

Área de Conocimiento de Tecnología de la  
Información y Comunicación

# **“Sistema inteligente de reconocimiento de placas vehiculares mediante redes neuronales para el control de acceso a la Universidad Nacional de Ingeniería”.**

Trabajo Monográfico para optar al título de  
Ingeniero en Computación

**Elaborado por**

Br. Yeferson José Berrios Moreno  
Carnet: 2017-0970U

**Tutor:**

MSc. Norwing Ernesto Madrigal Soza

## **DEDICATORIA**

A Dios, por ser guía constante, fuente de fortaleza, sabiduría y luz en cada etapa de este camino.

A mis padres, por su amor incondicional, el apoyo permanente y el ejemplo de esfuerzo que han representado a lo largo de mi vida. Su presencia ha sido esencial en cada logro alcanzado.

A mis tutores académicos, el Ing. Gabriel Lacayo y el Ing. Norwing Madrigal, por su orientación precisa, su paciencia y el compromiso demostrado con mi formación. Sus aportes han sido determinantes para la calidad de este proyecto.

A los docentes de la Facultad de Electrotecnia y Computación, por la entrega al compartir conocimientos, así como por contribuir con una base sólida que sustentó el desarrollo de este trabajo.

A todas las personas e instituciones que, de forma directa o indirecta, ofrecieron palabras de aliento, observaciones valiosas o gestos de apoyo. A cada uno, mi más sincero agradecimiento.

Este proyecto, enfocado en el diseño e implementación de un sistema de reconocimiento de placas vehiculares, representa el resultado de un esfuerzo colectivo y del acompañamiento recibido durante todo el proceso académico.

## RESUMEN

Este trabajo tiene como objetivo diseñar, implementar y evaluar un sistema automatizado de reconocimiento de placas vehiculares (ALPR) en el aparcadero de la Universidad Nacional de Ingeniería (UNI), utilizando inteligencia artificial y visión por computadora con OpenCV. La iniciativa surge ante la necesidad de mejorar el control, registro y seguridad vehicular dentro del campus, aplicando conocimientos adquiridos en el aula a un entorno real.

El sistema emplea la metodología CRISP-DM para abordar el problema de forma estructurada, desde la comprensión del dominio hasta la implementación. Se utilizaron imágenes captadas por cámaras IP en diversas condiciones ambientales para entrenar modelos de redes neuronales convolucionales (CNN), capaces de detectar y segmentar placas, y reconocer caracteres alfanuméricos. Se integraron herramientas como Tesseract OCR y técnicas de procesamiento de imagen (filtrado, binarización, detección de bordes, transformaciones morfológicas) para mejorar la precisión.

El prototipo desarrollado en Python logró una tasa de reconocimiento superior al 90% en condiciones favorables. Registra automáticamente la placa, fecha y hora de ingreso, y valida la autorización del vehículo. La información se almacena localmente y se consulta mediante una interfaz en Excel y un Dashboard interactivo.

Entre los beneficios destacan la reducción del tiempo de registro, mejora en la seguridad, trazabilidad digital y posibilidad de integración con otros sistemas institucionales. Se recomienda continuar con el perfeccionamiento del sistema mediante técnicas de aprendizaje profundo, ampliación de la base de datos y mejora de la infraestructura de hardware. Este trabajo representa un avance hacia la modernización de los procesos de control vehicular en la UNI y sirve como guía práctica para futuras generaciones de egresados.

# Índice de contenido

1. INTRODUCCION.....	1
2. ANTECEDENTES.....	3
2.1 Contexto Internacional.....	3
2.2 Contexto Nacional.....	5
3. JUSTIFICACION.....	6
4. OBJETIVOS.....	7
4.1 Objetivo General.....	7
4.2 Objetivos específicos.....	7
5. MARCO TEORICO CONCEPTUAL.....	8
5.1 Definiciones conceptuales.....	8
5.1.1 Placa de automóvil.....	8
5.1.2 Tipos de placas.....	9
5.1.3 Seguridad.....	9
5.1.4 Reconocimiento con Inteligencia Artificial (IA),.....	9
5.1.5 Inteligencia Artificial (IA).....	10
5.1.6 Algoritmo.....	10
5.1.7 Algoritmo de procesamiento de imágenes.....	10
5.1.8 Imagen.....	11
5.1.9 Procesamiento de imagen digital.....	11
5.1.10 Representación de una imagen digital.....	11
5.1.11 Tamaño de imagen.....	12
5.1.12 Binarizacion.....	12
5.1.13 Detección de bordes.....	13

5.1.14	OpenCV .....	13
5.2	Tesseract OCR .....	13
5.3	TensorFlow y PyTorch .....	14
5.4	EasyOCR.....	14
5.5	SQLAlchemy.....	14
5.6	NumPy .....	14
5.7	Matplotlib o PIL (Pillow) .....	14
5.8	Base de datos .....	15
5.9	Desarrollo web .....	15
5.10	Transformada de Fourier .....	15
5.11	Aplicativo web.....	16
5.12	Control y gestión del tráfico.....	16
5.13	Sistema de control vehicular .....	16
5.14	Prototipo de interfaz.....	17
5.15	Servidor.....	17
5.16	Pruebas de reconocimiento .....	19
5.17	Entorno controlado.....	20
5.18	Visión por computador .....	23
6.	DISEÑO METODOLOGICO .....	25
6.2	Población y muestra .....	25
6.3	Fases de la metodología.....	26
6.3.1	Fase 1: Preparación de los datos.....	26
6.3.2	Fase 2: Entendimiento de los datos .....	27
6.3.3	Fase 3: Entendimiento del negocio .....	28
6.3.4	Fase 4: Despliegue (deployment).....	29

6.3.5	Fase 5: Evaluación.....	29
6.3.6	Fase 6: Modelado.....	29
6.4	Ciclo de Vida del Software.....	30
6.5	Base de Datos Implementada en Macro Excel.....	30
7.	ANALISIS Y PRESENTACION DE RESULTADOS.....	32
	Fase 1: Preparación de los datos.....	35
	Fase 2: Entendimiento de los datos.....	49
	Fase 3: Entendimiento del negocio.....	52
	Fase 4: Despliegue.....	55
	Fase 5: Evaluación.....	57
	Fase 6: Modelado.....	60
8.	CONCLUSIONES.....	66
9.	RECOMENDACIONES.....	67
10.	BIBLIOGRAFIA.....	68
11.	ANEXOS.....	71

## Índice de Figuras

Figura 1. Diagrama de flujo de lo que sería procesamiento de imágenes digitales .....	12
Figura 2. Fases del reconocimiento de placas vehiculares .....	20
Figura 3. Proceso de la visión por computador .....	24
Figura 4. Vista frontal de la Universidad Nacional de ingeniería .....	25
Figura 5. Vista satelital del aparcadero de la universidad Nacional de Ingeniería. ....	26
Figura 6. Estructurado de la metodología. ....	31
Figura 7. Algoritmo para preparar datos a importarse a base de datos en Excel.....	36
Figura 9. Interfaz buscando vincularse a cámara IP .....	38
Figura 10. Vehículo siendo captado por la cámara, y preparación de imagen.....	39
Figura 11. Algoritmo de carga de imagen original del vehículo.....	40
Figura 12. Algoritmo, eliminando ruidos y preparación de escala de grises.....	41
Figura 13. Imagen del vehículo en escala grises .....	42
Figura 14. Algoritmo para suavizar imagen.....	42
Figura 15. Algoritmo detectando a una función específica, etiquetando al tipo de vehículo.....	43
Figura 16. Algoritmo detectando bordes, convirtiendo en color verde y preparando la detección de la placa.....	45
Figura 17. Imagen del vehículo, con detección de bordes, apoyado en Canny .....	46
Figura 18. Algoritmo que detecta bordes, Binarización OCR, y guarda imagen binarizada.....	46
Figura 19. Imagen con la detección de borde de la placa. Guardada .....	47
Figura 20. Algoritmo de la imagen procesada, devolviendo la ruta de esa imagen procesada .....	47
Figura 21. Binarización de la placa, y extracción de datos.....	48
Figura 22. Placa detectada y caracteres identificados .....	49
Figura 23. Imagen adelgazada con caracteres individuales.....	49
Figura 24. Algoritmo para exportar datos entendibles al humano en Excel para su posterior verificación de calidad de datos .....	51
Figura 25. Algoritmo para identificar al vehículo que ingresa al aparcadero .....	52

Figura 26. Login de ingreso al aparcadero una vez identificada la placa del vehículo aparece en monitor su respectiva identificación.....	54
Figura 27. Ventana de error, ingreso no registrado y ausente en base de datos .....	54
Figura 28. Reconocimiento de controles en Excel .....	56
Figura 29. Correo electrónico, ejecutando automáticamente respondiendo al informe para su revisión .....	57
Figura 30. Algoritmo para evaluar todo el modelo de visualización e identificación de placas vehiculares.....	58
Figura 31. Algoritmo para la elaboración de Dashboard de identificación de placas vehiculares .....	60
Figura 32. Diseño de la prueba, referido al control vehicular .....	61
Figura 33. Dashboard paso a paso para visualizar datos de importancia en la toma de decisiones .....	63
Figura 34. Filtro por propietario Vs fechas de visita al aparcadero .....	64
Figura 35. Evaluación de informes a través del Dashboard, validación de correcta transferencia de datos para la fase de evaluación en la metodología CRISP-DM .....	65

## Índice de Tablas

Tabla 1: Características de la cámara.....	33
Tabla 2. Comparación en la metodología CRISP-DM Vs el esquema de la propuesta de identificación vehicular .....	44
Tabla 3. Datos verificados, de acuerdo a la ejecución del proceso de la interfaz inicial, señala los datos extraídos de placas vehiculare .....	51
Tabla 4: Costos de Hardware y Software.....	73

# 1. INTRODUCCION

El presente trabajo monográfico se enmarca en el contexto del avance acelerado de la inteligencia artificial (IA), una tecnología que ha revolucionado la forma en que se abordan problemas complejos en diversos sectores. Gracias al aumento de la capacidad computacional y la disponibilidad de grandes volúmenes de datos, la IA ha evolucionado hacia sistemas más explicables, eficientes y adaptables. En particular, el reconocimiento de placas vehiculares (LPR, por sus siglas en inglés) ha pasado de ser una técnica limitada por condiciones ambientales a convertirse en una solución robusta y precisa gracias a la integración de algoritmos inteligentes.

En entornos institucionales como universidades, donde el flujo vehicular es constante, la necesidad de sistemas automatizados de control de acceso es cada vez más evidente. La Universidad Nacional de Ingeniería (UNI) enfrenta desafíos relacionados con la seguridad y la gestión eficiente del aparcamiento, lo que motiva el desarrollo de un sistema LPR basado en IA y visión por computadora. Este sistema no solo busca mejorar la precisión en la identificación de vehículos autorizados, sino también ofrecer trazabilidad, rapidez y compatibilidad con infraestructuras existentes.

El trabajo se apoya en la metodología CRISP-DM, ampliamente utilizada en proyectos de minería de datos, que permite estructurar el desarrollo del sistema desde la recolección de datos hasta su evaluación e implementación. Se emplean imágenes digitales de placas metálicas nicaragüenses, captadas en distintas condiciones ambientales, para entrenar modelos de redes neuronales convolucionales (CNN) capaces de detectar, segmentar y reconocer caracteres alfanuméricos. Además, se integran herramientas como Tesseract OCR y técnicas de procesamiento de imagen (filtrado, binarización, detección de bordes, transformaciones morfológicas) para mejorar la calidad de los datos y la precisión del reconocimiento.

El sistema desarrollado permite la identificación automática de placas en tiempo real, con almacenamiento local en Excel y visualización en Dashboards interactivos, facilitando el monitoreo y la toma de decisiones. Esta solución representa un avance significativo en la gestión vehicular de la UNI, elevando la calidad de sus procesos administrativos y fortaleciendo la seguridad institucional.

La estructura del trabajo incluye capítulos que abordan antecedentes teóricos y prácticos, justificación del proyecto, objetivos generales y específicos, marco conceptual, diseño metodológico, análisis de resultados, conclusiones y recomendaciones. Cada sección está orientada a demostrar cómo la aplicación de IA y OpenCV puede transformar la gestión vehicular en contextos universitarios, ofreciendo una base sólida para futuras investigaciones y desarrollos tecnológicos.

Finalmente, este trabajo no solo responde a una necesidad institucional, sino que también plantea nuevos desafíos para los profesionales en formación, al integrar conocimientos técnicos con aplicaciones reales. La implementación de este sistema en la UNI marca un paso hacia la modernización de sus procesos, promoviendo una cultura de innovación y eficiencia en la gestión de recursos.

## **2. ANTECEDENTES**

### **2.1 Contexto Internacional**

En los últimos años se ha venido observando el avance tecnológico sobre todo relacionado a la inteligencia artificial (IA), el uso de la IA, es estar un paso adelante en la solución, control y monitoreo de actividades sensibles a las instituciones o dependencias. De acuerdo a (Dang Minh, H Xiang Wang, Y Fen Li, Tan N Nguyen) (Dang Minh, 2022), en su libro “Inteligencia Artificial explicable: una revisión exhaustiva”, concluyen que existe un crecimiento exponencial de la potencia informática y a las grandes cantidades de datos, la inteligencia artificial (IA) ha sido testigo de avances notables en los últimos años, permitiendo la adopción propagada en la vida diaria de las personas. Los sistemas impulsados por la inteligencia artificial han aportado ventajas competitivas, la promoción de algoritmos que pueden mostrar un proceso interno y explicar cómo toman las decisiones en base a algo fundamentado.

El reconocimiento de placas de automóviles (LPR) por sus siglas en inglés, ha venido revolucionando y actualizándose cada vez transformándose de simples técnicas de procesamiento de imágenes a complejos sistemas haciendo uso de la transición de LPR hace algunas décadas utilizaban algoritmos básicos de visión por computadora para identificar caracteres en placas vehiculares, no obstante, estos métodos enfrentaban limitaciones significativas, especialmente en condiciones y medios de baja iluminación y placas dañadas o sucias. (Dang Minh, 2022).

Sin embargo, como lo afirma (Vasquez, 2014), la IA permite que los sistemas LPR no solo identifiquen caracteres con mayor exactitud, sino que también adapten su rendimiento a una variedad de condiciones ambientales y tipos de placas. En la actualidad se poseen distintos sistemas de seguridad en edificios, instituciones educativas, gubernamentales donde este método da lugar a esperar cuando se tiene un visitante y se tiene que autorizar su entrada, ya que contactar con el personal para confirmar es un poco tedioso, para los usuarios en ciertos casos puede ser un inconveniente perder su llave de acceso. La visión utilizando la inteligencia artificial automatiza el proceso de acceso de personas que son autorizados en tiempo real al acceso del mismo local basado en reconocimiento visual de la placa vehicular indudablemente la IA permite que los sistemas LPR no solo identifiquen caracteres

con mayor exactitud, sino que también adapten su rendimiento a una variedad de condiciones ambientales y tipologías de placas.

Por su parte (Miguel, 2024), fundamenta que:

“los estudios recientes sobre el reconocimiento de placas vehiculares, han demostrado que los sistemas basados en IA pueden alcanzar tasas de precisión superiores al 96% en condiciones controladas”.

En el año 1996 Daimler Chrysler en el mercado de la minería de datos, por la creciente ola de la industria, el comercio y la logística han automatizado y cuantificado en bases de datos toda la información, requiriendo respuestas en tiempo real para lograr conclusiones y decisiones críticas. Supone que los datos por si solos no serían más que registros sin significado alguno y no representan mayor información.

La metodología CRISP-DM (Cross Industry Standard Process for Data Mining), es un modelo de proceso de minería de datos que describe una manera en la que se puede abordar el problema a resolver.

En la implementación de esta tecnología en un negocio o proyecto, es necesaria una metodología que junto a experiencias similares se desarrollan en ambientes privados como públicos, consta de seis fases, en la que interactúan desde la preparación de los datos, hasta lograr la obtención de los datos, doble vía con el entendimiento de los datos, para luego el entendimiento del negocio u empresa o industria para luego tener doble vía con el despliegue, para posteriormente evaluar y modelar el modelo a seguir tal y como se explica en la figura 1.

Existe limitada literatura sobre el uso de la metodología CRISP-DM para el reconocimiento de placas, sin embargo, se ha encontrado algunas más en el ámbito de las investigaciones, entre estas se han encontrado métodos a través del procesamiento tradicional de imágenes, reconocimiento óptico de caracteres, métodos basados en aprendizajes automáticos, métodos utilizados en Deep learnig, pero muy pocas en la metodología CRISP-DM.

De acuerdo a (Cortines, 2015) existen empresas como IBM que ha integrado CRISP-DM en su producto SPSS Modeler, convirtiéndolo en una parte fundamental de soluciones de análisis y minería de datos. Agile Thought combina CRISP-DM con metodología Agile, demostrando flexibilidad y adaptabilidad en la gestión de proyectos de ciencia de datos.

## **2.2 Contexto Nacional**

De acuerdo a (Acosta Morales, 2020), el procesamiento de imagen en el dominio de la frecuencia es una herramienta útil e importante para el análisis de imágenes, implementado para la reducción de ruido y borrosidad de las imágenes con la aplicación de los filtros de este dominio y el dominio espacial. Dentro del desarrollo del sistema se logra reconocer las placas vehiculares, y es aplicando el procesamiento de imágenes haciendo uso de metodologías OpenCV, Numpy matplotlib; utilizando Python, donde la adquisición adecuada y la aplicación del reconocimiento óptico de caracteres es efectiva para extraer la información de las placas.

A pesar de los avances, existe una brecha en la literatura respecto a la implementación específica basada en la IA en contextos universitarios, poca literatura sobre este tema de identificación de placas vehiculares es muy poca o casi nula su existencia en los repositorios universitarios a nivel nacional en la utilización de metodología CRISP-DM, no obstante, existen algunas con OPEN-CV, y controles parciales sobre el acceso vehicular a distintos sectores con aparcaderos.

## **Contexto Local**

En la Universidad Nacional de Ingeniería, refiriendo al repositorio no existen temas similares, solamente de seguridad vial en carretera Rivas – Tola.

### 3. JUSTIFICACION

En una institución universitaria, dedicada a la ingeniería, se hace imperiosa la necesidad de implementar su mejora continua en todas las áreas sensibles a esta; una de ellas es la gestión del tráfico en los aparcaderos, así como la administración completa del estacionamiento. Ambas variables deben unirse a la tecnología avanzada mediante la inteligencia y visión artificial como fórmula *sine qua non* que la universidad cuente con seguridad y eficiencia en solución tecnológica para la buena gestión del acceso, ordenamiento al aparcadero y control sobre los estudiantes activos, visitantes particulares, personal docente y administrativo que tiene como propiedad o asignación un vehículo.

Al automatizar el proceso de identificación de placas vehiculares, se reduce el margen de error humano y por tanto se incrementa la rapidez en el control de acceso. Por lo tanto, esta posibilidad de acceso al “Reconocimiento de placas de automóviles, como una solución con inteligencia y visión artificial para la seguridad y la gestión universitaria utilizando los seis pasos con la metodología CRISP-DM.

La precisión en condiciones variables justifica el desarrollar algoritmos de Inteligencia Artificial (IA) robustos que manejen diferentes condiciones, tanto de iluminación, como procesamiento y calidad de imágenes utilizando OpenCV que mejore las imágenes capturadas. Además, se debe asegurar el cifrado de datos y accesos seguros, así como la implementación de políticas, normas y procedimientos de protección de datos para cumplir con las regulaciones de la universidad.

Se puede decir que la seguridad es insuficiente, debido a que el acceso no controlado de vehículos no autorizados, puede poner en riesgo la integridad física de los estudiantes, del personal docente y administrativo y a los visitantes

Además de dar un salto de calidad en la universidad en el uso de inteligencia artificial en el reconocimiento de placas de vehículos el beneficio que tendrán autoridades, personal docente, administrativo, estudiantes y visitantes particulares.

El propósito final del documento es dejar una base sólida para ejecución a posteriori del proyecto reconocimiento de placas vehiculares y que se continúe la investigación hasta su culminación por otros investigadores que deseen implicarse en la mejora continua de esta Alma Mater.

## **4. OBJETIVOS**

### **4.1 Objetivo General**

Desarrollar un sistema inteligente de reconocimiento de placas vehiculares mediante IA y OpenCV para el control del tráfico de la universidad Nacional de Ingeniería, Managua.

### **4.2 Objetivos específicos**

- Codificar algoritmos de procesamiento de imágenes que permita el reconocimiento de placas vehiculares para lecturas fiables desde una base de datos existentes.
- Diseñar un prototipo de interfaz que manipule y utilice los datos de reconocimiento de placas para optimizar el flujo vehicular en el campus de la UNI.
- Analizar el comportamiento del flujo vehicular permitiendo la toma de decisiones mediante un módulo de generación de cuadros de mandos (Dashboard).

## 5. MARCO TEORICO CONCEPTUAL

El reconocimiento de placas vehiculares, como herramienta importante para la aplicación de seguridad y control del tráfico, se ha convertido en algo fundamental para algunas instituciones; tal es el caso de la Universidad de Ingeniería en la cual se ha realizado este trabajo monográfico, apoyándose en la utilización de OpenCV, que es una biblioteca de visión por computadora de código abierto para desarrollar un sistema automatizado de detección y reconocimiento de placas; esto proporciona una variedad de funciones para el procesamiento de imágenes y la extracción de características, facilitando la implementación de algoritmos de reconocimiento de patrones.

Este trabajo se apoya de la metodología **CRISP-DM (Cross-Industry Estándar process for Data Mining)**, empleándose para estructurar el proceso de desarrollo del proyecto; pretendiendo abarcar desde la comprensión del problema y la preparación de los datos, hasta la modelación, evaluación y despliegue del sistema. Esta metodología es iterativa y al mismo tiempo flexible que asegura la eficiencia y efectividad en la construcción de modelos de reconocimiento de placas vehiculares, permitiendo mejoras continuas basadas en sus resultados.

### 5.1 Definiciones conceptuales

#### 5.1.1 Placa de automóvil

Es una estructura de metal que permite identificar a un automóvil, de forma única, conteniendo letras y números, asignada por las autoridades de tránsito, de forma permanente y válida para un país determinado. Permite obtener información del vehículo desde su matriculación hasta la actualidad y que algunas veces tienen dígitos anti falsificación. Para Nicaragua de acuerdo a (La Gaceta, 1949), en decreto de ley en su artículo 12 establece la definición de placa de automóvil para todo tipo de motorizados. “La Placa será de hojalata, de forma rectangular, pintada con pintura de aceite y del color y tamaño que indique el director de Ingresos. Leyenda: En la parte superior llevará el nombre «Nicaragua» seguido del año de vigencia a que corresponda dicha Placa, y en la parte inferior hacia la derecha, ostentará el número de orden respectivo, que deberá ser sucesivo para toda la República”.

Según (Motor), 2024), la revista motor giga, define placas de vehículo como la matrícula, placa o patente de un automóvil como “una combinación de caracteres alfabéticos o números que identifica e individualiza el vehículo respecto a los demás; son el documento de identidad de cada vehículo, las cuales permiten reconocer los derechos y obligaciones de los propietarios” efectivamente las placas de matrícula vienen a constituir el documento de identidad de cada vehículo que permite reconocer derechos y obligaciones, por ello es notable la importancia de las placas favoreciendo el desarrollo de la motorización por las implicaciones industriales, comerciales y de seguridad.

### 5.1.2 Tipos de placas



Placas de vehículos particulares. Placas blancas, con letra de la ciudad, y números en negro, establecido el escudo, el nombre del país y la región centroamericana.



Placa de vehículo de organismos y misiones internacionales con letras y números en negro

Estas placas son utilizadas al frente y parte de atrás de vehículos.

### 5.1.3 Seguridad

(Broncano, 2020), plantea que la seguridad en el control del flujo vehicular, es mucho más óptimo cuando se realiza con inteligencia artificial aprovechando las bibliotecas y software adecuados, que realizarlo de forma manual, la automatización, genera coordinación con base de datos pudiéndose identificar vehículos robados, o extraviados, por lo que genera a la institución la transparencia de no resguardar vehículos comprometidos. Por otro lado, plantea la seguridad del flujo de ingresos monetarios en el caso que se requiera de pago por resguardo, manteniendo la identificación de la placa vehicular.

### 5.1.4 Reconocimiento con Inteligencia Artificial (IA),

De acuerdo a (Omil, 2019), en su artículo de investigación, la inteligencia artificial plantea que es la habilidad y capacidad de un ordenador, red de ordenadores o red de robots controlados por ordenadores para realizar las tareas comunes asociadas a seres humanos inteligentes. Es una rama de la informática-computación que se ocupa de la simulación del comportamiento inteligente.

### **5.1.5 Inteligencia Artificial (IA)**

Define (Vigo, 2019), la inteligencia artificial como el objeto que los ordenadores hagan lo mismo que la mente humana. Con la ventaja de que puede articular sistemas automáticos que posibilitan la ejecución. La Inteligencia es una facultad cognoscitiva que facilita la comprensión y sobre ella se impulsa la capacidad de la interpretación y de la razón.

### **5.1.6 Algoritmo**

Afirma (Vivar, 2018), en su investigación, “Algoritmos, aplicaciones y Big data, nuevos paradigmas en el proceso de comunicación”, que el algoritmo es un flujograma de funciones trazadas con instrucciones bien definidas ordenadas y finitas que genera o permite realizar una actividad, mediante pasos sucesivos que no generen dudas para quien deba realizar dicha actividad

### **5.1.7 Algoritmo de procesamiento de imágenes**

Para (Morales A. , 2024), su afirmación sobre el diseño de algoritmos para procesar imágenes no es para simples fotográficas, sino datos, estructurados dentro de un abanico de posibilidades del espectro electromagnético y que para ser analizadas se requiere de procesamientos especializados a través de herramientas de software libres de uso. Donde el procesamiento de imágenes involucra técnicas y métodos para transformar, analizar y extraer información útil de las imágenes. Los algoritmos de procesamiento de imágenes pueden incluir tareas como:

- Pre procesamiento, de la eliminación de ruido, ajuste de contraste y brillo, filtrado, entre otras.
- Segmentación, que es la de dividir una imagen en regiones significativas, como identificar bordes o separar objetos dentro de la imagen.
- Detección de características que es la identificación de características relevantes como puntos clave, bordes, contorno etc.
- Transformaciones, donde se aplica la transformación geométrica o de color y finalizando con el
- Reconocimiento que es la identificación y clasificación de objetos o patrones dentro de una imagen

Para la investigación en cuestión un algoritmo de procesamiento de imágenes, puede describir los pasos y técnicas específicas utilizados para detectar y reconocer placas en las imágenes capturadas, incluyendo detalles sobre el uso de OpenCV para tareas como la detección de bordes, la extracción de regiones de interés, donde se encuentran las placas y el uso de técnicas de reconocimiento óptico de caracteres (OCR) para leer el contenido de las placas.

### **5.1.8 Imagen**

De acuerdo al diccionario de la Real Academia Española, imagen es la figura, representación, semejanza o apariencia de algo. Para (Morales R. J., 2020) imagen “Puede ser definida matemáticamente como una función de dos variables,  $f(x, y)$ , donde  $x$  e  $y$  son coordenadas espaciales (en un plano) y  $f$  en cualquier par de coordenadas es la intensidad o nivel de gris de la imagen en esas coordenadas.

### **5.1.9 Procesamiento de imagen digital**

De acuerdo a (Morales R. J., 2020), citando a (González and Woods, 2002) refieren que existen dos tipos de imágenes, y que el procesamiento de imagen se da “Si los valores  $x$  e  $y$  de  $f$  todos son cantidades finitas, discretas, se dice que la imagen es digital. Está compuesta de un número finito de elementos, cada uno con un lugar y valor específico, estos son llamados pels o píxeles. Una imagen digital puede ser definida matemáticamente como una función bidimensional,  $f(x, y)$  de donde  $x$  e  $y$  indican las coordenadas espaciales y el valor asignado a  $f$  en cualquier punto  $(x, y)$ , es proporcional a la luminosidad (intensidad) de la imagen en dicho punto”.

### **5.1.10 Representación de una imagen digital**

Según (Morales R. J., 2020), en su defensa monográfica establece que “la imagen digital se representa mediante una matriz donde “ $M$ ” son las filas que representa el valor asignado de la función y “ $N$ ” representa la columna y a su vez se asigna el nivel de intensidad para la imagen, definida por la matriz  $(I)$  de tamaño “ $M \times N$ ” También se entiende por representación de imagen lo que visualmente de algo o alguien vemos una representación que se asemeja a algo, que refleja la apariencia de algo, ya sea real o imaginario, concreto o abstracto. Donde efectivamente una imagen digital o gráfico digital es una representación bidimensional a partir de una matriz numérica frecuentemente en binario (unos y ceros). Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de una imagen matricial (o mapa de

bits) o de un gráfico vectorial aquí se representa por un diagrama de flujo según la Figura 1.

**Figura 1.**

*Diagrama de flujo de lo que sería procesamiento de imágenes digitales*



*Nota: Se observa en el gráfico el flujo de lo que sería el procesamiento de imágenes teniendo en cuenta la relación desde el pre procesamiento su relación con el procesamiento mismo y la segmentación correspondiente. Inspirado en la teoría de Morales, adaptado para este estudio como elaboración propia*

### **5.1.11 Tamaño de imagen**

Para (Morales R. J., 2020) citando a (González and Woods, 2002), refiere “El tamaño T de una imagen se refiere al número de píxeles que tiene una imagen. Las dimensiones físicas que adquiere la representación de una imagen dependen del medio de representación. Otra característica, es la resolución espacial definida como el número de píxeles, por unidad de área y relaciona entonces las dimensiones físicas de la representación con el tamaño de la imagen:  $D = T \times p$ .

### **5.1.12 Binarización**

De acuerdo (Molina E. O., 2016), plantea que “La Binarización de imágenes es una técnica básica en el procesamiento de imágenes digitales. Esta técnica convierte imágenes en escala de grises o a color en imágenes en blanco y negro.” En este caso la Binarización produce una imagen con tan solo dos niveles de gris y se puede utilizar para mejorar las funciones de una imagen seleccionando el umbral adecuado. La Binarización puede ser también el primer paso de un análisis de imagen. La conversión de la imagen digital en imagen blanco y negro, para que se preserven las

propiedades esenciales de la imagen. Uno de los métodos para poder binarizar una imagen digital es mediante el histograma de dicha imagen.

#### **5.1.13 Detección de bordes**

La universidad de Jaén, del departamento de ingeniería y sistemas, refiere que la detección de bordes es una herramienta fundamental en el procesamiento de imágenes y en visión por computadora, particularmente en las áreas de detección y extracción de características que tiene como objetivo la identificación de puntos en una imagen digital en la que el brillo de la imagen cambia drásticamente. (Universidad de Jaen, s/a)

#### **5.1.14 OpenCV**

(OpenCV, 2012) Pertenece a Intel, describe que esta es una biblioteca libre de visión artificial. Conteniendo más de 2500 algoritmos. Especializada en el sistema de visión artificial. Sus aplicaciones incluyen, características 2D y 3D, estimación de pose de cámara, reconocimiento facial, reconocimiento de gestos, interacción persona-computadora, robótica móvil, comprensión de movimientos, reconocimiento de objetos, segmentación, estereoscopia, *structure from motion* (SFM), Tracking, realidad aumentada. Pudiendo desarrollarse con alta eficiencia computacional con otros conectores para otros lenguajes como Python, Java, Matlab, Octave y Javascript.

Tesseract-OCR. De acuerdo a (Hewlett-Packard, 2024) en su página web, refiere que Tesseract es un motor OCR de código abierto que extrae texto impreso o escrito de las imágenes. Fue desarrollado originalmente por Hewlett-Packard, y su desarrollo fue adquirido después por Google. Por eso se conoce ahora como “Google Tesseract OCR” una interfaz de programación de aplicaciones (API).

### **5.2 Tesseract OCR**

Según Hewlett Packard, en el 2006 creó software como un motor de reconocimiento óptico de caracteres para varios sistemas operativos. Es software libre, liberado bajo la licencia Apache, Versión 2.02 y su desarrollo es financiado por Google desde el 2006, con el propósito: de reconocimiento óptico de caracteres (OCR), que es esencial para leer los números y letras de las placas.

### **5.3 TensorFlow y PyTorch**

(Paszke, 2024), y otros autores, junto a Facebook crearon una biblioteca de software que permite entrenar y hacer inferencia de redes neuronales. Es una buena opción para proyectos que requieren escalabilidad y producción a gran escala. Permite implementar modelos en múltiples plataformas, incluyendo móviles, navegadores y servidores en la nube. TensorFlow Lite y TensorFlow Serving optimizan el rendimiento en dispositivos de baja capacidad.

### **5.4 EasyOCR**

Jaidev IA, lo creo con el propósito, como una alternativa a Tesseract, es una biblioteca basada en PyTorch que tiene modelos pre entrenados para detectar textos en múltiples idiomas, incluidas las placas vehiculares.

### **5.5 SQLAlchemy**

Mike Bayer creó SQLAlchemy en el año 2005. SQLAlchemy es una biblioteca que se utiliza para acceder a bases de datos SQL en Python. Es una herramienta de mapeo relacional de objetos (ORM) que traduce clases Python a tablas en bases de datos relacionales. SQLAlchemy se utiliza para gestionar datos de bases de datos como SQL, PostgreSQL, SQLite, entre otro, con un único propósito para este trabajo monográfico, apoyarse en el caso de necesitar almacenar y consultar información relacionada con las placas, esta biblioteca permitirá interactuar con bases de datos SQL de manera eficiente.

### **5.6 NumPy**

Creado por Jim Hugunin con contribuciones de varios otros desarrolladores. En 2005, Travis Oliphant creó NumPy teniendo el propósito de ser una biblioteca para manejar matrices y realizar operaciones matemáticas rápidas. Es fundamental para el procesamiento de imágenes y otras operaciones computacionales en visión por computadora.

### **5.7 Matplotlib o PIL (Pillow)**

John D. Hunter, en el 2003, creo con el propósito para mostrar imágenes, gráficos y realizar algunos ajustes visuales en el procesamiento de imágenes.

## 5.8 Base de datos

(Espinoza, 2015), refiere que la base de datos es una colección de información que contiene datos relativos pertenecientes a un mismo contexto, compartiendo entre si algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto. Realizando una analogía una biblioteca puede considerarse una base de datos, compuesto por libros y documentos impresos indexados para su consulta. Existen sistemas dedicados a la gestión de base de datos que permiten almacenar datos de manera rápida y estructurada. Entre estos sistemas, los más conocidos son: Oracle, DB2 de IBM, Sql Server de Microsoft, MySQL (Oracle), Microsoft Access; para este trabajo monográfico se ha optado por el uso de MySQL ya que es un gestor de código abierto y no tiene costos de licenciamiento, es el más utilizado en aplicaciones web para comercio electrónico y aplicaciones dinámicas. Es compatible con lenguaje PHP para la implementación de páginas web.

## 5.9 Desarrollo web

Según (Suaza, 2023), El Desarrollo Web se puede definir como el proceso de creación de sitios web, aplicaciones o software. Este proceso implica llevar a cabo un estudio exhaustivo para determinar las necesidades del cliente y, a partir de esa información, diseñar e implementar una interfaz atractiva y funcionalidades que satisfagan las necesidades del usuario final.

## 5.10 Transformada de Fourier

Según (Molina E. M., 2016), “La Transformada de Fourier  $F$  es una poderosa herramienta capaz de ayudarnos a realizar la conversión de señales del espacio continuo al dominio de las frecuencias. La Transformada de Fourier  $F$  convierte una función  $f(t)$  al dominio de la frecuencia  $F(f(t)) = F(w)$ , donde  $w$  es la frecuencia en  $\text{Hz} = \text{s}^{-1}$ . A partir de este punto,  $F(w)$  es una función compleja en el espectro de la frecuencia que permite visualizar de manera sencilla diferentes porciones del espectro de la frecuencia. De este modo la Transformada de Fourier  $F$  de una función continua está dada por

$$F\{f(t)\} = F(w) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i w t} dt$$

### **5.11 Aplicativo web**

De acuerdo a (Suaza, 2023) citando a (Amazon, s.f.), que una aplicación web es un tipo de software que se ejecuta en un navegador web. Las empresas utilizan estas aplicaciones para intercambiar información y ofrecer servicios de manera remota, estableciendo una comunicación segura con sus clientes cuando sea necesario. Las funcionalidades más comunes en los sitios web, como carritos de compra, búsqueda y filtrado de productos, mensajería instantánea y canales de noticias en redes sociales, están diseñadas de forma similar a las aplicaciones web. Estas aplicaciones permiten acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software adicional.

(Suaza, 2023) Define citando a (CDSparking, s.f.) que el sistema de parqueadero es un conjunto de componentes y software diseñado para gestionar y administrar el estacionamiento de vehículos en un determinado espacio.

### **5.12 Control y gestión del tráfico**

(Perez, 2014), establece que el tránsito vehicular es como se denomina al flujo de automotores que van por una calle, autopista, o en su efecto como ingresa o egresa el automotor a determinado aparcadero. Entendiéndose como el número de vehículos que pasan, ingresan en un determinado tiempo. Existen muchas aproximaciones matemáticas que intentan modelar el flujo de tránsito vehicular en términos de alguna de sus características. Todas ellas responden con cierto grado de apego a la realidad, midiendo alguna o algunas características del flujo vehicular. Para el control y gestión del tráfico existen modelos macroscópicos que se enfocan en captar las relaciones globales del flujo de tránsito, tales como velocidad de los vehículos, flujo vehicular, densidad de tránsito. Estos modelos se enfocan en la descripción del comportamiento del flujo del tráfico vehicular a través de descripción de las entidades discretas individuales y atómicas que interactúan unas con otras (en este caso cada vehículo individual). Se dice que estos son generalmente modelos discretos.

### **5.13 Sistema de control vehicular**

Según (Perez, 2014), el sistema de control vehicular está conformado por los elementos de un sistema de control y que se gestiona a partir del acceso de estos

automotores a determinado lugar. Estos pueden variar entre diferentes parqueaderos o controles en carretera, pero los principales elementos son los siguientes:

- Barreras de paso
- Controles de recaudo
- Sensores de piso
- Lectores de tarjetas de proximidad

#### **5.14 Prototipo de interfaz**

Para (Tecnologicos, 2024) establecen que el prototipado web permite obtener una visualización previa de una página web, aplicación o interfaz en desarrollo, antes de llegar a su etapa final. La finalidad principal es anticipar errores y maximizar la experiencia final de uso. Todo el proceso el diccionario de la Real Academia Española define que es la conexión, física o lógica entre una computadora y el usuario, un dispositivo periférico o un enlace de comunicaciones. Así mismo la interfaz de usuario o UI (User Interface) es un concepto que abarca arquitectura de información, patrones y diferentes elementos visuales que permiten interactuar de forma eficaz con sistemas operativos y softwares de diversos dispositivos. Efectivamente interfaz es designada en informática, como la conexión física y funcional que se establece entre dos aparatos, dispositivos o sistemas que funcionan independientemente uno del otro. En este sentido, la comunicación entre la persona y una computadora se realiza por medio de una interfaz.

#### **5.15 Servidor**

De acuerdo a (Rouse, 2024), en su artículo de revista el servidor o server tiene dos acepciones o definiciones, por el uso que tiene en la informática. Por una parte, se refiere al programa que aloja los datos e información recibida de la red y además es el ordenador o equipo mediante el cual se gestionan los recursos para dicha información. Es una parte medular en la transmisión de datos en internet y que apoya la transmisión, mantenimiento, gestión y seguridad de los sistemas informáticos. Se trata de un servicio que funciona de manera continua, puesto que almacena y transmite la información a los usuarios de manera permanente. En la arquitectura cliente/servidor, existen dos tipos de ordenadores:

- Clientes son los ordenadores con los que interactúan los usuarios finales
- Los servidores son los ordenadores que proporcionan recursos y servicios a los clientes. Donde un único servidor puede satisfacer las necesidades de varios clientes, y un único-cliente puede acceder a varios servidores.

Los límites entre clientes y servidores se han vuelto más fluidos con la adopción generalizada de computación en la nube y micro servicios que pueden actuar como cliente y servidor en diferentes contextos.

Los servidores tienen varios tipos diferentes de formatos que se adaptan a diferentes escenarios de despliegue, necesidades de escalabilidad, limitaciones de espacio y requisitos de gestión. La elección de estas características depende de las necesidades específicas de un entorno informático.

Los servidores de torre son unidades autónomas que pueden colocarse en el suelo o en un escritorio. Se asemejan a los antiguos ordenadores torre de sobremesa. En cambio, los servidores de montaje en bastidor están diseñados para apilarse horizontalmente en armarios especiales para centros de datos llamados bastidores.

Este tipo de servidor viene en tamaños estándar que indican la altura del factor de forma en términos de unidades de rack. 1U mide aproximadamente 1,75 pulgadas (o 44,45 mm) de alto. Los tamaños más comunes para servidores de montaje en bastidor son:

- 1U: Este es el factor de forma más compacto en una configuración de montaje en bastidor. Un servidor 1U se conoce a menudo como servidor de caja de pizza porque mide 1,7 pulgadas de alto, 19 pulgadas de ancho y 17,7 pulgadas de profundidad.
- 2U: Estos servidores miden aproximadamente 3,5 pulgadas de alto y se suelen utilizar cuando se necesita más potencia de procesamiento o almacenamiento.
- 4U: Estos servidores miden aproximadamente 7 pulgadas de alto y se suelen utilizar para informática de alto rendimiento (HPC) y para ejecutar aplicaciones que consumen muchos recursos.

Los servidores Blade tienen un factor de forma delgado que facilita las implementaciones de servidores de alta densidad. Varios módulos de servidor se montan horizontalmente en el mismo chasis Blade para reducir el consumo de energía

y la complejidad del cableado. Los micros servidores son servidores de bajo consumo diseñados para cargas de trabajo ligeras. Son más pequeños que los servidores tradicionales y suelen tener menos componentes de hardware y funciones de gestión. Este factor de forma puede ser una opción rentable para aplicaciones específicas de Internet de las Cosas (IoT).

### **5.16 Pruebas de reconocimiento**

(Espinoza, 2015), establece que las pruebas de reconocimiento deben de estar dadas por la prueba misma de:

- Prueba de precisión de reconocimiento, que debe detallar la descripción, el procedimiento de conjunto de datos con imágenes de matrículas conocidas y su comparación, así como la métrica en la tasa de aciertos, de errores y tasa de falsos positivos y negativos.
- Prueba de velocidad de procesamiento, que mide el tiempo que tarda el sistema en procesar una imagen y generar el resultado, donde se registran los tiempos de procesamiento desde la captura de la imagen hasta generación del resultado, así como el tiempo medio de procesamiento por imagen
- Prueba de robustez ante condiciones adversas, que es el proceso el cual se evalúa el rendimiento del sistema en condiciones desfavorables como poca iluminación, matrículas sucias, ángulos difíciles y condiciones climáticas adversas, también aquí se prueba el sistema con imágenes que representan diferentes condiciones adversas, y se mide la tasa de aciertos bajo condiciones adversas.
- Prueba de escalabilidad, donde se determina la capacidad del sistema para manejar un gran volumen de imágenes en periodos corto, sometiendo al sistema a un alto número de imágenes para procesar en un tiempo limitado, tomando en cuenta la tasa de procesamiento por segundo, tasa de caída de rendimiento.
- Prueba de integración con otros sistemas, donde se verifica la integración del sistema de reconocimiento de matrículas con otros sistemas de gestión o seguridad, haciendo las pruebas de ser necesario
- Prueba de tolerancia a fallos, evaluando el comportamiento, simulando fallos, y pérdidas de conexión,

- Prueba de exactitud geográfica, donde se evalúa la capacidad del sistema para reconocer matriculas de diferentes regiones o países sea el caso.

**Figura 2.**

*Fases del reconocimiento de placas vehiculares*



*Nota: se observa en la figura el primer proceso de reconocimiento del entorno de la placa vehicular, llevándose a la Binarización, identificando tonalidades grises y escaneando los relieves de caracteres numéricos y letras, y la última fase que es el reconocimiento y segmentación de caracteres para reducir los pixeles.*

### **5.17 Entorno controlado**

(Aldas, 2016), refiere que el sistema debe en todo momento mantener un ambiente controlado y este incluye:

- Iluminación controlada, donde el uso de luces artificiales que eviten sombras y reflejos para asegurar una imagen clara de la placa.
- Cámaras de alta resolución, que capturen imágenes detalladas de las placas desde diferentes ángulos.
- Ubicación de las cámaras, colocadas en puntos estratégicos para captar la placa en el mejor Angulo posible.
- Fondo controlado, minimizando los fondos distractores o complicados que puedan dificultar la detección de la placa.
- Velocidad controlada del vehículo, donde la velocidad del vehículo es predecible o constante, facilitando la captura de imágenes.

En un ambiente controlado, las condiciones se ajustan para maximizar la precisión del sistema reduciendo la posibilidad de errores causados por variaciones en las condiciones ambientales.

### **Aplicación de los sistemas de procesamientos de imágenes para el reconocimiento de placas vehiculares:**

Las técnicas de procesamiento digital de imágenes proclives al reconocimiento de placas vehiculares tienen aplicaciones de relevancia, principalmente en lo relacionado con análisis de tráfico en carreteras, recaudación de derecho de paso o peaje, detección de vehículos robados o con infracciones o multas, estacionamiento de vehículos, multas impuestas a vehículos por exceso de velocidad, conteo de vehículos, estudios de flujo de tráfico, identificación de tipo de vehículos, identificación del registro vehicular (identificación del dueño de un vehículo), etc. (Espinoza, 2015).

### **Procesamiento de imágenes binarias**

Este método se basa en el principio de que el cambio de brillo en la región de la placa es más remarcable y frecuente que en cualquier otro lugar de la imagen. Un inconveniente de este método es su alta sensibilidad a bordes no deseados, mostrándolos con una elevada magnitud con el consiguiente perjuicio en la extracción exitosa de la región de interés de la placa. Los algoritmos deben ser capaces de localizar objetos pequeños con diferencia de brillo (Espinoza, 2015).

### **Análisis parcial de la imagen**

Este método permite escanear parcialmente imágenes de los vehículos con análisis repetidos en diferentes secciones para contar los bordes existentes. En este método no existen dependencias entre la distancia de la cámara y el tamaño de la placa vehicular (Espinoza, 2015).

### **Procedimiento de análisis de la placa**

Independientemente de la técnica aplicada, los pasos preparatorios del reconocimiento de los caracteres de la placa son los siguientes:

- Obtención de la imagen a reconocer, es decir, una subimagen de la placa vehicular.
- Detección de un rectángulo de imagen sin espacios irrelevantes.

- Reducción de escala.
- Conversión de los pixeles representados en la imagen como una matriz en una cadena (o “string”), haciendo un recorrido de filas y columnas.

### **Comparación con plantillas**

Este método permite escanear parcialmente imágenes de los vehículos con análisis repetidos en diferentes secciones para contar los bordes existentes. En este método no existen dependencias entre la distancia de la cámara y el tamaño de la placa vehicular (Espinoza, 2015). Esta técnica se basa en el principio de considerar a los caracteres y plantillas como matrices de “x” filas e “y” columnas. El reconocimiento se logra comparando un carácter de entrada con cada una de las plantillas y encontrar la mejor coincidencia.

### **Tareas de un sistema de reconocimiento de placas vehiculares.**

El reconocimiento se logra comparando un carácter de entrada con cada una de las plantillas y encontrar la mejor coincidencia. Las tareas fundamentales del sistema son las siguientes:

- Identificación de un usuario usando su placa vehicular.
- Presentación de información relacionada a la placa.
- Administración de usuarios que serán identificados posteriormente mediante placa vehicular.
- Garantía de acceso al parqueadero incluso en casos que no sea posible detectar una matrícula vehicular.
- Control de un prototipo de ingreso a un parqueadero.

(Espinoza, 2015)

### **Actores del sistema de reconocimiento de placas vehiculares.**

- **Usuario:** Es la persona que solicita el ingreso al parqueadero. Hay dos posibilidades: que esté registrado en el sistema o no.
- **Administrador:** Es la persona que maneja y gestiona las configuraciones en el sistema y el registro de usuarios y vehículos.
- **Sistema parante:** Es el mecanismo que permite abrir y cerrar el parante ubicado en el ingreso del parqueadero.

- **Base de datos:** Colección de información referente a los usuarios registrados en el sistema y a los administradores que pueden utilizarlo.

### **Principios de uso del sistema de reconocimiento de placas vehiculares.**

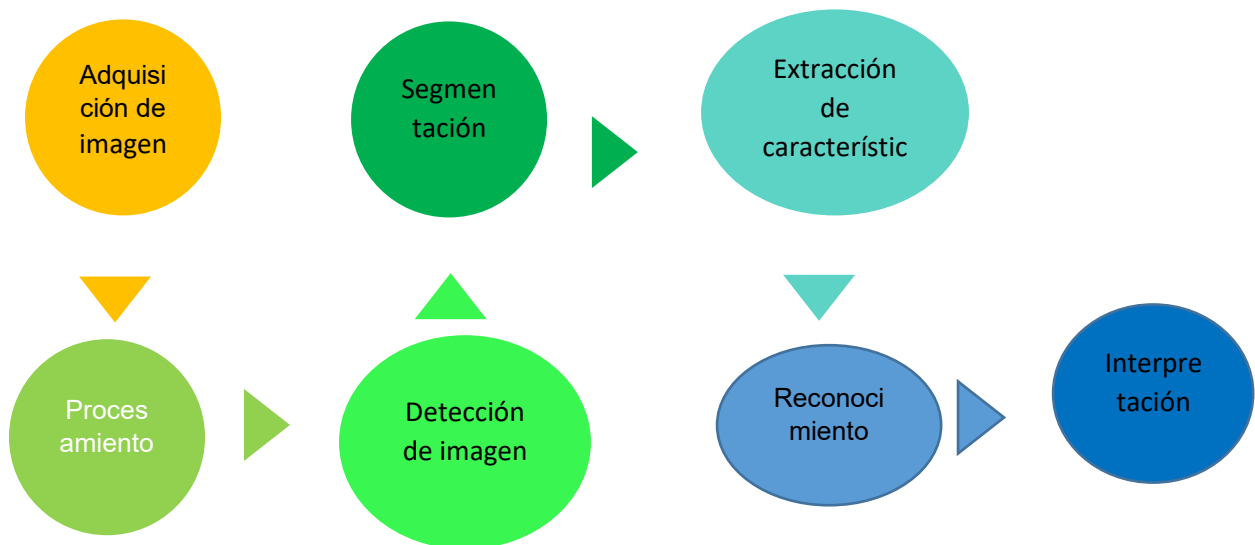
- **Comprensión:** Implica la detección de la placa vehicular, sus caracteres y los datos del usuario.
- **Aprendizaje:** El sistema debe ser sencillo y de fácil empleo para garantizar la operación prácticamente intuitiva del sistema de parte del usuario.
- **Operatividad:** El sistema debe contar con opciones de menús que hagan posible una operación eficaz y eficiente.
- **Atractividad:** La interfaz debe ser amigable al usuario y atractiva a la vista.

### **5.18 Visión por computador**

La visión por computador inicia con la realización de la primera fotografía por Joseph Nicephore Niepce, en el año 1825, en la cual se empleó una cámara oscura y una superficie fotosensible para fijar la imagen en esta. Luego en 1920 se utiliza el cable submarino para enviar y recibir imágenes entre Londres y Nueva York. Con la evolución en el sistema de las cámaras de video, en 1980, así como los microprocesadores y los computadores se logra desarrollar la visión por computador que permite la extracción de características relevantes, incluso que este procedimiento se pueda realizar de forma automática. La visión por computador realiza varios procesos (Somolinos, 2002); Adquiere la imagen, donde se captura la imagen digital, pre procesar mediante filtros, transformaciones que hacen posible obtener una imagen más nítida y con mayor claridad, la detección de bordes, donde se selecciona y verifica el área que se va a trabajar. Para luego segmentar que es la parte que comprende la partición de la imagen regiones con significado propio, de esta forma es posible interpretar y reconocer el área que se requiere identificar con colores distintivos, a diferencia del flujo de procesamiento de imágenes. Ver Figura 3

**Figura 3.**

*Proceso de la visión por computador*



Fuente: (Rodríguez, 2018),

### **Rendimiento optimo**

Mantiene que el rendimiento óptimo se consigue en este sistema propuesto, cuando la placa vehicular es detectada sin inconvenientes por cualquiera de los métodos implementados, y los caracteres de identificación de la placa son reconocidos en un cien por ciento de certeza, sin caracteres sobrantes o faltantes.

### **Rendimiento Bueno**

Continúa afirmando el mismo autor que el rendimiento es bueno cuando la placa vehicular es detectada por alguno de los métodos implementados, y el reconocimiento es realizado con un setenta por ciento de certeza o más, permitiendo pérdidas de hasta un carácter, o detección de un carácter adicional.

### **Rendimiento Malo**

En este mismo enfoque el rendimiento es malo cuando la placa vehicular se detecta total o parcialmente, con cualquiera de los métodos implementados en el sistema propuesto, pero los caracteres de la placa son reconocidos con un nivel de certeza del sesenta por ciento o menos, con pérdidas o incrementos de varios caracteres. Aquí entra en juego el estado físico de cada placa vehicular.

## 6. DISEÑO METODOLOGICO

La metodología a utilizar es CRISP-DM (*Cross-Industry Standard process for Data Mining*) altamente aceptada y utilizada debido a su enfoque estructurado pero flexible. Esta consta de seis fases cada una con tareas específicas que deben completarse una a una algunas con funciones bidireccionales. El resultado se espera sea de alta calidad para que se aplique y se cumplan con los objetivos formulados.

### 6.2 Población y muestra

El universo está compuesto por todos los vehículos que ingresan al campus universitario durante un periodo académico. Incluye vehículos livianos, como vehículos de servicio, que frecuentemente acceden al campus universitario. Para este caso se obtuvo una vista satelital del área arrojando una capacidad total de 101 vehículos. La muestra se ha obtenido con equipos, cámaras, y software en área libre fuera del aparcamiento de la universidad; debido a la privacidad que la universidad tiene, esta no presta sus instalaciones para la realización del montaje.

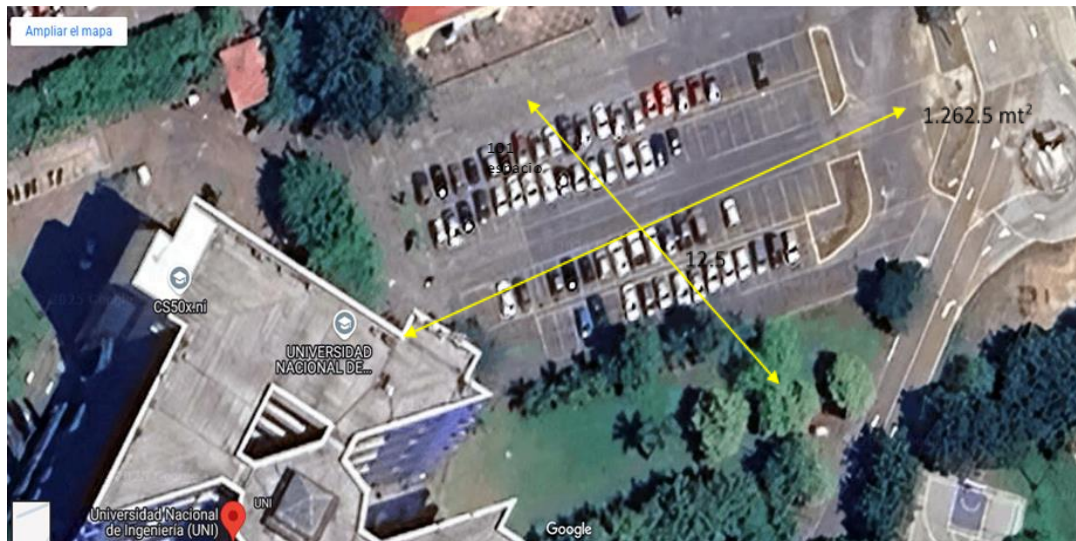
#### Figura 4.

*Vista frontal de la Universidad Nacional de ingeniería*



## Figura 5.

Vista satelital del aparcadero de la universidad Nacional de Ingeniería.



Nota: Tomado de google maps. En una escala de 50 pie/ 20 m

### 6.3 Fases de la metodología

#### 6.3.1 Fase 1: Preparación de los datos

- a. **Selección de datos:** En esta parte se da la captura de imágenes desde la cámara de vigilancia, en diferentes condiciones (día, noche, lluvia, polvo, neblina), descarga conjunto de datos públicos, buscados en bases de datos integradas, genera datos simulados a las placas vehiculares donde se incluyan tamaños, colores, descripción, figuras etc. Considerando también distancias y ángulos de captura; para posteriormente etiquetar las imágenes. Este paso metodológicamente ha sido para la visualización y revisión de literatura.
- b. **Limpieza de datos:** Aquí se ha identificado la calidad de las imágenes (resolución, condiciones de iluminación, ángulos de las cámaras) las que deberá excluir imágenes borrosas de baja calidad e irrelevantes. Reconocer patrones de tráfico (horas pico, días con mayor afluencia). Variabilidad en las placas, así como descartar imágenes que no sean vehículos y que no tengan placas.

- c. **Construcción de datos:** estará a cargo de las cámaras ubicadas en los accesos del campus. Construirá datos históricos de registros de vehículos (si existen). Y realizara la información administrativa (listas de vehículos autorizados). Se vincula la metodología CRISP DM vs los pasos del método científico para comprender los procesos de reconocimiento de placas de diferentes tamaños y además en coherencia con los distintos procedimientos hasta obtener un producto final.
- d. **Integración de datos:** En el proceso de integración de datos de diversos orígenes, teniendo un acceso fácil y rápido de donde diferentes usuarios autorizados puedan obtener de forma completa y consolidada donde varios sistemas puedan interpretar. Al facilitar estos se podrán tomar las decisiones rápidas y asertivas con respuestas en beneficio de la administración operativa del acceso vehicular a la universidad. Determinar el sistema de carga de los datos ya sea en la nube, o algún sistema de almacenamiento centralizado.
- e. **Formateo de datos:** en esta fase se cumple la estandarización de los datos, luego la conversión del tipo de datos, eliminación de consistencias, o corrección de las mismas y la normalización o ajuste de los datos a un formato común, como la conversión de cadenas o números representados; eliminación de espacios y caracteres innecesarios; o, conversión de datos textuales a categorías. Esto en dependencia de la mejor conveniencia a la universidad.

### 6.3.2 Fase 2: Entendimiento de los datos

- a. **Recopilación de datos:** en esta parte de la fase se debe recopilar que tipo de vehículos livianos (camionetas, suburvan, carros, camiones de carga, motocicletas (eléctricas, híbridas, gasolina).
- b. **Descripción de datos:** debe describir número de placa, hora de entrada y salida, tipo de usuario (estudiante, docente, visitante, administrativo, de servicio)
- c. **Exploración de datos:** Aquí se va a identificar y decidir sobre valores faltantes, se verificará si los datos tienen errores tipográficos o de formato. Este paso proporcionara la base necesaria para tomar decisiones informadas a las autoridades en la universidad, sobre cómo preparar los datos, de

acuerdo al modelo a utilizar y cómo abordar las tareas de minería de datos (como el reconocimiento de placas vehiculares).

- d. **Verificación de calidad de datos:** Se realizará la verificación, limpieza y calidad de los datos obtenidos, y el resultado deseado. De acuerdo al método científico se estaría recibiendo la experimentación de enfoque de vehículos, específicamente el diseño de placas, pasando por su Binarización y arrojando al programa seleccionado ya en texto y números entendibles al humano.

### 6.3.3 Fase 3: Entendimiento del negocio

- a. **Objetivos del negocio:** Garantizar en este paso la restricción o no de los vehículos autorizados, además de controlar y analizar datos del número y porcentajes de ingresos vehiculares. Re direccionar de acuerdo a la minería de datos obtenidos, dentro del campo de la estadística y aprovechando la inteligencia artificial referido al proceso de descubrir patrones en grandes volúmenes de conjunto de datos, recibidos los 365 días del año, tanto en la temporada de ciclo académico, como en ciclo de temporada baja donde solo el personal administrativo y operativo docente ingresan al local. Posteriormente debe consistir en la extracción de datos y transformarlas en algo comprensible para un uso posterior de ser necesario. En resumen, se debe implementar un sistema que registre automáticamente las placas de los vehículos que ingresan y salen del campus universitario para mejorar la seguridad y la administración del flujo vehicular.
- b. **Requisitos del negocio/ proyecto:** Para esta fase se ha garantizado la detección y registro del vehículo. Teniendo como en un proyecto de investigación la muestra obtenida en cada unidad de investigación como sería cada vehículo autorizado al acceso. Donde la muestra será reflejada con la letra "m", y la observación por parte de la cámara con la letra "o"; por tanto utilizando la referencia de (Elias, 2023), donde se garantiza una metodología flexible de enfoque mixto, cuantitativa (por los datos numéricos), cualitativo (por el análisis de los datos) y descriptiva (por los datos recogidos en cada muestra monitoreada).
- c. **Objetivos del proyecto:** ajuste entre la metodología Crisp-DM y el método científico que respalda la elaboración de este proyecto.

#### 6.3.4 Fase 4: Despliegue (deployment)

- a. **Planificación del despliegue:** Se podrá poner los pasos experimentales, así como, el modelo entrenado y evaluado desplegándose en un entorno real, implementándolo en un sistema de monitoreo de tráfico o seguridad, por eje.
- b. **Implementación:** Monitoreando el despliegue de los pasos efectuando, todo el proceso de verificación de placas vehiculares, metodológicamente es el resultado de prueba de la aplicabilidad del proyecto.
- c. **Monitoreo y mantenimiento:** no es más que mantener un seguimiento y monitoreo para el mantenimiento en tiempo real de anomalías que puedan ser resultas al momento de su hallazgo.
- d. **Revisión del proyecto:** La entidad se encargará de dar a conocer sus resultados y podrá, replicarlo en otras áreas adaptándolo a verificación de rostros u otras aplicabilidades, dando lugar a informes a otros niveles. En resumen, es la revisión de la mejora continua del proyecto.

#### 6.3.5 Fase 5: Evaluación

- a. **Evaluación del modelo:** Se pretende evaluar constantemente dentro de la fase anterior de revisión la evaluación periódicamente para ver su funcionalidad.
- b. **Revisión del proceso:** no se pretende hacer culminación de todo el ciclo, sino más bien mantener un proceso cíclico de sino revisar nuevos resultados, explorar mejorar y replantear nuevos procesos de calidad.
- c. **Determinación de próximos pasos:** En el caso que la evaluación arroje nuevos pasos, se deberá de elaborar nuevamente pasos y procesos dentro de la misma metodología en coherencia con el método científico.

#### 6.3.6 Fase 6: Modelado

- a. **Selección de técnicas de modelado:** en este proyecto se ha realizado las técnicas de modelado de acuerdo al clima, horas del día, pruebas OCR, análisis del modelo en visión nocturna, publicando a manera interna los resultados para redirigir y validar la efectividad de este frente a nuevas circunstancias que en la actualidad la tecnología y las necesidades van requiriendo.

- b. **Diseño de pruebas:** No es más que la revisión de pasos, su robustez y seguridad, velocidad de ingeniería de datos para su análisis, así como la identificación de errores en tiempo real.
- c. **Construcción del modelo:** en esta fase se ha realizado buscando el tipo de pruebas a realizar, pruebas de robustez, pruebas funcionales y pruebas de rendimiento.
- d. **Evaluación del modelo:** No es más que la evaluación al final de cierto periodo que la universidad deberá evaluar el modelo propuesto, comparándolo con los resultados.

#### **6.4 Ciclo de Vida del Software**

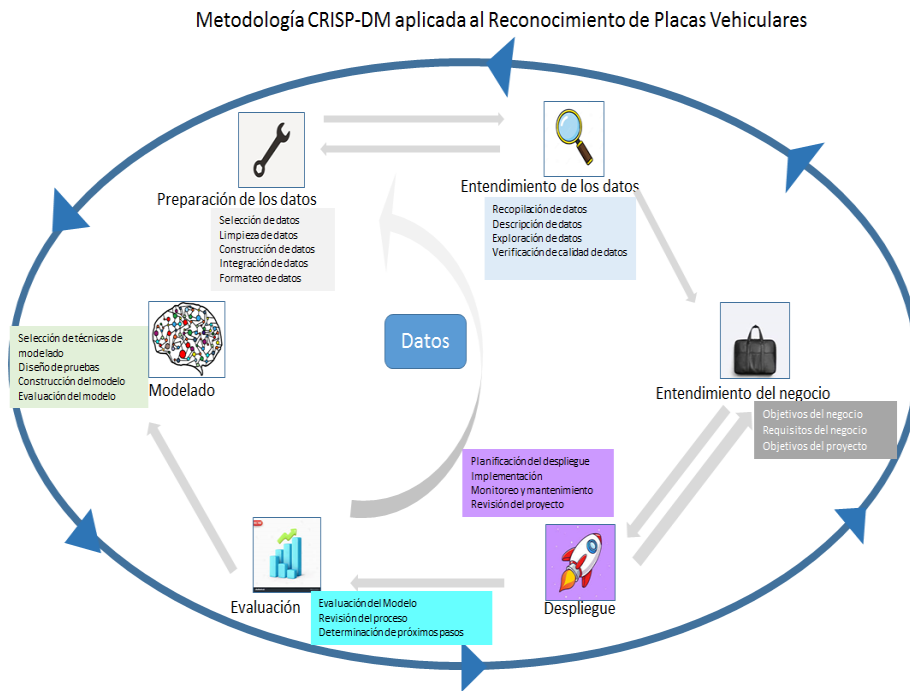
El desarrollo del sistema de reconocimiento de placas vehiculares se enmarca dentro de un ciclo de vida del software que comprende las siguientes etapas: análisis, diseño, implementación, pruebas, implantación y mantenimiento. Cada fase se vincula con la metodología CRISP-DM, garantizando la integración entre los procesos de ingeniería de software y la inteligencia artificial aplicada.

- Análisis: Identificación de requerimientos funcionales y no funcionales del sistema.
- Diseño: Arquitectura del sistema, diagramas de flujo, y base de datos en Excel.
- Implementación: Programación en Python con OpenCV, Tesseract OCR y macro en Excel.
- Pruebas: Evaluación de desempeño en diferentes condiciones ambientales.
- Implantación: Implementación del sistema en entorno real.
- Mantenimiento: Actualizaciones y mejora continua.

#### **6.5 Base de Datos Implementada en Macro Excel**

El sistema utiliza una macro en Microsoft Excel como base de datos funcional. Esta macro permite registrar, almacenar y consultar datos capturados por el sistema, incluyendo número de placa, propietario, hora de ingreso, hora de salida y tiempo total de permanencia. La macro automatiza cálculos y genera reportes que facilitan la gestión del flujo vehicular.

**Figura 6.**  
*Estructurado de la metodología.*



*Nota: En el grafico se observa el método con esta metodología para orientar los trabajos de minería de datos. Las descripciones de las fases de un proyecto y las tareas en cada fase, así como la indicación del flujo de una fase a otra, con una explicación de las relaciones entre tareas. (Cortines, 2015), tomado del Modelo Daimler Chrysler.*

## **7. ANALISIS Y PRESENTACION DE RESULTADOS.**

### **Contexto de la investigación**

El trabajo monográfico, se desarrolló para el aparcadero de la Universidad Nacional de Ingeniería, en el departamento de Managua, con un área aproximada de 1,262 mts<sup>2</sup>, con un espacio para 101 vehículos livianos y motocicletas. Considerando 2.5 mts para vehículo y 1.5 para motocicletas.

El aparcadero se encuentra controlado manualmente, mediante tabla con impresiones de hojas de block donde es anotado el número de matrícula y nombre del conductor en su ingreso; el egreso no es controlado a exactitud.

Actualmente no existe conocimiento de que hubiere algún sistema con inteligencia artificial en práctica o en ensayo. Al momento de esta investigación no se conoce algún sistema automatizado que tenga presencia en dicha institución.

### **Ventajas del sistema de reconocimiento de placa que se está proponiendo**

- Mejorar en la eficiencia del control de acceso
- Seguridad
- Reducción de costos operativos
- Mejora en la gestión del flujo vehicular
- Recopilación de datos para la toma de decisiones
- Accesibilidad para personas con discapacidades físicas

Si bien es cierto existen circunstancias benéficas a este sistema no se puede excluir las desventajas que podría presentarse.

- Problemas técnicos y de fiabilidad
- Altos costos iniciales de implementación
- Preocupaciones sobre la privacidad de los usuarios
- Resistencia al cambio por medio de usuarios como operarios / redistribución de tareas por parte del sujeto que actualmente maneja de forma manual el control
- Condiciones ambientales y mejora (gastos y costo) en la mejora de iluminación
- Mantenimiento y continua mejora

- Interrelación con otros programas en el caso que se pague el aparcadero
- Dependencia de la tecnología en este ámbito

### Técnicas e Instrumentos

En este trabajo se recurrió a enfoques más utilizados en la identificación de placas vehiculares, basándose en estudios previos y evaluando su aplicabilidad en los escenarios bajo la metodología CRISP-DM, y evidenciando la importancia de desarrollar métodos precisos y confiables para la detección y reconocimiento de placas vehiculares.

Para esto se fundamentó en el uso de instrumentos tecnológicos desde la observación digital, e informe extraído de la base de datos como resultado; tales como la cámara TP LINK VIGI exterior 3MP Bullet C3HP



Fuente: CICSA, Managua

Tabla 1: Características de la cámara.

Lente	6 mm
Detección	Movimiento, intrusión de área, cruce de línea,
Imagen	IR, WDR, 3d DNR
Visión	Día y noche robusto sistema de visión nocturna
Software	Compatible a todo sistema

Sistemas operativos	Compatible a Windows 10, Windows 8, Windows 7 MAC
Requisitos del sistema	CPU mínima = Core 2 dúo a 2,4 GHz CPU recomendada = Quad core a 3,06 GHz RAM mínima = 2 GB RAM recomendada = 4 GB

*Fuente: CICSA, Nicaragua*

### **Características de la computadora**

El procesador, tiene las siguientes características:

<b>Procesador</b>	Intel Core i5-2520M CPU 2.50 GHz
<b>Memoria RAM</b>	8,00 GB
<b>Tipo de Sistema operativo</b>	Windows 10 x 64 bits
<b>Monitor</b>	LED 20" LG 1080 px

### **Sistema operativo**

Microsoft Windows 10 con arquitectura de 64 bits

### **Software**

Microsoft visual studio code: es un editor de código fuente que permite a los desarrolladores escribir, depurar y gestionar código. Funciona en el escritorio y está disponible para Windows, macOS y Linux.

### **Lenguaje de programación**

Python: es un lenguaje de programación interpretado que funciona leyendo el código línea por línea. Esto facilita su depuración y uso. Para que el código Python se ejecute, el intérprete de Python lo traduce a instrucciones de bajo nivel que la CPU pueda entender

## **Biblioteca**

Opencv: Es una biblioteca libre de visión artificial desarrollada por Intel. Contiene más de 500 algoritmos con multiplicidad de funciones en el procesamiento de imágenes, descripciones geométricas, segmentación, reconocimiento facial, calibración de cámaras, visión estéreo visión robótica. Con algunas dependencias apoyándose con Numpy para manipular matrices y realizar operaciones matemáticas. Y Matplotlib Utilizado para visualizar imágenes procesadas en entornos interactivos.

## **Área de aplicabilidad**

El área de aplicabilidad es la seguridad y vigilancia, tratando lo que es la movilidad del tránsito de vehículos, transporte y logística.

## **Presentación de los resultados de acuerdo a la Metodología CRISP-DM.**

Se hizo uso de las etapas propuesta por la metodología para presentar los resultados de cada de sus fases, las cuales se detalla a continuación:

### **Fase 1: Preparación de los datos**

#### **A. Selección de datos**

En este segmento del trabajo monográfico, se preparó la fase de adquisición y procesamiento de datos conforme la metodología establecida, para ello, se ha definido de la selección del vehículo, el software, la biblioteca OpenCV (The Open Computer Vision), las librerías de apoyo así como componentes tecnológicos necesarios, incluyendo el servidor, la cámara y el monitor; el proceso de obtención de datos dio inicio desde el contacto de la imagen con la cámara, luego su identificación y posterior tratamiento. Según (Arévalo et al., 2016), la librería OpenCV permite el procesamiento de imágenes, así como su tratamiento pudiendo desarrollar aplicaciones de visión por computador. La entrada fue una imagen RGB, en formato PNG. En la imagen de la interfaz siguiente, se importan todas las librerías, y se crea una base de datos en Excel.

Para esto se realizó la programación de la interfaz para preparar datos a importarse dirigiéndose a preparar base de datos en Excel de acuerdo a la figura 7. Aquí se extrae el texto de la placa con pytesseract y guarda los datos en un archivo Excel.

### Figura 7.

*Algoritmo para preparar datos a importarse a base de datos en Excel*

```
1 import cv2
2 import pytesseract
3 import pandas as pd
4 import os
5
6 # Configuración de Tesseract (asegúrate de que esté instalado y configurado)
7 pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe' # Ajusta esta ruta en Wind
8
9 # Directorio de imágenes de placas (ajústalo a tu ruta)
10 directorio_imagenes = "imagenes_placas/"
11
12 # Cargar clasificador preentrenado de OpenCV para detectar placas
13 placa_cascade = cv2.CascadeClassifier(cv2.data.harcascades + "haarcascade_russian_plate_number.xml")
14
15 datos_placas = []
16
17 for archivo in os.listdir(directorio_imagenes):
18     if archivo.endswith(".jpg") or archivo.endswith(".png"):
19         ruta_imagen = os.path.join(directorio_imagenes, archivo)
20         imagen = cv2.imread(ruta_imagen)
21
22         # Convertir a escala de grises
23         gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
24
25         # Detectar placas
26         placas = placa_cascade.detectMultiScale(gris, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
27
28         for (x, y, w, h) in placas:
29             placa = imagen[y:y+h, x:x+w]
30             texto_placa = pytesseract.image_to_string(placa, config='--psm 8')
31
32             # Guardar datos en la lista
33             datos_placas.append([archivo, texto_placa.strip()])
34
35 # Crear DataFrame y exportar a Excel
36 df = pd.DataFrame(datos_placas, columns=["Imagen", "Placa Detectada"])
37 df.to_excel("placas_detectadas.xlsx", index=False)
38
39 print("Proceso completado. Datos exportados a 'placas_detectadas.xlsx'")
40
```

La simulación continuó con la cámara captando el auto, Kia Sportage con placa del departamento de León. Utilizándose cámara TP LINK VIGI exterior 3MP Bullet C3HP. En su captación arrojó imágenes claras identificando y visualizados los números y letras y que desde el inicio se alista para ser exportados los datos a una base de datos en Excel, los cuales serían la identificación de los campos seleccionados como:

- Automóviles todo tipo, marca, color y diseños: ID marca
- Nombre del propietario: ID Propietario
- Hora de ingreso: ID hora de ingreso
- Hora de egreso: ID hora de egreso
- Categoría: ID docente/administración/estudiante/proveedor

Aquí los datos fueron procesados y lanzados al servidor a través de OpenCV donde se envió mascarar simuladoras. Posteriormente el sistema manda a buscar conexión con una cámara IP. Tal y como lo señala la figura 8. Es escaneada la red para encontrar dispositivos con características de cámaras IP, donde se muestra las IP detectadas en una lista, luego permitió seleccionar una cámara y abrir su transmisión en vivo, donde se utilizó tkinter como parte de las librerías de apoyo para la interfaz gráfica.

**Figura 8.**

*Interfaz buscando vincularse a cámara IP*

```
45 # Interfaz para buscar cámaras IP
46 def escanear_red():
47     ip_red = "192.168.1.0/24" # Ajusta el rango según tu red
48     dispositivos = []
49
50     arp = ARP(pdst=ip_red)
51     ether = Ether(dst="ff:ff:ff:ff:ff:ff")
52     paquete = ether / arp
53     resultado = srp(paquete, timeout=3, verbose=False)[0]
54
55     for enviado, recibido in resultado:
56         ip = recibido.psrc
57         try:
58             respuesta = requests.get(f"http://{ip}", timeout=2)
59             if "camera" in respuesta.text.lower():
60                 dispositivos.append(ip)
61         except:
62             pass
63
64     lista_camaras.delete(0, tk.END)
65     for cam in dispositivos:
66         lista_camaras.insert(tk.END, cam)
67
68 def conectar_camara():
69     seleccion = lista_camaras.get(tk.ACTIVE)
70     if seleccion:
71         cap = cv2.VideoCapture(f"rtsp://{seleccion}/stream")
72
73         while True:
74             ret, frame = cap.read()
75             if not ret:
76                 break
77             cv2.imshow("Cámara IP", frame)
78             if cv2.waitKey(1) & 0xFF == ord('q'):
79                 break
80             cap.release()
81             cv2.destroyAllWindows()
82     else:
83         messagebox.showerror("Error", "No se ha seleccionado ninguna cámara")
84
85 # Interfaz gráfica
86 root = tk.Tk()
87 root.title("Buscador de Cámaras IP")
88 root.geometry("400x300")
89
90 btn_scan = tk.Button(root, text="Escanear Red", command=escanear_red)
91 btn_scan.pack(pady=10)
92
93 lista_camaras = tk.Listbox(root)
94 lista_camaras.pack(pady=10, fill=tk.BOTH, expand=True)
95
96 btn_connect = tk.Button(root, text="Conectar Cámara", command=conectar_camara)
97 btn_connect.pack(pady=10)
98
99 root.mainloop()
```

## **La limpieza de datos**

Aquí se inició el proceso de algoritmos para ver la resolución, ángulos, excluyendo imágenes borrosas de baja calidad, descartando algunas imágenes y anotado de placas. Tal es el caso de las imágenes abajo donde se identifica la imagen original, imagen borrosa, escala de grises y ya limpia la imagen es encuadrada la placa vehicular. En la limpieza de datos: las escalas a remarcar, los bordes tricolor rojo, verde y azul son transformados por los valores dados. Y son reducidos a un solo color verde que es el válido.

Aquí se ha capturado la imagen, desde la cámara TP LINK VIGI exterior 3MP Bullet C3HP y se procesó el conjunto de datos para cargar la imagen. De acuerdo a la carga de imagen se utiliza la biblioteca Opencv en Python para leer la imagen de acuerdo a la figura 9, y que posteriormente se presenta el algoritmo para el logro de la captación de imagen de acuerdo a la figura 10.

### **Figura 9.**

*Vehículo siendo captado por la cámara, y preparación de imagen*



## Figura 10.

### Algoritmo de carga de imagen original del vehículo

```
1 import cv2
2 import numpy as np
3 from tkinter import Tk, Button, Label, filedialog
4 from PIL import Image, ImageTk
5
6 # Cargar el clasificador de placas
7 placa_cascade = cv2.CascadeClassifier(cv2.data.harcascades + "haarcascade_russian_plate_number.xml")
8
9 def cargar_imagen():
10     global imgTk, img_cv
11     ruta_imagen = filedialog.askopenfilename(filetypes=[("Imágenes", "*.jpg;*.png;*.jpeg")])
12     if ruta_imagen:
13         img_cv = cv2.imread(ruta_imagen)
14         img_cv = cv2.cvtColor(img_cv, cv2.COLOR_BGR2RGB)
15         img = Image.fromarray(img_cv)
16         img = img.resize((500, 400))
17         imgTk = ImageTk.PhotoImage(img)
18         label_img.config(image=imgTk)
19
20 def detectar_placa():
21     global imgTk, img_cv
22     if img_cv is None:
23         return
24
25     img_gray = cv2.cvtColor(img_cv, cv2.COLOR_RGB2GRAY)
26     placas = placa_cascade.detectMultiScale(img_gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
27
28     for (x, y, w, h) in placas:
29         cv2.rectangle(img_cv, (x, y), (x + w, y + h), (0, 255, 0), 2)
30
31     img = Image.fromarray(img_cv)
32     img = img.resize((500, 400))
33     imgTk = ImageTk.PhotoImage(img)
34     label_img.config(image=imgTk)
35
36 # Configuración de la interfaz gráfica
37 root = Tk()
38 root.title("Reconocimiento de Placas Vehiculares")
39
40 btn_cargar = Button(root, text="Cargar Imagen", command=cargar_imagen)
41 btn_cargar.pack()
42
43 btn_detectar = Button(root, text="Detectar Placa", command=detectar_placa)
44 btn_detectar.pack()
45
46 label_img = Label(root)
47 label_img.pack()
48
49 root.mainloop()
50
```

En esta parte se ha creado una interfaz de tipo aplicativo, agregándose los botones que posee y todo lo que presenta la imagen de placas.

Para esto se debió de cargar la imagen, convirtiéndola a escala de grises, para ir depurando lo que se iba a identificar. En esta figura se ha identificado la calidad de las imágenes (resolución, condiciones de iluminación, ángulos de las cámaras) las que se debió excluir imágenes borrosas de baja calidad e irrelevantes. Procediendo a la eliminación de ruidos, preparándose para escala de grises. No obstante se hizo uso de filtro Gaussiano para suavizar la imagen y eliminar imperfecciones. En esta siguiente parte se redujo la complejidad de la imagen eliminando la información de color de acuerdo a la figura 11. Procesando la imagen la cual se presenta tal cual en la figura 12.

**Figura 11.**

*Algoritmo, eliminando ruidos y preparación de escala de grises*

```
import cv2
import numpy as np

# Cargar La imagen
imagen = cv2.imread("placa_vehicular.jpg")

# Convertir a escala de grises
imagen_gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)

cv2.imshow("Imagen en Escala de Grises", imagen_gris)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Procesada a imagen de acuerdo al algoritmo en la figura 12, se observa de la siguiente manera tal y como aparece la figura 13..

**Figura 12.**

*Imagen del vehículo en escala grises*



```
imagen_suavizada = cv2.GaussianBlur(imagen_gris, (5,5), 0)

cv2.imshow("Imagen Suavizada", imagen_suavizada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Posteriormente se dio el ajuste de contraste mejorando la visibilidad de los caracteres mediante ecualización de histograma. Y es aquí donde se hace la imagen ecualizada, resultando la imagen suavizada.

*imagen\_ecualizada = cv2.equalizeHist(imagen\_suavizada)*

### B. En la construcción de datos

Se vincularon los procesos de reconocimiento de placas de diferentes tamaños y además en coherencia con los distintos procedimientos hasta la obtención de un producto final. El proceso de extracción de la minería de datos y migración cumple los requerimientos y estructura necesaria en la codificación de origen y destino trazado.

## Figura 14.

*Algoritmo detectando a una función específica, etiquetando al tipo de vehículo*

```
7
8 # Función para identificar propietario y vehículo
9 v def identificar_vehiculo(placa):
10     placa = placa.upper().strip() # Convertimos a mayúsculas y eliminamos espacios
11 v     if placa in vehiculos:
12         datos = vehiculos[placa]
13         return f"Propietario: {datos['propietario']}, Vehículo: {datos['vehiculo']}"
14 v     else:
15         return "Placa no encontrada en la base de datos."
16
17 # Solicitar entrada al usuario
18 v if __name__ == "__main__":
19     placa_ingresada = input("Ingrese el número de placa: ")
20     print(identificar_vehiculo(placa_ingresada))
21
```

Lo que viene a responder a la visualización en pantalla del vehículo ya integrado a la base de datos. En este caso se puede ir agregando para este trabajo monográfico el login que se presenta el ingreso con ID placa; ID propietario; ID tipo de vehículo, para posteriormente si los datos están correctos se registra de acuerdo a la figura 26

En esta primera fase de preparación de datos, se finaliza con el formateo de los datos, en el servidor, existe la corrección de las consistencias identificadas, a través de algoritmos, detallados alimentando la base de datos en Excel del registro de las placas representativas de los vehículos identificados, para este trabajo monográfico solamente el vehículo de muestra.

### C. Integración de datos

En la integración de datos se ha podido determinar que toda la imagen que se ha adquirido, se ha venido realizado la detección de la imagen y se ha iniciado el procesamiento de esa imagen, comparándose con el esquema inicial de la metodología, según tabla 1.

**Tabla 2.**

*Comparación en la metodología CRISP-DM Vs el esquema de la propuesta de identificación vehicular*

Metodología CRISP DM	Esquema propuesto para la identificación de placa vehicular
I fase:  Selección de datos  Limpieza de datos  Construcción de datos  Integración de datos  Formateo de datos	<pre>graph LR; A((Adq. de image)) --&gt; B((Procesamiento)); B --&gt; C((Detec img));</pre>

#### D. Formateo de datos

En esta fase se cumple la estandarización de los datos, luego la conversión del tipo de datos, eliminación de consistencias, o corrección de las mismas y la normalización o ajuste de los datos a un formato común, como la conversión de cadenas o números representados; eliminación de espacios y caracteres innecesarios; o, conversión de datos textuales a categorías

Al darse la carga de la imagen en la preparación de datos, se ha utilizado la biblioteca OpenCV en Python, se ajusta el contraste mejorando la visibilidad de los caracteres mediante ecualización, Se realiza la conversión de la Binarización de OCR para mejorar la detección de caracteres que además de la detección de bordes, se aplicó el operador Canny para resaltar los bordes de los caracteres. Realizándose una segmentación de la placa.

## Figura 15.

*Algoritmo detectando bordes, convirtiendo en color verde y preparando la detección de la placa*

```
1 import cv2
2 import numpy as np
3
4 def detectar_bordes(ruta_imagen):
5     # Cargar la imagen
6     imagen = cv2.imread(ruta_imagen)
7     gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
8
9     # Aplicar desenfoque para reducir ruido
10    gris = cv2.GaussianBlur(gris, (5, 5), 0)
11
12    # Aplicar detección de bordes con Canny
13    bordes = cv2.Canny(gris, 50, 150)
14
15    # Convertir bordes a imagen en color
16    bordes_coloreados = cv2.cvtColor(bordes, cv2.COLOR_GRAY2BGR)
17    bordes_coloreados[np.where((bordes_coloreados == [255, 255, 255]).all(axis=2))] = [0, 255, 0] # Resalta
18
19    return imagen, bordes_coloreados
20
21 def detectar_placa(ruta_imagen):
22    imagen, bordes = detectar_bordes(ruta_imagen)
23
24    # Encontrar contornos
25    contornos, _ = cv2.findContours(cv2.cvtColor(bordes, cv2.COLOR_BGR2GRAY), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
26
27 for contorno in contornos:
28     perimetro = cv2.arcLength(contorno, True)
29     approx = cv2.approxPolyDP(contorno, 0.02 * perimetro, True)
30
31     # Si el contorno tiene 4 lados, puede ser una placa
32     if len(approx) == 4:
33         x, y, w, h = cv2.boundingRect(approx)
34         placa = imagen[y:y+h, x:x+w]
35         cv2.rectangle(imagen, (x, y), (x + w, y + h), (0, 255, 0), 2)
36         return imagen, bordes, placa
37
38     return imagen, bordes, None
39
40 # Ruta de la imagen a analizar
41 ruta_imagen = "placa_vehiculo.jpg"
42
43 # Ejecutar detección de placa
44 imagen_resultado, bordes_resaltados, placa_extraida = detectar_placa(ruta_imagen)
45
46 # Mostrar la imagen con los bordes resaltados y la detección de placa
47 cv2.imshow("Detección de Bordes", bordes_resaltados)
48 cv2.imshow("Detección de Placa", imagen_resultado)
49 cv2.waitKey(0)
50 cv2.destroyAllWindows()
51
```

Lo que dio como resultado la señalización de bordes en el vehículo en color verde de acuerdo a la figura 17.

**Figura 16.**

*Imagen del vehículo, con detección de bordes, apoyado en Canny*



El algoritmo se programó con apoyo de Canny, aplicando la umbralización para la Binarización OCR, y guardando esa imagen binarizada.

**Figura 17.**

*Algoritmo que detecta bordes, Binarización OCR, y guarda imagen binarizada*

```
# Detección de bordes con Canny
bordes = cv2.Canny(imagen_suavizada, 100, 200)

# Aplicar umbralización para binarización OCR
_, imagen_binaria = cv2.threshold(bordes, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

# Mostrar resultados
cv2.imshow("Imagen Binaria OCR", imagen_binaria)
cv2.waitKey(0)
cv2.destroyAllWindows()

# Guardar la imagen binarizada
cv2.imwrite("placa_binaria.png", imagen_binaria)
```

**Figura 18.**

*Imagen con la detección de borde de la placa. Guardada*



**Figura 19.**

*Algoritmo de la imagen procesada, devolviendo la ruta de esa imagen procesada*

```
# Guardar La imagen procesada
imagen_bordes_path = "/mnt/data/kia_sportage_bordes.png"
cv2.imwrite(imagen_bordes_path, bordes)

# Devolver La ruta de La imagen procesada
imagen_bordes_path
```

En la figura 18 el algoritmo ya refleja la identificación de la placa, la binariza, y muestra la placa; apoyándose en OpenCV y se muestra el umbral de la escala de grises lo que quedaría de la siguiente forma:

## Figura 20.

### Binarización de la placa, y extracción de datos

```
import cv2
import numpy as np

def binarizar_placa(imagen_placa):
    # Convertir la imagen a escala de grises
    gris = cv2.cvtColor(imagen_placa, cv2.COLOR_BGR2GRAY)

    # Aplicar umbral binario
    _, binarizada = cv2.threshold(gris, 120, 255, cv2.THRESH_BINARY)

    return binarizada

# Cargar la imagen de la placa extraída
ruta_placa = "placa_extraida.jpg"
imagen_placa = cv2.imread(ruta_placa)
```

```
# Binarizar la placa
imagen_binarizada = binarizar_placa(imagen_placa)

# Mostrar la imagen binarizada
cv2.imshow("Placa Binarizada", imagen_binarizada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

## Fase 2: Entendimiento de los datos

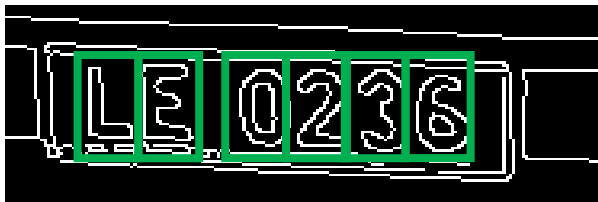
Para esta segunda fase de la metodología, ha sido comparado con el esquema dando lugar a la segmentación que es donde la imagen se fracciona en distintas partes para su análisis o validación.

### A. Recopilación de datos

Una vez segmentada, la placa es separada y extraída, han sido identificados los caracteres, corrigiendo los ángulos de la placa, que en OpenCV se logró mediante la función `getRotationMatrix2D` de acuerdo a la figura 22.

#### Figura 21.

*Placa detectada y caracteres identificados*



Posteriormente, los caracteres identificados y adelgazados de forma individual, son barridos de forma sucesiva, manteniendo su tipología y proporcionalidad, evitando las deformaciones.

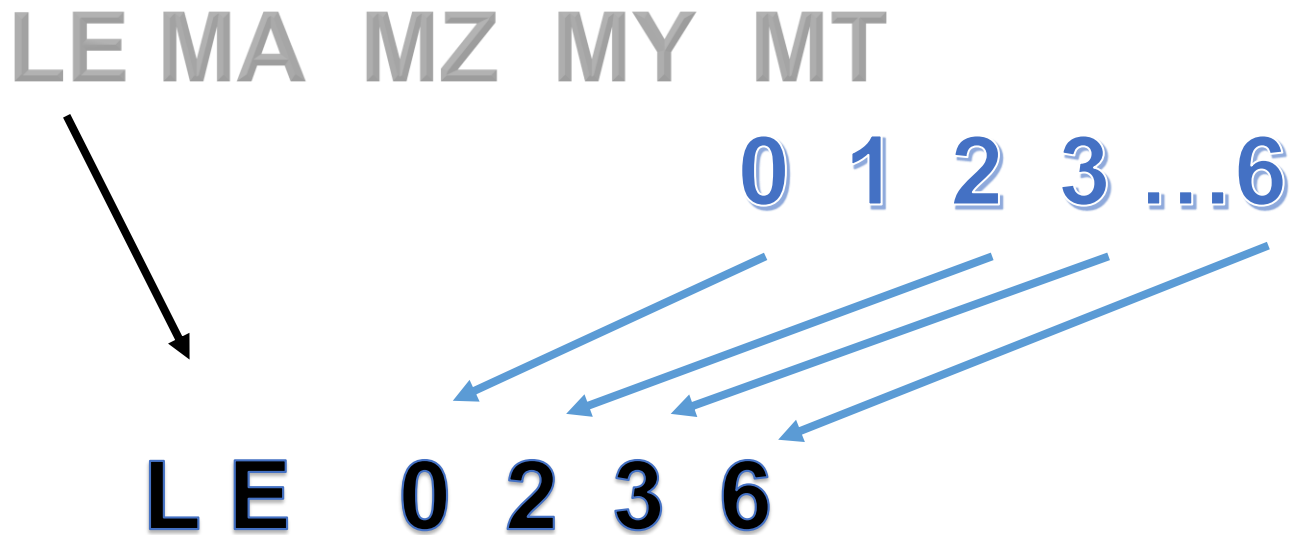
#### Figura 22.

*Imagen adelgazada con caracteres individuales*



### B. Descripción de los datos

En la descripción de los datos, y de acuerdo a (Navarrena, 2017), los caracteres descritos se empiezan a comparar con patrones establecidos ya en una base de datos. Que para este trabajo monográfico se establece con MySQL, con el método de proyecciones verticales y horizontales con la del patrón establecido, y aquí es donde se describió el carácter con mayor coincidencia. Por ejemplo



Donde las siglas en mayúsculas superior izquierda representan las dos primeras letras de cada departamento; caracteres numéricos guardados del 0 al 9 para su comparación, buscando similitud para luego realizar la identificación.

Para este caso, dentro del aislamiento de cada carácter se aplicó el reconocimiento óptico de caracteres (OCR), específico para placas vehiculares.

### C. Exploración de los datos

En la exploración de los datos o caracteres que encuentra un patrón, la identifica con una imagen y posteriormente extrae las características, al compararse con una forma similar al del patrón, se deduce de qué se trata de un número correcto. Para esto se apoyó en el algoritmo para enviar datos a través de configuración de rutas, para que estos sean guardados en una base entendible al humano en Excel en un servidor Windows. El cual se expresa en la figura 24.

### Figura 23.

*Algoritmo para exportar datos entendibles al humano en Excel para su posterior verificación de calidad de datos*

```
27
28 # Función para guardar los datos en Excel en un servidor Windows
29 v def guardar_en_excel(datos, ruta_salida):
30     df = pd.DataFrame(datos)
31     df.to_excel(ruta_salida, index=False)
32     print(f"Datos guardados en: {ruta_salida}")
33
34 # Configuración de rutas
35 directorio_imagenes = "C:\\ruta\\a\\imagenes"
36 ruta_excel = "C:\\ruta\\del\\servidor\\placas.xlsx"
37
38 # Ejecución del proceso
39 datos_extraidos = procesar_imagenes(directorio_imagenes)
40 guardar_en_excel(datos_extraidos, ruta_excel)
41
```

### Verificación de la calidad de los datos

El sistema exitoso fue verificado una vez segmentados y reconocidos los caracteres, con la base de datos ya elaborada mediante interfaz inicial que fue explicada en la preparación de datos, al importarse a Excel para el entendimiento humano leyéndose de acuerdo a la tabla 2.

### Tabla 3.

*Datos verificados, de acuerdo a la ejecución del proceso de la interfaz inicial, señala los datos extraídos de placas vehiculares*

Fecha	No. Placa	Marca	Propietario	Hora ingreso	Hora egreso
04/03/2025	LE 0236	Kia Sportage	Yeferson Berrios	08:30	12:30
05/03/2025	MY 1485	Toyota Yaris	Alexandra Dimas	11:00	14:00
06/03/2025	MA 6324	Switf Suzuki	Alejandro Peña	14:00	17:25

Aquí la consistencia de los valores, debe ser crítica, ya que se debe de garantizar la veracidad, o la necesidad de corrección de datos en los algoritmos.

### Fase 3: Entendimiento del negocio

#### A. Objetivo del negocio

En esta parte del trabajo monográfico, se estableció la capacidad de convertir todo el proceso adquirido en un problema de minería de datos, el cual su propósito es la base de datos en Excel, identificando las placas vehiculares. Que ingresan a determinado aparcadero, el cual puede:

- Garantizar seguridad
- Aviso del que no esté registrado
- Control administrativo en caso que se realice una anexión del sistema para cobro de utilización de parqueo

#### B. Requisitos del negocio

Para esto se ha tenido que realizar una ventana Login, en donde se obtuvo la identificación, del que ingresa, como propietario del vehículo, número de placa, marca del vehículo, ciudad, el cual estaría siendo contrastada en la base de datos. Si los datos son correctos aparecería en la pantalla del monitor su identificación de acuerdo a la figura 25, contrario a ello, la ventana arrojará un error según figura 26.

#### Figura 24.

*Algoritmo para identificar al vehículo que ingresa al aparcadero*

```
html
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Identificación / Login</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f5f5f5;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }
  </style>
</head>
</html>
```

```

}

.card {
  background: #fff;
  width: 300px;
  padding: 20px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.2);
  border-radius: 8px;
  text-align: center;
}

.card img {
  width: 100px;
  height: 100px;
  object-fit: cover;
  border-radius: 50%;
  margin-bottom: 15px;
}

.card h2 {
  margin: 10px 0;
  font-size: 18px;
  color: #333;
}

}

.info {
  text-align: left;
  margin: 0 auto;
  width: 90%;
  font-size: 14px;
  line-height: 1.5;
}

.info p {
  margin: 6px 0;
}

.info strong {
  color: #555;
}
</style>
</head>
<body>

<div class="card">
  <!-- Reemplaza 'foto.jpg' con la ruta o nombre real de la imagen adjunta -->
  
  <h2>Identificación</h2>
  <div class="info">
    <p><strong>Nombre:</strong> [Aquí el nombre que desees]</p>
    <p><strong>Propietario vehículo:</strong> Kia Sportage</p>
    <p><strong>Placa:</strong> LE 0236</p>
    <p><strong>Ubicación:</strong> Estudiante Sistemas</p>
  </div>
</div>

</body>
</html>

```

**Figura 25.**

*Login de ingreso al aparcadero una vez identificada la placa del vehículo aparece en monitor su respectiva identificación*



**Figura 26.**

*Ventana de error, ingreso no registrado y ausente en base de datos*



### C. Objetivos del proyecto/negocio

En el marco del reconocimiento de placas vehiculares, se torna de mucha importancia el desarrollar el sistema eficiente y preciso, permitiendo la identificación automática de matrículas en distintas condiciones ambientales, tal como se ha venido realizando en las interfaces aplicadas. Por tanto, el objetivo de la gestión del monitoreo del tráfico y además de la seguridad dentro del aparcadero, se fortalece por la optimización del flujo vehicular y minimización de errores de reconocimiento.

Dado el segundo objetivo específico de este trabajo monográfico, el diseño del prototipo de la interfaz, se demuestra con la visualización de la pantalla con los datos generales del vehículo que ingresa de acuerdo a la figura 25. Además del panel de administrador, que está en constante visualización con la cámara y donde consultar, sumado a los informes y estadísticas en tiempo real. En donde se ha permitido filtrar el tipo de vehículo que se desee buscar.

### **Fase 4: Despliegue**

#### A. Planificación del despliegue

En esta etapa, se ha aplicado la forma del funcionamiento de todo el proceso donde en el caso de la universidad, percibe de forma clara los objetivos e implementación, así como la ejecución de la identificación de placas vehiculares, además la implementación del control teniendo como soporte las estrategias de mantenimiento y recomendaciones a mejora continua. No obstante, se debe garantizar que todo individuo que ingrese en su vehículo debe facilitar la cedula de identidad y tarjeta de circulación del vehículo.

#### B. Implementación y monitores

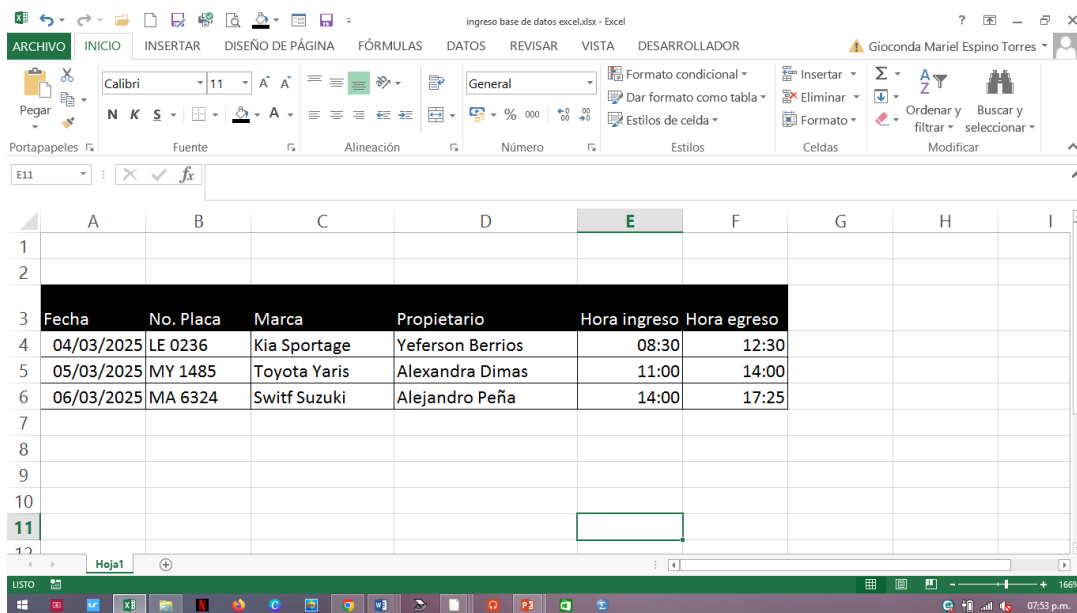
La explicación de un entorno real, conllevó la ejecución y equipos de monitores, para su visualización en tiempo real; y donde se integraron toda la base de datos y los modelos de reconocimiento, permitiendo una administración clara.

### C. Monitoreo y mantenimiento

En esta fase se permitió conocer que la validación, de forma permanente como un sistema de seguimiento prioritario e imprescindible de forma cíclica lo que garantizo además la aprobación de responsables quienes deben revisar que los reportes estén correctamente documentados como evidencia. Se ha sustentado en correos internos para la comunicación y solución inmediata a través de Gmail.com

**Figura 27.**

*Reconocimiento de controles en Excel*



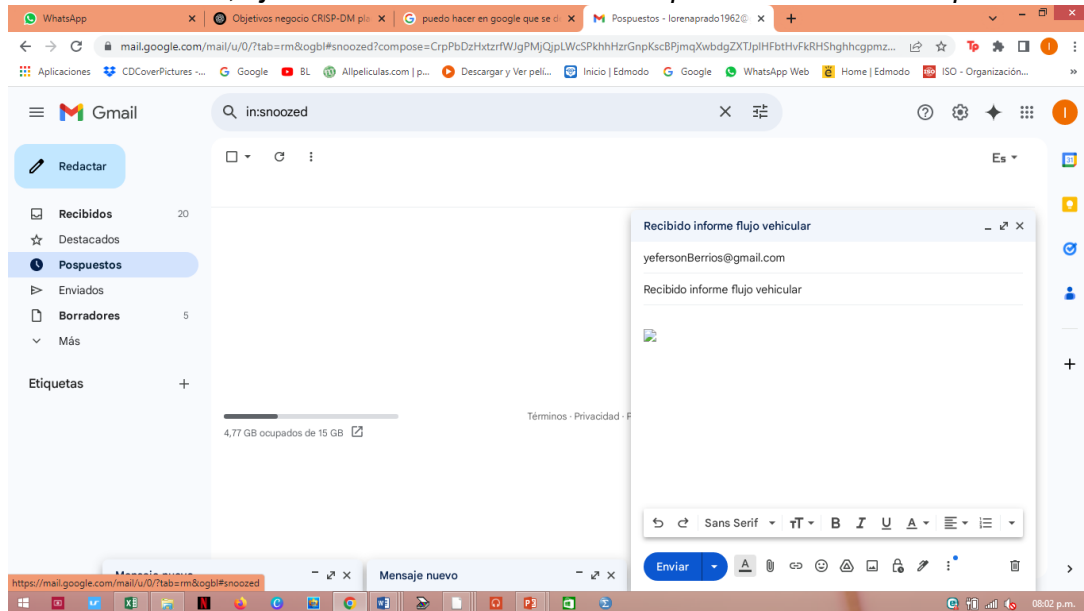
The screenshot shows an Excel spreadsheet with the following data:

Fecha	No. Placa	Marca	Propietario	Hora ingreso	Hora egreso
04/03/2025	LE 0236	Kia Sportage	Yeferson Berrios	08:30	12:30
05/03/2025	MY 1485	Toyota Yaris	Alexandra Dimas	11:00	14:00
06/03/2025	MA 6324	Swift Suzuki	Alejandro Peña	14:00	17:25

Mientras que en la figura 29, se envía la respuesta del informe recibido a través de correo electrónico interno.

**Figura 28.**

*Correo electrónico, ejecutando automáticamente respondiendo al informe para su revisión*



#### D. Revisión del proyecto

En esta parte la revisión a través de Excel y el Dashboard son las concordancias enlazadas, para la revisión y toma de decisiones. Donde claramente en Excel se observan los datos de los que ingresan al aparcadero.

### **Fase 5: Evaluación**

#### A. Evaluación del modelo

Como criterio de cumplimiento al tercer objetivo específico de desarrollar la propuesta de realización de informes, y manera de seguimiento para su evaluación del modelo desarrollado; se presenta el algoritmo que se encargara, de la precisión, ingreso de vehículo, el tiempo de identificación de la placa. Que se puede anexar al informe de Excel y el Dashboard, de acuerdo a la figura 29 y 30.

## Figura 29.

Algoritmo para evaluar todo el modelo de visualización e identificación de placas vehiculares

```
import cv2
import pytesseract
import time
from sklearn.metrics import precision_score, recall_score, f1_score
import numpy as np

# Configura el path de Tesseract
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe' # Can

def preprocess_image(image_path):
    """
    Preprocesa la imagen para mejorar la detección de placas.
    """
    # Cargar imagen en escala de grises
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    # Aplicar desenfoque para reducir ruido
    img = cv2.GaussianBlur(img, (5, 5), 0)
```

```
# Aplicar umbralización
_, img = cv2.threshold(img, 150, 255, cv2.THRESH_BINARY)
return img

def extract_plate_text(image):
    """
    Extrae el texto de una imagen utilizando Tesseract.
    """
    text = pytesseract.image_to_string(image, config='--psm 8') # El valor de psm 8 es ideal
    return text.strip()

def evaluate_model(true_labels, predicted_labels):
    """
    Calcula las métricas de precisión, recall y f1-score.
    """
    precision = precision_score(true_labels, predicted_labels, average='binary')
    recall = recall_score(true_labels, predicted_labels, average='binary')
```

```

f1 = f1_score(true_labels, predicted_labels, average='binary')
return precision, recall, f1

```

Copiar Editar

```

def main():
    # Lista de imágenes de prueba y sus etiquetas (planchas correctas)
    image_paths = ['test_image1.jpg', 'test_image2.jpg', 'test_image3.jpg'] # Agrega las rutas
    true_labels = ['ABC123', 'XYZ789', 'DEF456'] # Las etiquetas correctas (números de placa)

    predicted_labels = []
    start_time = time.time() # Marca el inicio del tiempo

    for image_path, true_label in zip(image_paths, true_labels):
        img = preprocess_image(image_path)
        predicted_text = extract_plate_text(img)
        predicted_labels.append(predicted_text)
        print(f'Placa Predicha: {predicted_text}, Placa Real: {true_label}')

```

```

end_time = time.time() # Marca el fin del tiempo
total_time = end_time - start_time
print(f'\nTiempo total de identificación: {total_time:.2f} segundos')

# Evaluar precisión, recall y F1-score
precision, recall, f1 = evaluate_model(true_labels, predicted_labels)
print(f'\nPrecisión: {precision:.2f}')
print(f'Recuerdo: {recall:.2f}')
print(f'F1-Score: {f1:.2f}')

if __name__ == "__main__":
    main()

```

Para el Dashboard, se ha generado el siguiente reflejado en la figura 30.

## Figura 30.

Algoritmo para la elaboración de Dashboard de identificación de placas vehiculares

```
def main():
    st.title("Dashboard de Identificación de Placas Vehiculares")

    # Subir archivo de imagen
    uploaded_file = st.file_uploader("Sube una imagen para identificar la placa", type=["jpg"])
    if uploaded_file is not None:
        # Mostrar imagen subida
        st.image(uploaded_file, caption="Imagen subida", use_column_width=True)

        # Preprocesar imagen y extraer texto
        img = preprocess_image(uploaded_file)
        predicted_text = extract_plate_text(img)

        # Mostrar texto identificado
        st.write(f"Texto Identificado: {predicted_text}")
```

### B. Revisión del proceso

En esta fase se permitió a los tomadores de decisiones, revisar, toda la metodología implementada, el cual trasmite un ciclo de mejoras en cada proceso, aterrizado al tratamiento de minería de datos a través de placas vehiculares, y que de acuerdo al sitio que está destinado podrán ser asistidos por computadora e inteligencia artificial para poder procesar y explorar esa parte.

### C. Determinación de próximos pasos

En esta fase, fue sencilla posterior a la revisión del proceso, se pudo validar y decidir que se tiene que hacer o que mejoras realizar, para la consecución de los objetivos propuestos por la institución académica.

## **Fase 6: Modelado**

### A. Selección de técnicas del modelado

Las técnicas del modelado para la identificación de placas vehiculares están dadas por la adquisición de la imagen, procesamiento de imagen, extracción de características, en el uso de estas técnicas que se ha apoyado por el uso de reconocimiento de caracteres a través de tesseract, camy, OpenCV, evaluación de

modelos sobre las métricas de tiempo de ingreso, número o cantidad de vehículos entre otros.

### B. Diseño de Pruebas

Se realizó su robustez y seguridad, así como la identificación de errores en tiempo real.

Para esto fue necesario crear el siguiente diseño en pasos:

1. Cargar el conjunto de datos de prueba
2. Para cada imagen en el conjunto de datos;
  - a. Aplicar el modelo de detección de placas
  - b. Comparar la placa detectada con la etiqueta real
  - c. Registrar si la detección fue correcta o incorrecta
3. Generar informe con los resultados y posibles mejoras

Así mismo se confirmó mediante la generación de datos de prueba, referido al resumen de cinco ventanas, el cual da por seguro el reconocimiento de los que ingresaron y fueron detectados e identificados de acuerdo a la figura 32. Permitiendo la generación de informe pudiéndose conocer número de vehículos, marcas, y permanencia en el aparcadero.

### **Figura 31.**

*Diseño de la prueba, referido al control vehicular*

CONTROL VEHICULAR UNI Del 04/03/2025 al 10/03/2025					
Placas detectadas	Propietario	Marca	Permanencia	H /I	H/E
LE 0236 MY 1485 MA 6324	yeferson Berrios Alexandra Dimas Alejandro Peña	kia sportage Toyota Yaris Suzuki Swift	4 hrs 3 hrs 3:25 min	08:30 12:30 11:00 14:00 14:00 17:25	

Para esto se genera algoritmo que genere la imagen anterior

```
1 import dash
2 from dash import html
3
4 # Inicializar la app Dash
5 app = dash.Dash(__name__)
6
7 # Layout del Dashboard
8 app.layout = html.Div([
9     html.H1("Control Vehicular - UNI", style={
10         "text-align": "center", "backgroundColor": "#0D6EFD", "color": "white", "padding": "15px"
11     }),
12
13     html.Div([
14         # Tarjeta de Placas Detectadas
15         html.Div([
16             html.H3("Placas Detectadas", style={"text-align": "center"}),
17             html.Ul([
18                 html.Li("LE 0236 - Kia Sportage - 08:00 AM"),
19                 html.Li("MY 1485 - Toyota Yaris - 09:30 AM"),
20                 html.Li("MA 6324 - Swift Suzuki - 10:15 AM")
21             ])
22         ], style={"background": "#f8f9fa", "padding": "20px", "border-radius": "10px", "box-shadow": "2px 2px"
23
24         # Tarjeta de Mapa del Campus
25         html.Div([
26             html.H3("Mapa del Campus", style={"text-align": "center"}),
27             html.Img(src="https://via.placeholder.com/200", alt="Mapa Campus UNI", style={"display": "block"
28         ], style={"background": "#f8f9fa", "padding": "20px", "border-radius": "10px", "box-shadow": "2px 2px"
29
30         # Tarjeta de Alertas
31         html.Div([
32             html.H3("Alertas", style={"text-align": "center"}),
33             html.P("¡Congestión detectada en la entrada principal!", style={"color": "red", "font-weight": "bold",
34         ], style={"background": "#f8f9fa", "padding": "20px", "border-radius": "10px", "box-shadow": "2px 2px 10px"
35         ], style={"display": "flex", "justify-content": "space-around", "margin-top": "20px"})
36     ])
37
38 # Ejecutar la aplicación
39 if __name__ == '__main__':
40     app.run_server(debug=True)
41
```

### C. Construcción del modelado

La construcción del modelado se remitió a la identificación de placas vehiculares, con las técnicas de procesamiento de imágenes, tomando en cuenta los equipos y la necesidad de automatizar a través de la IA.

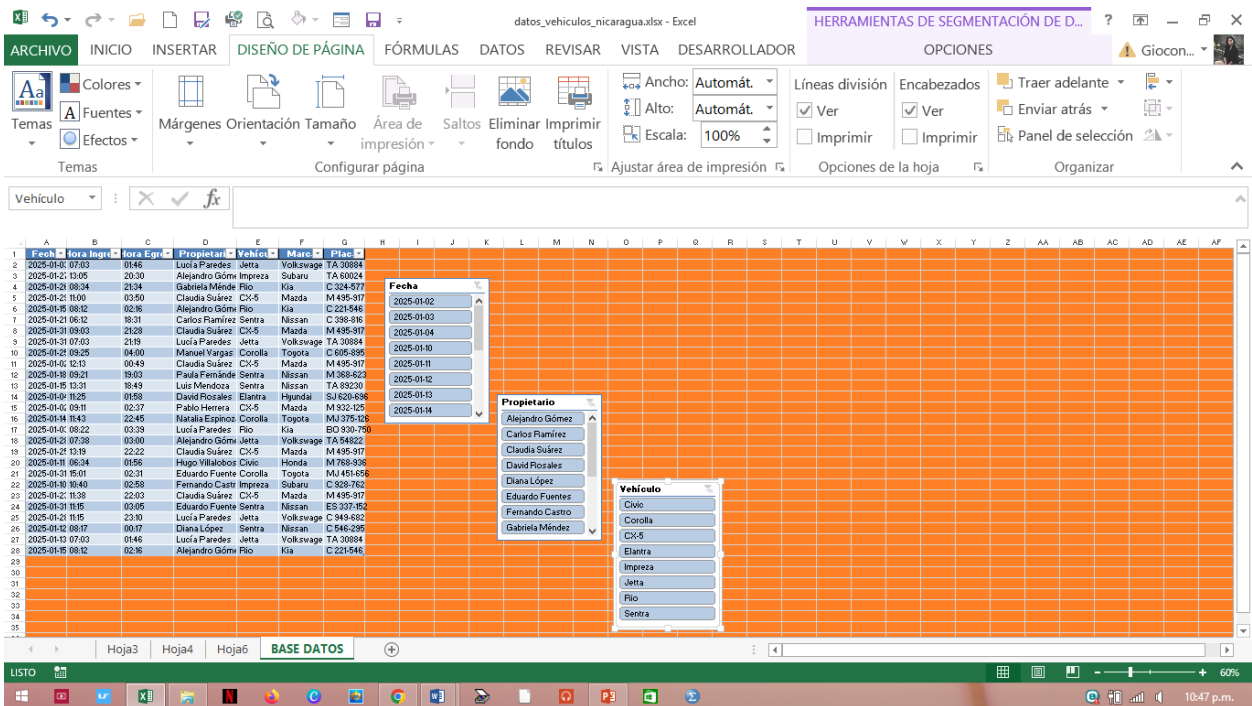
### D. Evaluación del modelado

En esta fase, ha sido fundamental seleccionar la metodología, y el modelo a ejecutar, determinando su viabilidad a través de algoritmos simulados. Sintiendo la robustez a través de los informes generados.

El Dashboard logro con una base de datos ya existente, filtrando segmentos como la fecha, propietario y el tipo de vehículo, para este trabajo a modo de ejemplo, de acuerdo a la figura se logra visualizar la base completa con fechas indistintas.

Figura 32.

*Dashboard paso a paso para visualizar datos de importancia en la toma de decisiones*

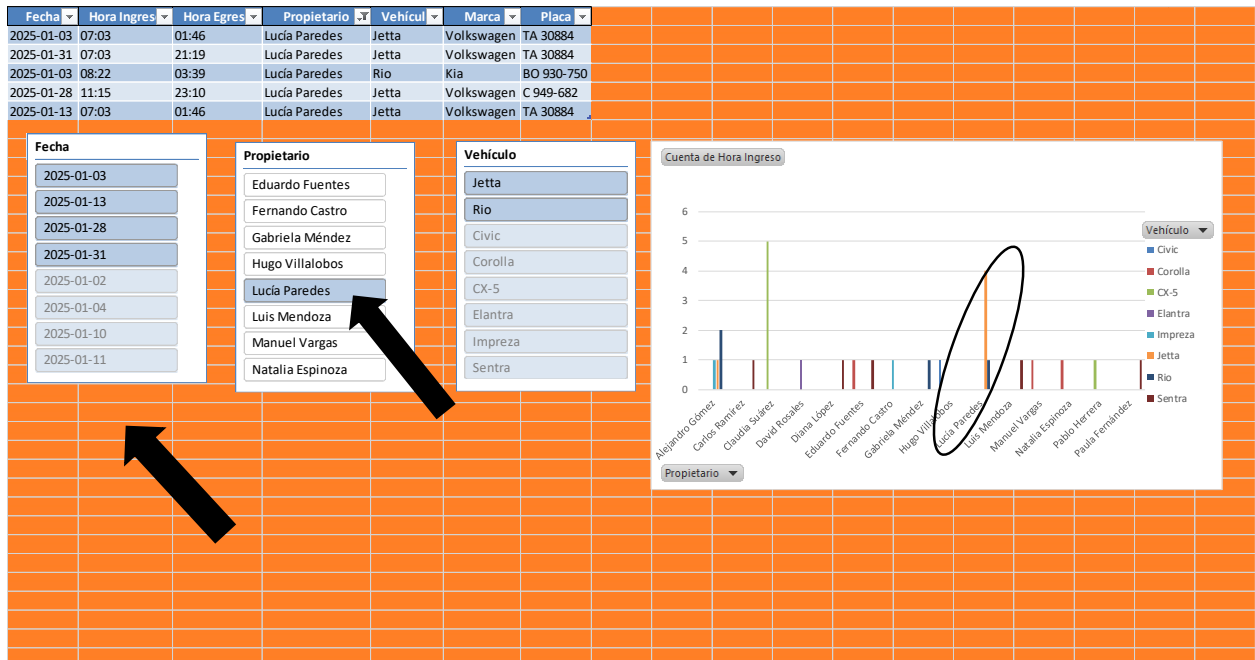


*Nota: Para la elaboración del Dashboard se tomaron de una base de datos existente, ficticia, elaborada por Yeferson Berrios, para la prueba del algoritmo; el cual sirvió para la elaboración de los cuadros de control a través de herramientas de segmentación en Excel.*

En la siguiente figura, se filtra por propietario, las veces que ha visitado el aparcadero de la universidad. Y se escogió a un sujeto de sexo femenino, con cuatro días en diferentes fechas, y dos tipos de vehículo, generado el gráfico dinámico.

**Figura 33.**

*Filtro por propietario Vs fechas de visita al aparcadero*



En la siguiente figura 35, el mismo sujeto ha presentado un horario de ingreso, no obstante, llama la atención que se reporta dos fechas con un horario de egreso del aparcadero hasta las 23 horas. Como ejemplo en la toma de decisiones y seguimiento en el monitoreo del informe, se demuestra que es funcional para poder verificar, mejorar de forma continua y brindar la seguridad a la institución académica.

**Figura 34.**

*Evaluación de informes a través del Dashboard, validación de correcta transferencia de datos para la fase de evaluación en la metodología CRISP-DM*

The dashboard displays a data table at the top left with columns: Fecha, Hora Ingreso, Hora Egreso, Propietario, Vehículo, Marca, and Placa. The table contains five rows of data for January 2025. Below the table is a bar chart titled 'Cuenta de Hora Ingreso' showing the number of entries per owner. The x-axis lists owners, and the y-axis shows the count from 0 to 6. A legend on the right identifies vehicle brands: Civic, Corolla, CX-5, Elantra, Impreza, Jetta, Rio, and Sentra. A red arrow points to the '23:10' entry in the 'Hora Egreso' filter on the right side of the dashboard.

Fecha	Hora Ingreso	Hora Egreso	Propietario	Vehículo	Marca	Placa
2025-01-03	07:03	01:46	Lucía Paredes	Jetta	Volkswagen	TA 30884
2025-01-31	07:03	21:19	Lucía Paredes	Jetta	Volkswagen	TA 30884
2025-01-03	08:22	03:39	Lucía Paredes	Rio	Kia	BO 930-750
2025-01-28	11:15	23:10	Lucía Paredes	Jetta	Volkswagen	C 949-682
2025-01-13	07:03	01:46	Lucía Paredes	Jetta	Volkswagen	TA 30884

**Cuenta de Hora Ingreso**

Vehículo: [Dropdown]

Propietario: [Dropdown]

Hora Egreso: [List: 01:46, 03:39, 21:19, 23:10, 00:17, 00:49, 01:56, 01:58]

Hora Ingreso: [List: 07:03, 08:22, 11:15, 06:12, 06:34, 07:38, 08:12, 08:17]

## **8. CONCLUSIONES**

La metodología propuesta, proporcionó una estructura clara y eficiente para abordar la posibilidad de agilizar el flujo vehicular, desde la comprensión del negocio hasta la implementación de un modelo funcional, que, a través de sus fases, se garantizaron pasos sistemáticos, como la preparación de datos, selección de modelo y la evaluación de los resultados lo que facilitó un sistema robusto y confiable.

El uso de OpenCV que permitió una implementación práctica y eficiente de técnicas avanzadas de procesamiento de imágenes, que son esenciales en el reconocimiento de placas vehiculares. Las tareas del análisis de patrones, detección de bordes segmentación de caracteres, dieron lugar a la conclusión eficaz que aporta a la capacidad del tratamiento de minería de datos virtuales, optimizando el proceso de identificación de placas bajo diferentes ambientes.

Se ha diseñado un prototipo de interfaz que utiliza los datos de reconocimiento de placas y optimiza el flujo vehicular en el campus y para completar el análisis y poder tomar decisiones asertivas en su evaluación, a través de ese comportamiento del flujo vehicular, mediante cuadros de mandos (Dashboard), para la continuidad de una mejora permanente.

## **9. RECOMENDACIONES**

Para la Universidad, crear ensayo piloto que provoque comparación de sistema de control vehicular a través de la IA vs control manual, en su aparcadero.

Incentivar el reconocimiento de la herramienta de la inteligencia artificial como parte de su efectividad y consolidación al conocimiento de nuevas generaciones.

Fomentar la interacción humana en función de la utilización de la IA, como medio y herramienta del crecimiento de la comunidad universitaria para mejorar el aprendizaje.

## 10. BIBLIOGRAFIA

- Acosta Morales, R. J. (s/d de agosto de 2020). Repositorio Institucional UNAN LEON. *Reconocimiento de placas vehiculares aplicando procesamiento de imagenes digitales en Python-OpenCV*. Leon, Leon , Nicaragua: Trabajo Monografico. Obtenido de <http://riul.unanleon.edu.ni:8080/jspui/handle/123456789/8177>.
- Aldas, D. (2016). Procesamiento de Imagenes con vision artificial en reconocimieto de placas vehiculares. (*Maester en gestion industrial y sistemas productivos*). Universidad de Ambato.
- Cortines, V. G. (2015). Aplicacion de la metodologia CRISP-DM a un proyecto de mineria de datos en el entorno universitario. En V. G. Cortines, *Aplicacion de la metodologia CRISP-DM a un proyecto de mineria de datos en el entorno universitario* (págs. 38-50). Madrid: CarlosIII de Madrid.
- Dang Minh, H. X. (2022). Inteligencia Artificial Explicable: una revision exhaustiva. En H. X. Dang Minh, *Inteligencia Artificial Explicable: una revision exhaustiv* (págs. 1-66). Países Bajos: Springer Países Bajos.
- Espinoza, D. (2015). *Desarrollo de un sistema de reconocimiento de placas vehiculares*. universidad de Azuay,, cuenca Ecuador.
- Hewlett-Packard. (30 de mayo de 2024). <https://www.klippa.com/es/blog/informativo/que-es-tesseract-ocr/>. Obtenido de <https://www.klippa.com/es/blog/informativo/que-es-tesseract-ocr/>: <https://www.klippa.com/es/blog/informativo/que-es-tesseract-ocr/>
- La Gaceta. (30 de agosto de 1949). Reglamento sobre el uso de placas para vehiculos motorizados. *Reglamento sobre el uso de placas para vehiculos motorizados*. Managua, Managua, Nicaragua: La Gaceta Diario Oficial.
- Llerena, R. (2008). Teoria de convolucion. En R. Llerena, *Teoria de convolucion* (pág. 13). Cartagenas.
- Miguel, J. J. (2024). Reconocimiento Automatico de matriculas para deteccion de vehiculos con reporte de robo. *Innovacion y Desarrollo Tecnologico. Revista Digital*, 305.
- Molina, E. M. (2016). Algoritmos de binarizaci ´on robusta de im ´agenes con. *CICESE*, 5. Obtenido de Algoritmos de binarizaci ´on robusta de im ´agenes con.
- Molina, E. O. (2016). Algoritmos de binarizacion robusta de imagenes con iluminacion no uniforme. *CICESE*, 5.
- Morales, A. (2024). Algoritmo para el procesamiento de imagenes, utilizando satelites del programa Copernicus de la Union Europea. *MappingGis*, 2.
- Morales, R. J. (2020). *Reconocimiento de placas vehiculares aplicando procesamiento de imágenes digitales en Python-OpenCV*. Leon : UNAN LEON/ Facultad de Matematicas estadística.

- Motor), M. G. (s/d de s/m de 2024). <https://diccionario.motorgiga.com/diccionario/placas-de-matricula-definicion-significado/gmx-niv15-con195156.htm>. Obtenido de <https://diccionario.motorgiga.com/diccionario/placas-de-matricula-definicion-significado/gmx-niv15-con195156.htm>: <https://diccionario.motorgiga.com/diccionario/placas-de-matricula-definicion-significado/gmx-niv15-con195156.htm>
- Omil, J. C. (2019). Inteligencia Artificial. *Mercado y Negocios / articulos de investigacion*, s/n.
- OpenCV, I. (Agosto de 2012). [Google.com/search?q=intel&sca\\_esv=986cfd32cb66a37f&sca\\_upv=1&sxsrf=ADLYWILTXO](https://www.google.com/search?q=intel&sca_esv=986cfd32cb66a37f&sca_upv=1&sxsrf=ADLYWILTXO). Obtenido de [Google.com/search?q=intel&sca\\_esv=986cfd32cb66a37f&sca\\_upv=1&sxsrf=ADLYWILTXO](https://www.google.com/search?q=intel&sca_esv=986cfd32cb66a37f&sca_upv=1&sxsrf=ADLYWILTXO): [Google.com/search?q=intel&sca\\_esv=986cfd32cb66a37f&sca\\_upv=1&sxsrf=ADLYWILTXO](https://www.google.com/search?q=intel&sca_esv=986cfd32cb66a37f&sca_upv=1&sxsrf=ADLYWILTXO)
- Perez, J. (2014). Reconocimiento de placas vehiculares mediante procesamiento de imágenes para optimizar el acceso a los parqueaderos de la UTA, Campus Huachi. (*tesis ingeniero en electronica y comunicaciones*). Universidad Tecnica de Ambato, Ambato.
- RAE, R. A. (27 de mayo de 2023). <https://dle.rae.es/ecualizar>. Obtenido de <https://dle.rae.es/ecualizar>: <https://dle.rae.es/ecualizar>
- Rossum, G. V. (2024). <https://docs.python.org/es/3/tutorial/>. Obtenido de <https://docs.python.org/es/3/tutorial/>: <https://docs.python.org/es/3/tutorial/>
- Rouse, M. (24 de Mayo de 2024). Servidor Informatico. *Servidor*. Techopedia.
- Suaza, A. E. (23 de Mayo de 2023). Desarrollo de aplicativo web con reconocimiento de placas vehiculares para parqueaderos . Villavicencio, Villavicencio, colombia: Universidad Corporativa.
- Tecnologicos, E. S. (4 de Julio de 2024). <https://www.significados.com/interfaz/>. Obtenido de <https://www.significados.com/interfaz/>: <https://www.significados.com/interfaz/>
- Universidad de Jaen. (s/d de s/m de s/a). [https://www4.ujaen.es/~satorres/practicas/practica3\\_vc.pdf](https://www4.ujaen.es/~satorres/practicas/practica3_vc.pdf). Obtenido de [https://www4.ujaen.es/~satorres/practicas/practica3\\_vc.pdf](https://www4.ujaen.es/~satorres/practicas/practica3_vc.pdf): [https://www4.ujaen.es/~satorres/practicas/practica3\\_vc.pdf](https://www4.ujaen.es/~satorres/practicas/practica3_vc.pdf)
- Vasquez, G. A. (s/d de Noviembre de 2014). *Sistema de Reconocimiento de patrones en placas vehiculares para el acceso automatico de visitas a un edificio*. Obtenido de <https://core.ac.uk/download/196534275.pdf>: <https://core.ac.uk/download/196534275.pdf>
- Vigo, U. d. (2019). Inteligencia Artificial. *Mercados y Negocios (articulos de investigacion)*, 1.

Vivar, J. M. (2018). Algoritmos, aplicaciones y Big data, nuevos paradigmas en el proceso de comunicacion. *Revista de comunicacion*, 279.

## 11. ANEXOS

### Anexo A: Estudio de factibilidad

El presente estudio de factibilidad determina la viabilidad técnica, operativa y económica para el desarrollo e implementación de un Sistema Inteligente de Reconocimiento de Placas (LPR) destinado a apoyar control de accesos, trazabilidad vehicular y analítica de movilidad en entornos universitarios y urbanos.

#### 1. Factibilidad Técnica

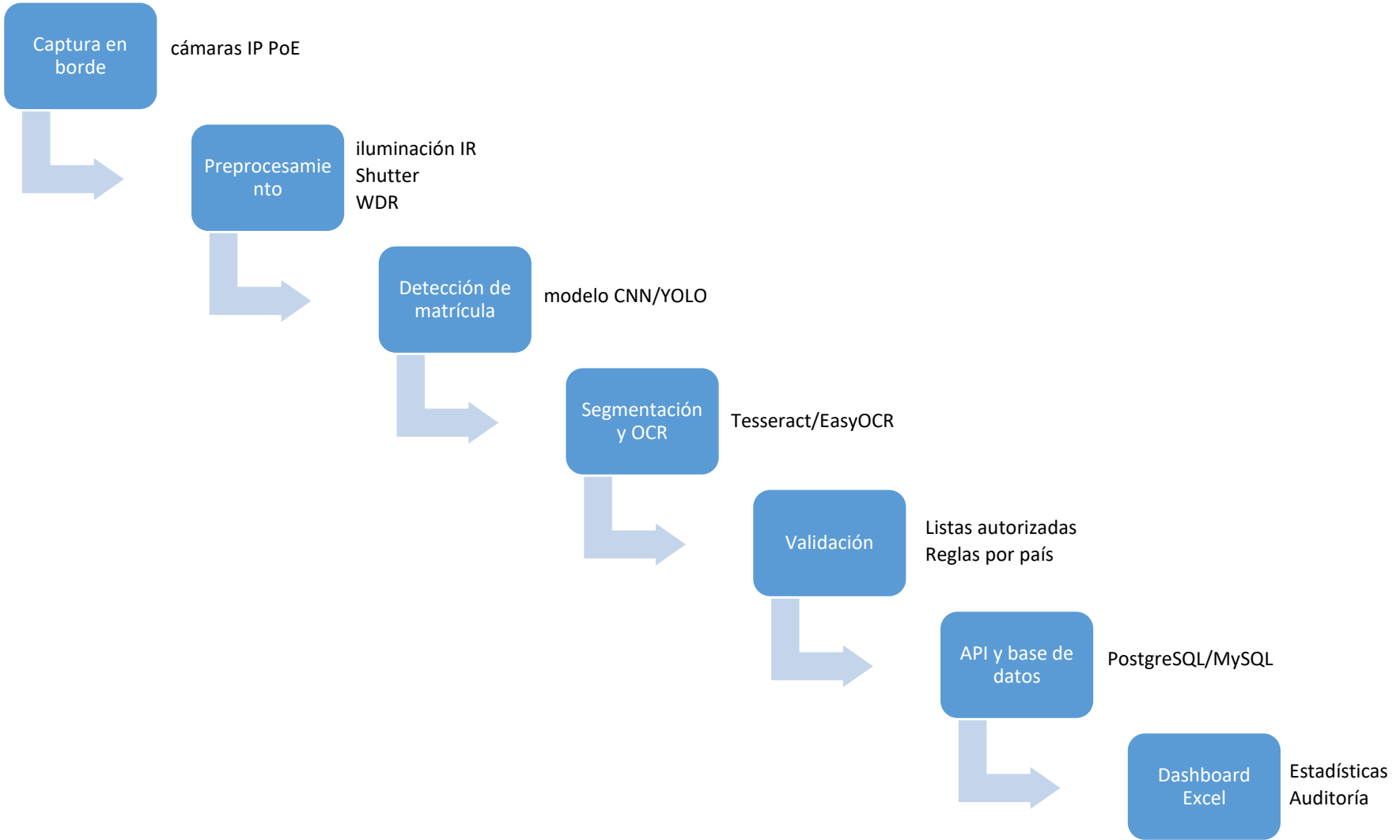
##### Tecnologías seleccionadas:

- **Capa de captura:** Cámaras IP PoE 4MP/5MP con IR, WDR, obturador 1/2000–1/8000 s, lente 6–12 mm.
- **Cómputo en borde:** NVIDIA Jetson Orin Nano/NX (CUDA/cuDNN, TensorRT) o mini-PC Intel i5/i7 con GPU.
- **Modelo de detección:** YOLOv8/YOLOv5 con OpenCV para mejoras (*dewarping, deblurring*).
- **OCR:** EasyOCR/Tesseract con whitelist alfanumérica y post-proceso (regex y checksum local).
- **Backend/API:** Python (FastAPI/Flask), Docker, PostgreSQL, Nginx/Apache.
- **Frontend:** HTML5/CSS/JS (Bootstrap), gráficos con Chart.js.
- **Integraciones:** Webhooks/REST con control de barreras, RFID/QR.

##### Riesgos técnicos y mitigaciones:

- **Baja iluminación/contraluz:** cámaras con IR + WDR; ubicación optimizada y ángulo  $\leq 30^\circ$ .
- **Velocidad del vehículo:** obturador alto, ROI de detección; reductores si corresponde.
- **Variabilidad/suciedad de placas:** *data augmentation* y reglas de validación.
- **Conectividad inestable:** cola local, reintentos y buffer en disco.
- **Juicio técnico:** Viable con equipamiento estándar y stack abierto; modular y escalable.

**Figura 1: Arquitectura propuesta (visión general):**



## 2. Factibilidad operativa

1. **Alineación con procesos:** Control de acceso automatizado, registro auditable de entradas/salidas y analítica operativa.
2. **Perfiles de usuario:** Administrador (parámetros y permisos), Operador (monitoreo y listas), Analista/Auditor (reportes y bitácora).
3. **Capacitación:** sesión de 6–8 horas (operativa + buenas prácticas de cámaras), manual de usuario y guía de incidentes; acompañamiento de arranque durante 2 semanas.
4. **Métricas objetivo:** precisión OCR  $\geq 95\%$  diurno ( $\geq 90\%$  nocturno con IR), tiempo de decisión  $\leq 1$  s, disponibilidad  $\geq 99\%$ .
5. **Juicio operativo:** viable con breve curva de aprendizaje.

## 3. Factibilidad económica

**Tipo de cambio de referencia asumido para conversión:** 1 USD = 36.62 NIO.

Se presentan costos estimados para un piloto de 2 cámaras y 30 días de retención.

**Tabla 4: Costos de Hardware y Software.**

Categoría	Cantidad	Unitario (USD)	Subtotal (USD)	Subtotal (NIO)
Cámaras IP PoE (4MP/5MP, IR, WDR, lente varifocal)	2	\$180.00	\$360.00	C\$13,176.00
Iluminación/Soporte (soportes, cajas, IR/reflectores)	2	\$40.00	\$80.00	C\$2,928.00
Switch PoE 8 puertos	1	\$150.00	\$150.00	C\$5,490.00
Edge/Servidor (Jetson Orin Nano o mini-PC i5)	1	\$450.00	\$450.00	C\$16,470.00

<b>Almacenamiento HDD 2 TB (o NAS básico)</b>	1	\$70.00	\$70.00	C\$2,562.00
<b>UPS 1500 VA</b>	1	\$160.00	\$160.00	C\$5,856.00
<b>Cableado e insumos (UTP Cat6, canaletas, conectores)</b>	—	\$120.00	\$120.00	C\$4,392.00
<b>Software OSS (Python, OCR, DB, Docker)</b>	—	\$0.00	\$0.00	C\$0.00
<b>Subtotal tecnología</b>			\$1,390.00	C\$50,874.00

**Tabla 2: Costos de recursos humanos.**

<b>Actividad</b>	<b>Duración</b>	<b>Costo (USD)</b>	<b>Costo (NIO)</b>
<b>Análisis y diseño (sitio, flujos, datos)</b>	1 semana	\$300.00	C\$10,980.00
<b>Instalación y montaje</b>	2 días	\$200.00	C\$7,320.00
<b>Ajuste/entrenamiento de modelos</b>	1 semana	\$350.00	C\$12,810.00
<b>Desarrollo backend/API y dashboard</b>	2 semanas	\$700.00	C\$25,620.00
<b>Pruebas y puesta en marcha</b>	1 semana	\$250.00	C\$9,150.00
<b>Subtotal recursos humanos</b>		\$1,800.00	C\$65,880.00

**Costo total estimado del piloto: \$3,190.00 | C\$116,754.00**

## 4. Beneficios del sistema

**1. Tangible:** reducción de tiempos de acceso, disminución de errores de registro, evidencia auditable, reportes automatizados.

**2. Intangible:** mejora de seguridad percibida, imagen de innovación, base para analítica avanzada, interoperabilidad futura.

## 5. Requerimientos funcionales

- 1) Captura y reconocimiento de placas en tiempo real (RTSP/ONVIF).
- 2) Gestión de listas (permitidos, restringidos, visitantes) con vigencia.
- 3) Reglas de decisión (apertura/alerta/denegación) y registro de eventos.
- 4) Evidencias con *snapshot* y metadatos (hora, cámara, confianza).
- 5) Dashboard y reportes (rango, placa, cámara, dependencia).
- 6) Auditoría de usuarios/acciones.
- 7) API REST para integraciones.

## 6. Requerimientos no funcionales

N.º	Requerimiento	Prioridad	Categoría
1	Disponibilidad $\geq 99\%$ en horario operativo	Alta	Confiabilidad
2	Latencia de decisión $\leq 1$ s por evento	Alta	Rendimiento
3	Precisión OCR $\geq 95\%$ diurno / $\geq 90\%$ nocturno (con IR)	Alta	Calidad
4	Autenticación por roles y cifrado en tránsito (HTTPS)	Alta	Seguridad
5	Interfaz web responsive y flujos en $\leq 3$ clics	Media	Usabilidad

<b>6</b>	Escalabilidad por canales sin rediseño	Media	Mantenibilidad
<b>7</b>	Logs inmutables de eventos y usuario	Alta	Auditoría

## Anexo B: Manual de Usuario – Sistema de Placas Vehiculares

### 1. Introducción

Este sistema ha sido desarrollado para la detección, registro y gestión de vehículos mediante reconocimiento óptico de caracteres (OCR) a partir de imágenes captadas por una cámara IP o webcam.

El software permite capturar placas vehiculares, asociarlas a un usuario y almacenar esta información en una base de datos, con opciones de exportación, generación de informes y control de accesos.

A continuación, se describe su funcionamiento paso a paso:

### 2. Requisitos Previos

- Python 3.10 o superior
- Tesseract OCR instalado (ruta configurada)
- Librerías como: OpenCV, pytesseract, SQLAlchemy, Tkinter, openpyxl, reportlab, folium

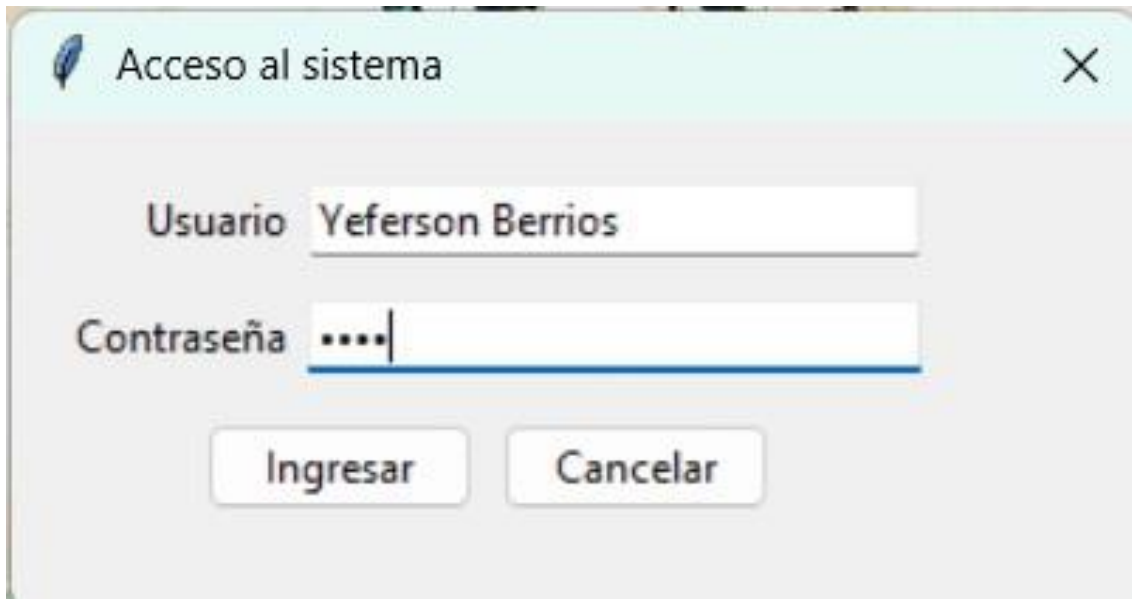
Tabla 5: Elementos que conforman el sistema

Elementos
Cámaras IP PoE (4MP/5MP, IR, WDR, lente varifocal)
Iluminación/Soporte (soportes, cajas, IR/reflectores)
Switch PoE 8 puertos
Edge/Servidor (Jetson Orin Nano o mini-PC i5)
Almacenamiento HDD 2 TB (o NAS básico)
UPS 1500 VA
Cableado e insumos (UTP Cat6, canaletas, conectores)
Software OSS (Python, OCR, DB, Docker)

### 3. Inicio del Sistema

- a) Al ejecutar el sistema
- b) Se muestra una pantalla de login.

*Figura 35 Usuario y contraseña ingresados*



Acceso al sistema

Usuario Yeferson Berrios

Contraseña ....

Ingresar Cancelar

Nota: Inicio de Sesión – Usuario y contraseña ingresados.

- c) El administrador puede crear y gestionar usuarios.
- d) Se inicializa la cámara (IP o webcam según configuración).

## 4. Reconocimiento de Placas

a) El sistema analiza el área central del video.

Figura 36: Cámara en vivo y registros

The screenshot displays a web-based application interface. At the top, there is a navigation menu with options: Archivo, Ver, Operación, Macro, Seguridad, Reportes, Ayuda. Below this, a control panel includes buttons for 'Iniciar turno', 'Finalizar turno', and 'Control c'. The user is identified as 'Usuario: Yeferson Berrios' and the camera is labeled 'Cámara: IP'. There are also buttons for 'Reconectar', 'Pausa OCR', 'Finalizar Programa', and 'Modo Oscuro'. The main interface is split into two sections: 'Cámara en Vivo' and 'Registros'. The 'Cámara en Vivo' section shows a black video feed with the text 'using droidcam.app' and a red rectangular bounding box in the center. The 'Registros' section features a search bar and a table with the following data:

Usuario	Placa	Departamento	Fecha Entrada	Fecha Salida
ANA RUIZ	M452156	Desconocido	2025-10-19 18:42:11	2025-10-19 18:42:23
Ana Berrios	R1422153	Rivas	2025-10-21 10:46:55	2025-10-21 10:47:23
Ana Berrios	R1145236	Rivas	2025-10-21 11:34:29	2025-10-21 13:12:50
MARIO RIVAS	M 45236	Desconocido	2025-10-21 11:59:06	2025-10-21 13:12:53
Carlos Lopez	R1 145236	Rivas	2025-10-21 12:04:08	2025-10-21 12:05:06
Pablo Rosales	R1 4235	Rivas	2025-10-21 12:12:18	2025-10-21 13:12:47
PABLO ROSALES	R1 45632	Rivas	2025-10-21 12:21:18	2025-10-21 12:21:47

Below the table, there are input fields for 'Nombre:', 'Placa:', 'Tipo:' (set to 'Vehículo'), and 'Marca y Año:', followed by an 'Agregar' button. At the bottom, there are four buttons: 'Registrar Entrada', 'Marcar Salida', 'Editar', and 'Borrar'.

b) Se procesa mediante OCR y se extrae la placa.

c) Si se detecta una placa válida, se registra automáticamente.

## 5. Registro y Consulta

- a) Los registros se guardan en una base de datos SQLite.
- b) Se pueden editar, marcar como salida o eliminar.

Figura 37: editar y borrar datos

The screenshot shows a software interface titled "Registros". At the top, there is a search bar labeled "Buscar:". Below it is a table with the following columns: "Usuario", "Placa", "Departamento", "Fecha Entrada", and "Fecha Salida". The table contains several rows of data, with the row for "PABLO ROSALES" highlighted in blue. An "Edit..." dialog box is open over the table, displaying "Usuario actual (PABLO ROSALES):" and a text input field. Below the dialog, there are four buttons: "Registrar Entrada", "Marcar Salida", "Editar", and "Borrar". A blue arrow points to the "Editar" button. At the bottom of the interface, there are input fields for "Nombre:", "Placa:", "Tipo:" (with a dropdown menu showing "Vehículo"), and "Marca y Año:", followed by an "Agregar" button.

Usuario	Placa	Departamento	Fecha Entrada	Fecha Salida
ANA RUIZ	M452156	Desconocido	2025-10-19 18:42:11	2025-10-19 18:42:23
Ana Berrios	RI422153	Rivas	2025-10-21 10:46:55	2025-10-21 10:47:23
Ana Berrios	RI145236	Rivas	2025-10-21 11:34:29	2025-10-21 13:12:50
MARIO RIVAS	M 45236	Desconocido	2025-10-21 11:59:06	2025-10-21 13:12:53
Carlos Lopez	RI 145236	Rivas	2025-10-21 12:04:08	2025-10-21 12:05:06
Pablo Rosales	RI 4235	Rivas	2025-10-21 12:12:18	2025-10-21 13:12:47
PABLO ROSALES	RI 45632	Rivas	2025-10-21 12:21:18	2025-10-21 12:21:47
Alvaro Navarro	MA 459602	Managua	2025-10-21 14:39:24	2025-10-21 14:40:06

*Nota: Acá seleccionamos el usuario a editar*

Figura 38: Función Borrar

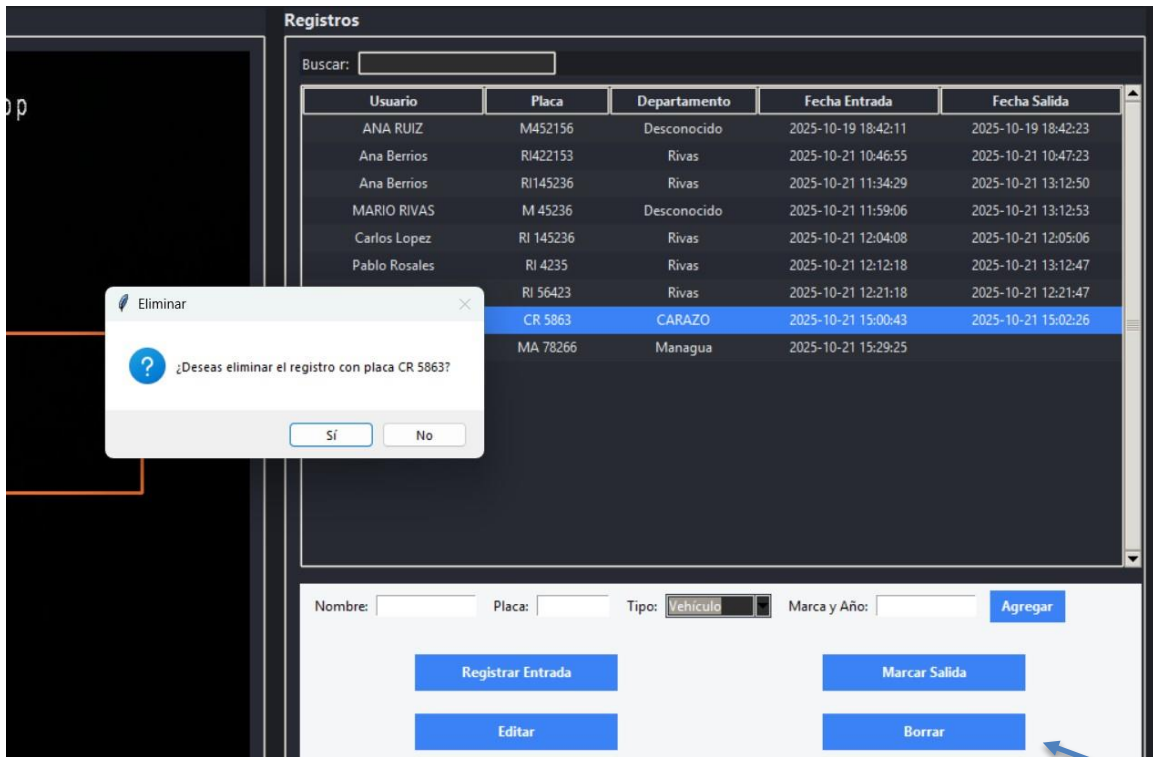
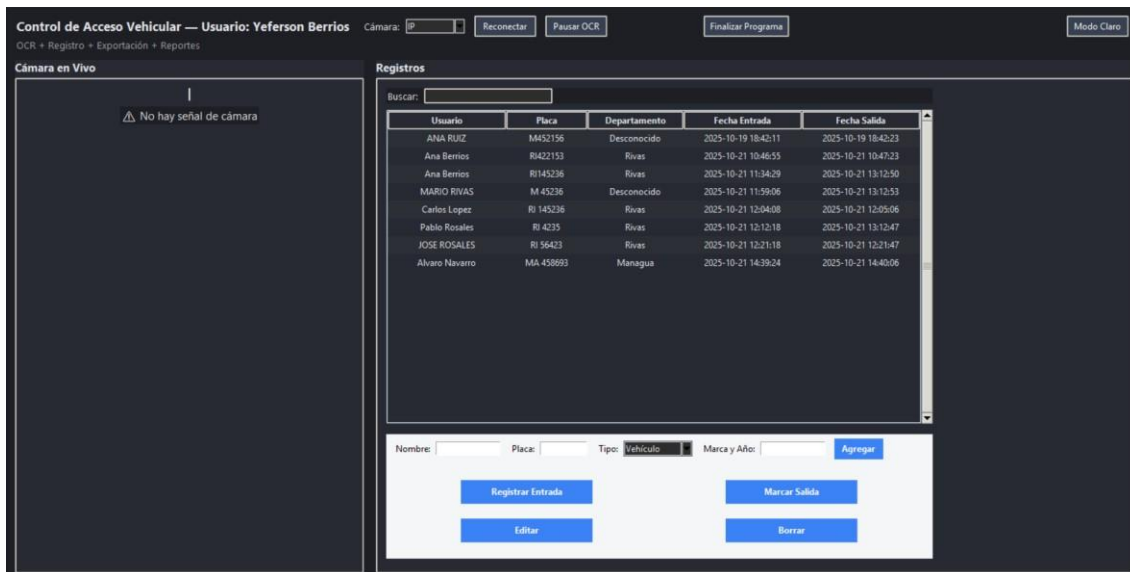


Figura 39: Vista con nuevo registro



c) Se filtran por nombre, placa, departamento o fecha.

Figura 40: Vista comprimida del panel registro

**Registros**

Buscar:

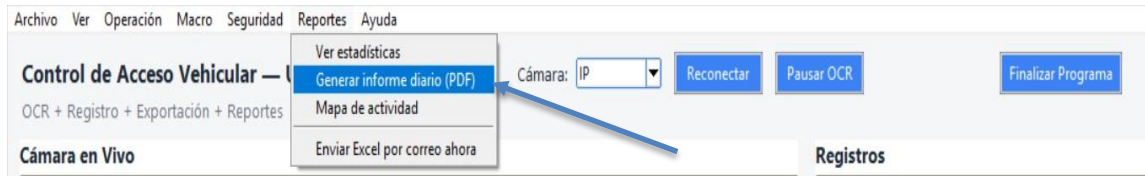
Usuario	Placa	Departamento	Fecha Entrada	Fecha Salida
ANA RUIZ	M452156	Desconocido	2025-10-19 18:42:11	2025-10-19 18:42:23
Ana Berrios	RI422153	Rivas	2025-10-21 10:46:55	2025-10-21 10:47:23
Ana Berrios	RI145236	Rivas	2025-10-21 11:34:29	2025-10-21 13:12:50
MARIO RIVAS	M 45236	Desconocido	2025-10-21 11:59:06	2025-10-21 13:12:53
Carlos Lopez	RI 145236	Rivas	2025-10-21 12:04:08	2025-10-21 12:05:06
Pablo Rosales	RI 4235	Rivas	2025-10-21 12:12:18	2025-10-21 13:12:47
JOSE ROSALES	RI 56423	Rivas	2025-10-21 12:21:18	2025-10-21 12:21:47

Nombre:  Placa:  Tipo:  Marca y Año:

## 6. Exportación e Informes

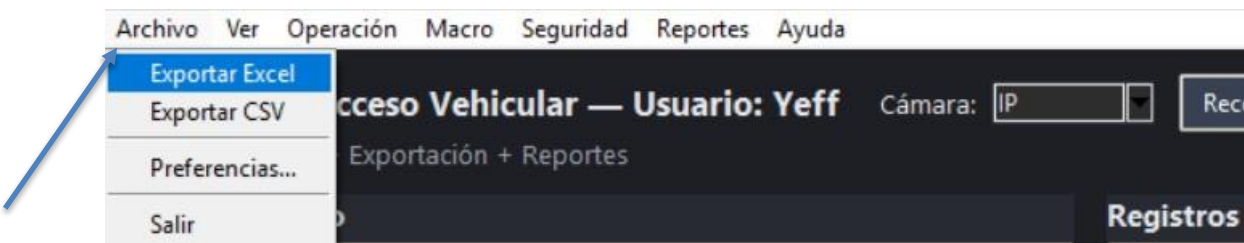
### a) Exportación de datos a Excel o CSV.

Figura 41: Menú Reportes – Estadísticas, informes y correo



*Nota: En selección PDF se nos descarga un documento con todos los datos obtenidos del día*

Figura 42: Ejecutar macro manualmente



*Nota: Desde acá podremos exportar un reporte detallado de los reportes del día a día.*

**b) Generación de informes PDF diarios o por turno.**

Figura 43: Informe Diario

### Informe Diario

Rango: 2025-10-21 00:00:00 a 2025-10-22 00:00:00

**Total de vehículos: 7**

Departamento	Conteo
Rivas	5
Desconocido	1
CARAZO	1

Nombre y apellido	Placa	Dpto	Entrada	Salida
Ana Berrios	RI422153	Rivas	2025-10-21 10:46:55	2025-10-21 10:47:23
Ana Berrios	RI145236	Rivas	2025-10-21 11:34:29	2025-10-21 13:12:50
MARIO RIVAS	M 45236	Desconocido	2025-10-21 11:59:06	2025-10-21 13:12:53
Carlos Lopez	RI 145236	Rivas	2025-10-21 12:04:08	2025-10-21 12:05:06
Pablo Rosales	RI 4235	Rivas	2025-10-21 12:12:18	2025-10-21 13:12:47
JOSE ROSALES	RI 56423	Rivas	2025-10-21 12:21:18	2025-10-21 12:21:47
Lucia Perez	CR 5863	CARAZO	2025-10-21 15:00:43	2025-10-21 15:02:26

**c) Envío automático por correo electrónico configurado.**

Figura 44: Usuario agregado

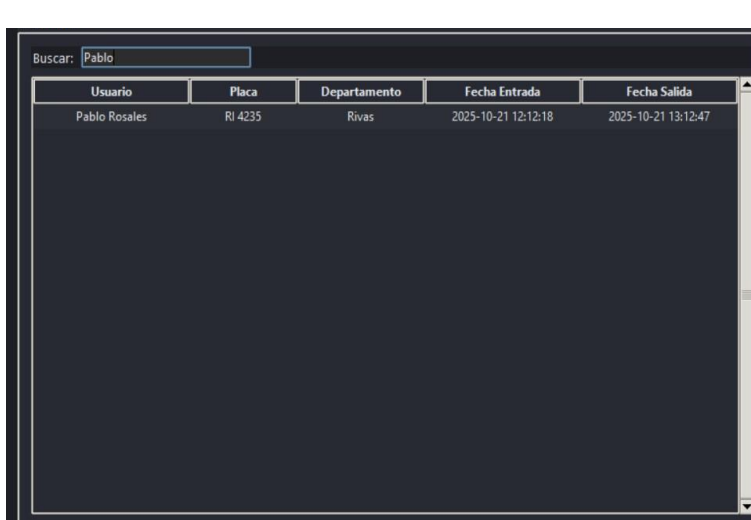
ESTHER LOPEZ      MA 78266      Managua      2025-10-21 15:29:25

Nombre:       Placa:       Tipo:       Marca y Año:

## 7. Seguridad

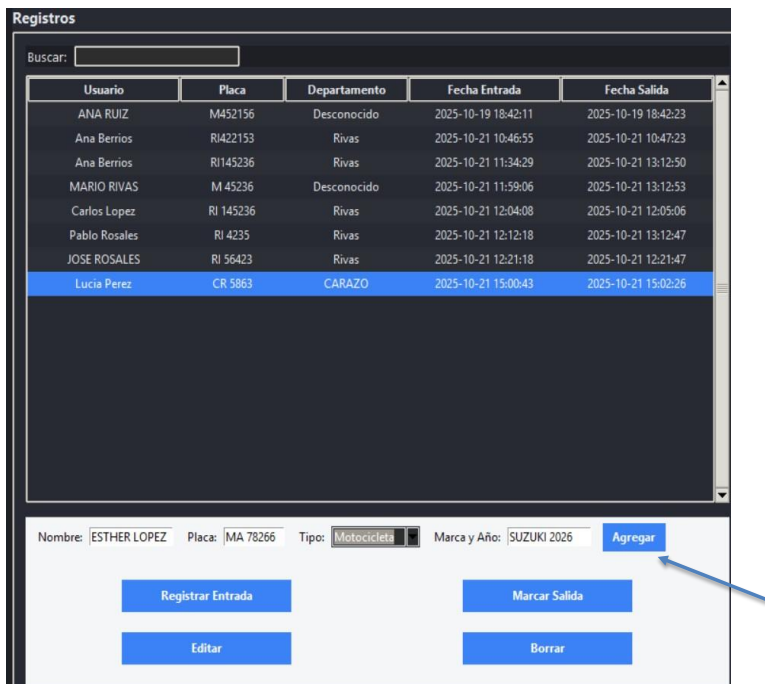
### a) Autenticación de usuarios con roles (admin/usuario)

Figura 45: Búsqueda de usuario



Usuario	Placa	Departamento	Fecha Entrada	Fecha Salida
Pablo Rosales	RI 4235	Rivas	2025-10-21 12:12:18	2025-10-21 13:12:47

Figura 46: Registro y agregado de usuarios



Usuario	Placa	Departamento	Fecha Entrada	Fecha Salida
ANA RUIZ	M452156	Desconocido	2025-10-19 18:42:11	2025-10-19 18:42:23
Ana Berrios	RI422153	Rivas	2025-10-21 10:46:55	2025-10-21 10:47:23
Ana Berrios	RI145236	Rivas	2025-10-21 11:34:29	2025-10-21 13:12:50
MARIO RIVAS	M 45236	Desconocido	2025-10-21 11:59:06	2025-10-21 13:12:53
Carlos Lopez	RI 145236	Rivas	2025-10-21 12:04:08	2025-10-21 12:05:06
Pablo Rosales	RI 4235	Rivas	2025-10-21 12:12:18	2025-10-21 13:12:47
JOSE ROSALES	RI 56423	Rivas	2025-10-21 12:21:18	2025-10-21 12:21:47
Lucia Perez	CR 5863	CARAZO	2025-10-21 15:00:43	2025-10-21 15:02:26

Nombre: ESTHER LOPEZ Placa: MA 78266 Tipo: Motocicleta Marca y Año: SUZUKI 2026 **Agregar**

Registrar Entrada Marcar Salida

Editar Borrar

*Nota: se registran nuevos usuarios con sus respectivos datos, sino están completos los datos el sistema dira que se tienen que completar todos los datos.*

## b) Listas de vigilancia y autorizados.

Figura 47: Listas y administración de usuarios



*Nota El usuario administrador puede agregar y eliminar nuevos usuarios.*

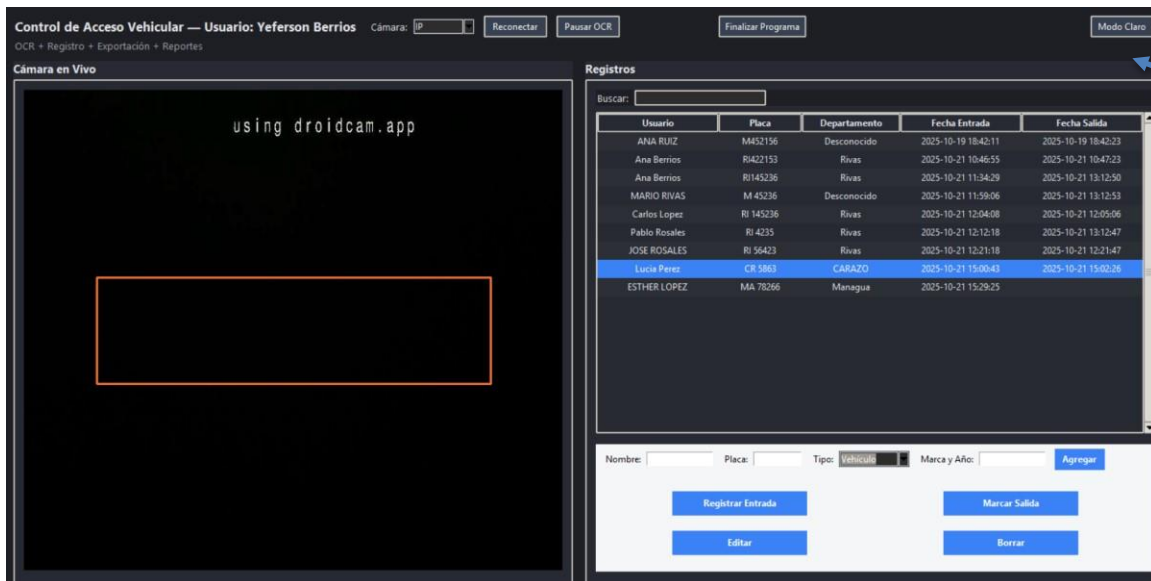
## c) Alertas por correo electrónico.

## 8. Personalización

### a) Configuración de OCR (OEM, PSM, ROI).

### b) Temas claros u oscuros.

Figura 48: Función modo claro oscuro.



*Nota: Desde acá se cambia a modo claro y oscuro para una mejor visualización.*

Figura 49: Función selección de cámaras conectadas



*Nota: Apartador para selección de cámaras en caso que hubiera varias conectadas*

**a) Macros de Excel automáticas.**

## **9. Finalización del Sistema**

**b) El sistema puede cerrarse de forma segura desde el menú.**

**c) Se libera la cámara y se cierra la aplicación.**