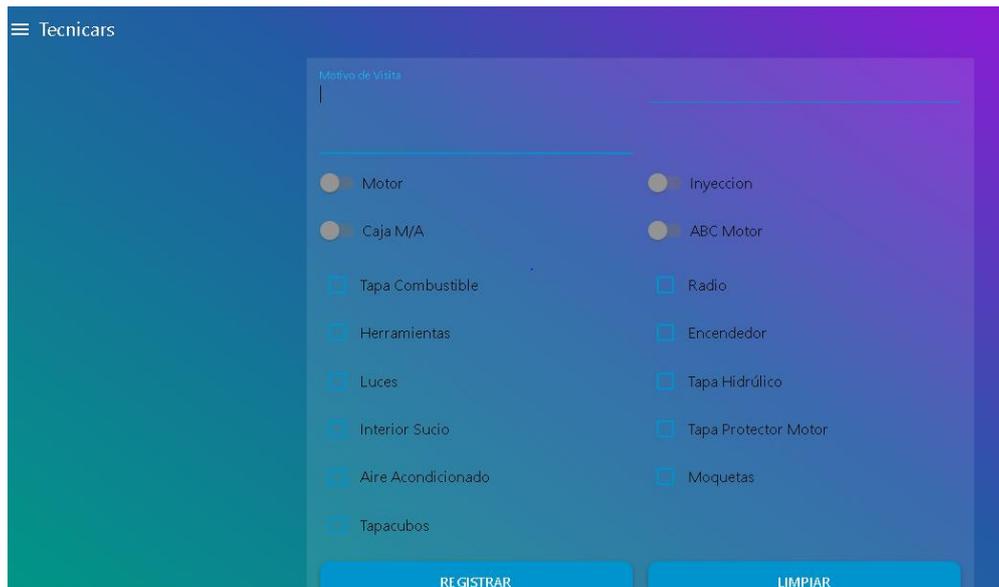
En la figura 44 se puede observar cómo está distribuida los datos de ingreso.

**Figura 44.** Interfaz del Dashboard para el ingreso de Usuarios

**Elaborado por:** El Investigador

En esta parte de la interfaz gráfica se lleva a cabo la introducción de los datos del usuario/cliente. Se pueden visualizar los detalles requeridos del vehículo, como el modelo y la marca. Una vez ingresados los datos, se ofrece la opción de registrar o limpiar los campos.

En la figura 45 se puede observar cómo está distribuido en el dashboard.



**Figura 45.** Interfaz del Dashboard /Control de ingreso de vehículos

**Elaborado por:** El Investigador

Esta parte contiene la sección de Motivo de visita en donde se detalla los trabajos requeridos para el auto. Después se tiene una serie de opciones para escoger como son Tapa combustible, herramientas, luces, aire acondicionado entre otros para verificar como ingresa el automóvil, esto se realizó acorde a la forma de registro manual que poseía el taller de Servicio “Técnicars”.

En la figura 46 se puede observar una proforma.

Proforma

CANT.	REPUESTOS	CANT.	MANO DE OBRA
#	Rep	#	#Obra
#	Rep	#	#Obra
#	Rep	#	#Obra
#	Rep	#	#Obra
#	Rep	#	#Obra
#	Rep	#	#Obra
#	Rep	#	#Obra
#	Rep	#	#Obra

REGISTRAR

LIMPIAR

**Figura 46.** Interfaz del Dashboard /Proforma requerimientos

**Elaborado por:** El Investigador

Esta proforma contiene espacios para ubicar la cantidad de repuestos y la mano de obra que requiere el vehículo.

En la figura 47 se observa un check list de vehículos para verificar el buen funcionamiento del vehículo.

**Check List de Calidad**

NIVELES DE LIQUIDO	Técnico Niveles
<input type="checkbox"/> Aceite	<input type="checkbox"/> Líquido frenos
<input type="checkbox"/> Líquido embrague	<input type="checkbox"/> Líquido refrigeración
<input type="checkbox"/> Líquido dir hidráulica	<input type="checkbox"/> Ajuste tapón cárter
<input type="checkbox"/> Ajuste tapones de transmisión	
INSPECCIONES FUGAS	Técnico Fugas
<input type="checkbox"/> Sistema de combustible mangueras/ductos/cañerías	<input type="checkbox"/> Sistema de admision mangueras/ductos/cañerías
<input type="checkbox"/> Sistema de liquido refrigerante	<input type="checkbox"/> Motor
<input type="checkbox"/> Transmision	<input type="checkbox"/> Caja de cambios
<input type="checkbox"/> Cañerías de freno	<input type="checkbox"/> Embrague

**Figura 47.** Interfaz del Dashboard /Check list de calidad

**Elaborado por:** El Investigador

El formulario de verificación de calidad contiene una lista exhaustiva de todos los componentes que deben ser inspeccionados para asegurar el óptimo rendimiento del automóvil.

Finalmente se muestra en la figura 48 un apartado de Cliente notas



**Figura 48.** Interfaz del Dashboard /Cliente notas

**Elaborado por:** El Investigador

Este apartado muestra las recomendaciones que se pueden dar a los clientes sobre los automóviles, además de llevar un historial de las llamadas suscitadas por los clientes.

### **3.3 Flujo de procesos del Sistema de Detección**

En esta sección se detalla el desarrollo de todos los procesos que componen el sistema de registro automático.

La conexión entre la tarjeta Raspberry PI4 y el servidor de Ubuntu Server en la nube de Azure se establece al detectar el dato de la placa del vehículo, el cual se visualiza en la interfaz del sistema como se muestra en la siguiente figura.

The image shows a web form titled "Usuario" with a purple-to-blue gradient background. The form contains the following fields:

- Fecha: A date input field.
- Hora Entrada: A time input field with a dropdown arrow.
- Hora Salida: A time input field with a dropdown arrow.
- Propietario: A text input field.
- R.U.C./C.I.: \* A text input field.
- Direccion: A text input field.
- Telf: A text input field.
- E-mail: A text input field.
- Conductor: A text input field.
- Modelo/Año: A text input field.
- Placa: \* PTR-756: A text input field containing the license plate number "PTR-756". A red arrow points to this field.
- Km: A text input field.
- Vehiculo: A text input field.
- VIN: A text input field.

At the bottom of the form, there are two buttons: "REGISTRAR" (left) and "LIMPIAR" (right).

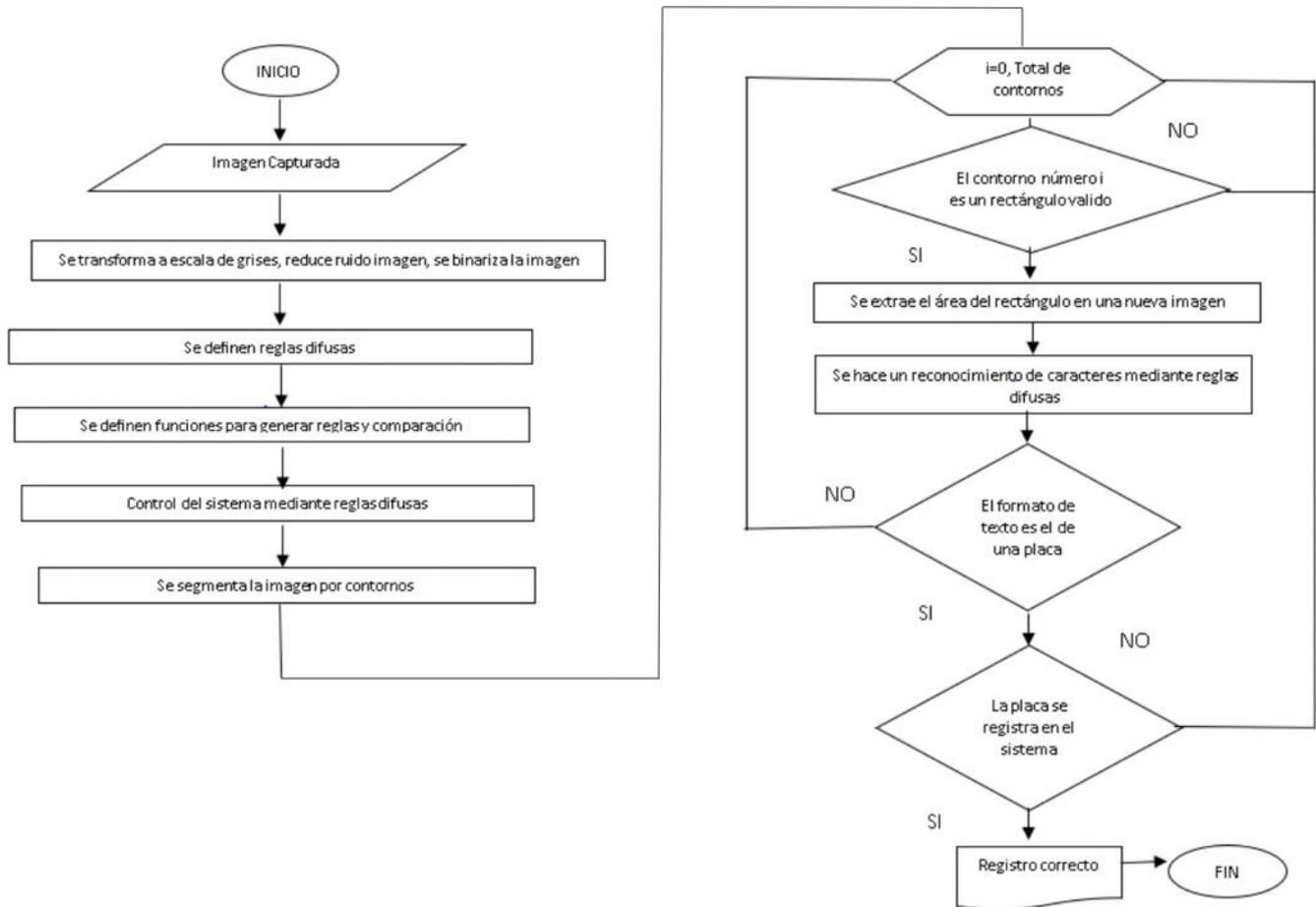
**Figura 49.** Dato de la placa enviado desde la Raspberry Pi4 al servidor

**Elaborado por:** El Investigador

El proceso comienza al encender la tarjeta Raspberry Pi4, la cual está equipada con el programa de registro automático en Python. En esta parte de la interfaz gráfica, se configura la conexión Wi-Fi de la tarjeta Raspberry Pi4 para conectarse a la red que tiene acceso a Internet, ingresando el nombre de la red y la contraseña correspondientes al Taller Tecnicars. Si la conexión se realiza con éxito, de acuerdo a la programación establecida, se establece la comunicación con el bróker MQTT ubicado en el servidor de Ubuntu Server en la nube de Azure, y se establece la conexión con el bróker Mosquitto y el Dashboard (tablero de control).

Para que se realice la transmisión de los datos de la placa del vehículo desde la tarjeta Raspberry Pi4 al servidor, es fundamental que la conexión sea exitosa. Los datos obtenidos de las placas de los vehículos forman parte de un proceso conocido como procesamiento de datos, en el cual la tarjeta Raspberry Pi4 se encarga de procesar la información para su posterior envío al bróker a través de la conexión Wi-Fi establecida. El bróker recibe el mensaje que contiene los datos de la placa. En la siguiente sección, los datos recibidos se presentan en la interfaz gráfica que permite

visualizar la información de la placa, tal como se muestra en la figura previa. Todo este proceso tiene lugar cuando el dispositivo está encendido y en funcionamiento.



**Figura 50.** Diagrama de flujo de Sistema de Registro

**Elaborado por:** El Investigador

El sistema inicia con la entrada de video, que proviene de la cámara, este dispositivo detecta movimiento y el programa captura la imagen. El programa está configurado para capturar un objeto de determinadas dimensiones, en este caso el vehículo, el sistema asume que es un vehículo y procesa digitalmente la imagen. A continuación, la imagen se procesa digitalmente, aplicando ajustes de perspectiva, correcciones y eliminación de ruido. Utilizando la Lógica Difusa, se busca y reconocen los caracteres de la placa del vehículo. Para obtener los datos del vehículo, se realiza una consulta al servidor, mientras que en la base de datos local se verifica si se han realizado trabajos

previos en ese auto. Finalmente, la información se muestra en la pantalla y se registra la entrada del nuevo vehículo en la base de datos del programa.

Una vez se han obtenido los caracteres de la placa, se realiza una consulta a la base de datos del servidor para obtener información sobre el usuario o cliente en caso de que el vehículo esté registrado.

### **3.4 Reconocimiento de caracteres de placa vehicular con Lógica Difusa**

El método de reconocimiento utiliza la transformación de características invariantes de escala integrado en conjunto con un algoritmo en base a Lógica Difusa.

La segmentación separa cada carácter en un área de placa para obtener la información requerida acerca de la placa.

En este trabajo de titulación se utilizó el algoritmo SIFT que es un método para extraer puntos característicos invariantes y distintivos de una imagen que pueden ser usados para mejorar la correspondencia entre dos vistas diferentes de un objeto o una escena [48], y permite la extracción de cada rasgo de las letras y números obtenidos de la placa. Las herramientas utilizadas para el reconocimiento de caracteres por medio de la Lógica Difusa analizan las características obtenidas por el algoritmo SIFT y detecta de manera correcta los caracteres.

El programa utiliza la segmentación de la imagen para poder ubicar el área en donde se encuentra la placa.

Internamente el programa realiza el proceso de localizar la placa mediante la siguiente ecuación

$$h = \text{CEILING}(0.5263(H)) \quad (1)$$

Donde h = Altura del carácter

H = Altura total de la placa

Se observa que en la ecuación (1), la altura del carácter de la placa es igual al 52,63% de la altura total de la placa. La palabra CEILING representa una función que se usa para redondear un número hacia arriba hasta el próximo número entero

Por otra parte, el ancho del carácter único en la placa es igual al 9.81% del ancho total de la placa, que da la ecuación:

$$w = \text{CEILING}(0.0981(W)) \quad (2)$$

Donde  $w$  = Ancho del carácter

$W$  = Ancho total de la placa

Una vez que se determinada la altura y el ancho de los caracteres, es importante asegurarse de que cada esquina de estos esté correctamente posicionada y eso se puede ubicar mediante la siguiente ecuación

$$L(x, y)_n = \begin{pmatrix} x = \varphi_n W \\ Y = 0.203 H \end{pmatrix} \quad (3)$$

Donde  $L(x, y)_n$  = ubicación de la esquina del caracter n

$\varphi_n$  = coeficiente de área de caracter n

$W$  = total del ancho de la placa

$H$  = total de alto de la placa

El coeficiente de área se determina experimentalmente y produce un valor con respecto a la relación de la posición de la placa y toda la dimensión de la placa en estudio.

### 3.4.1 La Transformación de Características Invariantes de Escala

La Transformación de Características Invariantes de Escala, conocida como SIFT, es una técnica utilizada para detectar, extraer y describir características locales en imágenes. Esta técnica es capaz de proporcionar características que son invariantes a la escala y la rotación, y en cierta medida, inmunes a los cambios en la iluminación [48].

Las características extraídas mediante SIFT se localizan tanto en el dominio espacial como en el de frecuencia. Esto se realiza con el objetivo de reducir las interferencias en la comparación de características clave causadas por el ruido, la oclusión y el desorden presentes en las imágenes [48].

El algoritmo SIFT comienza ejecutando una selección de puntos clave en el espacio de escala, lo que implica buscar computacionalmente puntos de interés potenciales que

sean invariantes a la escala y la orientación. Este proceso de búsqueda se lleva a cabo en diferentes escalas y ubicaciones de la imagen utilizando la diferencia de funciones gaussianas [48].

La diferencia de gaussianas (DoG) se calcula mediante la diferencia de las convoluciones de la imagen con dos niveles adyacentes de la función gaussiana utilizando un factor de escala "k". Para una imagen de placa "P(x,y)", el DoG se puede resolver utilizando la siguiente ecuación:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * P(x, y) \quad (3)$$

Donde : G = Función Gaussiana

P = Imagen de la placa

K = Factor de Escala

Donde  $G(x,y,\sigma)$  representa la convolución de la imagen con la función gaussiana de desviación estándar " $\sigma$ ".

Este cálculo del DoG es fundamental en el algoritmo SIFT, ya que permite detectar y resaltar regiones de la imagen que presentan cambios significativos en la escala y la orientación. Estas regiones son consideradas puntos de interés potenciales y sirven como base para la extracción y descripción de características locales utilizando el algoritmo SIFT [48].

El proceso de selección de puntos clave en el algoritmo SIFT implica comparar los valores de diferencia de Gaussian (DoG) con un umbral establecido. Estos valores se comparan con los máximos en cada nivel y también se comparan con los máximos de los píxeles vecinos, tanto en niveles superiores como inferiores [48].

La ubicación máxima hace referencia a las posiciones en las cuales se encuentran los puntos clave con características distintivas en la imagen. Estas ubicaciones máximas corresponden a lugares específicos en la imagen donde se ha identificado la presencia de un punto clave con características importantes para el proceso de detección y descripción de la imagen [48].

Los puntos clave se refiere a ubicaciones específicas en una imagen que son seleccionadas debido a sus características distintivas y estabilidad frente a cambios en la escala, rotación y orientación. Estos puntos clave son utilizados en el proceso de detección y descripción de características en la imagen, y son fundamentales para el reconocimiento y análisis de patrones [48].

En cada ubicación máxima, se utiliza un modelo detallado para determinar la ubicación, escala y contraste del punto clave. A partir de aquí, se seleccionan dichos puntos en función de su estabilidad ante variaciones en estos factores. Luego, se asignan una o más orientaciones a los puntos clave según las propiedades de la imagen. Esto se hace considerando la invariancia de las características con respecto a la escala, rotación y orientación. Una vez que se han asignado las orientaciones, se describen todos los puntos clave utilizando información local de la imagen. Esta descripción se basa en los gradientes medidos en regiones alrededor de cada punto clave. Los gradientes se transforman en una representación que ofrece invariancia a otras características, como la distorsión de la forma local y los cambios en la iluminación [48].

El algoritmo SIFT utiliza una serie de pasos para seleccionar puntos clave, asignarles orientaciones y describirlos utilizando características locales. Este proceso proporciona invariancia frente a la escala, rotación, orientación y otras variaciones en la imagen, lo que permite una detección y descripción robusta de características en diferentes condiciones.

Las características extraídas mediante el algoritmo SIFT serán utilizadas para comparar con las placas vehiculares. Sin embargo, es importante tener en cuenta que estas características no son valores discretos y pueden presentar desviaciones en diferentes instancias. Para detectar las características clave, que se refiere a las propiedades o atributos importantes a extraer de las imágenes de las placas de vehículos incluso con ciertas desviaciones, se propone la aplicación de herramientas de la Lógica Difusa.

La Lógica Difusa utiliza variables lingüísticas, que son las entradas y salidas del sistema, expresadas como cadenas de caracteres que describen un estado particular. El proceso de Lógica Difusa consta de tres etapas principales: fuzzificación, inferencia lógica y defuzzificación. En la etapa de fuzzificación, las variables de entrada se convierten en sus contrapartes difusas mediante el uso de funciones de membresía. Estas funciones asignan grados de pertenencia a los diferentes valores de entrada, transformando así la entrada no difusa en una forma difusa.

La inferencia lógica implica colocar las características extraídas en un conjunto de reglas para inferir los caracteres deseados. Estas reglas se evalúan utilizando operaciones de conjuntos borrosos, y se realizan las operaciones necesarias para compilar los resultados obtenidos. Finalmente, en la etapa de defuzzificación, se convierten los resultados difusos en valores concretos o crisp, utilizando diferentes tipos de funciones de pertenencia, como triangulares, trapezoidales, singleton, gaussianas y lineales por partes [49].

En resumen, la Lógica Difusa aprovecha variables lingüísticas y utiliza funciones de membresía para convertir las entradas y salidas en formas difusas. La inferencia lógica se basa en reglas y operaciones de conjuntos borrosos, y la defuzzificación convierte los resultados difusos en valores concretos. Estos procesos permiten manejar la imprecisión y la incertidumbre en el análisis de las características y obtener resultados más precisos y adaptados al contexto

Las herramientas de la Lógica Difusa proporcionan técnicas para construir sistemas que emulen el proceso de toma de decisiones humano. En este algoritmo, se percibe el razonamiento exacto como un limitante y se enfoca en el razonamiento aproximado. Las técnicas de la Lógica Difusa permiten manejar la imprecisión y la incertidumbre en los datos al asignar valores de pertenencia a cada característica, en lugar de tener una clasificación binaria [49].

### 3.4.2 Resultados de Segmentación de Imágenes

Antes de la extracción de las características locales, la imagen de la placa se someterá primero a la segmentación de la imagen.

El coeficiente de área de cada placa se deriva primero de la figura 51.

La figura 51 muestra los tamaños y dimensiones estándar de las placas en el Ecuador.



**Figura 51.** Dimensión de la placa de matrícula. [48]

La tabla 12 muestra los coeficientes de área que se utilizaron. Esto se derivó experimentalmente de la figura 52.

**Tabla 13.** Coeficientes de área. [48]

N	$\varphi_n$
1	0.072
2	0.193
3	0.305
4	0.490
5	0.607
6	0.721
7	0.838

**Elaborado por:** El Investigador

A continuación, se observa el algoritmo de segmentación de imágenes y se prueba en la figura 53.

La figura 53 se utilizó para probar todo el algoritmo y se utilizó en los pasos siguientes del programa.



**Figura 52.** Ejemplo de una placa. [48]

Después de aplicar el código para segmentar la imagen, se identificaron siete caracteres que conforman el número de la placa, como se muestra en la figura 53.



**Figura 53.** Caracteres Segmentados de la placa. [48]

Cada caracter de la placa se analizará utilizando la transformación de característica invariante de escala que extraerá la placa y la almacenará en una base de datos.

### **3.4.3 Resultados de Extracción de Características**

#### **3.4.3.1 Construcción de la pirámide gaussiana**

La construcción de la pirámide gaussiana es un proceso fundamental para visualizar y analizar una figura en diferentes niveles de escala. Antes de comenzar la construcción, es necesario seleccionar un factor de escala, que determinará la cantidad de niveles que se generarán en la imagen [48].

Para calcular el número de niveles, se utiliza la siguiente ecuación:

$$Levels = floor \left( min \left( \frac{\log \left( 2 * \frac{size(P)}{12} \right)}{\log(k)} \right) \right) \quad (4)$$

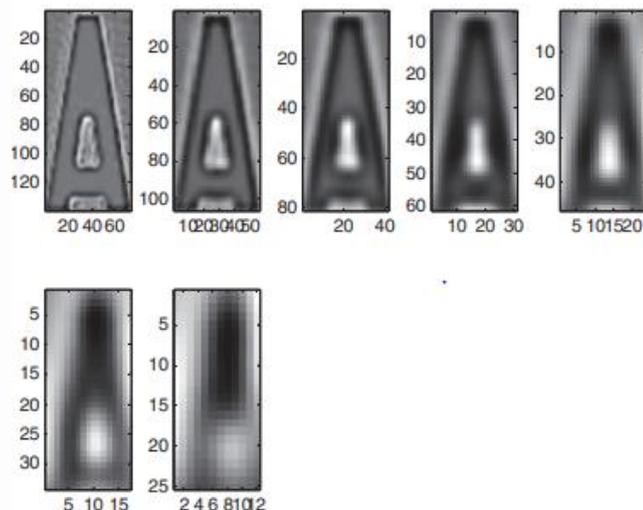
Donde: Niveles = Número de Niveles

P = Imagen de la placa

K = Factor de escala

La función "floor" se utiliza para redondear un número hacia abajo hasta el siguiente número entero, mientras que min es el menor valor dentro de un grupo de números.

En el programa, el número deseado de niveles es 7, por lo que el factor de escala que se utilizará es 1.3, que se reunió de forma experimental. A cada imagen se ampliará una vez y se reducirá en función del número de niveles, que se refiere a la cantidad de escalas diferentes en las cuales se muestra una imagen a medida que se va aumentando o disminuyendo su tamaño que luego se extraerán usando (3) para obtener las pirámides gaussianas. La figura 55 muestra las imágenes de la pirámide gaussiana.

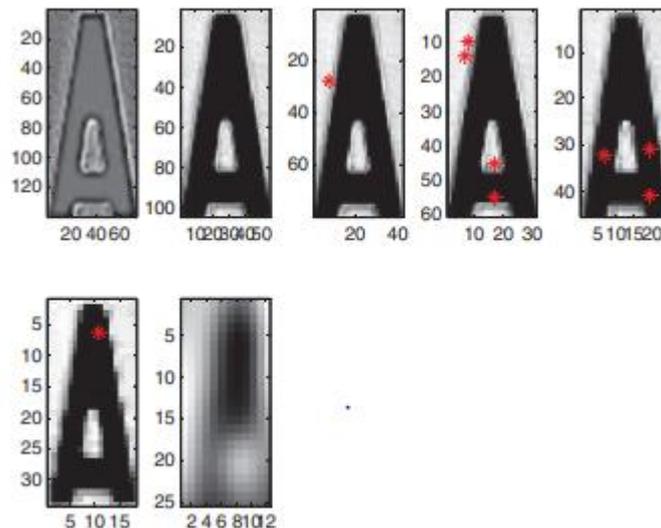


**Figura 54.** Imagen en escala de pirámide gaussiana para "A" [48].

### 3.4.3.2 Ubicación del punto máximo

La siguiente fase es localizar los máximos valores obtenidos de (3) a diferentes escalas. Este proceso implica la comparación de puntos que tienen diferentes valores de DoG en comparación con los puntos adyacentes en un radio especificado. El "radio específico" se refiere a la distancia o el tamaño de una vecindad en la que se comparan los puntos en el proceso de localización de los máximos de la diferencia de Gaussian (DoG). El radio utilizado en este trabajo es de 2 píxeles ya que el tamaño de cada caracter es muy pequeño. Los "2 píxeles", se refiere a que, en el proceso de localización de los máximos de la diferencia de Gaussian (DoG), se comparan los valores DoG de un punto con los valores DoG de los puntos vecinos que se encuentran a una distancia de 2 píxeles en todas las direcciones (arriba, abajo, izquierda y derecha). Es decir, se forma una vecindad cuadrada de 5x5 píxeles alrededor del punto, y se compara el valor DoG del punto central con los valores DoG de los puntos en esa vecindad para determinar si es un máximo local en esa escala específica. Esto ayuda a encontrar las ubicaciones máximas y destacadas en las características detectadas en la imagen. También se comparó con los niveles por encima y por debajo de él.

La figura 56 muestra los puntos máximos de "A" obtenidos en el algoritmo. Solo mostró puntos máximos en los niveles 3, 4, 5 y 6.

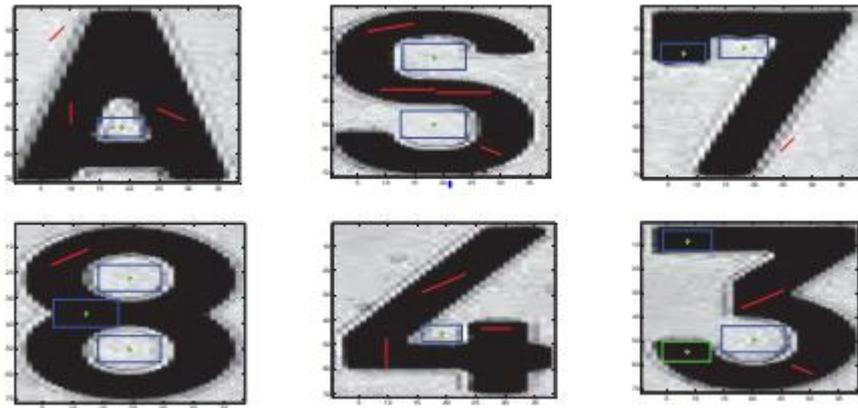


**Figura 55.** Puntos máximos extraídos para "A" [48].

### 3.4.3.3 Refinamientos de características

Las características obtenidas se refinan calculando varios parámetros, como el valor de escala local de la entidad, el tamaño de la entidad en la imagen original, la bandera que indica si es un borde o no, el ángulo de orientación del borde local y la curvatura de la respuesta a través del espacio de escala [48].

Dado que se han detectado las características, ahora está listo para mostrarse y usarse para el reconocimiento óptico de caracteres. La gráfica de las entidades detectadas en "A", "S", "7", "8", "4" y "3" se muestra en la figura 55.



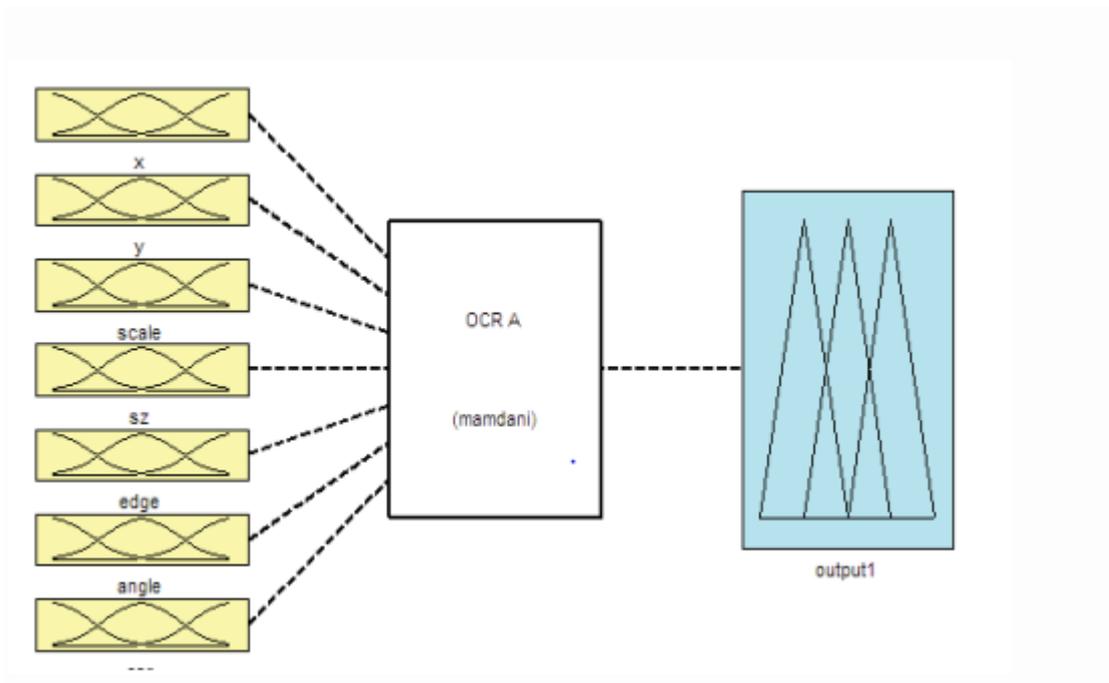
**Figura 56.** Características determinadas en los caracteres de la placa [32].

Dado que se ha creado un documento en donde se encuentran las letras con el grado de pertenencia de las características para los caracteres antes mencionados, se incorporará como entradas a la función de membresía o función de pertenencia de la Lógica Difusa para el reconocimiento de placas.

### 3.4.3.4 Resultados de la Lógica Difusa

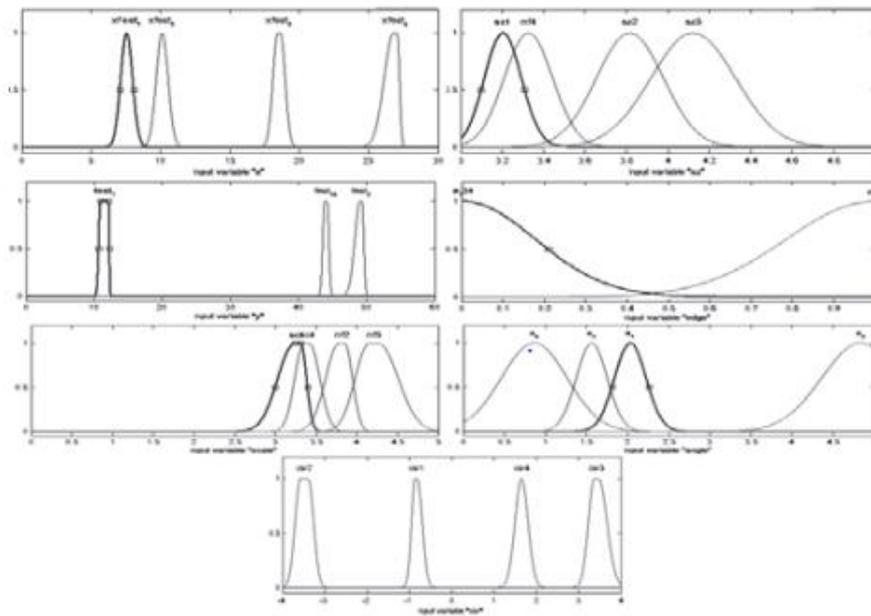
Las características extraídas en el algoritmo SIFT se utilizarán como valores en las funciones de pertenencia del sistema de Lógica Difusa. Cada característica tendrá su propio algoritmo de Lógica Difusa, y en cada conjunto de algoritmos se utilizarán 7 funciones de membresía.

En el contexto del formato de placas en el Ecuador, se han extraído los 7 caracteres específicos que pueden ser detectados por el sistema. Estos caracteres serán utilizados en el diseño de las funciones de pertenencia correspondientes en el sistema de Lógica Difusa construido.



**Figura 57.** Diagrama de bloques de Lógica Difusa [32].

El sistema de Lógica Difusa está compuesto por varias funciones de membresía organizadas en diferentes categorías, como x, y, escala, tamaño, borde, ángulo y CSR (relación de contraste y estructura). Estas funciones de membresía se utilizan para evaluar y procesar los datos obtenidos en imágenes que contienen placas de vehículos[32].

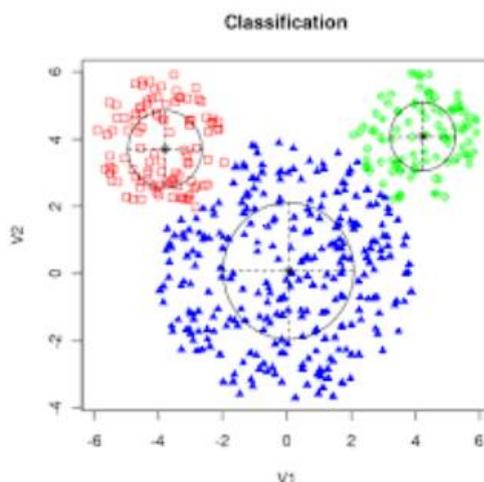


**Figura 58.** Membresía Función del sistema OCR [32].

### 3.5 Machine Learning aplicado al historial de la base de datos del sistema de registro

Se utilizó un algoritmo que posee la capacidad de aprender. Se alimentó al algoritmo con información acerca de proformas de autos que fueron revisados en el taller de servicios automotriz e identificó que hacer cuando se registra un auto que ya tiene previos trabajos realizados. Se utilizó Aprendizaje Supervisado por lo que el algoritmo empleado "Random Forest" para la base de datos del sistema cuenta con información de trabajos previos realizados en los autos y el programa puede hacer una predicción de los trabajos ya hechos en los autos cuando se registra la placa. El algoritmo analiza los datos asociados a las placas y encuentra patrones que le permiten asignarlas a un grupo correspondiente al historial de trabajos realizados. De esta manera, el sistema puede mostrar el historial de trabajos previos relacionados con una placa ya registrada.

El uso de Machine Learning en el sistema de registro automático brinda la capacidad de automatizar el proceso de identificar y rastrear los trabajos previos realizados en los autos, lo que facilita la toma de decisiones y proporciona un historial completo de los servicios realizados en cada vehículo.



**Figura 59.** Gráfico de un algoritmo de Clasificación. [49]

### 3.5.1 Programación del algoritmo en Python

El siguiente código utiliza el clasificador RandomForestClassifier para realizar una clasificación supervisada en un conjunto de características y etiquetas. Se divide el conjunto de datos en conjuntos de entrenamiento y prueba, se entrena el clasificador utilizando los datos de entrenamiento y se realizan predicciones en los datos de prueba. Finalmente, se calcula la precisión del clasificador comparando las etiquetas predichas con las etiquetas reales del conjunto de prueba.

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Crear conjunto de características y etiquetas
X = [[2, 4], [3, 1], [5, 6], [7, 8]] # Ejemplo de características
y = [0, 1, 1, 0] # Ejemplo de etiquetas (0 o 1)

# Dividir el conjunto de datos en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Crear el clasificador y entrenarlo
classifier = RandomForestClassifier()
classifier.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba
y_pred = classifier.predict(X_test)

# Calcular la precisión del clasificador
accuracy = accuracy_score(y_test, y_pred)
print("Precisión:", accuracy)

```

**Figura 60.** Librería Sklearn para clasificación Supervisada

**Elaborado por:** El Investigador

En código, se ha incluido un bloque try-except para manejar posibles errores durante el proceso, como errores de división de conjuntos o errores inesperados. También se ha agregado validación cruzada utilizando la función `cross_val_score` para obtener una evaluación más robusta del rendimiento del clasificador.

### 3.6 Funcionamiento del Sistema

#### 3.6.1 Procedimiento para la obtención de la placa del vehículo

El proceso empieza con la detección de movimiento en la entrada al taller. Para detectar el movimiento a través de la programación del algoritmo encargado de reconocer figuras rectangulares. Se asume que el primer cuadro es vacío y se utiliza como referencia para identificar cualquier cambio o presencia de movimiento.



**Figura 61.** Arquitectura del Sistema Automático de Registro

**Elaborado por:** El Investigador

Para detectar movimiento observamos el cuadro color rojo y éste servirá de referencia para detectar al auto.



**Figura 62.** Detección de Imagen en Movimiento

**Elaborado por:** El Investigador

El programa captura una imagen en movimiento y procesa la parte contenida dentro del recuadro rojo. Luego, se aplica una función de la librería OpenCV para resaltar los bordes de la imagen, y así poder procesar la parte rectangular de la placa.



**Figura 63.** Captura de imagen a procesar

**Elaborado por:** El Investigador

Ahora, el programa procede a realizar el procesamiento de la imagen basado en segmentación usando herramientas de la Lógica Difusa para extraer las letras de la imagen.



**Figura 64.** Parte de la imagen a procesar  
**Elaborado por:** El Investigador

### Procesamiento de Imagen y Extracción de caracteres



**Figura 65.** Imagen en movimiento capturada por la Cámara  
**Elaborado por:** El Investigador

Se aumenta el contraste de la imagen capturada a través del uso de la librería Open CV mediante la detección de movimiento y se le aplica un efecto de difuminación o suavizado Gaussiano.



**Figura 66.** Imagen transformada a escala de grises y detección de bordes

**Elaborado por:** El Investigador

La imagen se transforma a escala de grises y también se realiza la detección de bordes gracias a la librería Open CV con `cv2.canny()`.

Ahora se mejora la imagen mediante el filtrado de ruido presente gracias a `cv2.blur()`, y la imagen se visualiza de la siguiente figura:



**Figura 67.** Imagen mejorada con eliminación de Ruido

**Elaborado por:** El Investigador

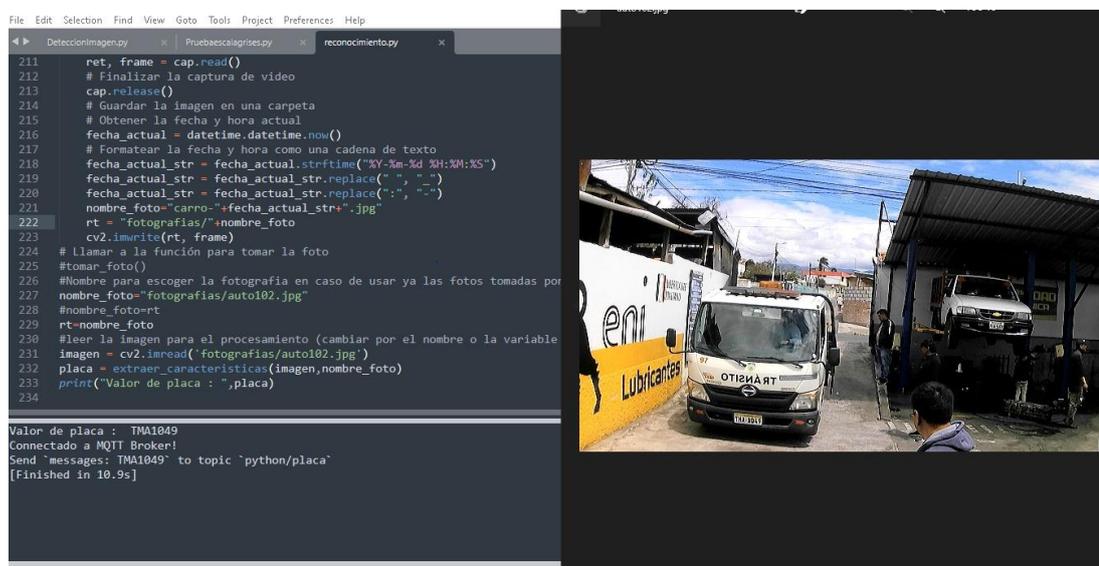
Ahora el programa procede a realizar el procesamiento del área del contorno de la placa.



**Figura 68.** Área para procesar con algoritmos de Lógica Difusa

**Elaborado por:** El Investigador

Los datos de la placa son sacados gracias al procesamiento realizado con el algoritmo construido basado en la Lógica Difusa. A continuación, se procede a observar el dato procesado por el programa.



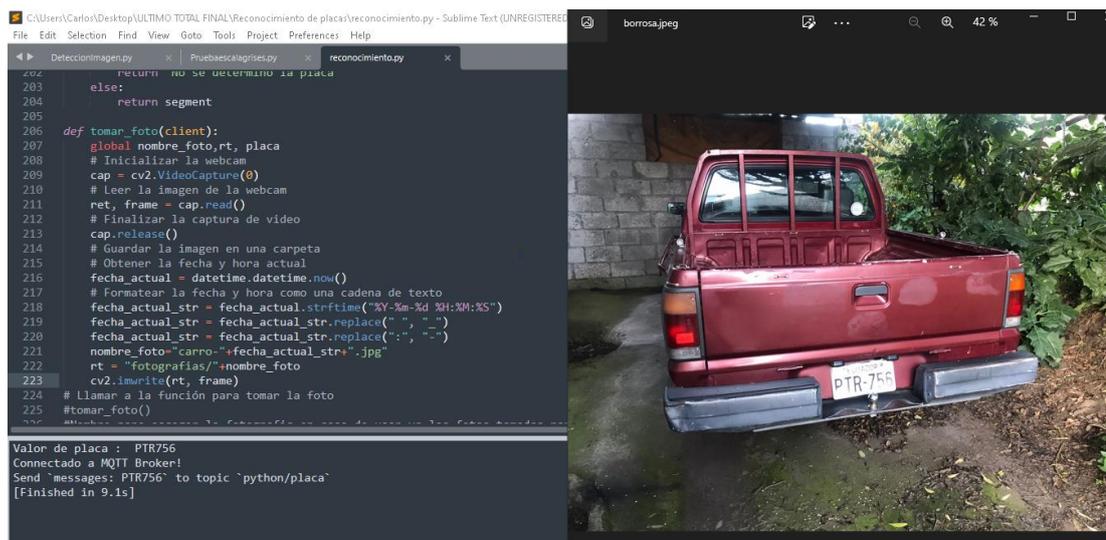
**Figura 69.** obtención de datos de placa de la imagen

**Elaborado por:** El Investigador

### 3.7 Pruebas de Funcionamiento

En esta sección se muestran los resultados de la detección de la placa del vehículo y la cantidad de aciertos logrados. Estos resultados se basan en el análisis de las imágenes capturadas en el taller de reparación de automóviles "Tecnicars David", utilizando una cámara 4 en 1 Bullet HD 1080p de 2 megapíxeles de color negro, colocada estratégicamente en la entrada principal del taller automotriz.

Durante las pruebas de funcionamiento, se obtuvieron ciertos resultados. Se planteó la pregunta de si era factible editar las placas vehiculares debido a que muchas de ellas estaban borrosas o en mal estado. Sin embargo, gracias a la programación del sistema basado en Lógica Difusa, éste es capaz de actuar sobre la placa identificando los caracteres. En caso de que los números y letras de la placa sean borrosos, el programa reconstruye y completa las letras y números de manera automática.



**Figura 70.** Prueba de funcionamiento con placa borrosa

**Elaborado por:** El Investigador

Como se observa obtenemos los datos de la placa de una imagen de un auto con una placa con letras y números incompletos.

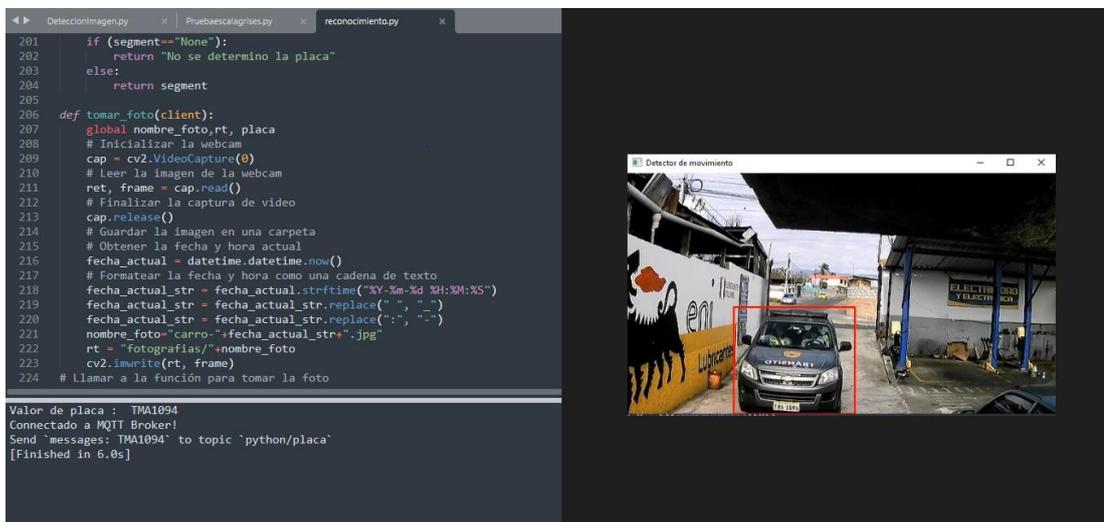
### 3.7.1 Verificación del funcionamiento del algoritmo programado

Para obtener los indicadores de aciertos, se puso en funcionamiento el sistema que combina la programación de la cámara para la detección de movimiento y captura de imágenes en conjunto con el procesamiento para extracción de caracteres con Lógica Difusa. Durante esta prueba se procesaron 20 vehículos. De estos, 18 vehículos pudieron ser identificados, mientras que los que no fueron identificados pasaron rápidamente al interior del taller. Los 18 vehículos identificados fueron sometidos al procesamiento de imagen, lo cual permitió obtener los caracteres de la placa vehicular de todos ellos. De los 18 vehículos con placa identificada, los 18 devolvieron correctamente los datos de su propietario.



**Figura 71.** Captura de imagen procesada para los indicadores

**Elaborado por:** El Investigador



**Figura 72.** Pruebas de Funcionamiento de la detección de placas

**Elaborado por:** El Investigador

Se observa que el programa extrae los datos de la placa del vehículo

### 3.7.2 Análisis de resultados

Para el análisis de los resultados, se detallará el procedimiento seguido con los 20 vehículos que ingresaron al taller de servicio automotriz. Es importante destacar que las imágenes de los vehículos fueron capturadas en un rango horario de 8:00 am a 6:00 pm, correspondiente al horario de trabajo del taller.

A continuación, se presenta la tabla de resultados de las placas vehiculares obtenidas durante el análisis:

**Tabla 14.** Tabla de Resultados de placas

Número de Placa	Vehículo foto	Resultado
<b>TMA1049</b>		<b>TMA1049</b>
<b>PTW675</b>		<b>PTW675</b>
<b>TBF2259</b>		<b>TBF2259</b>

TEC014		TEC014
FDT146		FDT146
PEO437		PEO437
PIM955		PIM945
PLO0015		PLO0015

<p><b>TMA273</b></p>		<p><b>TMA273</b></p>
<p><b>TEI1242</b></p>		<p><b>TEI1242</b></p>
<p><b>TEA735</b></p>		<p><b>TEA735</b></p>
<p><b><u>PEQ528</u></b></p>		<p><b>PEQ528</b></p>

<p><b>TBB1559</b></p>		<p><b>TBB1559</b></p>
<p><b>PPA6093</b></p>		<p><b>PPA6093</b></p>
<p><b>TMA1132</b></p>		<p><b>TMA1132</b></p>
<p><b>TMA1094</b></p>		<p><b>TMA1094</b></p>

<b>PKC0055</b>		<b>PKC0055</b>
<b>TBI6535</b>		<b>TBI6535</b>
<b>PIA593</b>		<b>PIA593</b>
<b>XBV0138</b>		<b>XBV0138</b>

Elaborado por: El Investigador

## Tabla de los vehículos que el sistema no detectó al 100%

Tabla 15. Tabla de Resultados erróneos

Número de Placa	Vehículo Foto	Resultado
PIM955		PIM945
PKC0055		PKO0055

**Elaborado por:** El Investigador

De acuerdo con los resultados presentados en la tabla 14, se observa que de los 20 vehículos sometidos a pruebas, hubo 2 casos en los que el sistema no logró detectar correctamente la placa vehicular en su totalidad. Esta situación se debe principalmente a la velocidad a la que dichos vehículos pasaron frente a la cámara de captura.

### Funcionamiento del sistema de base de datos con Machine Learning

El sistema de registro automático funciona en conjunto con la base de datos que utiliza un algoritmo de clasificación para reconocer a los vehículos que ingresaron al taller y que poseen un historial de trabajos realizados.

El funcionamiento del base de datos con Machine Learning es que identifica si el auto que ingresa ya posee trabajos previos realizados desplegando en la pantalla de la interfaz dichos datos, es decir reconoce que el vehículo ya se encuentra registrado y muestra los trabajos que ya se realizaron.

Por otra parte, si el vehículo ingresa por primera vez al taller, el algoritmo que enlaza la información del vehículo con la base de datos reconoce que es un vehículo nuevo para el taller y se despliega la pantalla con la interfaz para el registro.

Primero se procedió a entrenar al algoritmo clasificador de información dentro de la base de datos desarrollado en Python, en donde se completó información de vehículos que ya tienen trabajos previos realizados, así como también de vehículos nuevos para registrar, como se muestra en la figura 73.

Propietario	Cédula	Teléfono	Placa	Fecha Registro	Dashboard	Editar	Eliminar
Agencia de Regulación y Control de Energía	1768193080001	990344768	PEQ0371	12/07/2023, 22:58:13	HISTORIAL	EDITAR	ELIMINAR
Agencia de Regulación y Control de Energía	1768193080001	990344768	PEI5141	12/07/2023, 23:13:08	HISTORIAL	EDITAR	ELIMINAR
DIRECCIÓN DISTRITAL 18D04 PATATE SAN PEDRO DE PELILEO - SALUD.	1865000590001	32765324	TEA0735	12/07/2023, 23:25:24	HISTORIAL	EDITAR	ELIMINAR
MINISTERIO DEL TRABAJO REGIONAL 3, CENTRO DE LA SIERRA Y AMAZONIA.	1865023170001	984654922	PEQ0528	12/07/2023, 23:34:49	HISTORIAL	EDITAR	ELIMINAR
AGENCIA DE REGULACIÓN Y CONTROL DE ENERGÍA Y RECURSOS NATURALES NO RENOVABLES	1768193080001	990344768	PEI7522	12/07/2023, 23:46:25	HISTORIAL	EDITAR	ELIMINAR

**Figura 73.** Base de datos para entrenar el algoritmo de clasificación

**Elaborado por:** El Investigador

A continuación, en la siguiente figura se muestra el proceso de funcionamiento del sistema gestor de la base de datos.

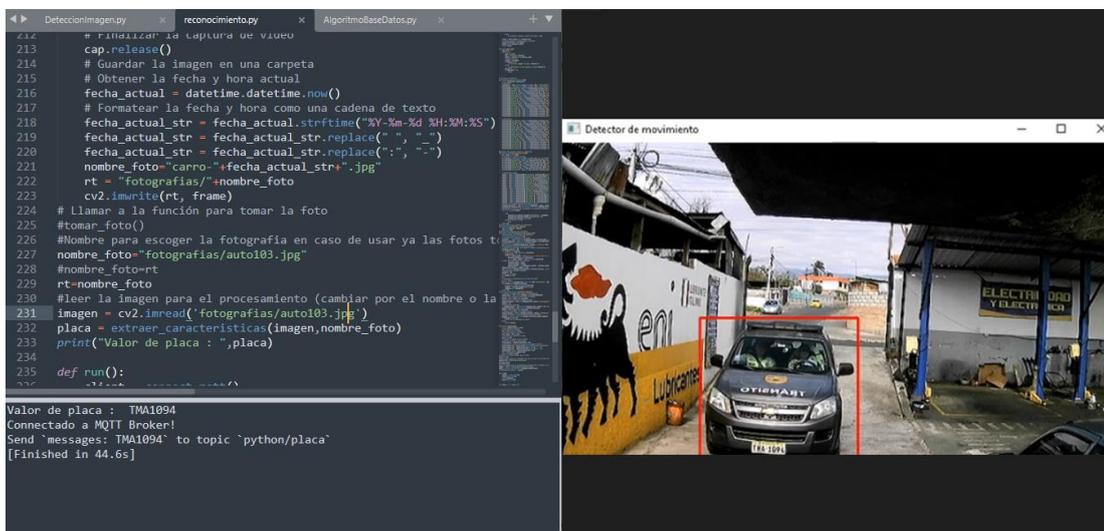
Inicialmente ingresa un vehículo el cual es captado por el sistema de registro automático a través de la cámara, la cual mediante el procesamiento de la tarjeta Raspberry envía el dato a la interfaz del programa como se observa en la figura 74.



**Figura 74.** Ingreso de Vehículo detectado por movimiento

**Elaborado por:** El Investigador

El Sistema identifica la placa y lo envía al Servidor por medio de MQTT como se muestra a continuación:



**Figura 75.** Reconocimiento de placa por el sistema

**Elaborado por:** El Investigador

Se observa que el sistema reconoce de manera correcta la placa. De igual forma envía el dato de la placa a la interfaz gráfica del sistema como se muestra en la figura 76.

```
Valor de placa : TMA1094
Connectado a MQTT Broker!
Send `messages: TMA1094` to topic `python/placa`
[Finished in 44.6s]
```

**Figura 76.** Envío de dato del sistema al Servidor

Elaborado por: El Investigador

Se puede observar que el dato fue enviado correctamente al servidor como se observa en la figura 77.

**Figura 77.** Dato de la placa recibido desde el sistema al servidor

Elaborado por: El Investigador

El dato de la placa de este vehículo ya cuenta con trabajos previos realizados por lo que el sistema de la base de datos reconoce automáticamente a este vehículo y despliega el historial de trabajos previos realizados como se muestra en las figuras 78 y 79.

Administración

Ver Datos

Tecnicars  Clientes

### Información de: GAD MUNICIPALIDAD DE AMBATO



**GAD MUNICIPALIDAD DE AMBATO**  
158236025000

**Información General**

Propietario	: GAD MUNICIPALIDAD DE AMBATO
Cedula	: 158236025000
Conductor	: Carlos Deleg
Placa	: messages:TMA1094
Vehículo y Kilometraje	: CHEVROLET / D-MAX CRDI AC 3.0 CD 4X2 TM DIESEL - 341266

**Figura 78.** Historial de trabajos previos realizados

**Elaborado por:** El Investigador

Administración

Ver Datos

Tecnicars  Clientes

**Información Adicional**

Dirección: Ambato

Fecha de registro: 13/07/2023, 09:11:16

### Historial del Cliente: GAD MUNICIPALIDAD DE AMBATO messages: TMA1094

ID Placa	Motivo de Visita	Fecha del trabajo
messages: TMA1094	REPUESTOS BOQUILLAS PARA FOCOS DE 1 FILAMENTO, FOCOS DOBLE FILAMENTO, FOCO 1 FILAMENTO, FOCO AMBAR, FOCO UÑA PEQUEÑO, FUSIBLES MASTER = 21 MANO DE OBRA REVISION DE LUCES = 5.60 TOTAL = 26.50	13/07/2023, 09:11:20

**Figura 79.** Historial de trabajos previos realizados

**Elaborado por:** El Investigador

En la figura 79 se observa que el sistema reconoce y muestra los datos de los trabajos realizados para ese vehículo.

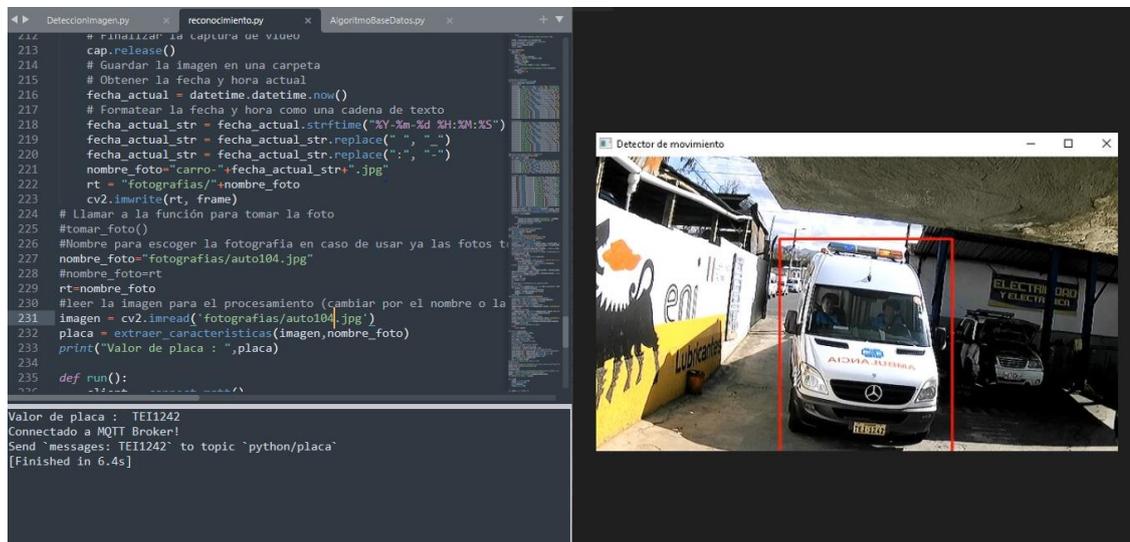
A continuación, se muestra otra prueba de funcionamiento con otro vehículo como se muestra en la figura 80.



**Figura 80.** Captura de imagen por movimiento

**Elaborado por:** El Investigador

Seguidamente, se muestran en la figura 81 los datos leídos de la placa por el sistema.



**Figura 81.** Datos de la placa detectados por el sistema

**Elaborado por:** El Investigador

Ahora, en la figura 82 se observa el envío del dato de la placa al servidor.

```
Valor de placa : TEI1242
Connectado a MQTT Broker!
Send `messages: TEI1242` to topic `python/placa`
[Finished in 6.4s]
```

**Figura 82.** Envío de dato de la placa al servidor

**Elaborado por:** El Investigador

Se observa a continuación en la figura 83 que el dato fue enviado correctamente a la interfaz del programa.

The screenshot shows a web application interface with a dark blue header and a light blue sidebar. The main content area is white and contains a registration form titled "Registro de Usuario". The form has two columns of input fields. The first column includes "Propietario: \*", "Direccion", "E-mail:", "Modelo/Año:", "Km:", and "VIN:". The second column includes "R.U.C./C.I: \*", "Telf: \*", "Conductor", "Placa: \*", and "Vehiculo:". The "Placa: \*" field contains the text "messages: TEI1242". Below the form are two buttons: "REGISTRAR" and "LIMPIAR". Below the form is a section titled "Motivo de Visita" with a text input field. At the bottom right, there is a confirmation message: "Placa: messages: TEI1242".

**Figura 83.** Datos correctos enviados al servidor

**Elaborado por:** El Investigador

Finalmente se observa que el sistema de la base de datos reconoce al auto y muestra automáticamente los trabajos realizados en el vehículo.

The screenshot shows a web interface with a navigation bar at the top containing 'Administración', 'Técnicos', and 'Clientes'. The main content area is divided into two sections. On the left, there is a logo for 'TECNICARS DAVID' featuring a cartoon mechanic, with the text 'CENTRO DE SERVICIO AUTOMOTRIZ MULTIMARCA' above it and 'DIRECCIÓN DISTRITAL 18D04 PATATE SAN PEDRO DE PELILEO - SALUD. 1865000590001' below. On the right, under the heading 'Información General', there is a table with the following data:

Propietario	:	DIRECCIÓN DISTRITAL 18D04 PATATE SAN PEDRO DE PELILEO - SALUD.
Cedula	:	1865000590001
Conductor	:	ING. MARCO ANDALUZ
Placa	:	TEI 1242
Vehículo y Kilometraje	:	MERCEDES BENZ / - 262738

Below this table, under the heading 'Información Adicional', the address is listed as 'Dirección: CALLE JUAN MONTALVO Y EUGENIO ESPEJO'.

**Figura 84.** Datos del vehículo mostrados por el algoritmo clasificador

**Elaborado por:** El Investigador

The screenshot shows the same web interface as Figure 84, but displaying the 'Historial del Cliente' for the vehicle 'DIRECCIÓN DISTRITAL 18D04 PATATE SAN PEDRO DE PELILEO - SALUD. TEI1242'. The history is presented in a table with the following data:

ID Placa	Motivo de Visita	Fecha del Trabajo
TEI1242	REPUESTOS FILTRO DE ACEITE MB SPRINTER OM 651 ,FILTRO DE AIRE DEPURADOR MB SPRINTER , FILTRO DE COMBUSTIBLE SPRINTER C/SENSOR, JUEGO DE PASTILLAS DE FRENO DELANTERAS Y POSTERIORES, PLUMA DERECHA LIMPIA PARABRISAS (JUEGO X 2), SENSOR MAF, TOMACORRIENTES HABITACULO = 925 LUBRICANTES ACEITE 15W40 LT = 108 MANO DE OBRA Alineación de ruedas , Arreglo Puerta Posterior , Balanceo de juego de ruedas, Calibración del motor, Cambio de Aceite y Filtro de motor, Cambio de MAF, Cambio de pastillas de freno delanteras y posteriores, Cambio de Pisos(Habitaculo), Cambio de Plumas, Cambio de zapatas de freno de mano, Cambio filtro de combustible , Desmontaje y montaje del tanque de combustible para limpieza, Mantenimiento elevador Puerta lateral derecha.	13/07/2023, 09:42:30
TEI1242	TOTAL = 2508.80 LUBRICANTES ACEITE 15W40 LT. = 108 TOTAL = 120.96	13/07/2023, 09:46:58

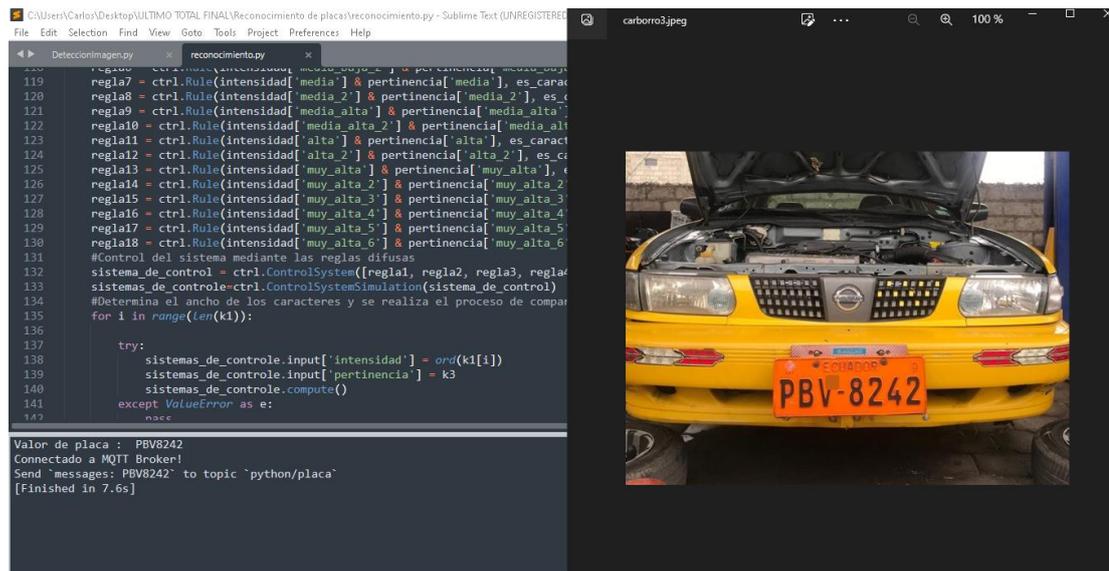
**Figura 85.** Historial del vehículo mostrado automáticamente por el sistema de base de datos

**Elaborado por:** El Investigador

Se observa que las pruebas realizadas dentro del sistema gestor de la base de datos entrenado con un algoritmo clasificador de Machine Learning funciona correctamente con una eficiencia del 100%.

### 3.7.3 Funcionamiento del sistema con letras y números borrosos e incompletos

A continuación, se procederá a mostrar los resultados de detección de placas borrosas para observar el funcionamiento del sistema cabe aclarar que se va a simular la esta parte editando las imágenes para obtener placas con números y letras incompletos, ya que en las pruebas de funcionamiento no se obtuvo este tipo de placas de los autos que ingresaron al taller.



**Figura 86.** Reconstrucción de letra V

**Elaborado por:** El Investigador

En la figura 86 se puede observar que el sistema reconstruye la letra V de la placa dando un resultado optimo.

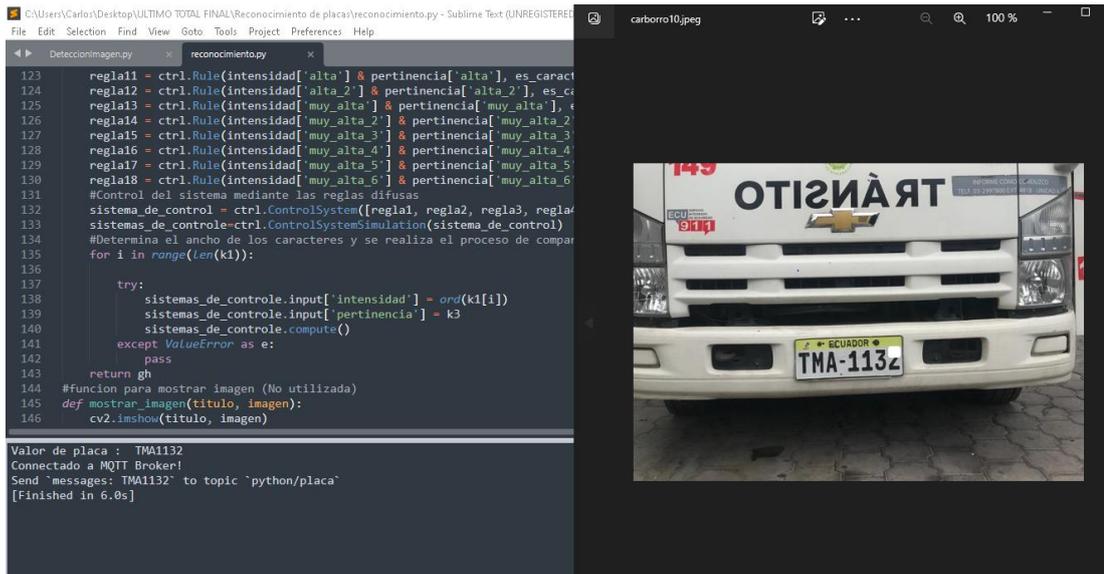


Figura 87. Reconstrucción de número 2

Elaborado por: El Investigador

En la figura 87 se puede observar que el sistema reconstruye el número 2 de la placa.

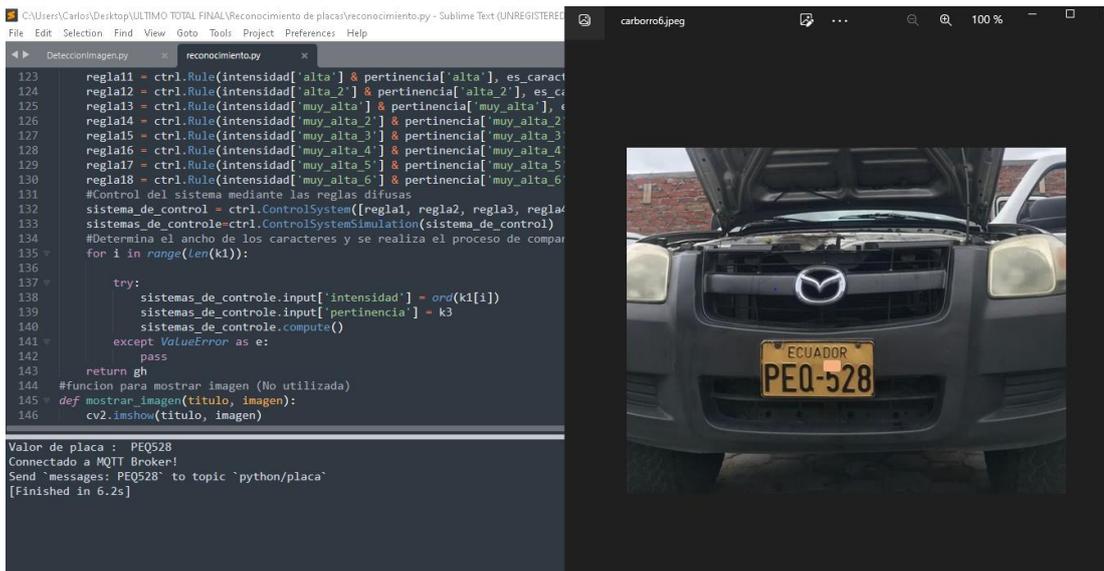
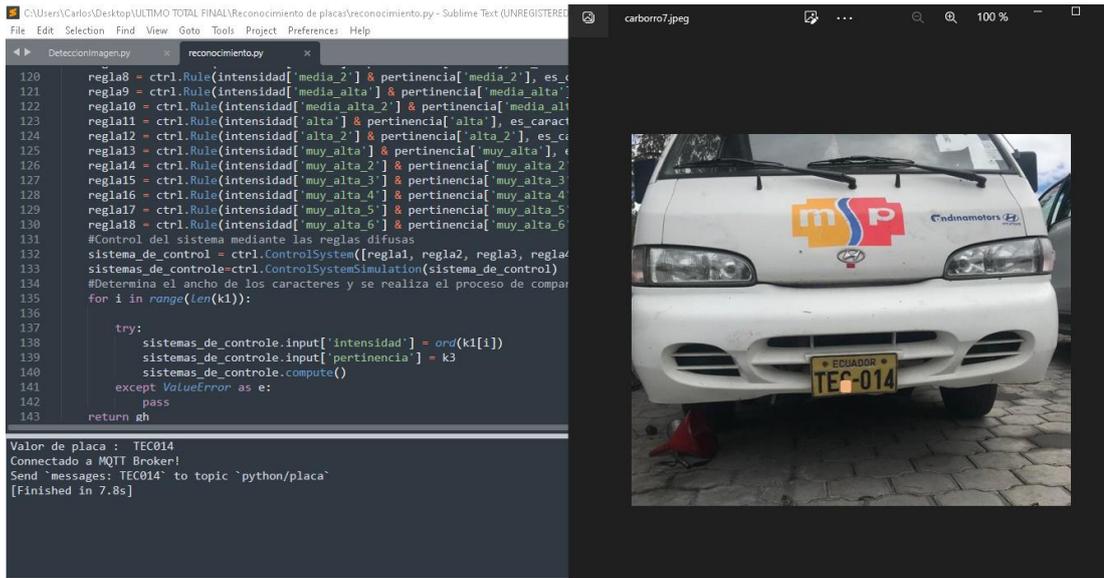


Figura 88. Reconstrucción de número 5

Elaborado por: El Investigador

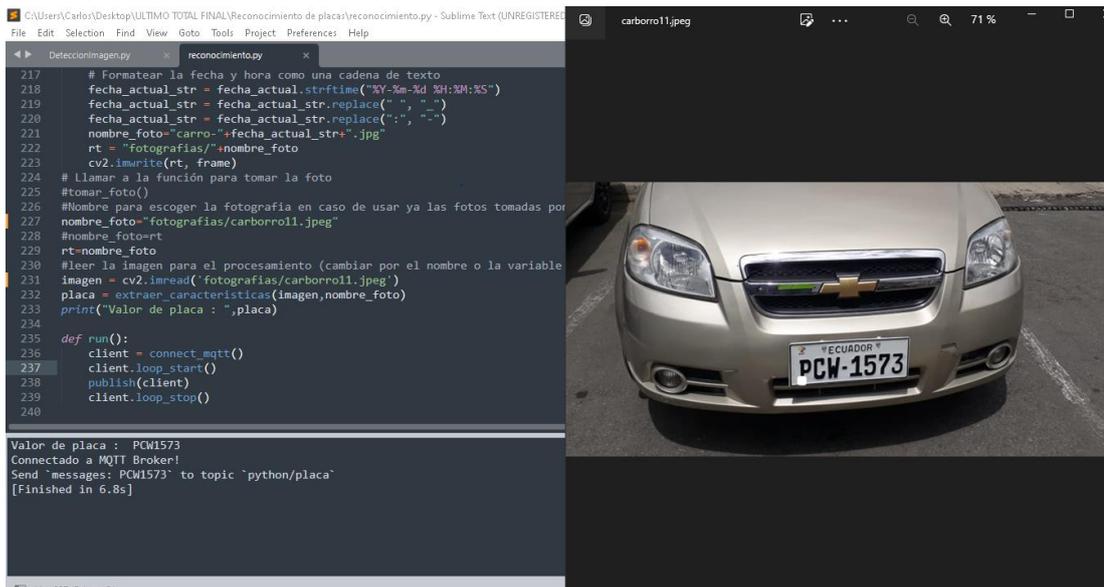
En la figura 88 se puede observar que el programa reconstruye el número 5 de la placa



**Figura 89.** Reconstrucción de letra C

**Elaborado por:** El Investigador

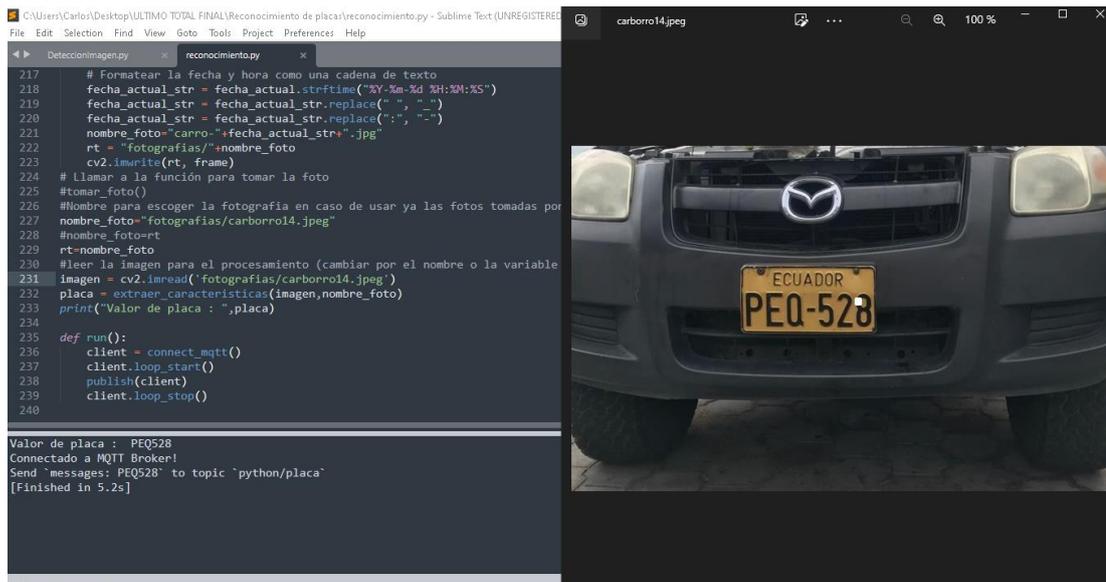
En la figura 89 se observa que el programa reconstruye la letra C de la placa



**Figura 90.** Reconstrucción de letra P

**Elaborado por:** El Investigador

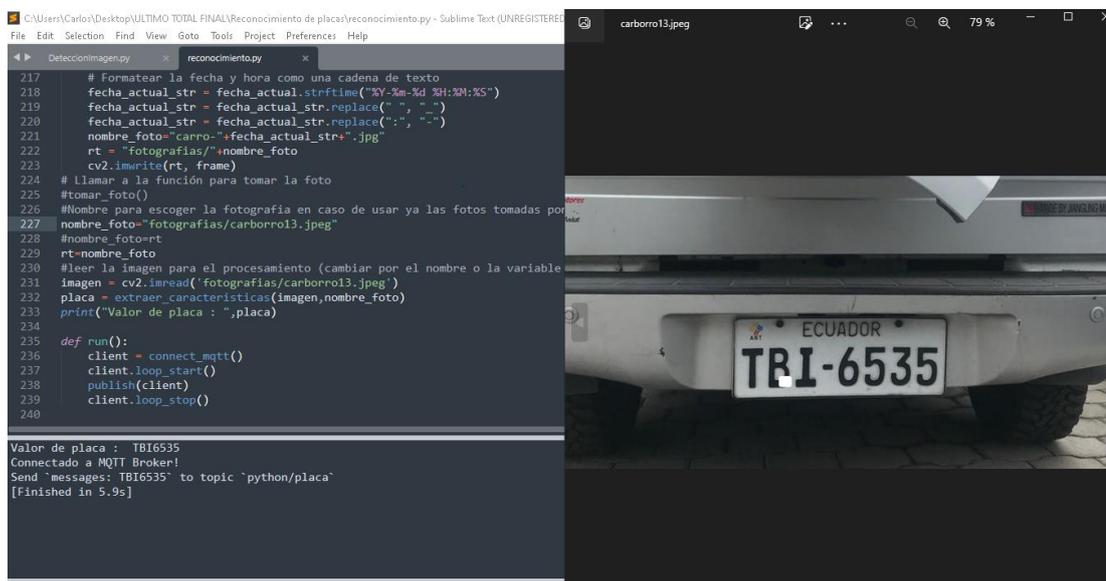
En la figura 90 se observa que el programa reconstruye la letra P de la placa



**Figura 91.** Reconstrucción de número 8

**Elaborado por:** El Investigador

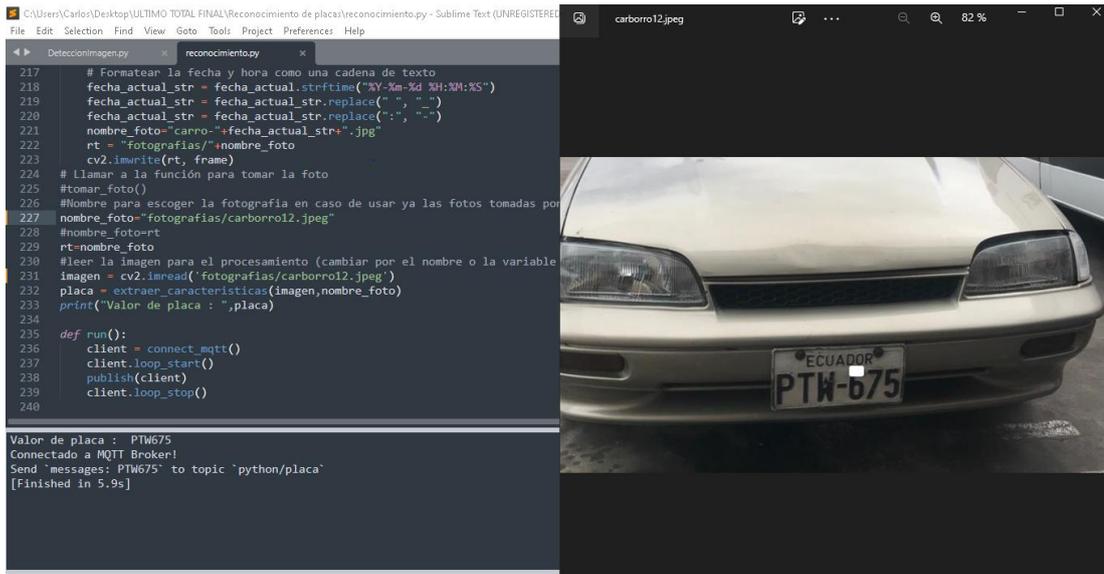
En la figura 91 se observa que el programa reconstruye el número 8 de la placa



**Figura 92.** Reconstrucción de letra B

**Elaborado por:** El Investigador

En la figura 92 se puede observar que el programa reconstruye la letra B de la placa



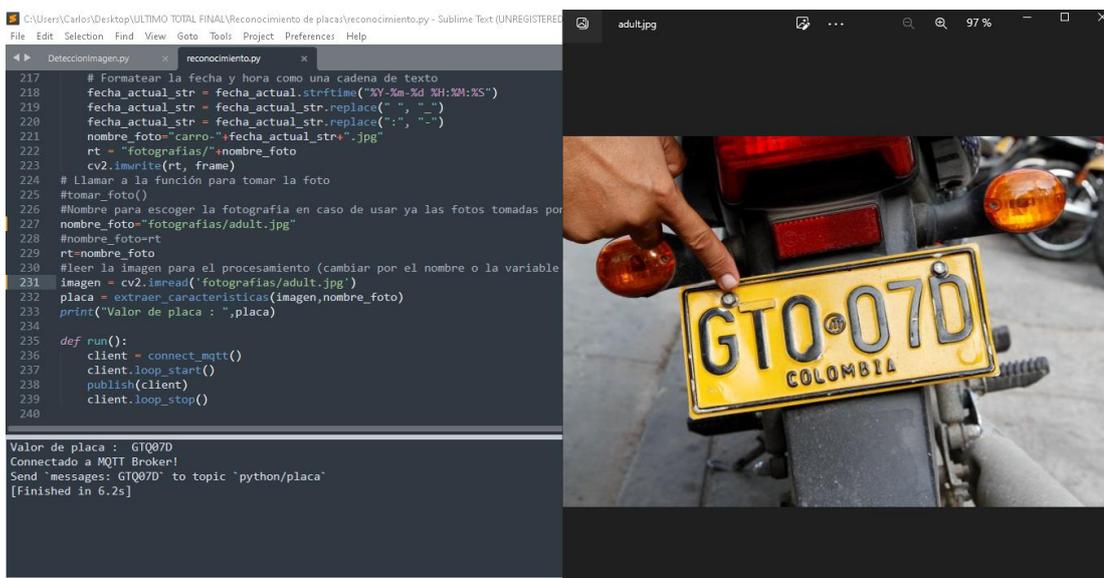
**Figura 93.** Reconstrucción de número 6

**Elaborado por:** El Investigador

En la figura 943se puede observar que el programa reconstruye el número 6 de la placa.

### 3.7.3 Realización de pruebas con placas reales borrosas

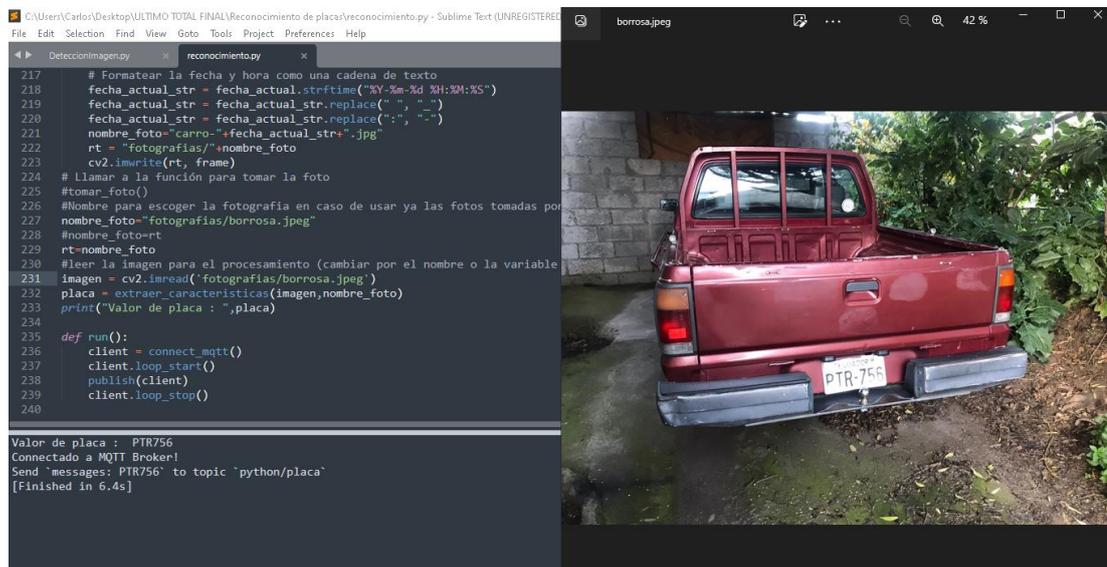
A continuación, se procedió a realizar pruebas de funcionamiento con imágenes reales.



**Figura 94.** Reconstrucción con imagen real letra T

**Elaborado por:** El Investigador

En la figura 94 se observa una placa del país Colombia que posee la letra T borrosa y se observa como el programa reconstruye la letra de la placa.



**Figura 95.** Reconstrucción imagen real letra P, números 5,6

**Elaborado por:** El Investigador

En la figura 95 se observa que el programa reconstruye la letra P y los números 5 y 6 de la placa.

### 3.7.4 Verificación de la precisión y eficiencia del sistema

Se llevaron a cabo pruebas exhaustivas con un total de 20 vehículos durante un período de dos semanas, en un horario comprendido entre las 8 de la mañana y las 6 de la tarde. Tras el análisis inicial, se determinó que las horas de mayor flujo vehicular hacia el interior del taller corresponden al intervalo de 9 a 11 de la mañana, así como de 3 a 5 de la tarde.

Con el objetivo de determinar la precisión y eficiencia del sistema, se ha elaborado una tabla con los resultados obtenidos durante las pruebas. De los 20 vehículos con placas distintas que ingresaron, se logró un reconocimiento correcto por parte del sistema en 18 de ellos. Sin embargo, en 2 vehículos se presentaron dificultades y no se obtuvo un reconocimiento preciso.



**Figura 96.** Representación de las pruebas correctas e incorrectas realizadas

**Elaborado por:** El Investigador

Al analizar el gráfico, se puede observar que el proceso de reconocimiento vehicular ha obtenido un resultado satisfactorio en un 90% de los casos, lo cual se refleja en la correcta identificación de 18 placas vehiculares. Sin embargo, se registra un 10% de placas mal identificadas, correspondiente a 2 vehículos.

Es importante destacar que el porcentaje del 10% de vehículos con resultados erróneos se atribuye a la velocidad del vehículo, ya que en estos casos los vehículos se desplazaron a gran velocidad frente a la cámara y así dificultó la captura de una imagen.

De igual forma durante las pruebas de funcionamiento relacionadas con la detección de información de vehículos previamente registrados, se obtuvo un resultado del 100% de efectividad. El sistema identificó de manera precisa y completa todos los datos almacenados previamente en la base de datos de vehículos registrados.

### 3.7.5 Probabilidad de Detección de placas

**Aciertos:** Detección correcta de placas.

**N:** Tamaño de muestra (20 autos).

$$\% \text{ De los aciertos de la detección de autos} = \frac{\text{Aciertos}}{N} * 100[\%]$$

$$\% \text{ De los aciertos de la detección de autos} = \frac{18}{20} * 100[\%]$$

$$\% \text{ De los aciertos de la detección de autos} = 90[\%]$$

### 3.7.6 Probabilidad de Aciertos de consulta de Datos con la base de datos

**Aciertos:** Detección correcta de consulta de datos de la placa con la base de datos.

**N:** Tamaño de la muestra (18 vehículos detectados).

$$\% \text{ Aciertos} = \frac{\text{Aciertos}}{N} * 100[\%]$$

$$\% \text{ Aciertos} = \frac{18}{18} * 100[\%]$$

$$\% \text{ Aciertos} = 100[\%]$$

### 3.7.8 Problemas encontrados

Durante la realización de este proyecto, se encontraron los siguientes problemas:

**Resolución de cámara:** inicialmente, se trabajó con una cámara de resolución 720p; sin embargo, la calidad de la imagen obtenida no fue satisfactoria. Por lo tanto, se decidió cambiar a una cámara para exteriores tipo bullet de color negro con una resolución de 1080p. Esta nueva cámara proporcionó una mejor calidad de imagen para el proyecto.

**Posicionamiento de la cámara:** se llevaron a cabo múltiples capturas de imagen utilizando diferentes ángulos de detección de placas. Se observó que hubo una mejora significativa al posicionar la cámara en una ubicación elevada, como en una parte superior. Esta posición permitió una mejor visualización y captura de las placas vehiculares, lo que contribuyó a una mayor precisión en la detección y reconocimiento de los caracteres de las placas.

### 3.8 Costos

**Tabla 16.** Tabla de costos de implementación del sistema

Ítem	Recursos Económicos	Cantidad	P. Unit. (USD)	P. Total (USD)
1	Material de oficina e impresiones	1	\$30,00	\$30,00
2	Dispositivos electrónicos (Cámara)	1	\$350,00	\$350,00
2	Dispositivos electrónicos de transmisión de datos	2	\$10	\$20
2	Dispositivos electrónicos de alimentación	2	\$14	\$28
4	Interfaz de visualización	1	\$100,00	\$100,00
5	Transporte	--	\$30,00	\$30,00
6	Tarjetas de Control	2	\$60,00	\$120,00
	Subtotal	-	-	678,00
	Imprevistos (5%)	-	-	33,90
	<b>TOTAL</b>	-	-	<b>711,90</b>

**Elaborado por:** El Investigador

### Presupuesto

Se consideraron dos aspectos fundamentales para calcular el presupuesto en la implementación del sistema de registro automático para la microempresa “Técnicars David”, el presupuesto de construcción y el de diseño. Para obtener el presupuesto de diseño, se tuvo en cuenta la cantidad de horas dedicadas a su diseño y construcción, y el salario de un Ingeniero en Telecomunicaciones, que según el Ministerio de Trabajo asciende a 858 dólares. Con base en un promedio de 21 días laborables, se utilizó una ecuación para calcular el salario diario del ingeniero [51].

$$Sueldo_{diario} = \frac{Sueldo_{mensual}}{Días_{laborables}}$$

$$Sueldo_{diario} = \frac{858}{21}$$

$$Sueldo_{diario} = \$40,8 [51]$$

De acuerdo con las regulaciones del Ministerio de Trabajo, la jornada laboral diaria se compone de 8 horas de trabajo. Utilizando la siguiente ecuación, es posible calcular el valor en dólares correspondiente a una hora de trabajo.

$$Sueldo_{hora} = \frac{Sueldo_{diario}}{Horas_{laborables}}$$

$$Sueldo_{hora} = \frac{40,8}{8}$$

$$Sueldo_{hora} = \$5.1 [51]$$

Se estima que se emplearon alrededor de 480 horas en actividades de investigación, diseño, programación, construcción y validación del funcionamiento. Para calcular el presupuesto de diseño, se utilizó la siguiente ecuación.

$$Presupuesto_{diseño} = Horas_{Investigación} * Sueldo_{hora}$$

$$Presupuesto_{diseño} = 480 * \$5,1$$

$$Presupuesto_{diseño} = \$2448$$

Utilizando la siguiente fórmula se calculó el monto total del presupuesto.

$$Presupuesto_{total} = Presupuesto_{diseño} + Presupuesto_{construcción}$$

$$Presupuesto_{total} = \$2448 + \$711,90$$

$$Presupuesto_{total} = \$3159,90$$

El presupuesto total del proyecto ascendió a \$ 3159,90 y fue completamente financiado por el tesista.

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1. Conclusiones

- Para este proyecto se utilizó herramientas con Lógica Difusa para manejar situaciones de incertidumbre o borrosidad en las imágenes de las placas, además de Machine Learning aplicado a la base de datos. La Lógica Difusa actúa sobre los caracteres de la placa, reconstruyendo y completando los números y letras en caso de que estén borrosos o ilegibles como se demostró en el proyecto. Esto, en combinación con el sistema de base de datos que funciona con un algoritmo de clasificación basado en árboles de decisión para el aprendizaje automático, permitió el registro de la información, con un manejo eficiente y seguro de dicha información de las placas de automóviles del Taller Automotriz “Técnicars David”.
- La arquitectura lógica del sistema se llevó a cabo en el lenguaje de programación Python. También, se implementó herramientas con Lógica Difusa para la extracción de datos de la imagen de la placa, permitiendo reconstruir letras y números incluso en casos de borrosidad o baja calidad de imagen. En cuanto al hardware, se empleó una cámara IP específicamente diseñada para su uso en exteriores, para ejecutar el programa del sistema desarrollado se utilizó la tarjeta Raspberry Pi4, conocida por su capacidad de procesamiento y flexibilidad en este tipo de aplicaciones. Este enfoque permite un funcionamiento eficiente y confiable del sistema, asegurando una detección precisa de placas de automóviles y un registro de información completo y seguro.
- La interfaz gráfica ha sido diseñada teniendo en cuenta los requerimientos específicos del taller, proporcionando una experiencia amigable al usuario y garantizando un registro completo de la información necesaria. Posee varias secciones sobre el registro de los vehículos, la plataforma permite ingresar los datos de la placa de forma manual si es el vehículo es nuevo para el taller y despliega de

forma automática los trabajos previos realizados a vehículos que poseen historial de trabajo.

- Después de llevar a cabo varias pruebas de funcionamiento, se concluye que el sistema de registro automático funciona de manera altamente eficiente. Los resultados revelaron que el sistema alcanzó un impresionante porcentaje de fiabilidad del 90% en el reconocimiento de las placas vehiculares. La precisión en el reconocimiento de las placas vehiculares mediante el uso de Lógica Difusa y el bajo porcentaje de errores obtenidos respaldan el desempeño exitoso del sistema en situaciones reales. Estos resultados validan la efectividad y utilidad del sistema de registro automático para agilizar y mejorar el proceso de gestión de automóviles en el taller automotriz.

## 4.2. Recomendaciones

- Se sugiere considerar la integración de esta técnica en otros aspectos del taller automotriz "Técnicars David". Dado el éxito y eficacia demostrados al utilizar herramientas de Lógica Difusa en este proyecto para abordar situaciones de incertidumbre en las imágenes de las placas. Explorar cómo la Lógica Difusa puede aplicarse en otros procesos, como la gestión de inventario, la programación de citas y la optimización de recursos, podría mejorar aún más la eficiencia y efectividad de las operaciones del taller.
- Se recomienda realizar una revisión a las fichas técnicas de los equipos que componen el sistema antes de proceder con su montaje. Es crucial conocer los requisitos de alimentación y la ubicación adecuada de cada equipo para garantizar su correcto funcionamiento. Al observar detalladamente estas especificaciones, se evitará la necesidad de repetir el trabajo y se ahorrarán tiempo y recursos. Al revisar las fichas técnicas, es importante verificar la fuente de alimentación necesaria para cada equipo, asegurándose de que se cumplan los requisitos de voltaje, corriente y tipo de conexión. Además, es esencial identificar la ubicación óptima para cada dispositivo, considerando factores como el acceso, la ventilación y la distancia con otros componentes.
- Se recomienda tomar medidas adicionales para mejorar la seguridad del sistema y proteger la información confidencial de los vehículos que utilizan el servicio. A pesar de contar con acceso rápido y sencillo a la información relevante, es fundamental salvaguardar los datos y prevenir posibles ataques o filtraciones. Para fortalecer la seguridad del sistema, se sugiere implementar medidas como el uso de encriptación de datos, contraseñas seguras y políticas de acceso restringido. Es importante asegurarse de que solo el personal autorizado tenga acceso a la información confidencial y que se establezcan niveles adecuados de permisos y privilegios.

- Es importante destacar que la velocidad a la que se desplaza un vehículo puede tener un impacto significativo en la calidad y nitidez de la imagen capturada. Para mejorar el proceso de detección y reconocimiento de placas vehiculares, se recomienda considerar la incorporación de un rompe velocidades en la entrada del taller. Esto permitirá reducir la velocidad de los vehículos al ingresar, lo que a su vez facilitará la captura de imágenes más claras y precisas. Evitará situaciones en las que los vehículos pasan rápidamente frente a la cámara, lo que podría resultar en imágenes borrosas o poco definidas, afectando la exactitud del sistema en la identificación de las placas. Con esta medida, se optimizará la eficiencia y confiabilidad del sistema de registro automático de vehículos en el taller.
- La aplicación de la lógica difusa en la reconstrucción de caracteres alfanuméricos de placas de automóviles se muestra viable, ya que esta técnica permite abordar imágenes menos nítidas con eficacia. Esto significa que es posible emplear una cámara de baja resolución para capturar las imágenes, lo que puede resultar en un considerable ahorro de costos en comparación con las cámaras especializadas en la detección de placas, que pueden tener un valor cercano a los 700 dólares.

## BIBLIOGRAFÍA

- [1] D. Automation, «Deloitte Insights,» 30 Junio 2022. [En línea]. Available: <https://www2.deloitte.com/us/en/insights/focus/technology-and-the-future-of-work/intelligent-automation-2022-survey-results.html>.
- [2] Mordor, «MordorIntelligence,» 16 Mayo 2022. [En línea]. Available: <https://www.mordorintelligence.com/es/industry-reports/latin-america-process-automation-market>.
- [3] J. Lucero, «Gestión de calidad en micro y pequeñas empresas de servicio automotriz ecuatoriano,» Revista Internacional, Quito, 2020.
- [4] A. S. y. C. Stalin, «" Prototipo de un sistema recomendador basado en Lógica Difusa para evaluar Obras de Relevancia de la Universidad de las Fuerzas Armadas ESPE",» Tesis de Ingenieria ESPE, Sangolquí, ECU,2022.
- [5] N. L. y. O. Sebastian, «"Diseño e implementación de un prototipo de sistema de seguridad integral con el fin de monitorear el acceso de automóviles, utilizando visión artificial y chatbot para el ingreso al conjunto residencial",» Tesis de grado de Ingenieria ESPE, Sangolquí, ECU, 2021.
- [6] L. Marquez, «"Diseño e implementación de un software de reconocimiento de placas vehiculares en tiempo real",» Tesis de grado, Univ. UNAD, Bogotá, COLOMBIA,2019.
- [7] S. A. Espinoza Darwin, «"Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares ",» Tesis de grado, Univ. del Azuay, Cuenca, ECU, 2019.
- [8] T. Milton, «"Reconocimiento Automático de la placa de un vehículo de Ecuador",» Tesis de grado, Univ. de UPS, Quito, Ecuador,2020.
- [9] J. P. Suquilanda Jaramillo, «Diseño y creación de una empresa de servicios automotrices aplicando la gestión por procesos,» Universidad De Las Américas , Quito, 2018.
- [10] O. Barcia, Reconocimiento automático a través de visión artificial, correlación estadística y Matlab., Guayaquil: Ingeniería Electrónica UPS, 2019.
- [11] J. J. Padiál, Técnicas de programación "Deep Learning", Málaga: Universidad de Malaga, 2019.

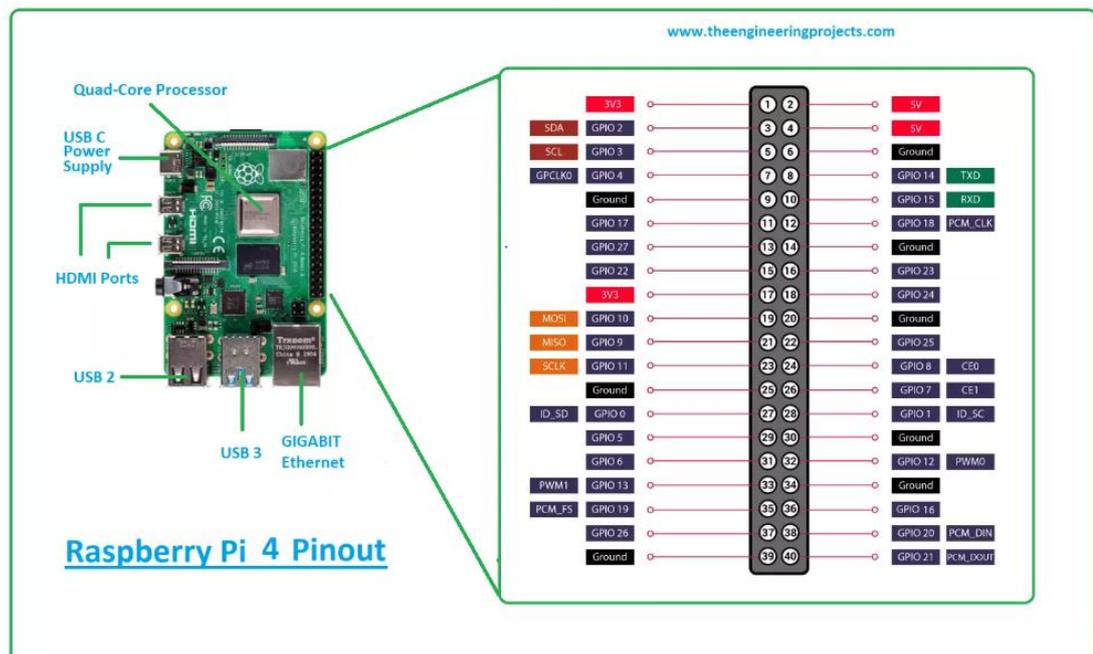
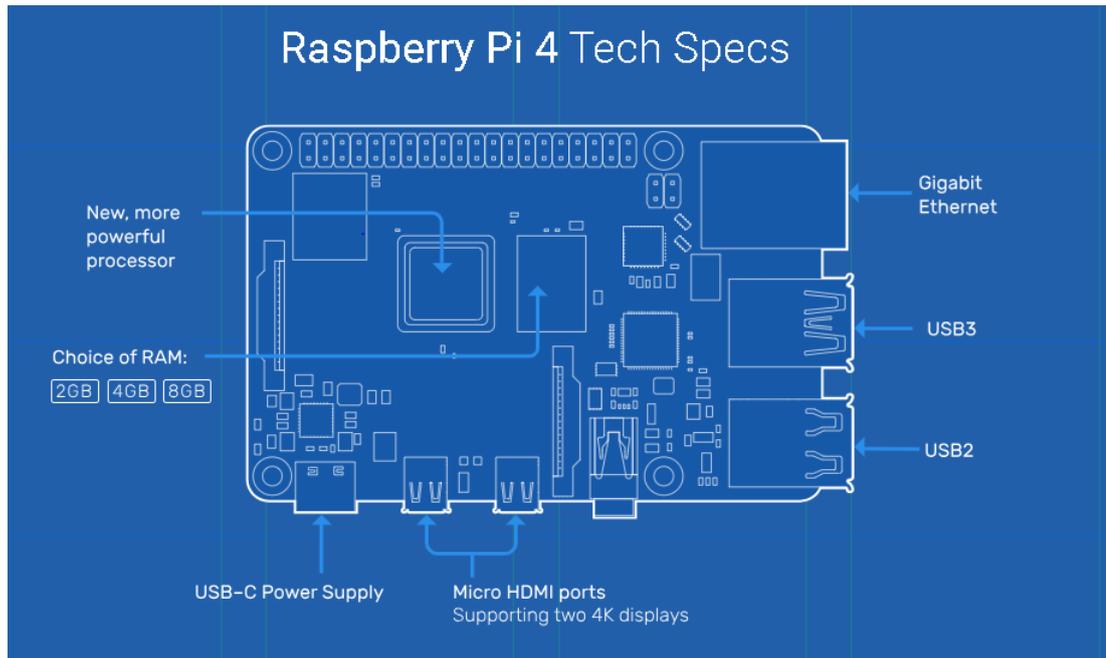
- [12] G. Albornoz, «Aplicacion del aprendizaje automático supervisado en el mantenimiento predictivo de los motores electricos de inducción en las empresas mineras del Perú,» Escuela Profesional de Ingeniería Eléctrica, Lima Peru, 2021.
- [13] C. Gonzalez, «"Lógica difusa una introducción Práctica",» Madrid, 2021.
- [14] T. J. Ross, «Fuzzy Logic with Engineering Applications,» Third Edition, Jhon Wiley & Sons, 2010.
- [15] F. R. y. Vermonden, «Lógica Difusa para la toma de decisiones y la selección de personal,» Rosario, Vol 5 , no. 9, pp2-7, Colombia, 2019.
- [16] J. H. Reyna Castillo, «Procesamiento Digital de Imágenes empleando filtros Espaciales,» Universidad Autónoma de Baja California, California, 2018.
- [17] R. G. d. Blas, «Reconocimiento de caracteres mediante imágenes en contadores de gas en entornos reales,» Departamento de Ingeniería Software e Inteligencia Artificial, Madrid, 2015.
- [18] A. Bermudez, «Red de telecomunicaciones con la tecnología de reconocimiento óptico de caracteres para la detección de infractores de gasolina en el municipio Maracaibo,» Trabajo de Grado, Maracaibo, 2018.
- [19] J. Porras, «Classification System based on computer vision,» Escuela Profesional de Ingeniería Electrónica Ricardo Palma, Lima, 2020.
- [20] J. Lira, La percepción remota: nuestros ojos desde el espacio, la Ciencia para todos, México: Secretaría de Educación Pública/Fondo de Cultura Económica, 1987.
- [21] G. Bradski, Learning OpenCV, Computer Vision with the OpenCV Library, O'Really Media Released, 2008.
- [22] Threshold, «Threshold pagina oficial Información Oficial,» [En línea]. Available: <https://imagej.net/plugins/auto-threshold>.
- [23] Torch3., «Torch3 pagina oficial,» [En línea]. Available: <http://torch3vision.idiap.ch>.
- [24] Vxl, «Vxl sitio oficial,» [En línea]. Available: <http://vxl.sourceforge.net>.
- [25] I. Ltilib, «LTilib,» [En línea]. Available: <http://Ltilib.sourceforge.net/doc/homepage/index.shtml>.

- [26] Opencv, «Opencv Información,» [En línea]. Available: <http://opencv.org/>.
- [27] A. I. LOTT, «DECIMO PRIMERO DEL REGLAMENTO GENERAL PARA LA APLICACIÓN DE LA LEY ORGÁNICA DE TRANSPORTE TERRESTRE, TRANSITO DE SEGURIDAD VIAL EN EL ECUADOR».
- [28] P. O. d. ANT, «Agencia Nacional de Transito y sus resoluciones,» [En línea]. Available: <http://www.ant.gob.ec/index.php/transito-7/resoluciones-2014>.
- [29] J. A. N. I. OCR, «OCR de las Implementaciones Tecnológicas e Informática,» [En línea]. Available: <http://www.iti.es/media/about/docs/tic/13/articulo2.pdf>.
- [30] P. e. Python, «Tokio school,» [En línea]. Available: <https://www.tokioschool.com/noticias/librerias-python-machine-learning/#:~:text=Pues%20porque%20se%20trata%20de,de%20Python%20para%20machine%20learning>.
- [31] A. Rosado, «Redes de comunicación Topología y enlaces,» Sistemas Industriales distribuidos, Valencia, 2017.
- [32] Azure, «"Azure",» [En línea].
- [33] A. Robledano, «OpenWebinars,» 24 Septiembre 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-mysql/>.
- [34] D. G. IONOS, «Cuestiones Técnicas,» 16 Febrero 2022. [En línea]. Available: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/oracle-database/>.
- [35] S. Borges, «Infranetworking,» 19 Noviembre 2021. [En línea]. Available: <https://blog.infranetworking.com/servidor-postgresql/>.
- [36] G. F., «Videovigilancia: CCTV,» Vértice, Málaga, 2010.
- [37] P. E. H., «Tecnologías y redes de transmisión de datos,» Limusa Noriega, México, 2003.
- [38] Jandei, «Jandei cámaras y seguridad,» 2021. [En línea]. Available: <https://jandei.com/camaras-4-1/3548-camara-bullet-4-en-1-1080p-exterior-28-12mm-lente-motorizada-ir-osd-blanca-8435523828527.html>.
- [39] Hikvision, «Hikvision Camaras,» [En línea]. Available: <https://www.hikvision.com/es-la/products/Turbo-HD-Products/Turbo-HD-Cameras/Value-Series/DS-2CE16C0T-IRPF/>.

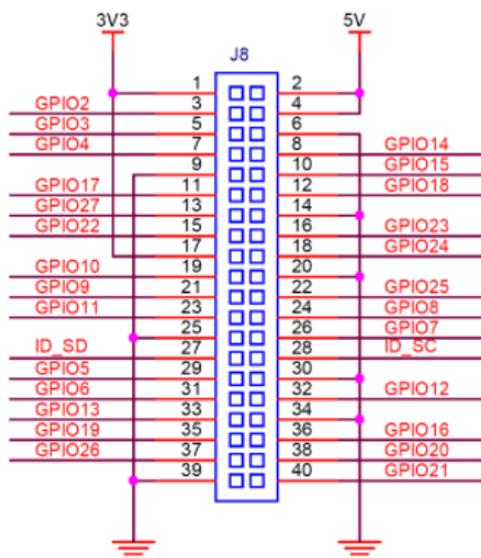
- [40] ADI, «ADI Soluciones de Seguridad,» [En línea]. Available: <https://www.adiglobal.es/Catalog/Brands/hanwha-vision/HCO-6070RP>.
- [41] xataka, «Raspberry Pi 4 es oficial,» 07 Mayo 2021. [En línea]. Available: <https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica>.
- [42] nvidia, «developer jetson nano,» 11 Junio 2022. [En línea]. Available: <https://developer.nvidia.com/embedded/jetson-nano>.
- [43] G. Hardware, «Guia Hardware, Comunicaciones Inalámbricas,» 11 Noviembre 2022. [En línea]. Available: <https://www.guiahardware.es/tecnologias-inalambricas/>.
- [44] T. M. a. R. C. P. Wankhede, «COMPARATIVE STUDY OF CLOUD PLATFORMS MICROSOFT AZURE, AMAZON EC2-AWS,» UG student, Departament of Computer Engineering, India 2020, 2020.
- [45] J. Narvaez, «Comparación de Protocolos de comunicación para internet de las cosas (IOT),» Universidad Politécnica Salesiana sede Quito, Quito, 2022.
- [46] Desarrolloweb.com, «Desarrolloweb Base de Datos,» 2012. [En línea]. Available: <https://desarrolloweb.com/articulos/840.php> .
- [47] S. Romerson, «Slideshare,» 04 07 2022. [En línea]. Available: <https://es.slideshare.net/RomersonSantana/oracle-database-100548612> .
- [48] D. G. Lowe., «Distinctive Image Features from Scale-Invariant Keypoints,» Columbia, University of British Vancouver, 2004.
- [49] Ocrad, «Ocrad Pagina Oficial,» [En línea]. Available: <http://www.gnu.org/software/ocrad/>.
- [50] T. OCR, «Tesseract OCR Pagina Oficial,» [En línea]. Available: <https://code.google.com/p/tesseract-ocr/>.

# **ANEXOS**

## Anexo 1. Datasheed Tarjeta Raspberry Pi4



## GPIO Pin Assignments



### ID\_SD and ID\_SC PINS:

These pins are reserved for HAT ID EEPROM.

At boot time this I2C interface will be interrogated to look for an EEPROM that identifies the attached board and allows automatic setup of the GPIOs (and optionally, Linux drivers).

**DO NOT USE** these pins for anything other than attaching an I2C ID EEPROM. Leave unconnected if ID EEPROM not required.

## GPIO Alternate Functions

GPIO	Default						
	Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
0	High	SDA0	SA5	PCLK	SPI3_CE0_N	TXD2	SDA6
1	High	SCL0	SA4	DE	SPI3_MISO	RXD2	SCL6
2	High	SDA1	SA3	LCD_VSYNC	SPI3_MOSI	CTS2	SDA3
3	High	SCL1	SA2	LCD_HSYNC	SPI3_SCLK	RTS2	SCL3
4	High	GPCLK0	SA1	DPL_D0	SPI4_CE0_N	TXD3	SDA3
5	High	GPCLK1	SA0	DPL_D1	SPI4_MISO	RXD3	SCL3
6	High	GPCLK2	SOE_N	DPL_D2	SPI4_MOSI	CTS3	SDA4
7	High	SPI0_CE1_N	SWE_N	DPL_D3	SPI4_SCLK	RTS3	SCL4
8	High	SPI0_CE0_N	SD0	DPL_D4	-	TXD4	SDA4
9	Low	SPI0_MISO	SD1	DPL_D5	-	RXD4	SCL4
10	Low	SPI0_MOSI	SD2	DPL_D6	-	CTS4	SDA5
11	Low	SPI0_SCLK	SD3	DPL_D7	-	RTS4	SCL5
12	Low	PWM0	SD4	DPL_D8	SPI5_CE0_N	TXD5	SDA5
13	Low	PWM1	SD5	DPL_D9	SPI5_MISO	RXD5	SCL5
14	Low	TXD0	SD6	DPL_D10	SPI5_MOSI	CTS5	TXD1
15	Low	RXD0	SD7	DPL_D11	SPI5_SCLK	RTS5	RXD1
16	Low	FL0	SD8	DPL_D12	CTS0	SPI1_CE2_N	CTS1
17	Low	FL1	SD9	DPL_D13	RTS0	SPI1_CE1_N	RTS1
18	Low	PCM_CLK	SD10	DPL_D14	SPI6_CE0_N	SPI1_CE0_N	PWM0
19	Low	PCM_FS	SD11	DPL_D15	SPI6_MISO	SPI1_MISO	PWM1
20	Low	PCM_DIN	SD12	DPL_D16	SPI6_MOSI	SPI1_MOSI	GPCLK0
21	Low	PCM_DOUT	SD13	DPL_D17	SPI6_SCLK	SPI1_SCLK	GPCLK1
22	Low	SD0_CLK	SD14	DPL_D18	SD1_CLK	ARM_TRST	SDA6
23	Low	SD0_CMD	SD15	DPL_D19	SD1_CMD	ARM_RTCK	SCL6
24	Low	SD0_DAT0	SD16	DPL_D20	SD1_DAT0	ARM_TDO	SPI3_CE1_N
25	Low	SD0_DAT1	SD17	DPL_D21	SD1_DAT1	ARM_TCK	SPI4_CE1_N
26	Low	SD0_DAT2	TE0	DPL_D22	SD1_DAT2	ARM_TDI	SPI5_CE1_N
27	Low	SD0_DAT3	TE1	DPL_D23	SD1_DAT3	ARM_TMS	SPI6_CE1_N



## Anexo 2. Datasheed Cámara



### CARACTERÍSTICAS

---

Estas cámaras cuentan con numerosas prestaciones y son perfectas para instalaciones de exterior que requieran una excelente calidad de imagen con un mínimo de ancho de banda. Además son estancas evitando la entrada de polvo, suciedad o agua y cuentan con un diseño moderno y con dos opciones distintas de lentes, estas cámaras son idóneas para una amplia gama de aplicaciones.

- CMOS de 1/2.7" o 1/2.8" de escaneado progresivo de 2 MP o 1/3" de 4 MP.
- Excelente rendimiento en escenas de muy baja iluminación.
- Iluminación IR de hasta 60 m, dependiendo de la reflectancia de la escena.
- WDR: hasta 140 dB.
- Excelente rendimiento con luz baja, hasta 0 lux BN (con los LED de IR encendidos).
- Hasta 3 flujos de vídeo.
- Resolución de 1080p a 25/30 fps o 4 MP a 20 fps.
- Lente con enfoque y zoom motorizados de 2,7 -12 mm o 5 - 50 mm, F1.4.
- Carcasa antivandálica de exterior con grado de protección IP67 y calificación de resistencia a impactos IK10.
- Visualización en modo pasillo.
- Almacenamiento de vídeo en la cámara: tarjeta micro SD hasta 128 GB (no incluida).
- Cumple el estándar ONVIF Perfil S y G.

ESTÁNDAR		ESPECIFICACIONES DE IP (CONTINUACIÓN)		
ESTÁNDAR DE VÍDEO	RAL	VELOCIDAD DE FOTOGRAMAS	FLUJO DE VÍDEO SECUNDARIO	<b>HBW2GR1/HBW2GR3/HBL2GR1:</b> 1-25 fps (D1-704 x 576, CIF-352 x 288) <b>HBW4GR1:</b> 1-20 fps (D1-704 x 576, CIF-352 x 288)
SISTEMA DE ESCANEADO	Escaneado progresivo		TERCER FLUJO DE VÍDEO	<b>HBW2GR1/HBW2GR3/HBL2GR1:</b> 720p 1-14 fps, D1 (704 x 576) 1-25 fps, CIF (352 x 288) 1-25 fps <b>HBW4GR1:</b> 1-20 fps (720p, D1-704 x 576, CIF-352 x 288) <b>HBL2GR1:</b> 1080p 1-20 fps, 1-25 fps (720p, D1-704 x 576)
SENSOR DE IMAGEN	<b>HBW2GR1/HBW2GR3:</b> 1/2.7" CMOS progresivo 2 MP <b>HBW4GR1:</b> 1/3" CMOS progresivo 4 MP	PROTODLOS*	IPv4/IPv6, HTTP, HTTPS, TCP/IP, UDP, UPnP, ICMP, IGMP, RTP, RTSP, SMTP, NTP, DHCP, DNS, PPPoE, DDNS, FTP, QoS, SNMP v2c/v3, LDAP (Cliente), NFS, RTCP, Bonjour	
NÚMERO DE PÍXELES (HxV)	<b>HBL2GR1/HBW2GR1/HBW2GR3:</b> 1920 x 1080 <b>HBW4GR1:</b> 2560 x 1440	COMPATIBILIDAD	ONVIF Profile S/G	
ILUMINACIÓN MÍNIMA	<b>HBL2GR1:</b> 0.005 lux color/0 lux con LED de IR encendidos a F1.4 <b>HBW2GR1/HBW2GR3/HBW4GR1:</b> 0.01 lux color/0 lux con LED de IR encendidos a F1.4	ACCESO MÁXIMO USUARIOS	20 usuarios	
RELACIÓN SEÑAL/RUIDO	50 dB o más	IDIOMAS	Español, inglés, portugués, alemán, árabe, checo, chino, coreano, francés, italiano, japonés, neerlandés, polaco, ruso, turco	
VELOCIDAD DEL OBTURADOR ELECTRÓNICO	Auto/Manual, 1/3(4)-1/10.000 s	ELECTRICAS		
DISTANCIA DE IR DÍA/NOCHE	Hasta 60 m dependiendo de la reflectancia de la escena Auto (OCR/Color/BN)	ALIMENTACIÓN	12 VCC +/-10 %, 24 VCA +/-10 %, PoE IEEE 802.3af	
COMPENSACIÓN DE CONTRALUCES	BLC/HLC	FRECUENCIA DE CA	50/60 Hz	
RANGO DINÁMICO AMPLIADO	<b>HBL2GR1:</b> 140 dB <b>HBW2GR1/HBW2GR3/HBW4GR1:</b> 120 dB	CONSUMO DE ENERGÍA	24 VCA 1.2 W máx (zoom/enfoque/IR/calentador activo), típico de 10.5 W PoE (802.3af) Clase 0 <b>HBL2GR1 PoE (802.3af)</b> 24 V CA 14.5 W Clase 4	
BALANCE DE BLANCOS	Auto/Manual	MECANICAS		
CONTROL DE GANANCIA	Auto/Manual	DIMENSIONES	281.0 x 110.0 mm	
REDUCCIÓN DE RUIDO	3DNR	PESO DEL PRODUCTO	2.0 kg	
ZONAS PRIVADAS	Hasta 4 zonas	PESO DEL EMBALAJE	2.3 kg	
MODO PASILLO	Si	MATERIAL DE CONSTRUCCIÓN	Carcasa de aluminio fundido	
LENTE	<b>HBL2GR1/HBW2GR1/HBW4GR1:</b> 2.7-12 mm, MFZ, F1.4 <b>HBW2GR3:</b> 5-50 mm, MFZ, F1.4	COLOR	RAL9003 (blanco) y RAL7022 (gris)	
ÁNGULO DE VISIÓN HORIZONTAL	<b>HBL2GR1:</b> 97°-34° <b>HBW2GR1/HBW4GR1:</b> 99°-34° <b>HBW2GR3:</b> 66°-6.2°	GRABACIÓN Y ALMACENAMIENTO	Tarjeta microSDHC de hasta 128 GB de clase 10 (no incluida)	
AUDIO/BIDIRECCIONAL	Línea de entrada/salida	AMBIENTALES		
ESPECIFICACIONES DE IP		TEMPERATURA	De -40 °C a 60 °C	
COMPRESIÓN DE VÍDEO	H.264/H.264H/H.264B/MJPEG	HUMEDAD RELATIVA	10 % a 100 %, con condensación	
RESOLUCIÓN	<b>HBL2GR1/HBW2GR1/HBW2GR3:</b> 1080p (1920 x 1080)/SxGA (1280 x 1024)/1.3M (1280 x 960)/720p (1280 x 720)/D1 (704 x 576)/704 + 480/CF (352 x 288/352 x 240) <b>HBW4GR1:</b> 4M (2688 x 1520 con salida de vídeo analógico apagada, o 2560 x 1440 con salida de vídeo analógico encendido)/3M (2304 x 1296)/1080p (1920 x 1080)/SxGA (1280 x 1024)/1.3M (1280 x 960)/720p (1280 x 720)/D1 (704 x 576)/704 + 480/CF (352 x 288/352 x 240)	PROTECCIÓN CONTRA LA ENTRADA DE AGUA	IP67	
VELOCIDAD DE FOTOGRAMAS	FLUJO DE VÍDEO PRINCIPAL	RESISTENCIA A ACTOS VANDÁLICOS	IK10	
<b>HBW2GR1/HBW2GR3/HBL2GR1:</b> 1-25 fps (1080p, SxGA-1280 x 1024, 1280 x 960, 720p, D1-704 x 576) <b>HBW4GR1:</b> 1-20 fps (2688 x 1520 con salida de vídeo analógico apagada, 2560 x 1440) 1-25 fps (2304 x 1296, 1080p, SxGA-1280 x 1024, 1280 x 960, 720p)		NORMATIVA		
		EMISIONES	FCC Parte 15, CE (EN 55022); cumple con RCM (AS/NZS)	
		INMUNIDAD	CE (EN 50130-4)	
		SEGURIDAD	UL: Norteamérica recogida en UL/CSA 60950-1, CE (EN 60950-1) UL: Norteamérica recogida en UL/CSA 60950-22, CE (EN 60950-22)	

**Anexo 3. Taller de servicio Automotriz "Técnicars David"**



**Anexo 4. Instalación de la cámara**



**Anexo 5.** Instalación Raspberry



## Anexo 6. Código en Python Sistema Automático de Registro

```
import cv2
#import imutils
import winsound
import datetime
print("Librerías Leídas...")

cap = cv2.VideoCapture(1)

while True:
    _, cam = cap.read()
    _, cam2 = cap.read()
    diff = cv2.absdiff(cam, cam2)
    gray = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    _, thresh = cv2.threshold(blur, 20, 255, cv2.THRESH_BINARY)
    dilated = cv2.dilate(thresh, None, iterations = 3)
    contours, _ = cv2.findContours(dilated, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    #cv2.drawContours(cam, contours, -1, (0, 0, 255), 2)
    for cnt in contours:
        if cv2.contourArea(cnt) < 10000:
            continue
        x, y, w, h = cv2.boundingRect(cnt)
        cv2.rectangle(cam, (x, y), (x+w, y+h), (0, 0, 255), 2)
        ret, frame = cap.read()
        fecha_actual = datetime.datetime.now()
        # Formatear la fecha y hora como una cadena de texto
        fecha_actual_str = fecha_actual.strftime("%Y-%m-%d %H:%M:%S")
        fecha_actual_str = fecha_actual_str.replace(" ", "_")
        fecha_actual_str = fecha_actual_str.replace(":", "-")
        nombre_foto="carro-"+fecha_actual_str+".jpg"
```

```

        rt = "fotografias/"+nombre_foto
        cv2.imwrite(rt,frame)
        #cv2.imwrite("imagen.jpg", frame)

    if cv2.waitKey(10) == ord("q"):
        break

    cv2.imshow("Detector de movimiento", cam)

#Importar librerias
import cv2
import numpy as np
import datetime
import matplotlib.pyplot as plt
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import copy
import subprocess
import random
import time
from paho.mqtt import client as mqtt_client

broker = 'broker.hivemq.com'
port = 1883
client_id = f'publish-{random.randint(0, 1000)}'
topic = "python/placa"

# Generate a Client ID with the publish prefix.
# username = 'emqx'
# password = 'public'

def connect_mqtt():
    def on_connect(client, userdata, flags, rc):

```

```

if rc == 0:
    print("Connectado a MQTT Broker!")
else:
    print("Fallo conexion, return code %d\n", rc)

```

```

client = mqtt_client.Client(client_id)
# client.username_pw_set(username, password)
client.on_connect = on_connect
client.connect(broker, port)
return client

```

```

def publish(client):
    msg_count = 1
    while True:
        time.sleep(1)
        msg = f"messages: {placa}"
        result = client.publish(topic, msg)
        # result: [0, 1]
        status = result[0]
        if status == 0:
            print(f"Send `{msg}` to topic `{topic}`")
        else:
            print(f"Failed to send message to topic {topic}")
        msg_count += 1
        if msg_count > 1:
            break

```

```

# Funciones de membresia
def funciones_membresia_iniciales():
    global intensidad, pertinencia
    # Intensidad

```

```

intensidad = ctrl.Antecedent(np.arange(0, 256), 'intensidad')
pertinencia = ctrl.Antecedent(np.arange(0, 101), 'pertinencia')
intensidad['muy_baja'] = fuzz.trimf(intensidad.universe, [0, 0, 12])
intensidad['muy_baja_2'] = fuzz.trimf(intensidad.universe, [4, 12, 20])
intensidad['baja'] = fuzz.trimf(intensidad.universe, [16, 32, 48])
intensidad['baja_2'] = fuzz.trimf(intensidad.universe, [40, 48, 56])
intensidad['media_baja'] = fuzz.trimf(intensidad.universe, [52, 64, 76])
intensidad['media_baja_2'] = fuzz.trimf(intensidad.universe, [72, 76, 80])
intensidad['media'] = fuzz.trimf(intensidad.universe, [76, 96, 116])
intensidad['media_2'] = fuzz.trimf(intensidad.universe, [108, 116, 124])
intensidad['media_alta'] = fuzz.trimf(intensidad.universe, [120, 132, 144])
intensidad['media_alta_2'] = fuzz.trimf(intensidad.universe, [140, 144, 148])
intensidad['alta'] = fuzz.trimf(intensidad.universe, [144, 160, 176])
intensidad['alta_2'] = fuzz.trimf(intensidad.universe, [168, 176, 184])
intensidad['muy_alta'] = fuzz.trimf(intensidad.universe, [180, 192, 204])
intensidad['muy_alta_2'] = fuzz.trimf(intensidad.universe, [200, 204, 208])
intensidad['muy_alta_3'] = fuzz.trimf(intensidad.universe, [204, 220, 236])
intensidad['muy_alta_4'] = fuzz.trimf(intensidad.universe, [232, 236, 240])
intensidad['muy_alta_5'] = fuzz.trimf(intensidad.universe, [236, 244, 252])
intensidad['muy_alta_6'] = fuzz.trimf(intensidad.universe, [248, 252, 255])

```

# Pertinencia

```

pertinencia['muy_baja'] = fuzz.trimf(pertinencia.universe, [0, 0, 12])
pertinencia['muy_baja_2'] = fuzz.trimf(pertinencia.universe, [4, 12, 20])
pertinencia['baja'] = fuzz.trimf(pertinencia.universe, [16, 32, 48])
pertinencia['baja_2'] = fuzz.trimf(pertinencia.universe, [40, 48, 56])
pertinencia['media_baja'] = fuzz.trimf(pertinencia.universe, [52, 64, 76])
pertinencia['media_baja_2'] = fuzz.trimf(pertinencia.universe, [72, 76, 80])
pertinencia['media'] = fuzz.trimf(pertinencia.universe, [76, 96, 116])
pertinencia['media_2'] = fuzz.trimf(pertinencia.universe, [108, 116, 124])
pertinencia['media_alta'] = fuzz.trimf(pertinencia.universe, [120, 132, 144])
pertinencia['media_alta_2'] = fuzz.trimf(pertinencia.universe, [140, 144, 148])
pertinencia['alta'] = fuzz.trimf(pertinencia.universe, [144, 160, 176])

```

```

pertinencia['alta_2'] = fuzz.trimf(pertinencia.universe, [168, 176, 184])
pertinencia['muy_alta'] = fuzz.trimf(pertinencia.universe, [180, 192, 204])
pertinencia['muy_alta_2'] = fuzz.trimf(pertinencia.universe, [200, 204, 208])
pertinencia['muy_alta_3'] = fuzz.trimf(pertinencia.universe, [204, 220, 236])
pertinencia['muy_alta_4'] = fuzz.trimf(pertinencia.universe, [232, 236, 240])
pertinencia['muy_alta_5'] = fuzz.trimf(pertinencia.universe, [236, 244, 252])
pertinencia['muy_alta_6'] = fuzz.trimf(pertinencia.universe, [248, 252, 255])

```

#funciones para generar reglas y comparacion

```
def generar_reglas(k1,k2,k3,k4,k5,k6):
```

```
    gh = copy.deepcopy(k1)
```

```
    # Definir las variables difusas
```

```
    es_caracter = ctrl.Consequent(np.arange(0, 101, 1), 'es_caracter')
```

```
    es_caracter['muy_bajo'] = fuzz.trimf(es_caracter.universe, [0, 0, 51])
```

```
    es_caracter['bajo'] = fuzz.trimf(es_caracter.universe, [25, 76, 127])
```

```
    es_caracter['medio_bajo'] = fuzz.trimf(es_caracter.universe, [102, 153, 204])
```

```
    es_caracter['medio'] = fuzz.trimf(es_caracter.universe, [178, 229, 255])
```

```
    es_caracter['medio_alto'] = fuzz.trimf(es_caracter.universe, [204, 255, 255])
```

```
    es_caracter['alto'] = fuzz.trimf(es_caracter.universe, [229, 255, 255])
```

```
    es_caracter['muy_alto'] = fuzz.trimf(es_caracter.universe, [253, 255, 255])
```

```
    # Definir las reglas
```

```
    regla1 = ctrl.Rule(intensidad['muy_baja'] & pertinencia['muy_baja'],
```

```
es_caracter['muy_bajo'])
```

```
    regla2 = ctrl.Rule(intensidad['muy_baja_2'] & pertinencia['muy_baja_2'],
```

```
es_caracter['muy_bajo'])
```

```
    regla3 = ctrl.Rule(intensidad['baja'] & pertinencia['baja'], es_caracter['bajo'])
```

```
    regla4 = ctrl.Rule(intensidad['baja_2'] & pertinencia['baja_2'], es_caracter['bajo'])
```

```
    regla5 = ctrl.Rule(intensidad['media_baja'] & pertinencia['media_baja'],
```

```
es_caracter['medio_bajo'])
```

```
    regla6 = ctrl.Rule(intensidad['media_baja_2'] & pertinencia['media_baja_2'],
```

```
es_caracter['medio_bajo'])
```

```
    regla7 = ctrl.Rule(intensidad['media'] & pertinencia['media'], es_caracter['medio'])
```

```

    regla8 = ctrl.Rule(intensidad['media_2'] & pertinencia['media_2'],
es_caracter['medio'])
    regla9 = ctrl.Rule(intensidad['media_alta'] & pertinencia['media_alta'],
es_caracter['medio_alto'])
    regla10 = ctrl.Rule(intensidad['media_alta_2'] & pertinencia['media_alta_2'],
es_caracter['medio_alto'])
    regla11 = ctrl.Rule(intensidad['alta'] & pertinencia['alta'], es_caracter['alto'])
    regla12 = ctrl.Rule(intensidad['alta_2'] & pertinencia['alta_2'], es_caracter['alto'])
    regla13 = ctrl.Rule(intensidad['muy_alta'] & pertinencia['muy_alta'],
es_caracter['muy_alto'])
    regla14 = ctrl.Rule(intensidad['muy_alta_2'] & pertinencia['muy_alta_2'],
es_caracter['muy_alto'])
    regla15 = ctrl.Rule(intensidad['muy_alta_3'] & pertinencia['muy_alta_3'],
es_caracter['muy_alto'])
    regla16 = ctrl.Rule(intensidad['muy_alta_4'] & pertinencia['muy_alta_4'],
es_caracter['muy_alto'])
    regla17 = ctrl.Rule(intensidad['muy_alta_5'] & pertinencia['muy_alta_5'],
es_caracter['muy_alto'])
    regla18 = ctrl.Rule(intensidad['muy_alta_6'] & pertinencia['muy_alta_6'],
es_caracter['muy_alto'])
    #Control del sistema mediante las reglas difusas
    sistema_de_control = ctrl.ControlSystem([regla1, regla2, regla3, regla4, regla5,
regla6, regla7, regla8, regla9, regla10, regla11, regla12, regla13, regla14, regla15,
regla16, regla17, regla18])
    sistemas_de_controle=ctrl.ControlSystemSimulation(sistema_de_control)
    #Determina el ancho de los caracteres y se realiza el proceso de comparacion con
los patrones
    for i in range(len(k1)):

        try:
            sistemas_de_controle.input['intensidad'] = ord(k1[i])
            sistemas_de_controle.input['pertinencia'] = k3
            sistemas_de_controle.compute()

```

```

    except ValueError as e:
        pass
    return gh
#funcion para mostrar imagen (No utilizada)
def mostrar_imagen(titulo, imagen):
    cv2.imshow(titulo, imagen)
    cv2.waitKey(0)
#Funcion para segmentar los caracteres
def segment_characters(erosion_image,imagen,rt):
    #Inicializamos las funciones de membresia
    funciones_membresia_iniciales()
    # Encontrar los contornos en la imagen
    contours, _ = cv2.findContours(erosion_image, cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)
    erosion_image_bgr = cv2.cvtColor(erosion_image, cv2.COLOR_GRAY2BGR)
    # Establecer los contornos en la imagen BGR
    cv2.drawContours(erosion_image_bgr, contours, -1, (0, 0, 255), 1)
    #cv2.imshow("Contornos",erosion_image_bgr)
    # Filtrar los contornos que tienen un tamaño y proporciones adecuadas para ser
caracteres
    filtered_contours = []
    for contour in contours:
        x, y, w, h = cv2.boundingRect(contour)
        aspect_ratio = w / h
        if 0.3 < aspect_ratio < 0.55 and 13 < w < 50 and 20 < h < 50: # segmentacion
            filtered_contours.append(contour)
            x, y, width, height = cv2.boundingRect(contour)
    # Medir las dimensiones del contorno
    contorno_width = width
    contorno_height = height
    aspect_ratio2 = float(contorno_width) / contorno_height
    image_with_contours = imagen.copy()
    cv2.drawContours(image_with_contours, filtered_contours, -1, (255, 255, 255), 2)

```

```

#inicializamos variable segments
segments = []
#Copia de la imagen
k = imagen.mean()
l=0
# Normalizar la intensidad en el rango [0, 255]
k = k / 255.0
pp=generar_reglas(subprocess.check_output(['./parametros',
rt]).decode().strip(),k,l,aspect_ratio,w,h)
for contour in filtered_contours:
    x, y, w, h = cv2.boundingRect(contour)
    segment = erosion_image[y:y+h, x:x+w]
    segments.append(segment)
return pp
def extraer_caracteristicas(imagen,imagen2):
    # Procesa la imagen para extraer características
    imagen_gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
    #matriz para realizar convolucion de imagenes
    kernel = np.ones((3, 3), np.uint8)
    #Realizar la conversion a imagen dilatada
    imagen_dilatada = cv2.dilate(imagen_gris, kernel, iterations=1)
    umbral = 127
    #Binarizacion de Imagen
    _, imagen_binaria = cv2.threshold(imagen_gris, umbral, 255,
cv2.THRESH_BINARY)
    imagen_invertida = cv2.bitwise_not(imagen_binaria)
    kernel = np.ones((3, 3), np.uint8)
    # Aplicar la erosión a la imagen binaria
    imagen_erodida = cv2.erode(imagen_binaria, kernel, iterations=1)
    #imagen_invertida = cv2.bitwise_not(imagen_erodida))
    segment = segment_characters(imagen_invertida,imagen,imagen2)
    if (segment=="None"):
        return "No se determino la placa"

```

```

else:
    return segment

def tomar_foto(client):
    global nombre_foto,rt, placa
    # Inicializar la webcam
    cap = cv2.VideoCapture(0)
    # Leer la imagen de la webcam
    ret, frame = cap.read()
    # Finalizar la captura de video
    cap.release()
    # Guardar la imagen en una carpeta
    # Obtener la fecha y hora actual
    fecha_actual = datetime.datetime.now()
    # Formatear la fecha y hora como una cadena de texto
    fecha_actual_str = fecha_actual.strftime("%Y-%m-%d %H:%M:%S")
    fecha_actual_str = fecha_actual_str.replace(" ", "_")
    fecha_actual_str = fecha_actual_str.replace(":", "-")
    nombre_foto="carro-"+fecha_actual_str+".jpg"
    rt = "fotografias/"+nombre_foto
    cv2.imwrite(rt, frame)
    # Llamar a la función para tomar la foto
    #tomar_foto()
    #Nombre para escoger la fotografia en caso de usar ya las fotos tomadas poner
    nombre_foto=rt
    nombre_foto="fotografias/borrosa.jpeg"
    #nombre_foto=rt
    rt=nombre_foto
    #leer la imagen para el procesamiento (cambiar por el nombre o la variable que sea la
    q toma la foto)
    imagen = cv2.imread('fotografias/borrosa.jpeg')
    placa = extraer_caracteristicas(imagen,nombre_foto)
    print("Valor de placa : ",placa)

```

```
def run():
    client = connect_mqtt()
    client.loop_start()
    publish(client)
    client.loop_stop()
```

```
if __name__ == '__main__':
    run()
```

**Anexo 7.** Código algoritmo de clasificación y diseño Interfaz gráfica y del

```
from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split, cross_val_score

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
confusion_matrix, roc_auc_score

import numpy as np

# Crear conjunto de características y etiquetas (aumentado con más datos si es posible)
X = [[2, 4], [3, 1], [5, 6], [7, 8], [6, 3], [2, 7], [8, 5], [4, 2]]
y = [0, 1, 1, 0, 0, 1, 1, 0]

# Dividir el conjunto de datos en entrenamiento y prueba (con validación cruzada
opcional)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Normalizar los datos (opcional dependiendo de las características)

# Aquí asumimos que no es necesario para este ejemplo.
```

```

# Crear el clasificador y entrenarlo (ajustando hiperparámetros si es necesario)

classifier = RandomForestClassifier(n_estimators=100, max_depth=3,
random_state=42)

classifier.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba

y_pred = classifier.predict(X_test)

# Calcular métricas de evaluación

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

conf_matrix = confusion_matrix(y_test, y_pred)

roc_auc = roc_auc_score(y_test, y_pred)

# Imprimir resultados

print("Precisión:", accuracy)

print("Precisión:", precision)

print("Recall:", recall)

print("F1-score:", f1)

print("Matriz de confusión:")

print(conf_matrix)

print("Área bajo la curva ROC:", roc_auc)

```

```
# Validación cruzada para obtener estimaciones más robustas del rendimiento
cv_scores = cross_val_score(classifier, X, y, cv=5, scoring='accuracy')
print("Precisión (Validación cruzada):", np.mean(cv_scores))
```

```
<style>
```

```
root, a, button, p, div {
  -webkit-user-select: none !important;
}
```

```
body {
  background: -webkit-linear-gradient(
    55deg,
    #009785 0%,
    #245aa5 50%,
    #b800e9 100%
  );
  -webkit-touch-callout: none !important;
}
```

```
body.nr-dashboard-theme md-toolbar,
body.nr-dashboard-theme md-content md-card {
  background-color: transparent !important;
  color: #FFFFFF;
}
```

```
ui-card-panel {
  background-color: rgba(255,255,255,0.1) !important;
  border-radius: 10px !important;
}
```

```
#Homey_Dashboard_Navigation {
  background-color: transparent !important;
  border-radius: 10px !important;
}
```

```

}

.md-card.md-default-theme, md-card {
  border-radius: 10px;
}

.nr-dashboard-switch.ng-scope {
  background-color: rgba(255,255,255,.2) !important;
}

.nr-dashboard-switch.ng-scope:hover {
  background-color: rgba(255,255,255,.5) !important;
}

.nr-dashboard-theme .nr-dashboard-button .md-button {
  background-color: rgba(255,255,255,.2);
}

.md-button {
  border-radius: 10px;
}

.nr-dashboard-cardpanel layout-column {
  background-color: rgba(255,255,255,0.2) !important;
  background-color: transparent !important;
  border-radius: 10px !important;
}

.nr-dashboard-theme ui-card-panel {
  border: none;
}

.nr-dashboard-template {

```

```
    overflow-y: visible;
}
.nr-dashboard-theme md-content md-card {
    background-color: transparent !important;
}

.nr-dashboard-theme ui-card-panel p.nr-dashboard-cardtitle {
    color: rgba(255, 255, 255, 0.5);
}
</style>

<script>
document.ontouchmove = function (e) {
    e.preventDefault();
}
</script>
```