



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
ESCUELA DE INFORMÁTICA E INTELIGENCIA ARTIFICIAL**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**SISTEMA DE CONTROL DE ACCESO VEHICULAR MEDIANTE
RECONOCIMIENTO DE PLACAS PARA EL PERSONAL DE LA PONTIFICIA
UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE IBARRA**

DANY RENE MEJIA SANDOVAL

TUTOR: ING. DARWIN PILLO

IBARRA – ECUADOR

FEBRERO, 2025

Ibarra, 3 de febrero de 2025

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de integración curricular titulado: **SISTEMA DE CONTROL DE ACCESO VEHICULAR MEDIANTE RECONOCIMIENTO DE PLACAS PARA EL PERSONAL DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE IBARRA**, presentado por el estudiante **DANY RENÉ MEJÍA SANDOVAL** con cédula de ciudadanía N°**1724675903**, para obtener el Título de INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.

28/1/25, 7:24

Turnitin - Informe de Originalidad - Trabajo de titulacion DANY RENE MEJIA SANDOVAL

Turnitin Informe de Originalidad					
Procesado el: 28-ene.-2025 07:21 -05 Identificador: 2569020587 Número de palabras: 12687 Entregado: 2					
Trabajo de titulacion DANY RENE MEJIA SANDOVAL Por DANY RENE MEJIA SANDOVAL	<table border="1"><thead><tr><th>Índice de similitud</th><th>Similitud según fuente</th></tr></thead><tbody><tr><td>6%</td><td>Internet Sources: 6% Publicaciones: 2% Trabajos del estudiante: 3%</td></tr></tbody></table>	Índice de similitud	Similitud según fuente	6%	Internet Sources: 6% Publicaciones: 2% Trabajos del estudiante: 3%
Índice de similitud	Similitud según fuente				
6%	Internet Sources: 6% Publicaciones: 2% Trabajos del estudiante: 3%				

(f:) 


Mgs. Darwin Marcelo Pillo Guanoluisa

C.C.: 1003319660

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El jurado examinador, aprueba el presente informe de investigación en nombre de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCESI):

(f): 
Mgs. Darwin Marcelo Pillo Guanoluisa
C.C.: 1003319660

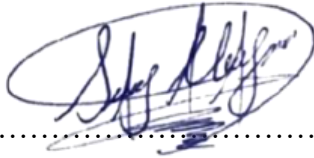
(f): 
Mgs. Álvaro Mauricio Cevallos Ramírez
C.C.: 1002494019

(f): 
Mgs. Stalin Marcelo Arciniegas Aguirre
C.C.: 1003496815

ACTA DE CESIÓN DE DERECHOS

Yo *Dany René Mejía Sandoval*, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilizaciones de sus obras o prestaciones, a título gratuito u oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 3 de Febrero de 2025



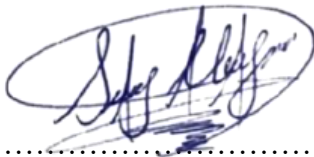
f):

Dany René Mejía Sandoval

C.C.:1724675903

AUTORÍA

Yo, *Dany René Mejía Sandoval*, portador de la cédula de ciudadanía N° 1724675903, declaro que la presente investigación es de total responsabilidad del (los) autor (es), y eximo expresamente a la Pontificia Universidad Católica del Ecuador Sede Ibarra de posibles reclamos o acciones legales.



f):

Dany René Mejía Sandoval

C.C.:1724675903

DEDICATORIA Y AGRADECIMIENTOS

Este trabajo es el reflejo de un esfuerzo constante y del apoyo invaluable de quienes han estado presentes en este camino.

A mi padre, un hombre experto en múltiples áreas y cuyo incansable trabajo ha sido un ejemplo para mí. Su habilidad para resolver cualquier desafío y su dedicación me han enseñado la importancia de ser perseverante y eficaz en la vida.

A mi madre, una mujer ejemplar, trabajadora incansable, que siempre estuvo a mi lado, brindándome su apoyo incondicional. En los momentos difíciles, sus sabios consejos y palabras de aliento fueron la fuerza que me permitió seguir adelante.

A mis hermanas, por su compañía constante, comprensión y apoyo incondicional. Su presencia ha sido un pilar fundamental en mi vida. A mis sobrinos, por llenar mis días de alegría y por recordarme que siempre hay una razón para sonreír, incluso en los momentos más complicados.

A los amigos que compartieron conmigo esta etapa de la carrera, enfrentando juntos los desafíos, las largas horas de estudio y disfrutando también de los momentos de descanso. Su amistad y apoyo hicieron de este proceso algo mucho más llevadero.

Aprecio a mis maestros, por su dedicación y paciencia al compartir sus conocimientos. Sus enseñanzas y su pasión por la educación han dejado una huella profunda en mi formación, tanto académica como personal.

A todas las personas que, con su aliento, consejos y compañía incondicional, me han acompañado a lo largo de este proceso y durante todos mis estudios. Cada gesto de confianza ha sido un impulso invaluable para alcanzar esta meta.

Finalmente, a la universidad por brindarme el espacio y las herramientas necesarias para desarrollarme y crecer, tanto en lo académico como en lo profesional.

A todos ustedes, mi más sincero y profundo agradecimiento.

ÍNDICE DE CONTENIDOS

Contenido	
CERTIFICACIÓN TUTOR.....	ii
PÁGINA DE APROBACIÓN DEL TRIBUNAL.....	iii
ACTA DE CESIÓN DE DERECHOS.....	iv
AUTORÍA.....	v
DEDICATORIA Y AGRADECIMIENTOS.....	vi
ÍNDICE DE CONTENIDOS.....	vii
INDICE DE TABLAS.....	x
INDICE DE FIGURAS.....	xi
RESUMEN.....	xii
ABSTRACT.....	xiii
INTRODUCCIÓN.....	1
CAPÍTULO 1.....	2
ESTADO DEL ARTE.....	2
1.1 ANTECEDENTES.....	2
1.2 MARCO TEORICO.....	4
1.2.1 Control de Acceso Vehicular.....	5
1.2.2 Tecnologías de Reconocimiento de Placas.....	5
1.2.2.1 OpenCV.....	6
1.2.2.2 EasyOCR.....	6
1.2.2.3 Machine Learning en el Reconocimiento de Imágenes.....	6
1.2.3 Desarrollo aplicación Web.....	7
1.2.3.1 Concepto de Framework.....	7
1.2.3.2 Nest.js como Framework para APIs.....	8
1.2.3.3 Quasar Framework para el Desarrollo de Aplicaciones.....	8
1.2.3.4 MySQL.....	9
1.2.4 Herramientas y materiales para el desarrollo de IoT.....	9
1.2.4.1 Definición de IoT.....	9
1.2.4.2 Arduino ID.....	10
1.2.4.3 Python para IoT.....	10
1.2.4.4 ESP32.....	11
1.2.4.5 Modulo Relé.....	11

1.2.4.6 Cámaras.....	12
CAPÍTULO II	13
MATERIALES Y MÉTODOS.....	13
2.1 MATERIALES Y MÉTODOS.....	13
2.1.2 Técnicas para la recolección de datos	14
2.1.3 Metodología.....	14
2.1.3.1 Desarrollo del Dispositivo IoT (Metodología de Prototipado)	15
2.1.3.2 Desarrollo de la página web (Metodología de Programación Extrema - XP)	22
2.1.4 Fase 1 (Planificación)	23
2.1.4.1 Historias de Usuario	23
2.1.4.2 Requisitos funcionales.....	26
2.1.4.3 Requisitos No Funcionales.....	28
2.1.4.4 Módulos	29
2.1.4.5 Roles y responsabilidades	32
2.1.4.6 Planificación de Actividades	33
2.1.4.7 Iteraciones.....	34
2.1.4.7.1 Iteración 1: Login al sistema (15 días)	34
2.1.4.7.2 Iteración 2: Gestión de Roles (5 días)	35
2.1.4.7.3 Iteración 3: Registro de vehículos (10 días).....	35
2.1.4.7.4 Iteración 4: Almacenamiento de datos en la BDD (20 días).....	35
2.1.4.7.5 Iteración 5: Monitoreo de ingreso/salida (25 días)	36
2.1.4.7.6 Iteración 6: Generación de reportes (10 días).....	36
2.1.4.7.7 Iteración 7: Exportación de reportes a Excel (5 días).....	36
2.1.5 Fase 2 (Diseño)	37
2.1.5.1 Arquitectura de microservicios	37
2.1.5.2 Funcionalidades del Sistema	37
2.1.5.2.1 Casos de Uso	37
2.1.5.2.2 Diagrama de Actividades.....	39
2.1.5.2.3 Base de Datos.....	41
2.1.5.2.4 Diseño de la Interfaz.....	42
2.1.6 Fase 3 (Codificación).....	43
2.1.7 Fase 4 (Pruebas)	44
3. Resultados Fase 1. Sistema Web	46
3.1.1 Vista Login	46
3.1.2 Vista Registro Personal.....	47
3.1.3 Vista Registrar Vehículo.....	47

3.1.4 Vista Seguimiento de Usuarios	48
3.1.5 Vista Reporte Diario	48
3.1.6 Vista Reporte Mensual	49
3.2 Resultados Fase 2. Prototipo lot.....	50
3.2.1 Prototipo Finalizado	50
3.3 Pruebas del sistema	51
CONCLUSIONES	53
1. Automatización del Control de Acceso:.....	53
2. Optimización en el Reconocimiento de Placas:	53
3. Seguridad y Control:.....	53
4. Mejora en la Experiencia del Usuario:.....	53
5. Escalabilidad y Flexibilidad:	53
6. Potencial para Implementación en Grandes Instalaciones:.....	53
RECOMENDACIONES.....	54
1. Mejora en la Precisión del Reconocimiento de Placas:	54
2. Implementación de Cámaras de Mayor Calidad:.....	54
3. Manejo de Condiciones Ambientales Adversas:.....	54
4. Expansión del Sistema a Zonas de Mayor Escala:.....	54
5. Optimización del Algoritmo de Reconocimiento:	54
6. Mejorar la Gestión de Usuarios:.....	54
7. Implementación de Cámaras para Identificación de Conductores:	54
8. Conteo de Espacios Disponibles:	54
Anexos.....	55
REFERENCIA BIBLIOGRÁFICA	57

INDICE DE TABLAS

Tabla 1: Requisitos Funcionales.....	17
Tabla 2: Requisitos no funcionales.....	17
Tabla 3: Historias de usuario	23
Tabla 4: Roles y responsabilidades	32
Tabla 5: Planificación de actividades	33
Tabla 6: Pruebas (Sistema).....	44
Tabla 7: Pruebas (Iot)	45
Tabla 8: Pruebas obtenidas (sistema)	51
Tabla 9: Pruebas obtenidas (Iot).....	52

INDICE DE FIGURAS

Figura 1: Primer caso de uso	18
Figura 2: Segundo caso de uso	18
Figura 3: Tercer Caso de Uso	18
Figura 4: Cuarto caso de uso	19
Figura 5: Quinto caso de uso	19
Figura 6: Diagrama de actividades del prototipo IoT	20
Figura 7: Arquitectura iot flujo de datos	21
Figura 8: Arquitectura iot flujo herramientas	21
Figura 9: Primer caso de uso (sistema).....	37
Figura 10: Segundo caso de uso (sistema).....	38
Figura 11: : Tercer caso de uso (sistema)	38
Figura 12: : Cuarto caso de uso (sistema).....	38
Figura 13: Diagrama de actividades (registro vehículo)	39
Figura 14: Diagrama de actividades (ingreso salida de vehículos)	39
Figura 15: Diagrama de actividades (Exportación de reportes)	40
Figura 16: Diagrama de la base de datos	41
Figura 17: Interfaz preliminar.....	42
Figura 18: Vista Login.....	46
Figura 19: Vista Registro Personal	47
Figura 20: Vista Registrar Vehículo	47
Figura 21: Vista Seguimiento de Usuarios	48
Figura 22: Vista Reporte Diario	48
Figura 23: Vista Reporte Mensual.....	49
Figura 24: Prototipo Finalizado	50
Figura 25: Objeto percibido por la cámara	55
Figura 26: La cámara está en espera para percibir un objeto	55
Figura 27: Modelo detecta vehículo	55
Figura 28: Extracción de placa / Ingreso al sistema	56
Figura 29: Registro de entrada del vehículo al sistema	56
Figura 30: Registro de salida del vehículo al sistema	56

RESUMEN

Este informe detalla el desarrollo del prototipo de un sistema de control de acceso vehicular basado en el reconocimiento automático de placas mediante visión por computadora e inteligencia artificial. El sistema está diseñado para capturar imágenes de los vehículos mediante una cámara de seguridad, procesarlas con OpenCV para la detección y segmentación de la placa, y emplear EasyOCR y TensorFlow para la extracción y reconocimiento de caracteres.

Una vez identificada la matrícula, la información es comparada con una base de datos alojada en MySQL, donde se registran los vehículos autorizados y los eventos de ingreso y salida. Para la visualización y gestión de los registros en tiempo real, se implementó una interfaz web desarrollada en Quasar Framework, la cual permite a los usuarios monitorear el acceso vehicular de manera remota desde cualquier dispositivo con conexión a la red.

El sistema también incorpora un ESP32 y un módulo de relé, los cuales permiten la automatización del mecanismo de apertura y cierre de una barrera o portón eléctrico, activándose únicamente cuando se reconoce una matrícula autorizada. Esta integración de visión por computadora, aprendizaje profundo con TensorFlow e IoT proporciona una solución eficiente y de bajo costo para optimizar la seguridad en estacionamientos y zonas restringidas.

Las pruebas realizadas con el prototipo muestran que el sistema es funcional y capaz de reconocer matrículas en diferentes condiciones de iluminación y ángulos de captura. Estos resultados permiten evaluar su desempeño y plantear mejoras para futuras versiones con el fin de aumentar su precisión y adaptabilidad en entornos reales.

Palabras clave: reconocimiento de placas, OpenCV, EasyOCR, TensorFlow, control de acceso, ESP32, Quasar Framework, visión por computadora, IoT.

ABSTRACT

This report details the development of a prototype for a vehicle access control system based on automatic license plate recognition using computer vision and artificial intelligence. The system captures vehicle images through a security camera, processes them with OpenCV for plate detection and segmentation, and utilizes EasyOCR and TensorFlow for character extraction and recognition.

Once the license plate is identified, the information is compared with a database stored in MySQL, where authorized vehicles and access events are recorded. For real-time monitoring and data management, a web interface developed with Quasar Framework was implemented, allowing users to remotely supervise vehicle access from any network-connected device.

The system also integrates an ESP32 microcontroller and a relay module, enabling the automation of a barrier or electric gate, which is activated only when an authorized license plate is recognized. This combination of computer vision, deep learning with TensorFlow, and IoT provides an efficient and cost-effective solution to enhance security in parking lots and restricted areas.

Tests conducted with the prototype demonstrate that the system is functional and capable of recognizing license plates under different lighting conditions and viewing angles. These results allow for performance evaluation and potential improvements in future versions to increase accuracy and adaptability in real-world environments.

Keywords: license plate recognition, OpenCV, EasyOCR, TensorFlow, access control, ESP32, Quasar Framework, computer vision, IoT.

INTRODUCCIÓN

El control de acceso vehicular se ha vuelto esencial para mejorar la seguridad y la eficiencia en la gestión del tránsito en instituciones educativas y empresas. Un sistema automatizado que controle el flujo de vehículos mediante el reconocimiento de placas vehiculares presenta múltiples ventajas, tales como la reducción de tiempos de espera, la disminución de errores humanos y el fortalecimiento de las medidas de seguridad.

Este trabajo de titulación tiene como finalidad desarrollar un prototipo de sistema de control de acceso vehicular para la Pontificia Universidad Católica del Ecuador, sede Ibarra. El sistema empleará las librerías *OpenCV* y *EasyOCR*, que integran algoritmos de *Machine Learning* para el procesamiento de imágenes y el reconocimiento óptico de caracteres. Esto permitirá registrar automáticamente las placas de los vehículos que acceden al campus. La implementación de esta tecnología busca optimizar la gestión del tránsito interno, disminuir la congestión vehicular y mejorar la seguridad en los accesos.

El enfoque principal de este proyecto es desarrollar un sistema funcional que capture y registre automáticamente la información de las placas vehiculares, y permita su visualización y gestión a través de una plataforma web, facilitando un control más eficiente del flujo de vehículos dentro del campus universitario. Además, se evaluarán y caracterizarán los sensores y dispositivos necesarios para garantizar el funcionamiento adecuado del prototipo en el garaje del Edificio 4.

Para alcanzar estos propósitos, se han planteado los siguientes objetivos específicos:

- Realizar una revisión bibliográfica sobre tecnologías de control de acceso vehicular, desarrollo web y reconocimiento óptico de caracteres.
- Analizar la situación actual del flujo vehicular y las medidas de seguridad en la Pontificia Universidad Católica del Ecuador.
- Caracterizar los sensores y dispositivos necesarios para el sistema de control de acceso vehicular.
- Desarrollar un prototipo de reconocimiento de matrículas mediante *Machine Learning* con *OpenCV* y *EasyOCR*, complementado con una plataforma web de gestión.

CAPÍTULO 1

ESTADO DEL ARTE

El desarrollo del estado del arte implica la recopilación y análisis de información relevante sobre las tecnologías aplicadas al control de acceso vehicular, centrándose en el reconocimiento de matrículas. Se lleva a cabo una revisión bibliográfica que abarca estudios y desarrollos recientes relacionados con **OpenCV** y **EasyOCR**, bibliotecas que integran técnicas de **Machine Learning** para reconocer y procesar los caracteres.

La investigación se centra en identificar las metodologías empleadas en proyectos similares y examinar los algoritmos de aprendizaje automático que sustentan estas herramientas. Se exploran casos de uso que demuestran la eficacia de estas tecnologías para automatizar y brindar una mayor facilidad al momento de monitorear.

Además, se destaca las limitaciones y desafíos que enfrentan estos sistemas en implementaciones previas, lo que permite fundamentar la propuesta del prototipo para la Pontificia Universidad Católica del Ecuador, sede Ibarra. A través de esta revisión, se establecen las bases teóricas necesarias para el desarrollo de un sistema que no solo es funcional, sino también innovador en la aplicación de tecnologías de vanguardia en el ámbito del control de acceso vehicular.

1.1 ANTECEDENTES

Según Aguilar Anaya (2022), el trabajo de Ashrafee et al. se centra en el desarrollo de un sistema para el reconocimiento de matrículas de vehículos en tiempo real, optimizado para operar en condiciones de recursos limitados. Según Aguilar Anaya (2022), el trabajo de Ashrafee et al. se centra en el desarrollo de un sistema para el reconocimiento de matrículas de vehículos en tiempo real, optimizado para operar en condiciones de recursos limitados. En su investigación, enfatizan la necesidad de crear soluciones que sean capaces de detectar, localizar y reconocer caracteres en matrículas a partir de secuencias de video. Para alcanzar estos objetivos, proponen un enfoque de detección de dos etapas que integra una API de visión, lo que permite obtener velocidades de inferencia en tiempo real y mantener un alto nivel de precisión en la detección y el reconocimiento. Priorizando

las detecciones de alta confianza para minimizar el tiempo de inferencia. Asimismo, utilizan una técnica de separación temporal de marcos para identificar varias matrículas dentro de un solo clip de video. Debido a la falta de conjuntos de datos públicos de matrículas en bengalí, los investigadores generan su propio conjunto de datos compuesto por imágenes y videos. Como resultado, logran entrenar modelos que alcanzan una puntuación AP (0.5) del 86 % y muestran un rendimiento razonable en la detección y reconocimiento, con una tasa de detección del 82.7 % y un puntaje OCR F1 del 60.8 %, además de una velocidad de procesamiento de 27.2 fotogramas por segundo.

Asimismo, el trabajo de Niño-Rondón et al. (2021) presenta una metodología complementaria en la detección de placas vehiculares, centrada en la implementación de un modelo de clasificador en cascada tipo Haar, similar al utilizado por Ashrafee et al. Sin embargo, el enfoque de Niño-Rondón et al. se basa en imágenes capturadas en una zona urbana con condiciones de iluminación y tráfico típicas, utilizando una Raspberry Pi para la adquisición de datos y OpenCV para el desarrollo del modelo. A diferencia de la solución de Ashrafee et al., que optimiza el rendimiento para entornos de recursos limitados mediante la integración de una API de visión y un enfoque de separación temporal de marcos, el estudio de Niño-Rondón et al. pone énfasis en la evaluación de falsos positivos y falsos negativos, así como en el tiempo de respuesta y uso de memoria del sistema. Estas diferencias reflejan los distintos contextos y objetivos de ambos estudios, contribuyendo cada uno con un enfoque único en el campo del reconocimiento de matrículas vehiculares.

Peña Morán (2023) propone un sistema de reconocimiento de placas salvadoreñas para la Universidad de El Salvador, seleccionando y evaluando diferentes opciones de software libre en los repositorios más populares de internet. Los modelos seleccionados fueron OpenALPR y la combinación de YOLOv5 con PaddleOCR, los cuales se instalaron en un entorno de prueba. Para la evaluación, se preparó un conjunto de imágenes de placas salvadoreñas que facilitó la comparación del rendimiento de los modelos en cuanto a precisión., tiempo de procesamiento y nivel de certeza. Los resultados demostraron que la combinación de YOLOv5 con PaddleOCR obtuvo el mejor desempeño, convirtiéndose en la propuesta final para implementar en la universidad. Este sistema destaca por su

capacidad de reconocer placas con alta precisión y rapidez, optimizando el uso de los recursos disponibles.

1.2 MARCO TEORICO

El marco teórico constituye una parte esencial en la estructuración de cualquier proyecto de investigación, ya que establece el contexto y la base conceptual sobre la cual se desarrolla el estudio. En el caso del presente trabajo, se enfoca en el diseño y desarrollo de un sistema de control de acceso vehicular mediante el reconocimiento de placas en la Pontificia Universidad Católica del Ecuador. Este sistema tiene como objetivo principal optimizar la seguridad y la eficiencia en la gestión del tránsito dentro del campus universitario, un aspecto crucial en la actualidad debido al aumento del tráfico y las preocupaciones de seguridad.

Para abordar este problema, se exploran conceptos clave relacionados con el control de acceso vehicular y las tecnologías que lo hacen posible. En particular, se analiza el uso de OpenCV y EasyOCR, herramientas fundamentales que permiten el procesamiento y reconocimiento óptico de caracteres a partir de imágenes. Estas librerías utilizan técnicas de machine learning para mejorar la precisión en la identificación de vehículos, lo que contribuye a la automatización del proceso, minimizando la intervención humana y los errores asociados.

Asimismo, se discute la implementación de APIs utilizando Nest.js, que proporciona una arquitectura eficiente y escalable para la comunicación entre el sistema de control de acceso y la base de datos. La interfaz de usuario, desarrollada con el framework Quasar, ofrecerá una plataforma intuitiva para la gestión y visualización de los datos recopilados, facilitando así la toma de decisiones informadas por parte de los administradores del campus.

A través de una revisión bibliográfica exhaustiva de estudios previos y tecnologías existentes, este marco teórico no solo contextualiza el problema de investigación, sino que también identifica oportunidades de mejora en las soluciones actuales. Esto fundamenta los objetivos específicos del proyecto y establece un sólido cimiento teórico que guía el desarrollo del prototipo, subrayando la importancia de integrar diversas tecnologías en un único sistema de control de acceso vehicular.

1.2.1 Control de Acceso Vehicular

El control de acceso vehicular se refiere a un sistema diseñado para gestionar la entrada y salida de vehículos en un área determinada. Este sistema emplea diversas tecnologías, como lectores de matrículas y sensores de proximidad, para identificar y autenticar vehículos. Entre los principios fundamentales se encuentran la verificación de la identidad del vehículo, el registro de su acceso y la emisión de permisos o restricciones según las políticas establecidas. La integración de tecnologías de **machine learning** en estos sistemas permite mejorar la precisión en la identificación de matrículas y optimizar la gestión del tráfico, ya que se adaptan y aprenden de patrones previos de acceso.

1.2.2 Tecnologías de Reconocimiento de Placas

El reconocimiento óptico de caracteres (OCR) es una tecnología que convierte imágenes de texto, como las de documentos escaneados o fotografías, en un formato digital procesable por máquinas. Este proceso utiliza algoritmos que identifican y traducen la estructura de los caracteres presentes en la imagen a texto legible electrónicamente. El OCR se aplica en diversas áreas, como la digitalización de documentos, la automatización de tareas y la mejora de la accesibilidad a la información, facilitando así la gestión y búsqueda de datos en formato digital.

Amazon Web Services (s. f.) señala que el OCR transforma las imágenes de texto en un formato compatible con software empresarial, lo que optimiza el manejo de información. Esta tecnología mejora la eficiencia al convertir documentos físicos en archivos editables y buscables, lo que favorece la accesibilidad y la organización de datos.

1.2.2.1 OpenCV

OpenCV es muy útil para el desarrollo de sistemas de reconocimiento de matrículas vehiculares, debido a su capacidad para detectar objetos y procesar imágenes con alta precisión. Utiliza técnicas avanzadas como la detección de bordes y el reconocimiento óptico de caracteres (OCR) para identificar matrículas en imágenes o videos, lo que facilita su uso en aplicaciones como el control de acceso a garajes en universidades. Su eficiencia permite procesar imágenes en tiempo real, incluso en condiciones adversas, y es ampliamente utilizada en sistemas de seguridad y monitoreo (Rodríguez, 2021).

1.2.2.2 EasyOCR

EasyOCR es una biblioteca de reconocimiento óptico de caracteres (OCR) desarrollada en Python que facilita la extracción de texto de imágenes mediante técnicas avanzadas de visión por computadora y aprendizaje automático. Esta herramienta es capaz de reconocer texto en múltiples idiomas y se basa en redes neuronales profundas, lo que garantiza una alta precisión en la detección y conversión de caracteres impresos y escritos a mano (Reconocimiento Óptico de Caracteres con EasyOCR y Python, 2024). En el contexto de este proyecto, EasyOCR será fundamental para procesar las imágenes capturadas de los vehículos, permitiendo la digitalización automática de las matrículas y su integración en una base de datos para facilitar la gestión y el control de acceso

1.2.2.3 Machine Learning en el Reconocimiento de Imágenes

El reconocimiento de imágenes ha evolucionado significativamente con la incorporación de técnicas de Machine Learning, particularmente a través de las redes neuronales convolucionales (CNN). Estas arquitecturas permiten a las máquinas aprender automáticamente a identificar patrones en imágenes, lo que las hace especialmente útiles en aplicaciones como el reconocimiento de matrículas. Modelos como YOLO (You Only Look Once) han demostrado ser efectivos al combinar velocidad y precisión en la detección de objetos en tiempo real, mientras que métodos

tradicionales como los Support Vector Machines (SVM) ofrecen enfoques alternativos que, aunque útiles, son menos eficientes comparados con las técnicas de deep learning (Hussain et al., 2022; Viso, 2024).

Sin embargo, el uso de Machine Learning para el reconocimiento óptico de caracteres presenta varios desafíos. La necesidad de conjuntos de datos grandes y etiquetados, la variabilidad en las condiciones de captura de imágenes y la capacidad limitada de los modelos para generalizar a nuevas situaciones son limitaciones significativas que deben ser consideradas al desarrollar sistemas de reconocimiento de matrículas (Khan et al., 2022; Viso, 2024).

Por su parte, MobileNetV2 mejora la eficiencia mediante el uso de convoluciones separables en profundidad y la estructura de residuales invertidos, lo que permite que el modelo sea ligero y rápido, lo cual es esencial para aplicaciones en tiempo real, como el monitoreo del tráfico y el reconocimiento de matrículas. En combinación con SSD, que discretiza el espacio de salida de las cajas delimitadoras, este enfoque ha demostrado ser altamente efectivo en la localización y clasificación de vehículos en imágenes (Liu et al., 2016; TensorFlow, 2024).

A pesar de sus ventajas, el uso de SSD MobileNetV2 también enfrenta los desafíos inherentes al entrenamiento en grandes conjuntos de datos etiquetados y la capacidad del modelo para adaptarse a diferentes condiciones de iluminación y ángulos de captura. La generalización del modelo a nuevas situaciones y contextos sigue siendo un área activa de investigación (Zhou et al., 2023).

1.2.3 Desarrollo aplicación Web

1.2.3.1 Concepto de Framework

Un framework es una estructura que provee un conjunto de herramientas, reglas y convenciones diseñadas para optimizar el desarrollo de software, especialmente en proyectos complejos o de gran escala. Estos marcos de trabajo simplifican el proceso de creación de aplicaciones al ofrecer una base funcional que los desarrolladores pueden utilizar como punto de partida, lo cual reduce el tiempo y los recursos necesarios. En

lugar de diseñar una arquitectura desde cero, el framework facilita la reutilización de componentes y soluciones ya probadas, lo que permite a los desarrolladores centrarse en el diseño de características específicas y únicas para su proyecto. Así, "permiten desarrollar software de manera más eficiente y rápida, al proporcionar una estructura básica que ahorra tiempo y esfuerzo" (Lucena, 2023). Además, los frameworks abordan problemas técnicos comunes y proporcionan una estructura modular que facilita la integración de nuevos componentes, mejorando la escalabilidad y el mantenimiento del software a largo plazo.

1.2.3.2 Nest.js como Framework para APIs

NestJS es un framework innovador para Node.js que utiliza TypeScript, diseñado para desarrollar aplicaciones del lado del servidor con un enfoque en la escalabilidad y la mantenibilidad. Su estructura modular permite una organización eficiente del código, facilitando la creación de componentes que se pueden reutilizar en diferentes partes del proyecto. Además, NestJS se integra sin dificultad con múltiples bibliotecas y herramientas, lo que apoya la implementación de APIs RESTful y microservicios. En definitiva, NestJS se presenta como una solución avanzada y eficaz para el desarrollo de aplicaciones en el ecosistema de JavaScript y TypeScript.

Nest (NestJS) es un marco para crear aplicaciones escalables y eficientes del lado del servidor en Node.js. Utiliza JavaScript progresivo, está creado con TypeScript y es totalmente compatible con él (pero permite a los desarrolladores codificar en JavaScript puro) y combina elementos de OOP (programación orientada a objetos), FP (programación funcional) y FRP (programación reactiva funcional) (NestJS, s. f.).

1.2.3.3 Quasar Framework para el Desarrollo de Aplicaciones

Quasar es un framework versátil que facilita el desarrollo de aplicaciones de monitoreo y gestión vehicular, permitiendo a los desarrolladores crear interfaces modernas y funcionales utilizando Vue.js como lenguaje de programación base. Su capacidad para

construir aplicaciones web, móviles y de escritorio lo convierte en una opción ideal para proyectos que requieren una integración fluida de diferentes plataformas y dispositivos.

Al utilizar Quasar, es posible implementar componentes visuales atractivos que mejoran la experiencia del usuario y optimizan la visualización de datos en tiempo real. La reactividad y la simplicidad de Vue.js aseguran que los desarrolladores puedan manejar de manera efectiva la manipulación de datos, un elemento crucial en la gestión de flotas vehiculares. Como se menciona en la documentación, "Quasar permite crear aplicaciones web, móviles y de escritorio desde un solo código base usando Vue.js" (Quasar Framework, s. f.).

1.2.3.4 MySQL

MySQL es un sistema de gestión de bases de datos relacional muy popular en el desarrollo de aplicaciones y sistemas de información. Debido a su arquitectura de código abierto y a su soporte para SQL, facilita el almacenamiento, la recuperación y la manipulación de datos de manera eficiente. Su estructura modular permite realizar operaciones complejas sobre datos de manera escalable y segura, lo cual lo hace adecuado tanto para pequeñas aplicaciones como para sistemas empresariales de gran envergadura. Además, su compatibilidad con diversas plataformas y su integración con otros lenguajes de programación lo convierten en una herramienta versátil y confiable en entornos de desarrollo de software (Oracle Corporation, 2023).

1.2.4 Herramientas y materiales para el desarrollo de IoT

1.2.4.1 Definición de IoT

El Internet de las Cosas (IoT) es una tecnología que conecta dispositivos físicos a internet para compartir información entre sí y con sistemas de control, facilitando el monitoreo y la automatización en tiempo real. Según Amazon Web Services (s. f.), técnicamente, se refiere a la red de dispositivos interconectados y a la tecnología que facilita la

comunicación entre estos y la nube, así como entre los propios dispositivos. Este avance ha sido posible gracias a la aparición de chips económicos y redes de alta capacidad, lo que permite que miles de millones de dispositivos. Aunque el IoT trae importantes beneficios en términos de eficiencia y gestión de recursos, también presenta desafíos relacionados con la seguridad y la privacidad debido al intercambio constante de datos.

1.2.4.2 Arduino ID

El uso de la plataforma Arduino se ha popularizado en el ámbito del desarrollo de proyectos electrónicos y de Internet de las Cosas (IoT). Uno de los factores clave que ha contribuido a su éxito es la accesibilidad de su entorno de desarrollo, conocido como IDE (Integrated Development Environment). Este software proporciona un conjunto de herramientas que simplifican el proceso de programación y permiten a los usuarios, tanto principiantes como experimentados, interactuar fácilmente con el hardware de Arduino.

El IDE es un conjunto de herramientas de software que permiten a los programadores desarrollar y grabar todo el código necesario para hacer que nuestro Arduino funcione como queramos. El IDE de Arduino nos permite escribir, depurar, editar y grabar nuestro programa (llamados “sketches” en el mundo Arduino) de una manera sumamente sencilla; en gran parte, a esto se debe el éxito de Arduino, a su accesibilidad (Aguayo, 2022).

1.2.4.3 Python para IoT

Python se ha convertido en una herramienta clave en el desarrollo de proyectos de Internet de las Cosas (IoT) gracias a su facilidad de uso y flexibilidad. Su código claro permite a los programadores conectar y controlar dispositivos de forma eficiente, como sensores y microcontroladores, que son fundamentales en IoT. Python facilita la recopilación y el envío de datos a otros dispositivos o a la nube, permitiendo la interacción entre múltiples elementos de manera sencilla. Además, es compatible con plataformas como ESP32 lo que hace posible el desarrollo de sistemas conectados y su monitoreo remoto.

Python es ampliamente reconocido como una excelente opción para el desarrollo en IoT, gracias a su facilidad de aprendizaje, sintaxis limpia y una comunidad sólida que lo respalda. Además, Python ofrece una gran cantidad de bibliotecas

útiles para el trabajo con dispositivos IoT, lo que permite a los desarrolladores desarrollar y probar prototipos rápidamente. Incluso en proyectos que se reescriben en otros lenguajes como C o Java para mejorar el rendimiento, Python es una herramienta eficaz para el desarrollo de dispositivos y la reducción de datos en la nube (Internet of Things with Python - Javatpoint, s. f.).

1.2.4.4 ESP32

En el ámbito del Internet de las Cosas (IoT), el microcontrolador ESP32 ha demostrado ser una solución eficiente y versátil. Gracias a sus capacidades de conectividad WiFi y Bluetooth, es posible implementar sistemas de control y automatización que operen de manera remota y en tiempo real. Estas características hacen del ESP32 una herramienta esencial para proyectos que buscan un equilibrio entre alto rendimiento y bajo consumo energético. Su arquitectura de doble núcleo y su compatibilidad con protocolos como SPI, I2C y UART lo convierten en una excelente opción para aplicaciones de monitoreo y control a distancia.

El ESP32, al integrar múltiples protocolos y ofrecer opciones de ahorro de energía, se adapta a diversas aplicaciones, garantizando la eficiencia y funcionalidad en escenarios de operación continua (Espressif Systems, 2023).

1.2.4.5 Modulo Relé

El módulo de relé de 2 canales es una herramienta fundamental en proyectos de electrónica, ya que permite controlar dispositivos de corriente alterna (AC) o de corriente continua (DC) que requieren mayor potencia. Este tipo de relé actúa como un interruptor controlado electrónicamente, lo que facilita que un microcontrolador, como un Arduino o un ESP32, pueda activar o desactivar dispositivos de alto voltaje, tales como luces, motores y electrodomésticos. En el módulo, los pines de control ubicados en el lado derecho se conectan al microcontrolador para gestionar los relés, mientras que los terminales en el lado izquierdo permiten la conexión de los dispositivos a ser controlados. Como se menciona en el artículo de Tindie (2021), “los relés funcionan como

interruptores controlados electrónicamente, permitiendo que un microcontrolador... active o desactive dispositivos de mayor voltaje o corriente” (Tindie, 2021).

1.2.4.6 Cámaras

Las cámaras son componentes esenciales en los sistemas de reconocimiento de placas vehiculares, ya que permiten capturar imágenes claras y detalladas de las placas en diversas condiciones de luz y movimiento. Para maximizar la precisión en la lectura de las placas, se suelen emplear cámaras de alta resolución, a menudo con tecnología infrarroja (IR) para asegurar visibilidad nocturna, y cámaras de alta velocidad que evitan el desenfoque en vehículos en movimiento. Además, se pueden integrar tecnologías de procesamiento de imágenes, como OpenCV y EasyOCR, que analizan el texto capturado, logrando una identificación eficiente de los vehículos en aplicaciones de control de acceso, seguridad y monitoreo en carreteras.

CAPÍTULO II

MATERIALES Y MÉTODOS

Este capítulo describe la metodología empleada para el desarrollo de un sistema de reconocimiento de matrículas vehiculares, realizado en el contexto de la Pontificia Universidad Católica del Ecuador, Sede Ibarra, como parte del proyecto de titulación. Se detallan los procedimientos implementados para alcanzar los objetivos establecidos, abarcando la recolección, procesamiento y análisis de datos relacionados con las matrículas vehiculares.

Además, se explican las técnicas utilizadas para garantizar la precisión y funcionalidad del sistema, alineadas con los principios metodológicos establecidos. Este apartado también justifica las decisiones tomadas durante el desarrollo, asegurando que las estrategias aplicadas sean coherentes con los objetivos planteados y con el contexto académico e institucional en el que se enmarca el proyecto.

2.1 MATERIALES Y MÉTODOS

Este apartado abordó los aspectos fundamentales que guiaron la investigación, centrada en el diseño e implementación de un sistema automatizado para el control de acceso vehicular. El proyecto combinó conocimientos teóricos y prácticos, utilizando tecnologías como OpenCV, EasyOCR, y frameworks modernos como NestJS y Quasar. Estas herramientas permitieron desarrollar un sistema eficiente y funcional, adaptado a las características y necesidades del campus universitario.

Se adoptó un enfoque cualitativo que facilitó comprender las necesidades del personal de seguridad y los usuarios del sistema. Este enfoque se fundamentó en la recolección de datos a través de entrevistas no estructuradas, lo que permitió profundizar en las expectativas y preocupaciones asociadas al control de acceso vehicular. Estos insumos fueron clave para definir las funcionalidades y características principales del sistema.

La elección de este enfoque metodológico cualitativo se justificó por la importancia de explorar en detalle las experiencias y contextos del flujo vehicular dentro del campus universitario. Este análisis enriqueció la toma de decisiones durante el diseño del

prototipo, asegurando que la solución tecnológica respondiera a los requerimientos operativos del personal y ofreciera una experiencia de usuario intuitiva y efectiva.

El propósito central de esta investigación aplicada fue ofrecer una solución práctica para optimizar la seguridad y la gestión del tránsito vehicular en el campus, integrando datos y tecnologías de vanguardia para atender necesidades reales de los usuarios.

2.1.2 Técnicas para la recolección de datos

Para obtener información detallada sobre las necesidades del personal de seguridad y los usuarios del sistema, se realiza una entrevista no estructurada con el jefe del área de seguridad de la Pontificia Universidad Católica del Ecuador, Sede Ibarra (PUCE-I), el Sr. Carlos Lima. Esta entrevista permite un enfoque flexible y abierto, en el cual se exploran diversas áreas relacionadas con el funcionamiento del sistema de seguridad, sin un guion rígido que limite las respuestas del entrevistado. Durante la conversación, se abordan temas como el uso actual del sistema de monitoreo, los problemas que enfrenta el personal de seguridad en el día a día, las expectativas en cuanto a la precisión y eficacia del sistema, y los aspectos operativos que deben mejorarse. La naturaleza no estructurada de la entrevista permite profundizar en aspectos específicos que surgen de la conversación, como las condiciones de iluminación, el flujo de vehículos y otras situaciones particulares que afectan la efectividad del sistema de reconocimiento de matrículas. La información recabada de esta manera se utiliza para entender mejor las necesidades reales del personal de seguridad y diseñar un sistema que sea fácil de usar y eficiente en el contexto específico de la universidad. Los datos obtenidos en esta entrevista son fundamentales para tomar decisiones informadas sobre el diseño e implementación del sistema y realizar ajustes según sea necesario durante el desarrollo del proyecto.

2.1.3 Metodología

En este proyecto se utilizará una combinación de metodologías ágiles, específicamente Extreme Programming (XP) y Prototipado. Estas metodologías se seleccionan por su capacidad para adaptarse a los cambios constantes que pueden surgir durante el desarrollo de sistemas interactivos como el de reconocimiento de matrículas, además de su enfoque en la mejora continua y la satisfacción del usuario final.

La metodología XP se aplicará para asegurar que el desarrollo del sistema sea flexible, con una estrecha colaboración entre el equipo de trabajo y el cliente, en este caso, el personal de seguridad de la PUCE-I. XP fomenta ciclos de desarrollo cortos e iterativos, donde se realiza un trabajo incremental y se obtienen retroalimentaciones constantes. Esto permitirá realizar ajustes frecuentes en el sistema de acuerdo con los comentarios del usuario, garantizando que las funcionalidades y características del sistema se alineen con las necesidades reales de los usuarios.

Por otro lado, la metodología de Prototipado se utilizará para crear versiones iniciales del sistema que permitan visualizar y probar rápidamente las funcionalidades del sistema antes de su implementación final. A través de la creación de prototipos, se podrán identificar y corregir posibles problemas de usabilidad y diseño desde las primeras etapas del proyecto. Esta metodología también facilita la validación continua del sistema por parte de los usuarios, asegurando que el producto final sea efectivo y fácil de utilizar.

En conjunto, XP y Prototipado proporcionan un marco flexible y enfocado en el usuario, lo que facilitará el desarrollo de un sistema de monitoreo de matrículas eficiente, adaptado a las necesidades del personal de seguridad y con un enfoque práctico y continuo hacia la mejora.

2.1.3.1 Desarrollo del Dispositivo IoT (Metodología de Prototipado)

La metodología de prototipado fue empleada en el desarrollo del sistema de control de acceso vehicular, ya que permite construir y refinar un modelo funcional de manera iterativa. Este enfoque se utilizó para garantizar que el sistema cumpliera con los requisitos funcionales y no funcionales definidos, a la vez que se adaptaba a las necesidades del personal de seguridad y usuarios finales.

El prototipado permitió visualizar tempranamente las funcionalidades del sistema, identificar posibles fallos y realizar ajustes oportunos antes de la implementación final. La participación activa de los usuarios durante el proceso fue clave para lograr un prototipo que no solo resultara técnicamente viable, sino también intuitivo y adecuado para el contexto de uso en la Pontificia Universidad Católica del Ecuador, sede Ibarra.

El desarrollo del sistema de control de acceso vehicular mediante reconocimiento de placas siguió varias fases:

1. **Recolección de Requisitos:** Se recopilaron las necesidades y expectativas del personal de seguridad a través de entrevistas y observaciones en el lugar de implementación. Esta información se utilizó para definir los objetivos y las funcionalidades principales del sistema.
2. **Diseño del Prototipo:** Se elaboró un diseño preliminar que incluyó la selección de tecnologías, la arquitectura del sistema y diagramas de flujo para representar los procesos de captura de imágenes, reconocimiento de placas y gestión de datos.
3. **Desarrollo y Pruebas Iniciales:** Se construyó una versión funcional inicial del sistema, integrando cámaras con algoritmos de reconocimiento óptico de caracteres, junto con el desarrollo de una plataforma web. Esta versión fue probada para validar su funcionamiento básico.
4. **Iteración y Refinamiento:** Se realizaron ajustes según los resultados de las pruebas y la retroalimentación recibida de los usuarios. Este proceso incluyó mejoras en la precisión del reconocimiento de placas y la estabilidad de la comunicación entre dispositivos.
5. **Implementación Final:** Una vez refinado, el sistema fue implementado en un entorno real, asegurando su correcto funcionamiento y cumplimiento de los requisitos planteados.

Se llevaron a cabo entrevistas con el jefe del área de seguridad para obtener los requisitos funcionales y no funcionales del prototipo IoT. Estos requisitos están detallados en la Tabla 1.

Tabla 1: Requisitos Funcionales

Requisitos Funcionales	
Código	Descripción
RF-01	El prototipo debe capturar imágenes de las placas vehiculares de forma continua o al detectar movimiento.
RF-02	El prototipo debe procesar las imágenes capturadas y extraer los caracteres mediante técnicas de OCR.
RF-03	El prototipo debe transmitir los datos obtenidos al sistema central en tiempo real.
RF-04	El prototipo debe registrar eventos con información como fecha y hora de autorización.
RF-05	El prototipo debe interactuar con el mecanismo del portón para permitir o denegar el acceso según las instrucciones recibidas.
RF-06	El prototipo debe notificar al personal de seguridad sobre accesos no autorizados o fallos técnicos.

De igual manera, se listan los requisitos no funcionales, según se observa en la Tabla 2.

Tabla 2: Requisitos no funcionales

Requisitos No Funcionales	
Código	Descripción
USABILIDAD	La interfaz de la página web debe ser intuitiva y fácil de usar, permitiendo al personal de seguridad operar el sistema sin capacitación técnica avanzada..
EFICIENCIA	El sistema debe procesar las imágenes de las placas y enviar los resultados al servicio en menos de 25 segundos para garantizar una validación rápida.
COMPATIBILIDAD	La página web debe ser accesible desde navegadores modernos, como Chrome, Firefox y Edge, sin necesidad de complementos adicionales.

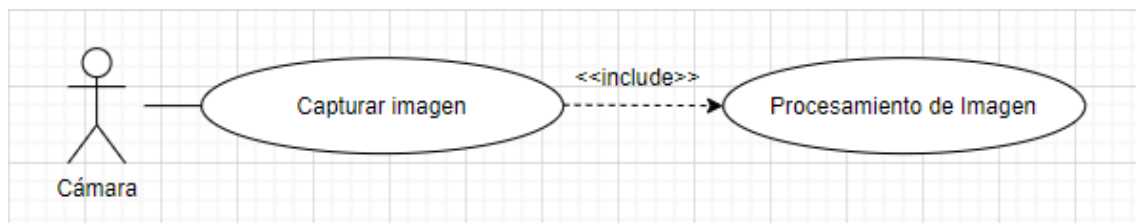
Limitaciones: El prototipo debe capturar imágenes de las placas vehiculares de manera óptima utilizando una cámara fija. Su rendimiento puede verse afectado por condiciones de iluminación extremas, como reflejos intensos o baja luz, y por la velocidad a la que se aproxime el vehículo al portón.

Propósito: El objetivo fue crear un prototipo IoT capaz de capturar imágenes de las placas vehiculares y procesarlas para el reconocimiento de caracteres, enviando los datos a un sistema central para gestionar el acceso vehicular de manera automatizada y eficiente.

Diseño rápido: Se realizó un prototipo preliminar de la arquitectura, definiendo funcionamiento, así mismo haciendo uso de herramientas como una cámara de alta resolución para gestionar la captura y transmisión de datos.

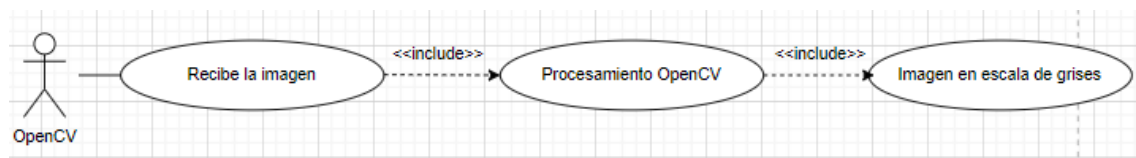
Casos de uso. La Figura 1 muestra cómo el prototipo IoT captura la imagen del vehículo cuando un vehículo se aproxima al portón y procesa con el modelo `ssd_mobilenet_v2`

Figura 1: Primer caso de uso



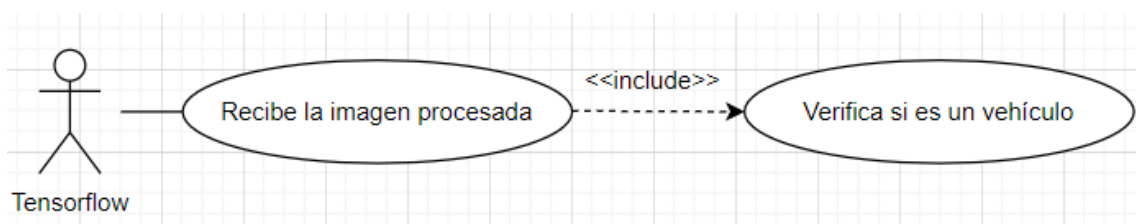
La Figura 2 ilustra el procesamiento local de la imagen

Figura 2: Segundo caso de uso



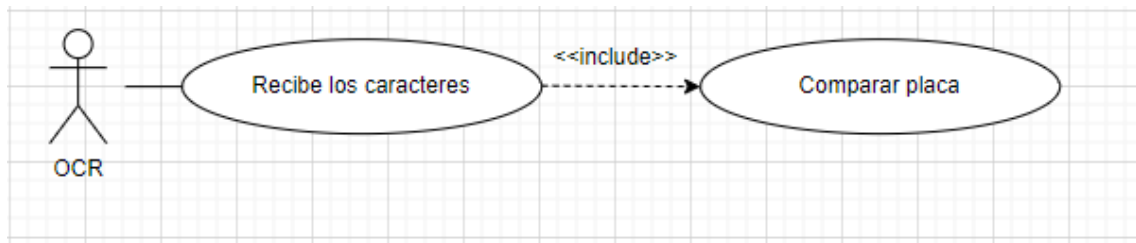
La Figura 3 muestra como Tensorflow reconoce si es un vehículo

Figura 3: Tercer Caso de Uso



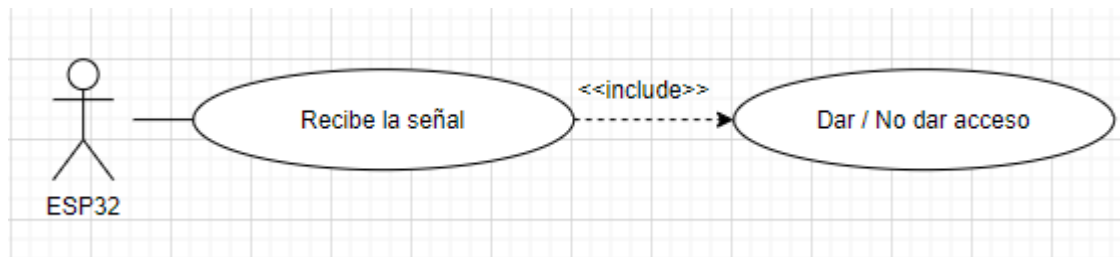
La Figura 4 muestra el uso de EasyOCR en el prototipo IoT para extraer los caracteres de la placa

Figura 4: Cuarto caso de uso



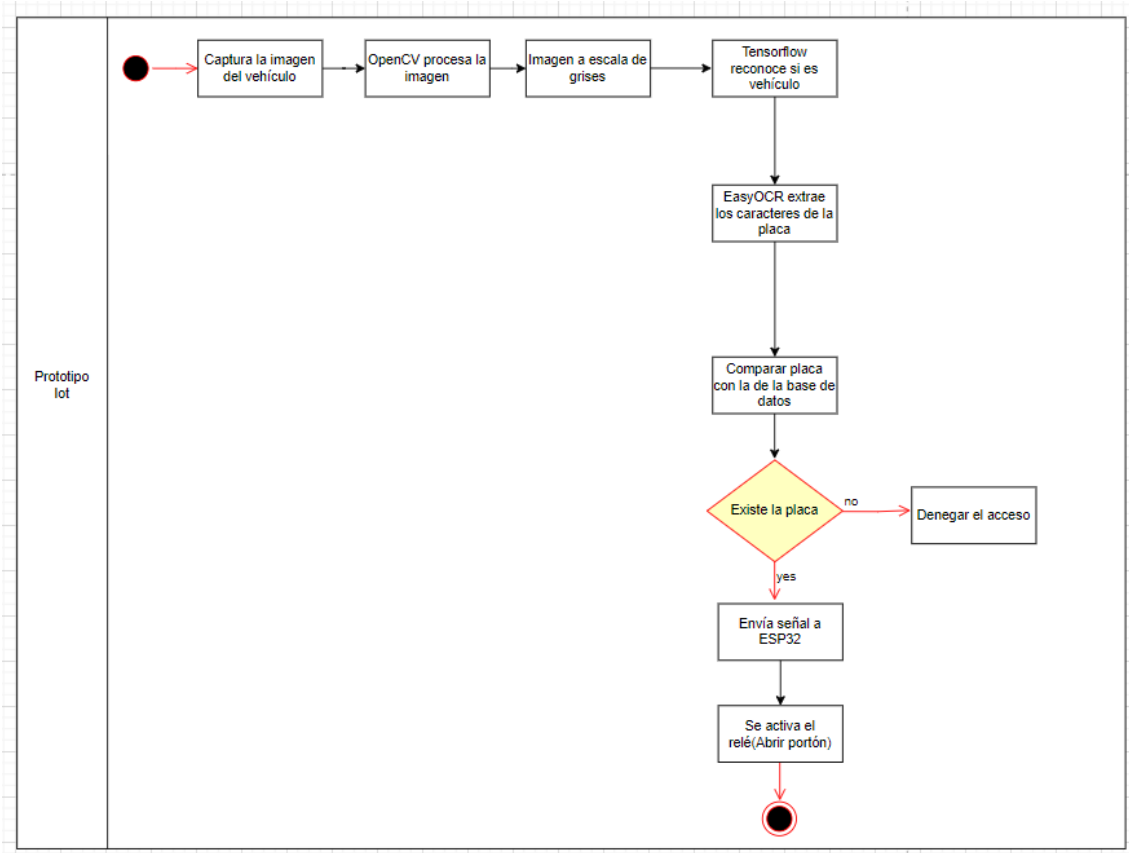
La Figura 5 ilustra la comparación entre la placa detectada y las registradas en la base de datos

Figura 5: Quinto caso de uso



La Figura 6 muestra el diagrama de actividades del prototipo IoT, detallando el flujo de acciones desde la captura de la imagen hasta la autorización del acceso. Incluye las etapas de procesamiento de imagen, reconocimiento de caracteres y la validación de la placa para determinar si se debe abrir el portón.

Figura 6: Diagrama de actividades del prototipo IoT



En la figura 7 se puede apreciar el flujo de la información mediante un diagrama de imágenes y en la figura 8 podemos observar el mismo flujo pero con las herramientas utilizadas:

Figura 7: Arquitectura iot flujo de datos

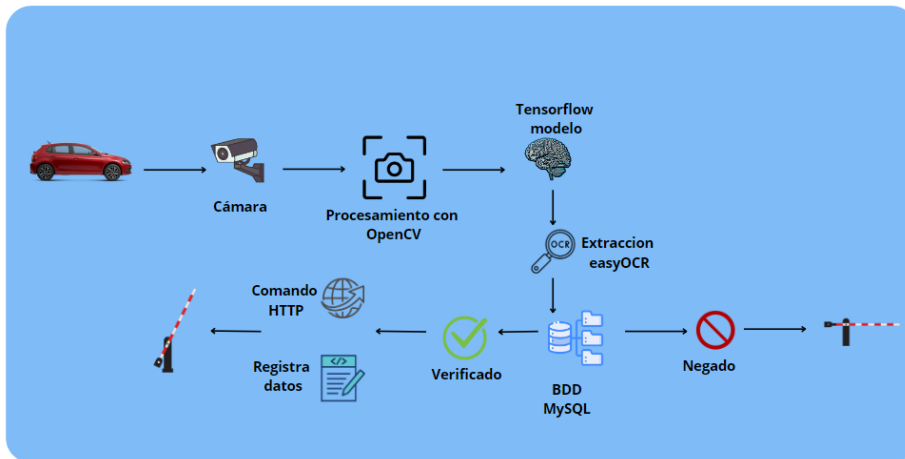


Figura 8: Arquitectura iot flujo herramientas



Fases de la Metodología de Prototipado

Fase 1: Construcción.

Se realizó la integración de los componentes del prototipo, conectando ,la cámara capturaba las imágenes y estas eran enviadas a un modelo preentrenado y a un algoritmo para su procesamiento. Posteriormente, el ESP32 recibía una señal con el resultado del análisis, permitiendo la ejecución de las acciones correspondientes según la información procesada.

Fase 2: Evaluación.

Se llevaron a cabo pruebas funcionales del prototipo en un entorno real con las condiciones que lo demanda, evaluando la capacidad de la cámara para capturar imágenes claras, la precisión del procesamiento de placas con OpenCV y el desempeño del reconocimiento de caracteres con EasyOCR. La validación incluyó la comparación de placas registradas y no registradas.

Fase 3: Refinamiento e iteración.

Basado en los resultados de la evaluación, se ajustaron los parámetros del procesamiento de imágenes, se optimizó el tiempo de respuesta del sistema y se mejoró la precisión del reconocimiento de placas. Estas iteraciones se realizaron hasta garantizar el cumplimiento de los requisitos funcionales establecidos.

2.1.3.2 Desarrollo de la página web (Metodología de Programación Extrema - XP)

Para la creación de la página web destinada a la gestión del sistema de control de acceso vehicular, se aplicó la metodología ágil XP. Este enfoque fomenta un desarrollo iterativo e incremental, permitiendo una colaboración constante con los usuarios y una rápida capacidad de respuesta ante cambios. A través de prácticas como la integración continua y pruebas frecuentes, se buscó garantizar la calidad del software y entregar un producto funcional, confiable y eficiente.

2.1.4 Fase 1 (Planificación)

2.1.4.1 Historias de Usuario

Las historias de usuario son un recurso esencial en el desarrollo de software y productos tecnológicos, particularmente en metodologías ágiles como la Programación Extrema (XP). Estas historias ofrecen una manera sencilla y comprensible de recoger los requisitos y necesidades de los usuarios finales, funcionando como una guía práctica para la planificación y ejecución del proyecto. En la Tabla 3 se detallan las historias de usuario identificadas para la página web, clasificadas según las principales funcionalidades del sistema.

Tabla 3: Historias de usuario

Página Web						
ID	NOMBRE	ESTIMACIÓN ESFUERZO (DÍAS)	IMPORTANCIA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DEPENDENCIAS
1	Login al sistema	15	95/100	Permitir que los usuarios inicien sesión en el sistema mediante credenciales válidas.	<ul style="list-style-type: none">• Los usuarios pueden iniciar sesión con credenciales válidas.• Se muestra un mensaje de error en caso de datos incorrectos.	Ninguna
2	Gestión de Roles	12	90/100	Permitir que el sistema asigne diferentes roles a los usuarios, donde el	<ul style="list-style-type: none">• El superadministrador puede acceder a todas las funciones del sistema, incluyendo la creación, edición y eliminación de usuarios.	1

				<p>superadministrador tenga privilegios completos, incluyendo la creación de usuarios, y el administrador solo pueda visualizar el monitoreo de vehículos.</p>	<ul style="list-style-type: none"> ● El administrador solo puede visualizar el monitoreo de vehículos. ● Los permisos asignados a cada rol se aplican correctamente y restringen funciones no autorizadas. 	
3	Registro de vehículos	10	90/100	Registrar un vehículo con datos como placa, marca y nombre del estudiante, almacenándolos en la BDD.	<ul style="list-style-type: none"> ● Todos los campos obligatorios son validados antes de guardar. ● El sistema notifica si la placa ya está registrada. ● Los datos se guardan correctamente en la BDD. 	1
4	Almacenamiento de datos en la BDD	20	100/100	Garantizar que los datos ingresados y generados se almacenen correctamente en la base de datos.	<ul style="list-style-type: none"> ● La conexión a la base de datos es estable y confiable. ● Los datos ingresados, como vehículos, ingresos y salidas, se almacenan sin pérdida ni duplicación. 	1, 3
5	Monitoreo de ingreso/salida	25	100/100	Registrar el ingreso y salida de vehículos con fecha y hora, actualizando automáticamente en la BDD.	<ul style="list-style-type: none"> ● El ingreso se registra con fecha y hora al acceder. ● La salida se actualiza al momento de salir. ● Los datos son actualizados en tiempo real. 	1, 3, 4

6	Generación de reportes	10	85/100	Permitir a los usuarios generar reportes filtrando por usuario	<ul style="list-style-type: none"> ● Los reportes muestran resultados claros y organizados. ● El sistema valida fechas correctas para evitar errores. ● Los filtros funcionan adecuadamente. 	1, 5
7	Exportación de reportes a Excel	5	80/100	Exportar los reportes generados a un archivo Excel descargable.	<ul style="list-style-type: none"> ● Los datos exportados reflejan los filtros seleccionados. ● El archivo Excel incluye toda la información necesaria y es descargado sin problemas. 	1, 6

2.1.4.2 Requisitos funcionales

A continuación, se detallan los requisitos funcionales del sistema basado en el análisis de las historias de usuario.

RF-01: Login al sistema

- **RF-01.1:** El sistema debe permitir que los usuarios inicien sesión con credenciales válidas.
 - **Criterios de aceptación:**
 - El usuario tiene la posibilidad de introducir su nombre de usuario y contraseña en los campos designados.
 - El sistema verifica la validez de las credenciales proporcionadas.
 - En caso de que las credenciales sean válidas, se permite el acceso del usuario a la página principal.
 - Si las credenciales son incorrectas, el sistema genera un mensaje de error para informar al usuario.

RF-02: Gestión de Roles

- **RF-02.1:** El sistema debe permitir la asignación de roles diferenciados para usuarios superadministradores y administradores.
 - **Criterios de aceptación:**
 - El superadministrador puede acceder a todas las funciones del sistema, incluyendo la creación, edición y eliminación de usuarios.
 - El administrador solo puede visualizar el monitoreo de vehículos.
 - Los permisos asignados a cada rol se aplican correctamente y restringen el acceso a las funcionalidades no autorizadas.

RF-03: Registro de vehículos

- **RF-03.1:** El sistema debe permitir registrar un vehículo con su placa, marca y nombre del estudiante.
 - **Criterios de aceptación:**
 - El usuario puede ingresar los datos requeridos (placa, marca y nombre del estudiante).
 - El sistema valida que todos los campos obligatorios están completos.
 - Si la placa ya está registrada, el sistema muestra un mensaje informativo.
 - Los datos del vehículo se guardan correctamente en la base de datos.

RF-04: Almacenamiento de datos en la base de datos

- **RF-04.1:** El sistema debe guardar los datos de registro y monitoreo de los vehículos en una base de datos de manera segura.
 - **Criterios de aceptación:**
 - El sistema debe establecer una conexión confiable y segura con la base de datos.
 - Los datos de los vehículos, incluyendo la placa y la información de ingreso y salida, se almacenan correctamente en la base de datos.
 - El sistema debe evitar la duplicación de registros.

RF-05: Monitoreo de ingreso y salida de vehículos

- **RF-05.1:** El sistema debe registrar la entrada y salida de los vehículos, incluyendo la fecha y hora exactas.
 - **Criterios de aceptación:**
 - El sistema registra la fecha y hora exacta del ingreso y salida de un vehículo.
 - El registro se actualiza correctamente cada vez que un vehículo entra o sale del campus.
 - La información de ingreso y salida se almacena correctamente en la base de datos.

RF-06: Generación de reportes

- **RF-06.1:** El sistema debe permitir generar reportes filtrados por estudiante y por rango de fechas.
 - **Criterios de aceptación:**
 - El usuario puede seleccionar un estudiante y un rango de fechas (desde/hasta).
 - El sistema genera un reporte con los registros de ingreso y salida del estudiante dentro del rango de fechas seleccionado.
 - Los reportes son claros, organizados y fáciles de leer.

RF-07: Exportación de reportes a Excel

- **RF-07.1:** El sistema debe permitir exportar los reportes generados a un archivo Excel descargable.
 - **Criterios de aceptación:**
 - El usuario puede descargar el reporte como un archivo Excel.
 - El archivo Excel incluye los datos filtrados correctamente.
 - El archivo es descargado sin errores y contiene toda la información necesaria.

2.1.4.3 Requisitos No Funcionales

RNF-01: Usabilidad

- **RNF-01.1:**

La interfaz de la página web debe ser sencilla e intuitiva, permitiendo a los usuarios realizar acciones como registrar vehículos y generar reportes en un tiempo menor a 5 minutos, sin necesidad de ayuda técnica.

RNF-02: Seguridad

- **RNF-02.1:**

El sistema debe gestionar las sesiones de los usuarios de manera segura, garantizando que cada sesión sea autenticada correctamente y que las contraseñas almacenadas en la base de datos estén encriptadas de forma adecuada.

RNF-03: Eficiencia

- **RNF-03.1:**

El sistema debe procesar y registrar los datos de ingreso y salida de vehículos en un tiempo inferior a 10 segundos, asegurando que la información se actualice en tiempo real para los usuarios.

2.1.4.4 Módulos

Módulo de Autenticación

Roles:

- Desarrollador

Actividades:

- Implementar la funcionalidad de login para autenticar a los usuarios.
- Desarrollar la interfaz de inicio de sesión donde los usuarios ingresen sus credenciales (nombre de usuario y contraseña).
- Validar las credenciales ingresadas por el usuario.
- Gestionar sesiones de usuario una vez autenticado.

Módulo de Registro de Vehículos

Roles:

- Desarrollador

Actividades:

- Crear la interfaz de registro donde el personal de seguridad ingrese los datos del vehículo (placa, marca, nombre del estudiante).
- Validar que todos los campos obligatorios estén completos antes de guardar la información.
- Implementar la verificación de unicidad de la placa para asegurarse de que no se registren placas duplicadas.
- Almacenar la información del vehículo en la base de datos de manera segura y confiable.
- Desarrollar un sistema de notificaciones en caso de que la placa ya esté registrada.

Módulo de Gestión de Roles

Roles:

- Desarrollador

Actividades:

- Crear la interfaz para que el superadministrador asigne roles a los usuarios (superadministrador o administrador).
- Implementar la funcionalidad para que el superadministrador pueda crear, editar y eliminar usuarios.
- Configurar restricciones en el sistema para que el administrador solo pueda visualizar el monitoreo de vehículos.
- Validar que los permisos asignados a cada rol se apliquen correctamente y que las funcionalidades no autorizadas estén restringidas.
- Realizar pruebas para garantizar que los roles se gestionen y apliquen de manera correcta en todo el sistema.

Módulo de Monitoreo de Ingreso/Salida

Roles:

- Desarrollador

Actividades:

- Implementar la funcionalidad de monitoreo para registrar la fecha y hora de ingreso de los vehículos al campus.
- Desarrollar la lógica de actualización para registrar la salida de los vehículos, con la fecha y hora correspondiente.
- Asegurar que los registros de ingreso y salida se actualicen en tiempo real en la base de datos.
- Validar que no haya duplicados en los registros de ingreso/salida para un mismo vehículo.
- Implementar pruebas de precisión para garantizar que las fechas y horas sean correctas al registrar el ingreso y la salida.

Módulo de Reportes

Roles:

- Desarrollador

Actividades:

- Desarrollar la interfaz de generación de reportes, permitiendo a el personal de seguridad filtrar los registros por estudiante y rango de fechas (desde/hasta).
- Implementar la lógica de generación de reportes, donde el personal de seguridad pueda ver los registros filtrados.
- Asegurar que el sistema valide las fechas correctas para los filtros (desde/hasta).

Módulo de Exportación de Reportes a Excel

Roles:

- Desarrollador

Actividades:

- Desarrollar la funcionalidad de exportación de reportes a un archivo Excel descargable.
- Asegurar que los datos filtrados por el personal de seguridad se reflejen correctamente en el archivo Excel.
- Garantizar que el archivo Excel sea descargado correctamente y que la información se visualice correctamente en un formato accesible y organizado.

2.1.4.5 Roles y responsabilidades

Tabla 4: Roles y responsabilidades

Nombre	Rol	Responsabilidad
Carlos Lima	Jefe de Seguridad	<p>Proporcionar los requisitos y necesidades de monitoreo de vehículos.</p> <p>Supervisar la correcta implementación del sistema de control de acceso.</p> <p>Cumplir con los estándares de seguridad</p> <p>Aprobar la implementación final del sistema.</p>
Dany Mejía	Desarrollador	<p>Desarrollar el prototipo IoT</p> <p>Desarrollar el sistema de control de acceso, incluyendo el registro de vehículos y monitoreo de ingreso/salida.</p> <p>Implementar la funcionalidad de generación de reportes y exportación a Excel.</p> <p>Realizar pruebas de funcionalidad y solucionar problemas.</p> <p>Documentar el proyecto y sus avances.</p>

2.1.4.6 Planificación de Actividades

Las historias de usuario previamente definidas fueron desglosadas en actividades específicas, como se muestra en la tabla 5, las cuales representan las tareas necesarias para el desarrollo de la página web de control de acceso vehicular. Cada historia de usuario se dividió en una serie de actividades detalladas, asignándoles sus respectivas duraciones y fechas planificadas de inicio y finalización.

Tabla 5: Planificación de actividades

Historia de Usuario	Duración (Días)	Actividades	Duración de las Actividades (Días)	Inicio	Fin
1	15	Desarrollar el formulario de login para ingresar las credenciales (usuario y contraseña).	3	2/10/2024	5/10/2024
		Implementar la validación de las credenciales ingresadas.	2	06/10/2024	07/10/2024
		Configurar la gestión de sesiones para usuarios autenticados	10	08/10/2024	08/10/2024
2	5	Desarrollar la interfaz para que el superadministrador asigne roles a los usuarios.	1	18/10/2024	18/10/2024
		Implementar la lógica para la creación, edición y eliminación de usuarios por parte del superadministrador.	2	19/10/2024	19/10/2024
		Configurar las restricciones para que el administrador solo pueda visualizar el monitoreo de vehículos.	2	21/10/2024	21/10/2024
3	10	Crear la interfaz de registro de vehículos (placa, marca y nombre del estudiante).	2	23/10/2024	23/10/2024
		Implementar la validación de campos obligatorios antes de guardar.	1	25/10/2024	25/10/2024
		Desarrollar la lógica para verificar que la placa no esté duplicada	6	26/10/2024	26/10/2024

		Almacenar la información del vehículo en la base de datos.	1	01/11/2024	01/11/2024
4	20	Configurar la base de datos para almacenar la información de vehículos, ingresos y salidas.	5	02/11/2024	06/11/2024
		Implementar la lógica para almacenar los registros de ingreso y salida en tiempo real.	15	07/11/2024	21/11/2024
5	25	Desarrollar la funcionalidad para registrar el ingreso y salida de vehículos.	10	27/11/2024	01/12/2024
		Implementar la actualización en tiempo real de los registros de entrada y salida.	10	02/12/2024	11/12/2024
		Realizar pruebas de actualización en tiempo real.	5	12/12/2024	16/12/2024
6	10	Desarrollar la interfaz para filtrar reportes por estudiante y fechas (desde/hasta).	2	17/12/2024	18/12/2024
		Implementar la lógica para mostrar los reportes de entrada/salida de vehículos.	5	19/12/2024	23/12/2024
		Probar la funcionalidad de generación de reportes y validación de fechas.	3	24/12/2024	26/12/2024
7	5	Desarrollar la funcionalidad para exportar reportes a Excel.	3	27/12/2024	29/12/2024
		Probar la generación del archivo Excel y la descarga correcta.	2	30/12/2024	31/12/2024

2.1.4.7 Iteraciones

Se detallan las actividades necesarias para desarrollar la página web de control de acceso vehicular. La fase de iteración se ejecutará mediante un conjunto de iteraciones sucesivas.

2.1.4.7.1 Iteración 1: Login al sistema (15 días)

Desarrollar la funcionalidad de login al sistema para permitir a los usuarios ingresar con sus credenciales.

Crear la interfaz de inicio de sesión para que los usuarios ingresen su nombre de usuario y contraseña.

Implementar la validación de credenciales para asegurar que los usuarios solo puedan acceder con información correcta.

Realizar pruebas de autenticación de usuarios y manejo de errores en caso de datos incorrectos.

2.1.4.7.2 Iteración 2: Gestión de Roles (5 días)

Desarrollar la interfaz para que el superadministrador asigne roles a los usuarios.

Implementar la creación, edición y eliminación de usuarios por parte del superadministrador.

Configurar las restricciones para que el administrador solo pueda visualizar el monitoreo de vehículos.

2.1.4.7.3 Iteración 3: Registro de vehículos (10 días)

Crear la interfaz de registro de vehículos para ingresar la placa, marca y nombre del estudiante.

Validar los campos obligatorios antes de guardar los datos.

Implementar la verificación de duplicados de la placa para evitar registros duplicados.

Almacenar la información del vehículo en la base de datos.

Realizar pruebas de registro de vehículos y validación de datos.

2.1.4.7.4 Iteración 4: Almacenamiento de datos en la BDD (20 días)

Configurar la base de datos para almacenar la información de vehículos, ingresos y salidas.

Implementar la lógica para almacenar registros de ingreso y salida en la base de datos en tiempo real.

Realizar pruebas de confiabilidad y seguridad en el almacenamiento de datos.

2.1.4.7.5 Iteración 5: Monitoreo de ingreso/salida (25 días)

Desarrollar la funcionalidad de monitoreo de ingreso y salida de vehículos, registrando la fecha y hora.

Implementar la actualización en tiempo real de los registros de ingreso y salida en la base de datos.

Realizar pruebas de precisión de registros y validación de la actualización en tiempo real.

2.1.4.7.6 Iteración 6: Generación de reportes (10 días)

Desarrollar la interfaz de generación de reportes para que los usuarios puedan filtrar por estudiante y fechas.

Implementar la lógica para generar reportes basados en los filtros seleccionados.

Validar la exactitud de los datos en los reportes generados.

Realizar pruebas de funcionalidad de filtros y generación de reportes.

2.1.4.7.7 Iteración 7: Exportación de reportes a Excel (5 días)

Desarrollar la funcionalidad de exportación de los reportes generados a Excel.

Asegurar que los datos exportados reflejen los filtros seleccionados y estén organizados correctamente en el archivo Excel.

Realizar pruebas de descarga de archivos y validación de la información contenida en el archivo Excel.

2.1.5 Fase 2 (Diseño)

2.1.5.1 Arquitectura de microservicios

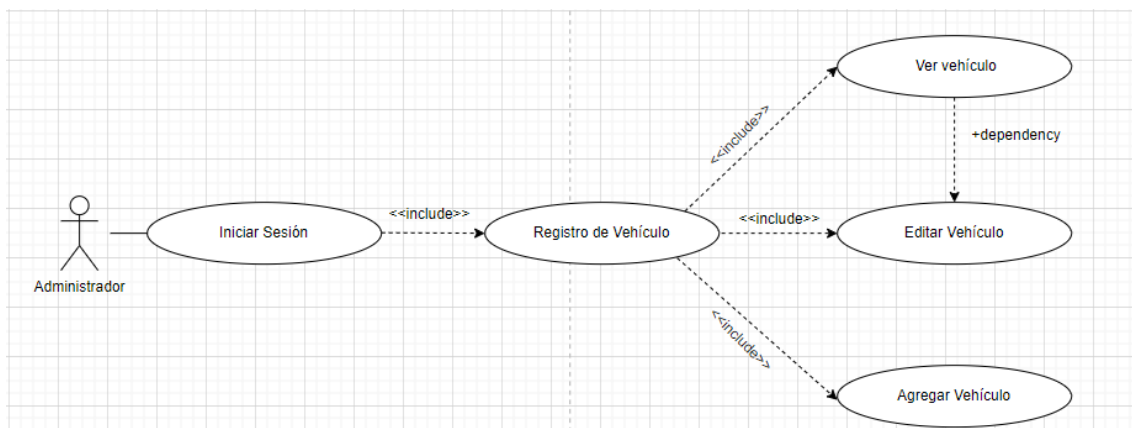
La arquitectura de microservicios fue seleccionada para este proyecto debido a su capacidad para ofrecer escalabilidad y modularidad. Al fragmentar el sistema en servicios independientes, cada uno dedicado a una función específica, se simplifica tanto la gestión como el mantenimiento general del sistema. En el caso de la página web diseñada para el control de acceso vehicular, esta arquitectura permite que cada módulo sea desarrollado, probado e implementado de forma individual, lo que acelera el proceso de desarrollo y garantiza que las actualizaciones o mejoras en un servicio no afecten el funcionamiento del resto del sistema.

2.1.5.2 Funcionalidades del Sistema

2.1.5.2.1 Casos de Uso

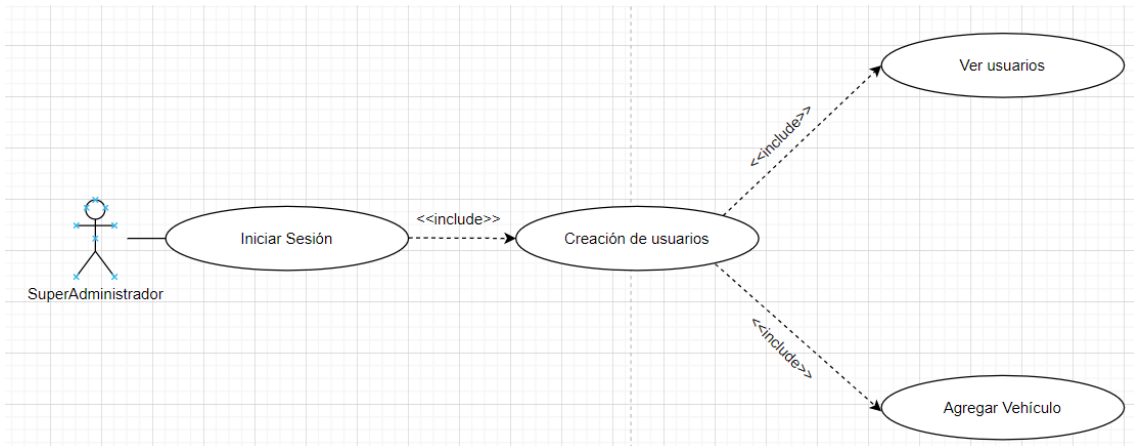
En la Figura 9 se presenta el caso de uso para el ingreso de los datos del vehículo y del propietario:

Figura 9: Primer caso de uso (sistema)



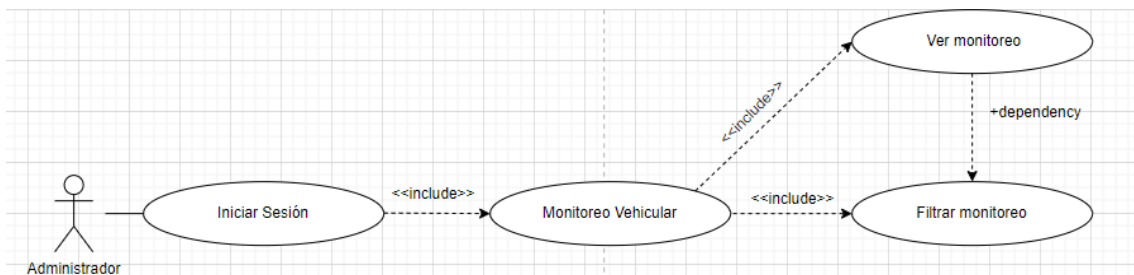
En la Figura 10 se presenta el caso de uso para la creación de administradores mediante el rol de superadministrador:

Figura 10: Segundo caso de uso (sistema)



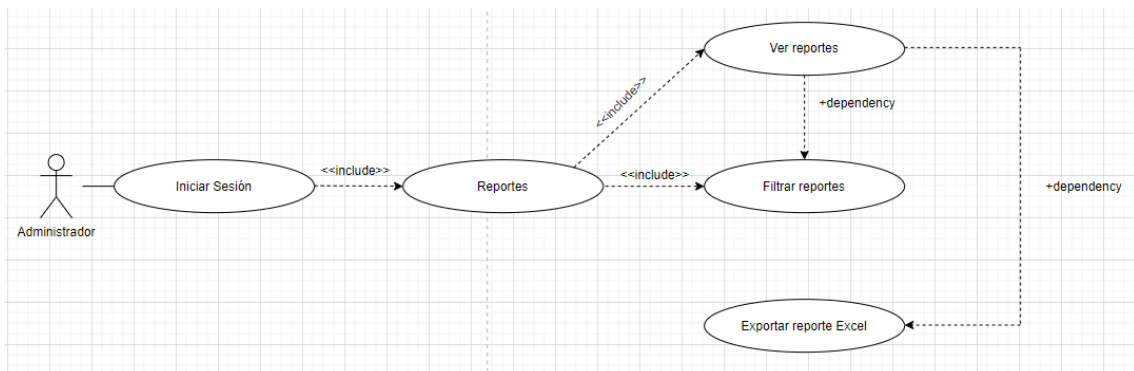
En la Figura 11 se presenta el caso de uso para la visualización del monitoreo vehicular:

Figura 11: : Tercer caso de uso (sistema)



En la Figura 12 se presenta el caso de uso para visualizar reportes dependiendo los filtros y la exportación en formato .xlsx:

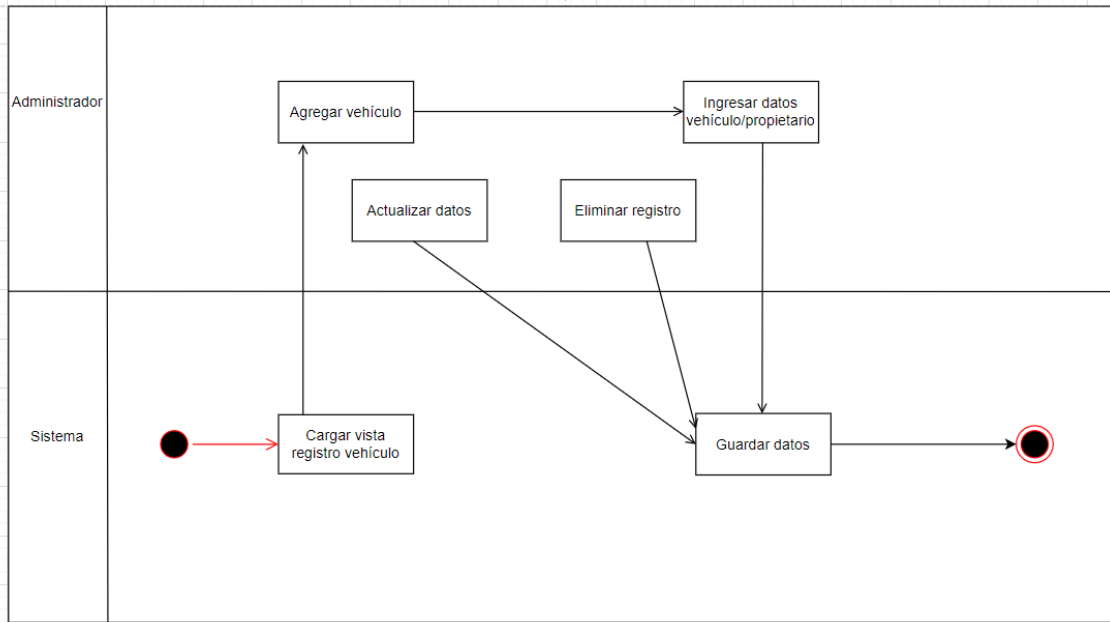
Figura 12: : Cuarto caso de uso (sistema)



2.1.5.2.2 Diagrama de Actividades

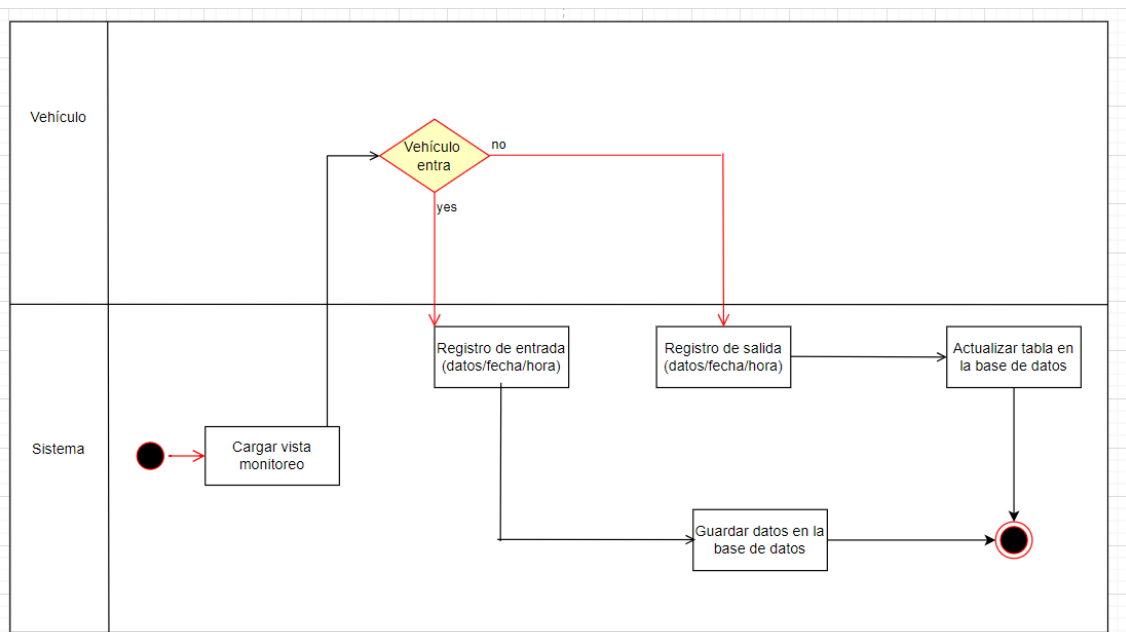
A continuación, se presentan los diagramas que describen las decisiones y acciones que realizará el usuario al interactuar con el sistema. La Figura 13 representa el flujo de actividades para el registro de vehículos en el sistema.

Figura 13: Diagrama de actividades (registro vehículo)



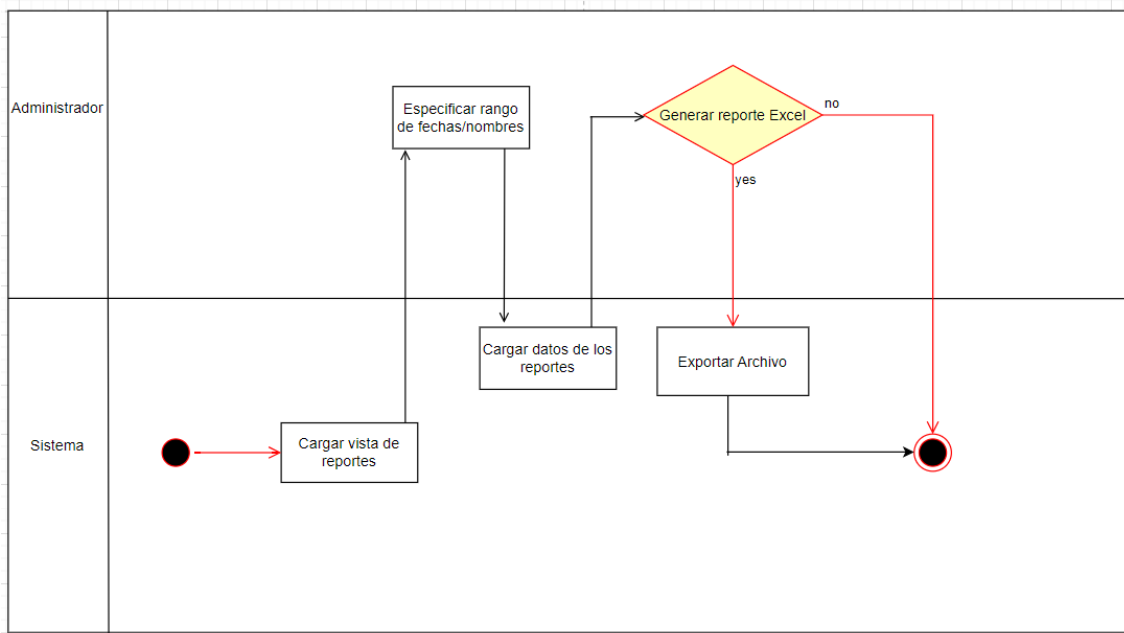
La Figura 14 representa el flujo de actividades para el monitoreo de ingreso y salida de vehículos.

Figura 14: Diagrama de actividades (ingreso salida de vehículos)



La figura 15 representa el flujo de actividades para la generación y exportación de reportes.

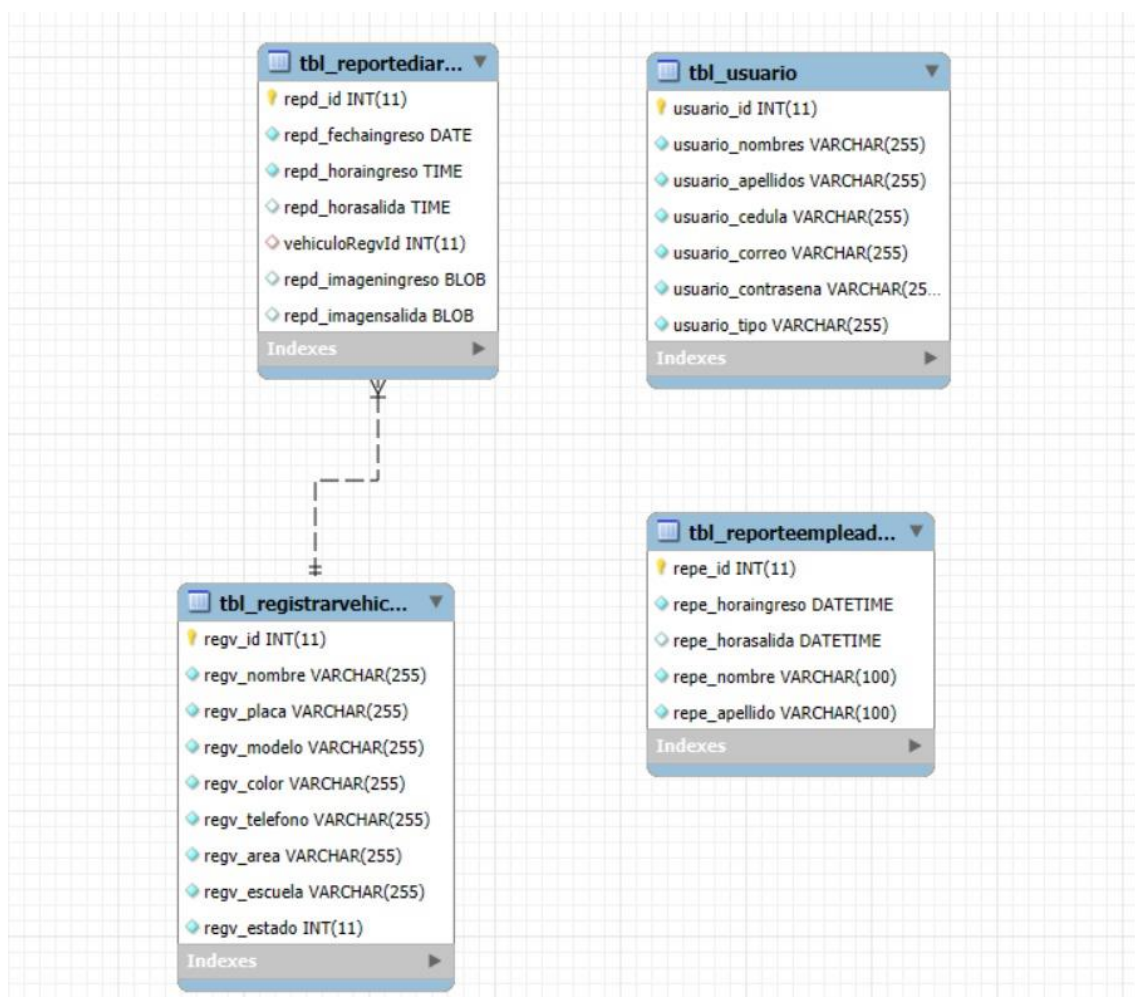
Figura 15: Diagrama de actividades (Exportación de reportes)



2.1.5.2.3 Base de Datos

MySQL fue la base de datos relacional seleccionada para este proyecto debido a su robustez, eficiencia y amplia adopción en el desarrollo de sistemas web. MySQL permitió almacenar y gestionar datos estructurados de manera eficiente, lo cual se ajustó perfectamente a las necesidades del sistema, como el registro de vehículos, monitoreo de ingresos y salidas, y generación de reportes. Su estructura basada en tablas relacionales garantizó la integridad y consistencia de los datos, permitiendo relaciones claras entre las entidades, como vehículos, estudiantes e ingresos o salidas. Además, MySQL ofreció compatibilidad con múltiples herramientas de desarrollo y una excelente integración con lenguajes de backend, lo que facilitó la implementación y escalabilidad del sistema, como se observa en la Figura 16.

Figura 16: Diagrama de la base de datos

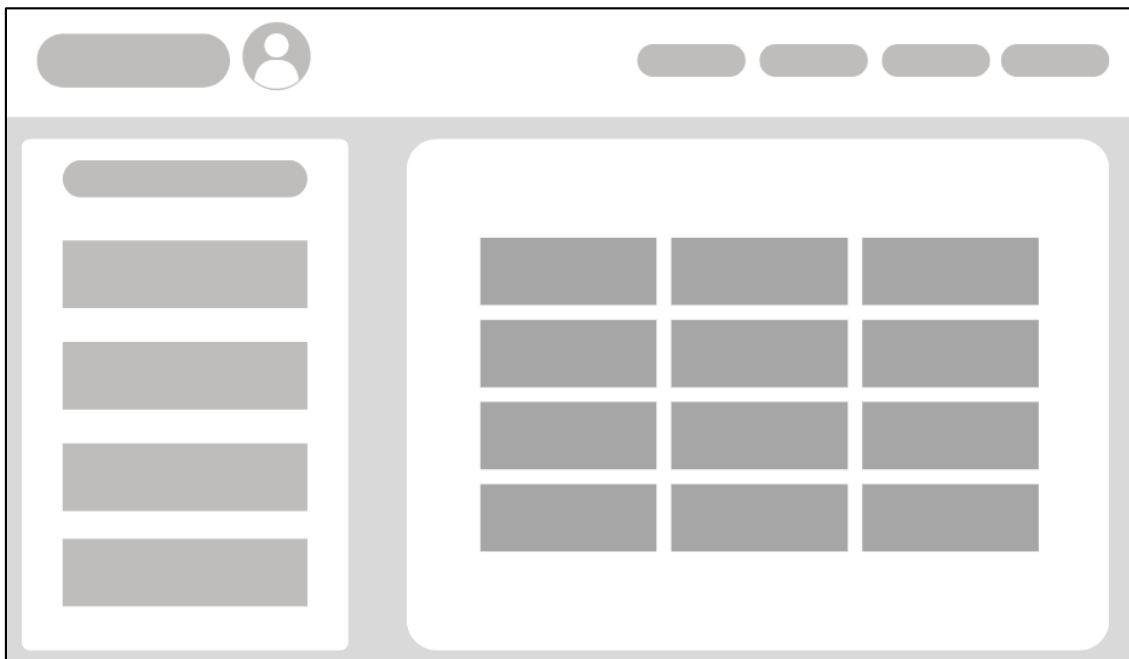


2.1.5.2.4 Diseño de la Interfaz

Esta interfaz cuenta con 6 pantallas principales: Inicio, Iniciar Sesión, Registro de Usuario, Registro de Vehículo, Reporte Diario y Reporte Mensual. Cada una de estas pantallas está diseñada para facilitar la interacción del usuario con el sistema, brindando acceso rápido y organizado a las funcionalidades correspondientes.

En la Figura 17, se presenta el prototipo de la interfaz general, que muestra cómo se estructuran las funciones señaladas anteriormente.

Figura 17: Interfaz preliminar



2.1.6 Fase 3 (Codificación)

Para el desarrollo del backend, se eligió TypeScript como lenguaje de programación debido a sus capacidades avanzadas, como el tipado estático y otras funcionalidades modernas que amplían las características de JavaScript. Además, se empleó NestJS, un framework progresivo basado en Node.js, que facilita la implementación de patrones de diseño y arquitecturas altamente escalables. Esta combinación permitió construir una API robusta y mantenible, adecuada para gestionar el registro de vehículos, el monitoreo de ingresos y salidas, y la generación de reportes de manera eficiente.

En cuanto al procesamiento de imágenes, se integraron herramientas especializadas para garantizar la precisión en el reconocimiento de placas vehiculares. Se utilizó OpenCV, una biblioteca de código abierto para el procesamiento de imágenes, que permitió recortar y detectar las placas en tiempo real. Posteriormente, se empleó EasyOCR, una herramienta de reconocimiento óptico de caracteres (OCR), para extraer los textos de las placas y enviarlos al backend para su validación.

En el caso del frontend, se seleccionó el framework Vue.js, una herramienta progresiva para construir interfaces de usuario interactivas y responsivas. Vue.js facilitó el desarrollo de una interfaz intuitiva y amigable, permitiendo a los administradores y estudiantes interactuar de manera eficiente con el sistema. Su capacidad para integrarse con otras bibliotecas y herramientas complementarias garantizó una experiencia de usuario fluida y un desarrollo ágil del proyecto.

Esta arquitectura, basada en TypeScript y NestJS para el backend, OpenCV y EasyOCR para el procesamiento de placas, y Vue.js para el frontend, aseguró una solución modular, escalable y fácil de mantener, cumpliendo con los requisitos funcionales y no funcionales establecidos para el sistema de control de acceso vehicular.

2.1.7 Fase 4 (Pruebas)

Con el objetivo de garantizar el correcto funcionamiento del sistema, se elaboró un plan de pruebas que abarcó tanto el sistema como el prototipo IoT, tal como se muestra en la tabla 6 y la tabla 7.

Tabla 6: Pruebas (Sistema)

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado
1	Iniciar Sesión	Verificar que el sistema permita iniciar sesión con credenciales válidas	Ingreso de credenciales correctas (Usuario: admin, Contraseña: 1234)	El sistema permite acceso a la aplicación web.
			Ingreso de credenciales incorrectas (Usuario: admin, Contraseña: 12345)	El sistema muestra un mensaje de error indicando que las credenciales no son válidas
			Campos de credenciales vacíos	El sistema muestra un mensaje de error indicando que los campos no pueden estar vacíos.
2	Registro de un Nuevo Vehículo	Verificar que se pueda registrar un vehículo con datos correctos y que el sistema no permita duplicados.	Placa no registrada (Placa: ABC123, Marca: Toyota, Estudiante: Juan)	El vehículo se registra correctamente en la base de datos.
			Placa duplicada (Placa: ABC123, Marca: Nissan, Estudiante: Ana)	El sistema muestra un mensaje de error indicando que la placa ya está registrada.
			Campos de datos vacíos	El sistema muestra un mensaje de error indicando que los campos no pueden estar vacíos.
3	Visualización del Monitoreo de Ingreso/Salida	Verificar que el sistema registre la fecha y hora de ingreso y salida de los vehículos y que la base de datos se actualice en tiempo real.	Ingreso de vehículo registrado (Placa: ABC123)	El sistema registra correctamente la hora de ingreso en la base de datos.
			Salida de vehículo registrado (Placa: ABC123)	El sistema registra correctamente la hora de salida en la base de datos haciendo una actualización.
			Ingreso de vehículo no registrado	El sistema muestra un mensaje de error indicando que el vehículo no está registrado.
4	Generación de Reportes	Verificar que el sistema genere reportes correctamente al aplicar filtros de estudiante y/o rango de fechas.	Filtros correctos (Estudiante: Juan, Fechas: 01/01/2024 - 31/01/2024)	El sistema genera el reporte con los registros correspondientes a los filtros.
			Filtros incorrectos (Estudiante: Pedro, Fechas: 01/01/2024 - 31/01/2024)	El sistema muestra un mensaje de error indicando que el estudiante no está registrado.

Tabla 7: Pruebas (Iot)

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado
1	Procesamiento de Imágenes	Verificar que OpenCV detecte el objeto que entra en rango de la cámara	El objeto entra al rango de la cámara	El objeto es percibido por la cámara
			El objeto no está dentro del rango de la cámara	Si no haya un objeto la cámara sigue registrando
2	Reconocimiento de vehículo	Mediante el uso de Tensorflow verificar si es un vehículo	El objeto es detectado como un vehículo	El modelo entrenado enmarca el objeto como un vehículo
			El objeto no es detectado como un vehículo	Sigue en espera hasta que detecta que es un vehículo
2	Reconocimiento de Caracteres	Verificar que EasyOCR extraiga correctamente	Imagen con placa registrada (Placa: ABC123)	La placa es extraída y se muestra en la pantalla
3	Validación de Placas	Verificar que el sistema compare correctamente las placas extraídas con las registradas en la base de datos.	Placa extraída (Placa: ABC123)	Anuncia la hora en la que el vehículo entra al parqueadero

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

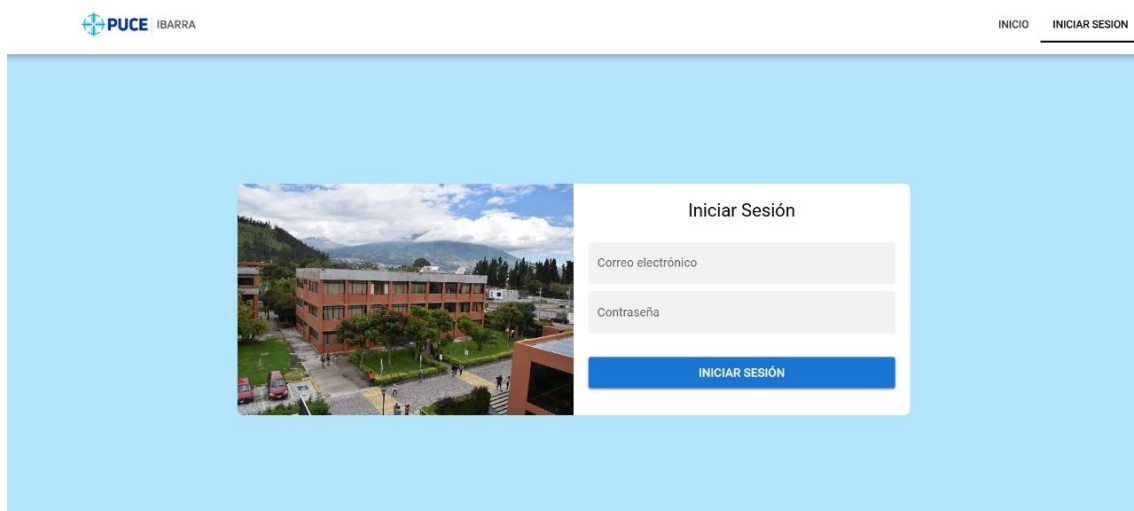
En este capítulo se presentan los resultados obtenidos durante el desarrollo del sistema de control de acceso vehicular, implementado mediante un prototipo IoT y una página web. Se describen los hallazgos derivados de las pruebas realizadas en condiciones reales, así como la validación del cumplimiento de los requisitos definidos en la etapa inicial del proyecto. Además, se analizan los retos y limitaciones enfrentados a lo largo del desarrollo, junto con las soluciones implementadas y las mejoras realizadas para optimizar el funcionamiento del sistema

3. Resultados Fase 1. Sistema Web

3.1.1 Vista Login

En la figura 18 pantalla inicial para que los usuarios ingresen sus credenciales (usuario y contraseña) y accedan al sistema.

Figura 18: Vista Login



3.1.2 Vista Registro Personal

En la figura 19 podemos observar la interfaz para el ingreso de datos personales de los usuarios, asegurando una correcta identificación en el sistema.

Figura 19: Vista Registro Personal

PUCE IBARRA admin

REGISTRAR VEHÍCULO REGISTRAR PERSONAL SEGUIMIENTO USUARIOS REPORTE DIARIO REPORTE MENSUAL Cerrar Sesión

Crear Usuario

Nombres

Apellidos

Número de Cédula

Correo Electrónico

Tipo de Usuario administrador

Contraseña

Confirmar Contraseña

CREAR CUENTA

3.1.3 Vista Registrar Vehículo

En la figura 20 se muestra la pantalla para registrar los vehículos, donde se ingresa información relevante como la placa, marca y nombre del estudiante

Figura 20: Vista Registrar Vehículo

PUCE IBARRA Dany

REGISTRAR VEHÍCULO REPORTE DIARIO REPORTE MENSUAL Cerrar Sesión

Registrar Vehículo

Nombres Completos

Placa Vehicular

Modelo del Vehículo

Color del Vehículo

Teléfono

Área de Pertenencia

Escuela a la que pertenece

AGREGAR VEHÍCULO

Vehículos Registrados

Nombres	Placa Vehicular	Modelo	Color	Teléfono	Área	Escuela	Acciones
Dany Rene Mejia Sandoval	PDL4175	Chevrolet Corsa	Blanco	0996240609	Estudiante	Ingeniería	
Mikaela Cartagena Paez	PD18165	Scoda	Blanco	0963490102	Estudiante	Ciencias de la Salud	

Records per page: 5 1-2 of 2

3.1.4 Vista Seguimiento de Usuarios

En la figura 21 se muestra la interfaz para monitorear las acciones y datos de los usuarios dentro del sistema, permitiendo un control eficiente.



















Figura 21: Vista Seguimiento de Usuarios

Nombre	Apellido	Hora Ingreso	Hora Salida
admin	admin	18 de enero de 2025, 12:34 p. m.	Fecha no disponible
Dany	Mejia	18 de enero de 2025, 12:32 p. m.	18 de enero de 2025, 12:34 p. m.
admin	admin	18 de enero de 2025, 12:30 p. m.	18 de enero de 2025, 12:32 p. m.
Dany	Mejia	16 de enero de 2025, 12:33 a. m.	18 de enero de 2025, 12:29 p. m.
admin	admin	16 de enero de 2025, 12:30 a. m.	16 de enero de 2025, 12:32 a. m.
admin	admin	15 de enero de 2025, 12:22 p. m.	Fecha no disponible
admin	admin	14 de enero de 2025, 08:55 p. m.	15 de enero de 2025, 12:22 p. m.
admin	admin	14 de enero de 2025, 08:54 p. m.	14 de enero de 2025, 08:35 p. m.
admin	admin	9 de enero de 2025, 03:31 p. m.	Fecha no disponible
Dany	Mejia	9 de enero de 2025, 03:17 p. m.	9 de enero de 2025, 03:30 p. m.
admin	admin	9 de enero de 2025, 03:14 p. m.	9 de enero de 2025, 03:17 p. m.
Alexandra	Sandoval	9 de enero de 2025, 03:03 p. m.	9 de enero de 2025, 03:14 p. m.
admin	admin	9 de enero de 2025, 03:00 p. m.	9 de enero de 2025, 03:02 p. m.

3.1.5 Vista Reporte Diario

En la figura 22 tenemos la pantalla donde se visualizan los reportes diarios de actividad, facilitando el acceso a información puntual.

Figura 22: Vista Reporte Diario

ID Reporte	Fecha Ingreso	Hora Ingreso	Hora Salida	Placa	Acciones
1	2025-01-15	12:28:42	12:29:40	PDL4175	 
2	2025-01-15	12:30:40	12:31:15	PDL4175	 
3	2025-01-15	17:53:24	17:54:30	PDI8165	 
4	2025-01-15	17:55:48	17:57:29	PDI8165	 
5	2025-01-15	18:09:56		PDI8165	 
6	2025-01-15	18:18:41		PDL4175	 
7	2025-01-15	18:22:59		PDL4175	 
8	2025-01-15	18:41:16	18:42:38	PDL4175	 
9	2025-01-15	18:43:43	18:48:06	PDL4175	 

3.1.6 Vista Reporte Mensual

En la figura 23 se presenta la interfaz para acceder a los reportes mensuales, proporcionando una visión más amplia de la actividad a lo largo del mes.

Figura 23: Vista Reporte Mensual

The screenshot displays the 'Reporte mensual' interface. At the top, there is a navigation menu with the following items: REGISTRAR VEHÍCULO, REGISTRAR PERSONAL, SEGUIMIENTO USUARIOS, REPORTE DIARIO, REPORTE MENSUAL (highlighted), and Cerrar Sesión. The main content area is titled 'Reporte mensual' and contains a search bar labeled 'Buscar placa...' and a date range selector labeled 'Seleccionar rango de fechas'. Below these is a table with the following data:

Placa	Ingresos	Acciones
PDL4175	7	EXPORTAR EXCEL
POB165	4	EXPORTAR EXCEL

At the bottom right of the table, there is a pagination control: 'Records per page: 5' and '1-2 of 2'.

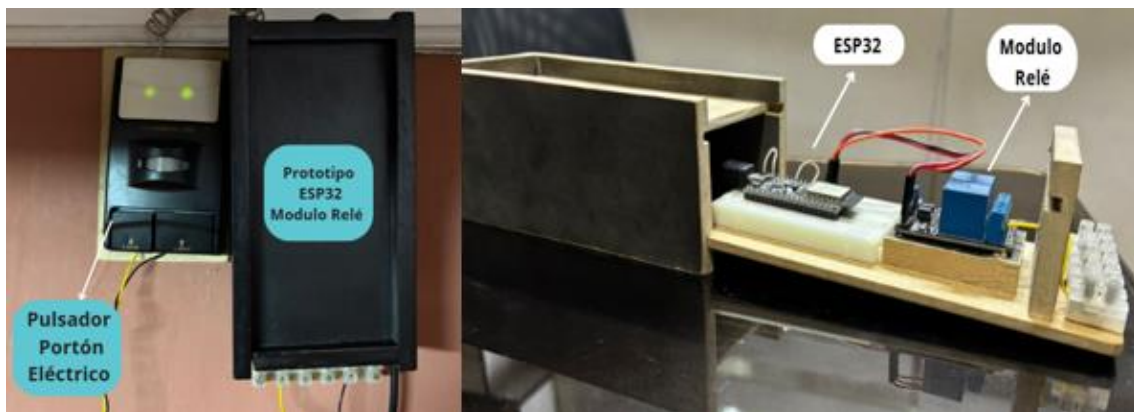
3.2 Resultados Fase 2. Prototipo Iot

3.2.1 Prototipo Finalizado

El desarrollo del proyecto para la apertura de portones mediante la identificación de placas de vehículos se fue mejorando bajo iteraciones. Inicialmente, se utilizó una cámara para capturar las imágenes de las placas de los vehículos para que posterior mediante el algoritmo y el modelo entrenado se envíe una señal al microcontrolador ESP32. Este dispositivo permitió recibir la señal para poder activar el relé y abrir el portón.

Una vez validado el funcionamiento básico del sistema, se optimizó el diseño para integrar de manera más compacta la cámara y el ESP32, lo que permitió reducir el tamaño del dispositivo y mejorar su eficiencia. En esta etapa, también se conectó el sistema a un relé, que se activa cuando la placa es validada, permitiendo la apertura del portón de manera automática. En la figura 24 podemos observar el desarrollo del prototipo

Figura 24: Prototipo Finalizado



3.3 Pruebas del sistema

Se realizaron una serie de pruebas en el sistema con el objetivo de verificar su funcionalidad y asegurar que cumpla con los requisitos establecidos. Las pruebas incluyeron escenarios como el inicio de sesión con credenciales válidas e inválidas, el registro de vehículos con datos correctos y la validación de placas duplicadas, así como el monitoreo de ingresos y salidas de vehículos, y la generación de reportes con filtros por estudiante y fechas. En todos los casos, el sistema respondió correctamente a los diferentes escenarios, mostrando mensajes de error cuando era necesario y funcionando conforme a lo esperado. Los resultados obtenidos se plasman en la tabla 8 confirmaron que el sistema es estable y capaz de manejar las operaciones de manera eficiente y confiable.

Tabla 8: Pruebas obtenidas (sistema)

Número de Prueba	Nombre	Descripción	Escenario	Resultado Obtenido
1	Iniciar Sesión	Verificar que el sistema permita iniciar sesión con credenciales válidas	Ingreso de credenciales correctas (Usuario: admin, Contraseña: 1234)	El sistema permitió el acceso correctamente.
			Ingreso de credenciales incorrectas (Usuario: admin, Contraseña: 12345)	El sistema mostró el mensaje de “Error al iniciar sesión. Por favor verifica tus credenciales”
			Campos de credenciales vacíos	El sistema marca como “requeridos” a los campos
2	Registro de un Nuevo Vehículo	Verificar que se pueda registrar un vehículo con datos correctos y que el sistema no permita duplicados.	Placa no registrada (Placa: ABC123, Marca: Toyota, Estudiante: Juan)	El vehículo se registró correctamente en la base de datos y se visualiza en el listado que está en la tabla.
			Placa duplicada (Placa: ABC123, Marca: Nissan, Estudiante: Ana)	El sistema mostró el mensaje de “El vehículo ya se encuentra registrado”
			Campos de datos vacíos	El sistema marca como “requeridos” a los campos
3	Visualización del Monitoreo de Ingreso/Salida	Verificar que el sistema registre la fecha y hora de ingreso y salida de los vehículos y que	Ingreso de vehículo registrado (Placa: ABC123)	El sistema registró la hora de ingreso correctamente.
			Salida de vehículo registrado (Placa: ABC123)	El sistema registró la hora de salida correctamente.

		la base de datos se actualice en tiempo real.	Ingreso de vehículo no registrado	El sistema no registra al no ser un vehículo con placa registrada
4	Generación de Reportes	Verificar que el sistema genere reportes correctamente al aplicar filtros de estudiante y/o rango de fechas.	Filtros correctos (Estudiante: Juan, Fechas: 01/01/2024 - 31/01/2024)	El sistema generó el reporte correctamente
			Filtros incorrectos (Estudiante: Pedro, Fechas: 01/01/2024 - 31/01/2024)	El sistema muestra una tabla sin datos al ver que el filtro no es correcto

Durante el proceso de desarrollo del prototipo, se llevaron a cabo diversas pruebas para validar su capacidad en el reconocimiento y validación de placas vehiculares. Estas pruebas incluyeron el procesamiento de imágenes, el reconocimiento de caracteres, y la comparación de placas extraídas con las registradas en base de datos. A continuación, se detallan los escenarios de prueba, los resultados esperados y los resultados obtenidos:

Tabla 9: Pruebas obtenidas (Iot)

Número de Prueba	Nombre	Descripción	Escenario	Resultado Obtenido
1	Procesamiento de Imágenes	Verificar que OpenCV detecte el objeto que entra en rango de la cámara	El objeto entra al rango de la cámara	En la sección de anexos en la figura 25 podemos observar como el objeto es percibido por la cámara
			El objeto no está dentro del rango de la cámara	En la sección de anexos en la figura 26 podemos observar como así no haya un objeto la cámara sigue registrando
2	Reconocimiento de vehículo	Mediante el uso de Tensorflow verificar si es un vehículo	El objeto es detectado como un vehículo	En la sección de anexos en la figura 27 podemos observar como el modelo entrenado detecta el vehículo
			El objeto no es detectado como un vehículo	Sigue en espera hasta que detecta que es un vehículo
2	Reconocimiento de Caracteres	Verificar que EasyOCR extraiga correctamente	Imagen con placa registrada (Placa: ABC123)	En la sección de anexos en la figura 28 podemos observar como la placa está siendo extraída

3	Validación de Placas	Verificar que el sistema compare correctamente las placas extraídas con las registradas en la base de datos.	Placa extraída (Placa: ABC123)	En la sección de anexos en la figura 28 podemos observar como si la placa está registrada nos anuncia la hora en la que el vehículo entra/sale al parqueadero
---	----------------------	--	--------------------------------	---

CONCLUSIONES

1. Automatización del Control de Acceso:

La integración del backend con Nest.js y MySQL facilita la gestión automatizada de los registros de entrada y salida de vehículos, eliminando la necesidad de intervención manual.

2. Optimización en el Reconocimiento de Placas:

Se ha logrado una detección y validación efectiva de placas en tiempo real, asegurando que solo los vehículos registrados puedan ingresar, lo que mejora la seguridad y la eficiencia operativa.

3. Seguridad y Control:

El sistema mejora la seguridad al verificar el registro de los vehículos antes de permitir su acceso y mantiene un historial detallado de las entradas y salidas para futuras consultas.

4. Mejora en la Experiencia del Usuario:

La interfaz en Quasar proporciona una experiencia de usuario intuitiva y accesible, con visualización en tiempo real y notificaciones claras sobre el estado de los accesos.

5. Escalabilidad y Flexibilidad:

El sistema es escalable y flexible, lo que permite agregar funcionalidades adicionales, como la gestión de usuarios o la integración con otros sistemas de seguridad en el futuro.

6. Potencial para Implementación en Grandes Instalaciones:

El prototipo es adecuado para garajes pequeños y tiene un gran potencial de expansión para ser implementado en instalaciones más grandes o zonas de control vehicular.

RECOMENDACIONES

1. Mejora en la Precisión del Reconocimiento de Placas:

Se recomienda mejorar la precisión del reconocimiento bajo condiciones difíciles, como baja calidad de imagen o placas sucias, mediante el uso de cámaras de mayor resolución o la optimización del modelo de TensorFlow.

2. Implementación de Cámaras de Mayor Calidad:

Incorporar cámaras de mayor resolución para mejorar la calidad de la imagen capturada, lo que ayudaría a mejorar la precisión en la detección de las placas y en la visualización en tiempo real.

3. Manejo de Condiciones Ambientales Adversas:

Se podría implementar un sistema de mejora en las condiciones de iluminación o la adaptación a condiciones climáticas adversas (como lluvia o niebla) para mejorar el rendimiento del sistema en diferentes ambientes.

4. Expansión del Sistema a Zonas de Mayor Escala:

Se recomienda considerar la implementación del sistema en estacionamientos o instalaciones más grandes, integrando múltiples cámaras y unidades de control para abarcar un mayor número de accesos.

5. Optimización del Algoritmo de Reconocimiento:

Afinar el algoritmo de reconocimiento para reducir el tiempo de procesamiento de cada frame, lo que permitirá una mayor velocidad en la detección y en la validación de las placas en tiempo real.

6. Mejorar la Gestión de Usuarios:

Para futuras actualizaciones, se podría incluir un sistema de gestión de usuarios para asignar roles y permisos, lo que permitiría un control más detallado sobre quién puede modificar o visualizar la información del sistema.

7. Implementación de Cámaras para Identificación de Conductores:

Instalar cámaras adicionales para capturar imágenes de los conductores, aumentando la seguridad y trazabilidad de los accesos.

8. Conteo de Espacios Disponibles:

Agregar sensores o procesamiento de imagen para monitorear en tiempo real la cantidad de puestos de estacionamiento libres..

Anexos

Figura 25: Objeto percibido por la cámara

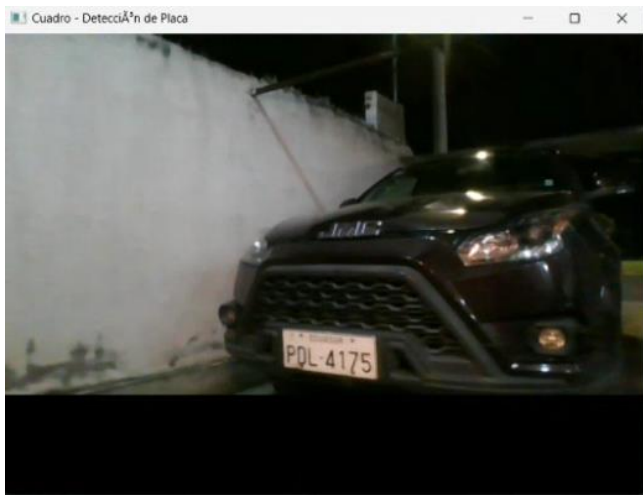


Figura 26: La cámara está en espera para percibir un objeto

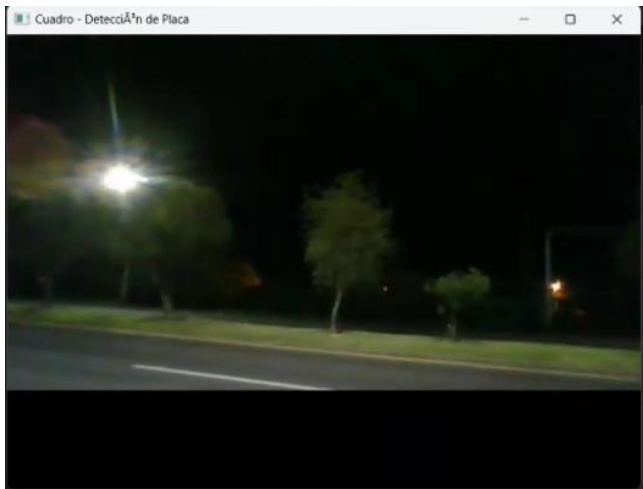


Figura 27: Modelo detecta vehículo

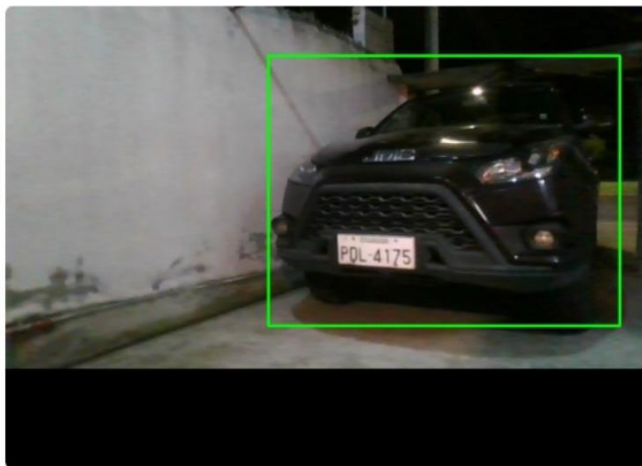


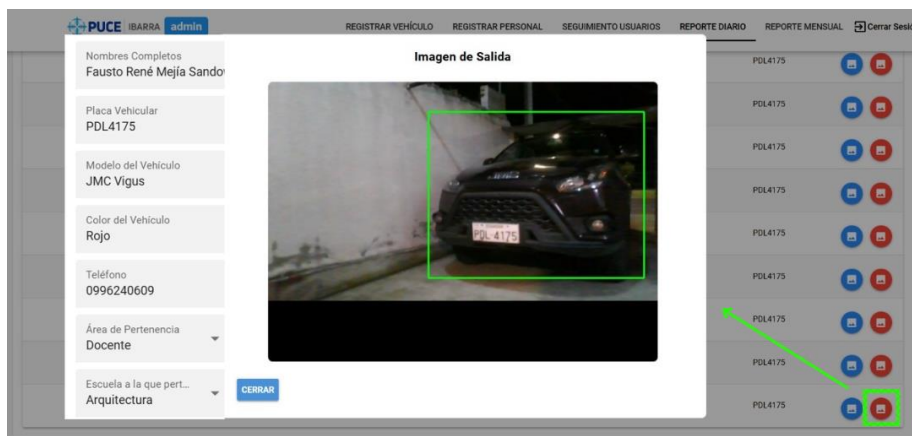
Figura 28: Extracción de placa / Ingreso al sistema



Figura 29: Registro de entrada del vehículo al sistema



Figura 30: Registro de salida del vehículo al sistema



REFERENCIA BIBLIOGRÁFICA

- Aguayo, P. (2022, 8 julio). Software de Arduino | Arduino.cl - Compra tu Arduino en Línea. Arduino.cl - Compra Tu Arduino En Línea.
<https://arduino.cl/programacion/#:~:text=El%20IDE%20es%20un%20conjunto,nuestro%20Arduino%20funcione%20como%20queramos>.
- Aguilar Anaya, J. (2022). Prototipo de reconocimiento de placas vehiculares para detección de vehículos alertados en el complejo de control aduanero de Tomasiri, Tacna - 2022 [Tesis de licenciatura]. Universidad Nacional Jorge Basadre Grohmann.
- Amazon Web Services. (s. f.). ¿Qué es IoT? - Explicación del Internet de las cosas. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/iot/>.
- Amazon Web Services, Inc. (s. f.). ¿Qué es el OCR? - Explicación del reconocimiento óptico de caracteres. <https://aws.amazon.com/es/what-is/ocr/>.
- Espressif Systems. (2023). ESP32 Series Datasheet. Recuperado de <https://www.espressif.com/en/products/socs/esp32>.
- Hussain, M., Raza, K., & Khan, S. (2022). Image recognition using machine learning: A comprehensive survey. *International Journal of Computer Applications*, 181(16), 1-12. <https://doi.org/10.5120/ijca20229226080>.
- Internet of Things with Python - Javatpoint. (s. f.). Javatpoint. <https://www.javatpoint.com/internet-of-things-with-python>.
- Khan, R., Iqbal, T., & Shah, M. (2022). A review on machine learning techniques for image recognition. *Journal of Computer Science and Technology*, 37(1), 1-19. <https://doi.org/10.1007/s11390-022-2188-x>.
- Liu, W., et al. (2016). SSD: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 21-37.

- Lucena, P. (2023, mayo 6). ¿Qué es el framework? Maestrías y MBA. CESUMA.
<https://www.cesuma.mx/blog/que-es-el-framework.html#:~:text=En%20el%20%C3%A1mbito%20de%20la,manera%20m%C3%A1s%20eficiente%20y%20r%C3%A1pida.>
- NestJS. (s. f.). Documentation | NestJS - A progressive Node.js framework.
<https://docs.nestjs.com/>.
- Niño-Rondón, C. V., Castellano-Carvajal, D. A., Castro-Casadiago, S. A., Medina-Delgado, B., & Guevara-Ibarra, D. (2021). Detección de placas vehiculares mediante modelo de clasificador en cascada basado en lenguaje Python. *Eco Matemático*, 12(1), 41-53.
- Oracle Corporation. (2023). MySQL documentation. Oracle Corporation.
<https://dev.mysql.com/doc/>.
- Peña Morán, J. C. (2023). Propuesta de un sistema de reconocimiento de placas salvadoreñas para la Universidad de El Salvador. *Revista de Investigación de la Escuela de Matemática y Física*, 2(1), 25-35.
<https://revistas.ues.edu.sv/index.php/remi/article/view/2755/3104>.
- Quasar Framework. (s. f.). Documentation. Recuperado el 29 de octubre de 2024, de <https://quasar.dev>.
- Reconocimiento óptico de caracteres con EasyOCR y Python. (2024, 20 febrero).
<https://www.toolify.ai/es/ai-news-es/reconocimiento-ptico-de-caracteres-con-easyocr-y-python-1188329>.
- Rodríguez, H. (2021, 28 abril). ¿Qué es OpenCV?: ¡Descubre todo acerca de la visión artificial! Crehana. <https://www.crehana.com/blog/transformacion-digital/que-es-opencv/>.
- TensorFlow. (2024). SSD MobileNetV2: TensorFlow Model Zoo. TensorFlow GitHub Repository. Recuperado de <https://github.com/tensorflow/models>.

Tindie. (2021). 2-Channel Relay Module with Opto-Isolation.

<https://www.tindie.com/products/techsck/2-channel-relay-module-with-opto-isolation/>.

Viso. (2024). Image recognition in 2024: A comprehensive guide. Recuperado de

<https://viso.ai/computer-vision/image-recognition/>.

Zhou, X., et al. (2023). Improving license plate recognition with deep learning models:

A comprehensive survey. *IEEE Transactions on Image Processing*, 72, 3112-3129.

Unir, V. (2023, September 27). Framework: qué es, para qué sirve y algunos ejemplos.

UNIR FP. <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>



CARTA DE ACEPTACIÓN

Ibarra, 14 de febrero de 2025

Por medio de la presente, yo **José Luis Yépez Cifuentes**, con cedula 1002751012 y ocupando el puesto de **Jefe de Seguridad Física (encargado) de la PUCE-I**, hago constar que el **Sr. Dany René Mejía Sandoval**, con cedula 1724675903, realizó el proyecto de titulación denominado **SISTEMA DE CONTROL DE ACCESO VEHICULAR MEDIANTE RECONOCIMIENTO DE PLACAS PARA EL PERSONAL DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE IBARRA**.

Este proyecto fue desarrollado en beneficio de nuestra institución, permitiéndonos llevar a cabo un mejor control de acceso.

Es importante mencionar que el proyecto cumplió con las expectativas y satisfacción de nuestra institución.

Además, el autor, Sr. Dany René Mejía Sandoval, demostró profesionalismo y compromiso durante todo el desarrollo del mismo.

Atentamente,

José Luis Yépez
JEFE DE SEGURIDAD FISICA PUCE-I (E)