



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
MECATRÓNICA

TEMA:

**“SISTEMA DE SEGURIDAD CON VISIÓN ARTIFICIAL EN EL PARQUE
ACUÁTICO ARAQUE”**

AUTOR: Willan Santiago Yaselga Antamba

DIRECTOR: MSc. Cosme Mejía

IBARRA-ECUADOR

2019



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TECNICA DEL NORTE

1. IDENTIFICACIÓN DE OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100404099-2		
APELLIDOS Y NOMBRES:	YASELGA ANTAMBA WILLAN SANTIAGO		
DIRECCIÓN:	SAN PABLO CALLE SUCRE Y ATAHUALPA		
EMAIL:	wsyaselgaa@utn.edu.ec		
TELÉFONO FIJO:	062-919 173	TELÉFONO MÓVIL	0988377132

DATOS DE LA OBRA	
TÍTULO:	“SISTEMA DE SEGURIDAD CON VISIÓN ARTIFICIAL EN EL PARQUE ACUÁTICO ARAQUE”
AUTOR:	YASELGA ANTAMBA WILLAN SANTIAGO
FECHA:	09-12-2020
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE	Ingeniero en Mecatrónica
OPTA:	
DIRECTOR:	MSc. Cosme Mejía

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 23 días del julio de 2021

EL AUTOR:





UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN

En calidad de director del trabajo de grado “SISTEMA DE SEGURIDAD CON VISIÓN ARTIFICIAL EN EL PARQUE ACUÁTICO ARAQUE”, ha sido desarrollado en su totalidad por el Sr. Yaselga Antamba Willan Santiago, con cédula de identidad 1004040992, bajo mi supervisión para lo cual firmo en constancia.



MSc. Cosme Mejía

DIRECTOR DEL PROYECTO

Agradecimiento

El éxito no es casualidad si no el resultado de un arduo trabajo, el mismo que he culminado con éxito gracias a la fortaleza que Dios puso en mi a través del apoyo incondicional de tres mujeres esenciales en este largo camino: mi madre, mi esposa y mi hermana, un camino que al principio fue duro y parecía interminable, pero gracias a su amor, aliento y apoyo incondicional me impulsaron a lograr culminar con este objetivo.

Quiero agradecer a mi padre que con sus buenos consejos siempre estuvo allí en todo momento y a toda mi familia que con un granito de arena me han apoyado para convertirme en lo que ahora soy.

También quiero agradecer a mis docentes que con sus enseñanzas no solo forjaron mi carácter si no también mi conocimiento, a mi tutor que con su guía complemento el éxito de este proyecto y a mis compañeros que a lo largo de la carrera estuvieron en los buenos y malos momentos que compartimos.

Willan Santiago Yaselga Antamba

Dedicatoria

Este logro académico lo dedico a mi madre quien con su ejemplo de lucha incansable ante las adversidades y quien ha sido mi motor desde niño, quien sabido guiarme en los buenos y malos momentos hasta convertirme en un profesional.

Willan Santiago Yaselga Antamba

Resumen

El presente trabajo de titulación está destinado a espacios turísticos con la finalidad de mejorar la seguridad y agilizar el tráfico vehicular al ingreso y salida del mismo. Este proyecto se basa en algoritmos de programación en un lenguaje abierto (Open CV) que permiten el cobro de los peajes de manera rápida y eficiente mediante un lector de tarjetas y un sistema RFID conectado a un sistema arduino programado para la activación del actuador, la implementación de este sistema permite adquirir los datos de la placa mediante una cámara y un proceso de visión artificial en el cual se incluyen pre-procesamiento de imágenes, entrenamiento de redes neuronales (Deep Learning para el vehículo y la placa, OCR para reconocimiento de caracteres), los datos extraídos son almacenados en una base de datos creada en Python y SQL donde se llenan los datos requeridos para un posterior análisis, además este sistema cuenta con una red de cámaras Hik-Vision que permiten monitorear el lugar mediante la aplicación Hi-Look instalada en un celular Android y almacenar las grabaciones en un DVR de alta capacidad para su revisión en caso de ser necesario, la ventaja de este sistema de seguridad es su fácil instalación y costo accesible permitiendo que los turistas y propietarios de los bienes inmuebles del sector turístico tenga una confianza y seguridad eficiente.

Summary

The present degree work is intended for tourist spaces in order to improve safety and streamline vehicular traffic at the entrance and exit of the same, this project is based on programming algorithms in an open language (Open CV) that allow the collection of tolls quickly and efficiently through a card reader and an RFID system connected to an arduino system programmed to activate the actuator, the implementation of this system allows us to acquire the data from the plate by means of a camera and an artificial vision process which includes image pre-processing, neural network training (Deep Learning for the vehicle and the plate, OCR for character recognition), the extracted data is stored in a database created in Python and SQL where it is filled the data required for further analysis, in addition this system has a network of Hik-Vision cameras that allow monitoring the place through the Hi-Look application installed on an Android cell phone and store the recordings in a high capacity DVR for review if necessary, the advantage of this security system is its easy installation and affordable cost allowing tourists and owners of the real estate of the tourism sector has an efficient confidence and security.

Índice General

Agradecimiento.....	5
Dedicatoria.....	6
Resumen.....	7
Abstract.....	8
Índice General.....	9
Índice de Figuras.....	13
Índice de Tablas.....	15
Introducción	
Planteamiento del problema.....	16
Objetivo General.....	17
Objetivos Específicos.....	17
Alcance.....	17
Justificación.....	17
Capítulo 1. Marco Teórico	
1.1 Visión Artificial.....	19
1.2 Componentes Principales.....	20
1.2.1 Fuente de iluminación.....	21
1.2.2 Cámara.....	21
1.2.3 Digitalizador.....	21
1.2.4 Computador.....	22
1.3 Fases del proceso de visión por computador.....	22
1.3.1 Adquisición de la imagen.....	22
1.3.2 Pre procesamiento.....	22
1.3.3 Detección de bordes.....	22
1.3.4 Segmentación.....	23
1.3.5 Extracción de características.....	23
1.3.6 Reconocimiento y localización.....	23
1.3.7 Interpretación.....	23
1.4 Tratamiento digital de imágenes.....	24

1.4.1 Manipulación de píxeles.....	24
1.4.2 Operaciones Matemáticas.....	25
1.4.3 Operaciones Aritmetológicas.....	25
1.4.4 Operaciones Geométricas.....	26
1.4.5 Detección de bordes.....	29
1.5 Sistema de control de acceso vehicular.....	30
1.6 Sistemas de cobro de peajes.....	31
1.6.1 Métodos de cobro.....	31
1.6.2 Sistema de cobro por TAGS.....	32
1.7 Tecnología RFID.....	32
1.7.1 Componentes.....	33
1.7.2 Tipos RFID.....	34
1.8 Detección de placas.....	36
1.8.1 ANPR.....	36
1.8.2 Placas de vehículos en Ecuador por provincias.....	37
1.9 Algoritmos de programación.....	39
1.10 Open CV.....	39
1.11 Machine Learning.....	41
1.12 Métodos.....	43
1.13 Deep Learning.....	44
1.14 OCR.....	45
1.15 Base de datos.....	45
1.16 Sistemas de video vigilancia.....	46
1.17 Tipos de cámaras.....	47
1.18 Características del lugar turístico.....	49
Capítulo II. Metodología	
2.1 Elaboración CAD del Lugar.....	50
2.2 Análisis y selección de sistemas de cobro de peajes.....	50
2.3 Implementación del sistema.....	56
2.3.1 Esquema de conexión.....	58

2.3.2 Diagrama de flujo del funcionamiento.....	58
2.4 Selección de componentes para la detección de placas vehiculares.....	58
2.4.1 Adquisición de la imagen.....	59
2.4.2 Deep Learning para detectar un vehículo	60
2.4.3 Deep Learning para detectar una placa.....	62
2.4.4 OCR para detectar caracteres.....	63
2.4.5 Creación base de datos.....	64
2.5 Análisis de los componentes del sistema de seguridad.....	65
2.6 Instalación de los componentes de seguridad.....	67
Capítulo III. Pruebas, análisis y resultados	
3.1 Pruebas de funcionamiento RFID.....	68
3.2 Pruebas de funcionamiento del detector de placas.....	70
3.5 Validación del sistema de cámaras.....	79
Capítulo IV. Conclusiones y recomendaciones	
4.1 Conclusiones.....	80
4.2 Recomendaciones.....	81
Bibliografía.....	82
Anexos.....	84

Índice de Figuras

Figura 1. Componentes de un sistema de visión artificial.....	20
Figura 2. Etapas de procesamiento de imágenes.....	23
Figura 3. Imagen en escala de grises.....	24
Figura 4. Imagen con filtro por umbral.....	24
Figura 5. Resultado de una operación matemática.....	25
Figura 6. Operación AND entre dos píxeles.....	25
Figura 7. Operación OR entre dos píxeles.....	25
Figura 8. Operación de negación en un pixel.....	26
Figura 9. Transformación geométrica de rotación en un tablero de ajedrez.....	26
Figura 10. Operación erosión en una imagen.....	27
Figura 11. Operación dilatación en una imagen.....	27
Figura 12. Operación suavizado en una imagen.....	28
Figura 13. Operación Gaussiano en una imagen.....	28
Figura 14. Operación sustracción de fondo de una imagen.....	29
Figura 15. Operación canny en una imagen.....	29
Figura 16. Operación sobel en una imagen.....	29
Figura 17. Sistema de acceso vehicular tradicional.....	30
Figura 18. Funcionamiento de un sistema RFID.....	33
Figura 19. Formato de placas en Ecuador.....	36
Figura 20. Proceso para entrenar un sistema autónomo y eficiente.....	43
Figura 21. Estilo de aprendizaje dentro de las redes neuronales.....	44
Figura 22. Implementación toolbox neuronal en network Matlba.....	45
Figura 23. Cámara de red fija.....	47
Figura 24. Cámara de red fija tipo domo.....	48
Figura 25. Cámara de red PTZ.....	48
Figura 26. Cámara de red PTZ no mecánica.....	49
Figura 27. Cámara de red domo PTZ.....	49
Figura 28. Planos CAD del Parque Acuático Araque.....	50

Figura 29. Funcionamiento de las Tags.....	51
Figura 30. Diagrama de un monedero electrónico.....	52
Figura 31. Lector código QR.....	54
Figura 32. Componentes de un sistema RFID.....	55
Figura 33. Funcionamiento de un sistema RFID.....	57
Figura 34. Diagrama de conexión RFID y Arduino NANO.....	58
Figura 35. Paradero Turístico “Don Pablo”.....	59
Figura 36. Tratamiento de imágenes por binarización,,.....	61
Figura 37. Deep Learning para detectar vehículos.....	61
Figura 38. Deep Learning para detectar placas.....	62
Figura 39. Extracción de placas en formato PNG.....	62
Figura 40. Extracción de imagen PNG.....	63
Figura 41. Placa procesada por el método OCR.....	63
Figura 42. Parámetros de la base de datos.....	64
Figura 43. Interfaz del sistema de parqueo.....	65
Figura 44. Verificación del sistema de ingreso vehicular.....	65
Figura 45. Conexión del sistema RFID con el servomotor.....	68
Figura 46. Tarjetas TAG para el sistema RFID.....	69
Figura 47. Servomotor OFF activado por tarjeta la RFID.....	69
Figura 48. Servomotor ON activado por tarjeta la RFID.....	70
Figura 49. Vehículo 1 ingresando al lugar turístico	71
Figura 50. Vehículo 2 ingresando al lugar turístico	71
Figura 51. Vehículo 3 ingresando al lugar turístico	71
Figura 52. Vehículo 4 ingresando al lugar turístico	72
Figura 53. Vehículo 5 ingresando al lugar turístico	72
Figura 54. Vehículo 6 ingresando al lugar turístico	73
Figura 55. Base de datos de la red neuronal.....	73
Figura 56. Imagen con el ángulo adecuado para la extracción de la imagen.....	77
Figura 57. Deep Learning detección de placa.....	77
Figura 58. OCR placas vehiculares.....	78

Figura 59. Enlace con la base de datos.....	78
Figura 60. Componentes del sistema de seguridad.....	79
Figura 61. Validación del sistema de cámaras.....	79

Índice de Tablas

Tabla 1. Características de los RFID.....	35
Tabla 2. Letras de placas por provincia en Ecuador.....	37
Tabla 3. Costo de un sistema de peaje manual.....	51
Tabla 4. Características de los TAGS.....	52
Tabla 5. Características de un monedero electrónico.....	53
Tabla 6. Características de un lector QR.....	55
Tabla 7. Componentes RFID.....	56
Tabla 8. Análisis de los factores en los sistemas de cobros de peaje.....	56
Tabla 9. Presupuesto para un sistema de seguridad convencional.....	66
Tabla 10. Análisis de un sistema de seguridad HikVision.....	66
Tabla 11. Análisis de Deep Learning para detección de vehículos.....	74
Tabla 12. Análisis Deep Learning para detección de placas vehiculares.....	75
Tabla 13. Análisis OCR para detección de caracteres.....	76
Tabla 14. Análisis del sistema completo.....	76

Introducción

Planteamiento del problema

La relación de seguridad entre el visitante y el medio turístico es un factor importante para el desarrollo del turismo a nivel nacional, por lo cual se debería considerar importante proteger al visitante, medio cultural y comerciante local. La seguridad se ve afectada principalmente por un escenario inseguro en el cual el ser humano (turista, comunero o comerciante) se desplaza de forma permanente o transitoria [1].

Según estudios (Llaguno 2016) a un delincuente promedio lo toma alrededor de 12 segundos abrir, prender y llevarse un automóvil mientras esta parqueado, existen varias maneras de evitarlo como dispositivos antirrobo ineficientes, bloqueos que entorpecen el manejo, alarmas inservibles y trucos poco convencionales y eficaces [2].

Actualmente el “Parque Acuático Araque” carece de recursos tecnológicos por lo cual la seguridad del turista, comerciante y comunero es baja, en el lugar han existido robos a los turistas y daños a las propiedades de los comerciantes. Por esto es necesario implementar sistemas de seguridad, con la finalidad de salvaguardar la integridad humana y material en el sector turístico [3].

El sector turístico hoy en día constituye un factor importante dentro de la sociedad por lo que la digitalización e innovación tecnológica es necesaria, una de las ventajas y retos es hacer que los destinos turísticos sean más seguros y que garanticen el desarrollo sostenible del sector turístico, una excelencia en cuanto a la accesibilidad del lugar, la integración de turista con el entorno y el aumento de calidad de su experiencia [4] [5].

Objetivo General

Diseñar un sistema de seguridad con visión artificial en el “Parque Acuático Araque”

Objetivos Específicos

- Determinar las condiciones adecuadas para la adquisición de imágenes a utilizar en la visión por computadora.
- Diseñar un sistema de parqueo inteligente.
- Diseñar un sistema de seguridad para la zona “Deportes Acuáticos”.
- Realizar pruebas y validación de los sistemas de seguridad.

Alcance

Se diseñará un sistema de seguridad en el “Parque Acuático Araque”, el cual guardará un registro de las placas vehiculares para su control y monitoreo, también tendrá un sistema de cobro de peajes. Además, se diseñará un sistema de seguridad para la zona de “Deportes Acuáticos”

Justificación

El desarrollo de un sistema de seguridad en el “Parque Acuático Araque” es importante debido a que permitirá prevenir robos a los turistas y también a las propiedades de los comerciantes de la zona.

Actualmente los beneficios que se obtiene en los sistemas de seguridad son muchos, pero el que se aplicará en esta zona turística está basado en el control y monitoreo a través de sensores y cámaras brindando seguridad a los turistas y comerciantes.

El cambio en el sistema de cobro de peaje beneficiará tanto a los turistas como a la comunidad ya que es un sistema que agilizará el cobro de la tarifa de ingreso y determinará un lugar de parqueo el cual contará con la señalética y seguridad respectiva, además que está

incluido un sistema de visión artificial que archivará los números de placas para cualquier evento o posterior uso por parte de los administradores del lugar turístico.

Es importante implementar un sistema de seguridad en la zona de “Deportes Acuáticos” con la finalidad de salvaguardar los bienes materiales de los comerciantes debido a que en la noche están expuestos a robos o daños por personas ajenas a la zona.

CAPÍTULO I

Marco Teórico

1.1 Visión Artificial

La visión artificial, también conocida como la visión por computadora es una rama de tantas que posee la inteligencia artificial, esta rama comprende varios procesos como capturar, procesar e interpretar videos o imágenes obtenidas del mundo real y estos a su vez comprenden etapas conocidas como captación, procesamiento, segmentación, descripción, reconocimiento e interpretación [6].

La etapa de captación es un método o proceso por el cual se obtiene la imagen, esta también puede ser extraída de videos en tiempo real de manera digital [6].

En la etapa de procesamiento existen algunos métodos para filtrar la imagen y adecuarla para el beneficio del sistema [6].

En la segmentación se delimita los patrones a usar o que sean del interés para que en la descripción sean extraídas características necesarias del objeto [6].

En el reconocimiento se compara con los objetos planteados al inicio del proceso y en la interpretación se lo asocia con un entorno real comparándolo con objetos en 3D [6].

Estos sistemas también son aplicados al campo industrial, pero para esto es necesario que sean más robustos, tengan más fiabilidad, precisión, alta resistencia al entorno o a los cambios de temperatura y estabilidad mecánica al momento de procesar las imágenes, estas son aplicadas por el gobierno y las fuerzas armadas debido a que sus costos son bajos y los resultados son excelentes gracias a que actualmente cuentan con sensores digitales, cámaras ópticas especializadas para imágenes y con los softwares necesarios para procesarlas y analizarlas de mejor manera [7].

1.2 Componentes Principales

La visión artificial tiene como objetivo mejorar, brindar seguridad y servicios a la comunidad, por esta razón se puede apreciar como la tecnología evoluciona día a día con el fin de mejorar las herramientas informáticas (hardware y software) [8].

Para esto existen varias compañías encargadas de crear componentes y la vez mejorar a sus predecesores, entre los que están (sensores, cámaras, tarjetas digitalizadoras, computadoras, algoritmos o librerías informáticas) con el objetivo de facilitar el trabajo a realizar y optimizar el mismo [6].

Las mejoras en los dispositivos permiten tener una imagen de alta calidad y una mayor velocidad al momento de realizar el procesamiento de imágenes [6].

Otros componentes importantes son los externos como la escena en donde se captura la imagen ya que esta puede poseer varios objetos a su alrededor que interfieran en la percepción del objeto a buscar o la fuente de luz que esta posee ya que no es lo mismo una fotografía en el día que al anochecer. [6] Los componentes principales son indispensables para este proceso como se muestra en la Fig. 1.



Figura 1 Componentes de un sistema de visión artificial
Fuente [9]

1.2.1 Fuente de iluminación.

La iluminación es un factor importante y determina el resultado del proceso de visión artificial, existen 2 tipos: el ambiente controlado y el ambiente no controlado [6].

El primero se encuentra en lugares cerrados como parqueaderos subterráneos donde la luminosidad siempre será la misma a cualquier hora del día, pero en el segundo caso cambia de acuerdo con la luminosidad natural que se disponga del ambiente determinada por los factores de lluvia, hora del día, sombra, etc [6].

Este proceso es importante porque influye en la nitidez de la imagen obtenida por la cámara para el proceso de visión artificial [6].

1.2.2 Cámara.

La cámara de visión artificial juega un papel importante porque cumple la función de ojos en el sistema y es la encargada de captar y recibir la imagen, este dispositivo tiende a tener varios aspectos como la resolución, tipos de conexión y controlador para las que están diseñadas [6].

1.2.3 Digitalizador.

Este dispositivo no es más que un conversor análogo digital encargado de convertir la señal eléctrica en un código binario para que sea interpretado por la computadora y posteriormente cumpla con los procesos de estudio necesarios o requeridos por el comprador, la tarjeta digitalizadora convierte la señal analógica de la cámara en una señal digital para almacenar la información en la memoria, al igual que las cámaras estas tarjetas son de diferentes características que varían según la necesidad del comprador, existen desde las más básicas hasta las que muestran en pantalla la imagen [6].

1.2.4 Computador.

Actualmente esta es una herramienta de trabajo muy usado por todos a diario con el fin de facilitarnos el trabajo, el hardware y software que estas poseen varían de acuerdo a las condiciones y necesidades de uso del comprador, por lo general para procesos de visión por computador se utiliza las de gama alta con una memoria RAM y procesador de alta velocidad [6].

Complementario a esto tenemos los programas y algoritmos de programación que permiten procesar y extraer e interpretar las imágenes capturadas por la cámara [6].

1.3 Fases del proceso de visión por computador

1.3.1 Adquisición de la imagen.

Consiste en capturar la imagen de una escena o entorno mediante una cámara digital y enviarla a un ordenador para almacenarla y procesarla [10].

1.3.2 Pre procesamiento.

En esta etapa se toma la imagen y se aplica los tratamientos, transformaciones y filtros necesarios para optimizarla eliminando partes innecesarias con el fin de facilitar las etapas posteriores [10].

1.3.3 Detección de bordes.

Permite delimitar de manera detallada la imagen y la parte que se va a utilizar de la misma, por esto es muy importante realizar un pre procesamiento correcto [10].

1.3.4 Segmentación.

Posterior a la selección de la parte de la imagen a utilizar se procede a aislar los elementos y seccionar las regiones de pixeles necesarios para la extracción futura [10].

1.3.5 Extracción de características.

Es un proceso de representación matemática aplicada las partes de las imágenes seleccionadas [10].

1.3.6 Reconocimiento y Localización.

Se agrupan los objetos con algunas características en común y se compara que se asemejen al objeto localizado en 3D, para esto se aplica una técnica de triangulación de los espacios que dará como resultado una única imagen [10].

1.3.7 Interpretación.

Es la etapa final del proceso, aquí se procede a interpretar la escena final tomando en consideración la escena inicial y los cambios encontrados y generados durante los diversos procesos que atravesó la imagen [10].

El procesamiento de imágenes siempre es el mismo como se puede apreciar en la Fig. 2

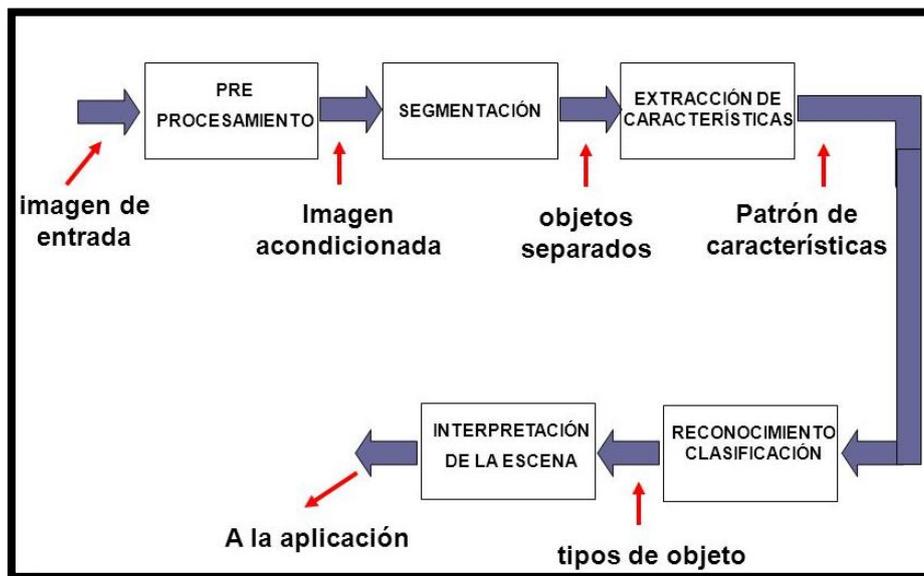


Figura 2 Etapas del procesamiento de imágenes
Fuente [11]

1.4 Tratamiento digital de imágenes

1.4.1 Manipulación de píxeles.

1.4.1.1 Escala de grises.

Esta transformación permite tener la imagen en una resolución con píxeles más adecuados para el procesamiento que se realizará [12]. Como se muestra en la Fig. 3



Figura 3 Muestra una imagen en escala de grises

1.4.1.2 Binarización por Umbral.

Esta transformación permite tener una imagen con detalles más definidos que pueden ser usados para la detección de bordes [12]. Como se muestra en la Fig. 4



Figura 4 Muestra una imagen con un filtro por umbral

1.4.2 Operaciones Matemáticas.

Al igual que en las matemáticas existen las mismas Operaciones básicas: suma (caracterizada o fusionar o sumar dos imágenes en una sola), resta (resta una imagen B de una A), multiplicación (multiplica los valores de pixeles de una imagen por la otra), división (divide los pixeles de una imagen por la otra) [12]. En la Fig. 5 se observa la operación suma entre dos pixeles y a su vez dividiéndose el resultado obtenido

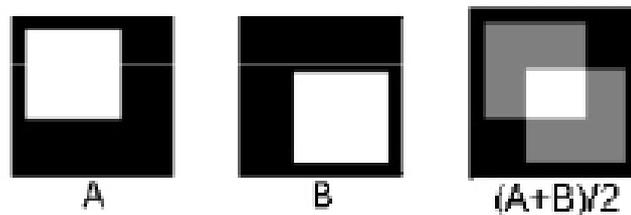


Figura 5 Muestra el resultado de una operación matemática

1.4.3 Operaciones Aritmetológicas.

1.4.3.1 Conjunción.

Operación AND entre los pixeles de la imagen y es usada para borrar pixeles de una imagen. [12] Como se muestra en la figura 6.

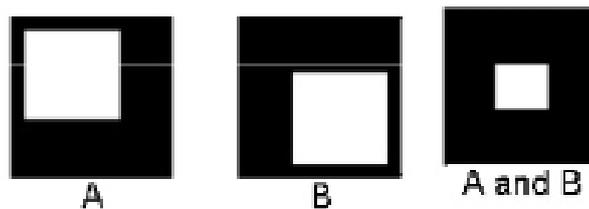


Figura 6 Operación AND entre dos píxeles

1.4.3.2 Disyunción.

Operación OR entre los pixeles de la imagen y es usada para añadir pixeles en una imagen. [12]. Como se muestra en la Fig. 7

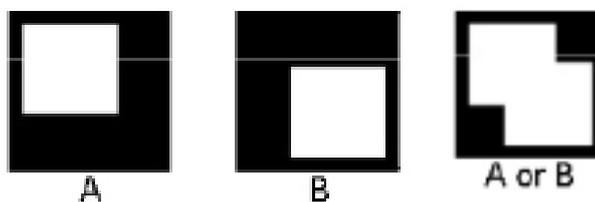


Figura 7 Operación OR entre píxeles

1.4.3.3 Negación.

Inversión de bits de una imagen se utiliza para obtener el negativo de la imagen [12].

Como se muestra en los píxeles de la Fig. 8

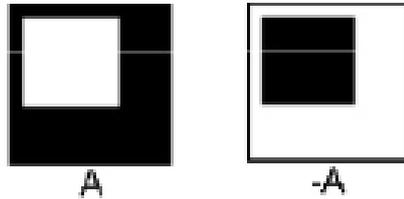


Figura 8 Muestra la negación en un píxel

1.4.4 Operaciones Geométricas.

1.4.4.1 Rotación.

Simula la rotación de la cámara en el objeto, utiliza los parámetros de centro de rotación y ángulo de giro, radio de giro y posición inicial angular [12]. La Fig. 9 muestra la rotación de un tablero de ajedrez.

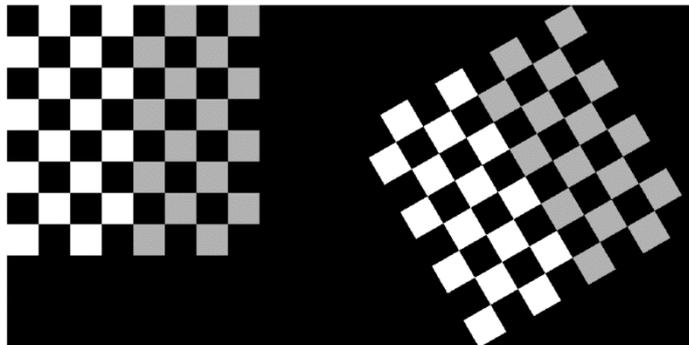


Figura 9 Muestra la transformación geométrica de rotación en un tablero de ajedrez
Fuente: [12]

1.4.4.2 Erosión.

Es la comparación local de una forma, llamada elemento estructural, con el objeto que será transformado. Si, cuando posicionado en un punto dado, el elemento estructural está

incluido en el objeto entonces este punto aparecerá en el resultado de la transformación como se muestra en la Fig. 10, en otro caso no. [12]



Figura 10 Muestra el operador erosión en una imagen
Fuente [12]

1.4.4.3 Dilatación.

Es la comparación local de una forma, llamada elemento estructural, con el objeto a ser transformado. Cuando el elemento estructural es posicionado en un punto dado y toca al objeto como se muestra en la Fig. 11, entonces este punto aparecerá en el resultado de la transformación [12].



Figura 11 Muestra el proceso de dilatación de una imagen
Fuente [12]

1.4.4.4 Suavizado.

Permite reducir la cantidad de variación de intensidad entre píxeles, ayuda a la mejor detección de bordes [12]. Como se muestra en la Fig. 12



Figura 12 Muestra el operador suavizado en una imagen

1.4.4.5 Gaussiano.

Permite emborronar imágenes y eliminar el ruido similar al filtro de mediana [12]. Como se muestra en la Fig.13

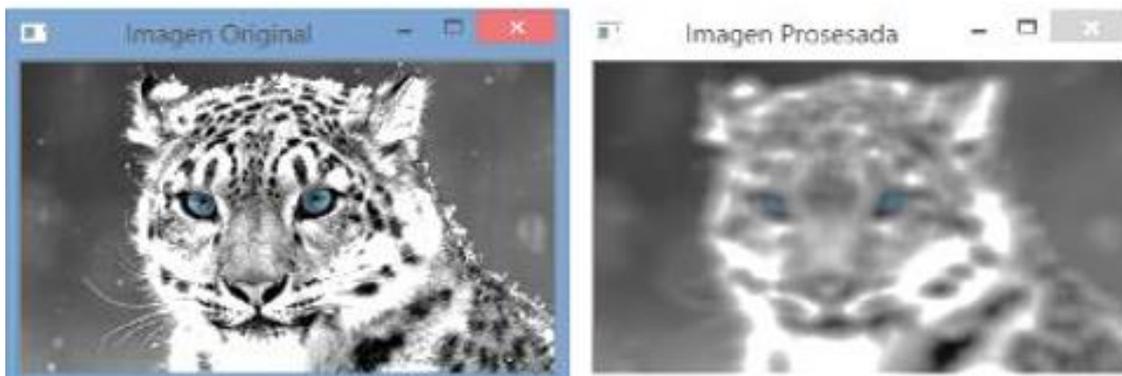


Figura 13 Muestra el operador Gaussiano en una imagen

1.4.4.6 Sustracción de fondo.

Es una técnica para detectar objetos en movimientos a partir de las cámaras estáticas [12]. Como se muestra en la Fig. 14

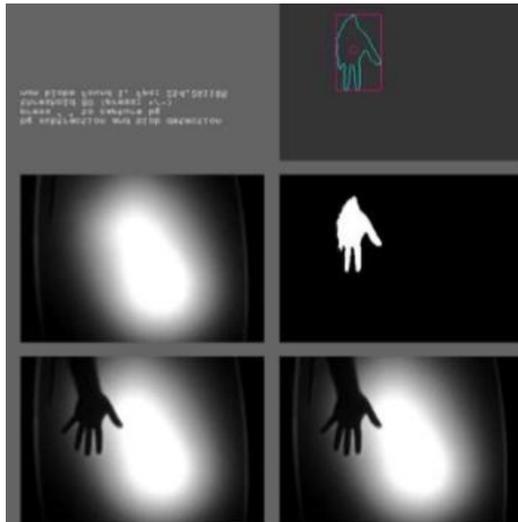


Figura 147 Muestra el proceso de sustracción de fondo en la imagen de una mano
Fuente [12]

1.4.5 Detección de bordes.

1.4.5.1 Operador Canny.

Es una técnica que permite extraer una amplia gama de bordes [12]. Como se muestra en la Fig. 15



Figura 85 Muestra el operado canny en una imagen
Fuente [12]

1.4.5.2 Operador Sobel.

Es una técnica que permite calcular y la aproximación al gradiente de la función de intensidad de una imagen [12]. Como se muestra en la Fig. 17



Figura 96 Muestra el operador sobel en la imagen
Fuente [12]

1.5 Sistema de control de acceso vehicular

Los sistemas de control de acceso vehicular hoy en día se puede encontrarlos en cualquier sitio turístico, condominio, o entradas a diversos lugares de afluencia masiva, estos sistemas tienen elementos sincrónicos que permiten el flujo ordenado de vehículos de entrada o salida a dichos lugares evitando el desorden o el ingreso restringido a personas particulares, así como también el cobro por su uso [13].

Estos son implementados con el fin de tener más seguridad y tener un control en el espacio ya sea público o privado, monitorear el ingreso y salida de vehículos para un posterior registro o análisis correspondiente por parte de los dueños de los inmuebles [13].

Las técnicas que se puede utilizar para este control son varias, sin embargo, han ido evolucionando a través de los años, al principio el ingreso o salida de los vehículos se lo hacía de manera manual, es decir con la ayuda de una persona que registre el mismo en un papel, pero ahora todo ha cambiado [13].

Actualmente este registro se lo hace de manera digital, en las personas es muy común el uso biométrico o huella digital, y en los automóviles con la ayuda de los tags, sistemas RFID, y demás sistemas de cobros automatizados [13]. Un ejemplo claro de cobro de peajes actuales se muestra en la Fig. 17



*Figura 17 Muestra un sistema de acceso vehicular tradicional
Fuente: [14]*

1.6 Sistemas de cobro de peajes

Los sistemas de cobros de peajes son un elemento que permite el control de acceso vehicular y restringir el paso de vehículos no autorizados, la inversión en la implementación de este sistema siempre es recuperada con las tarifas de cobro que imponen a cada uno de los propietarios del inmueble, además estos sistemas son un instrumento que permiten guiar el tránsito de modo confiable y evita el tráfico. Cabe mencionar que existen varias empresas que brindan la instalación de este servicio, así como también su mantenimiento entre otros [15].

Todos estos proyectos primero son sometidos a un análisis económico y un estudio del lugar a ser implementado, además el MTOP es el encargado de la regulación de valores en el peaje para que estos sean justos [16].

Los sistemas de cobro de peajes existentes son: manual, automáticos telepeaje o por medios electrónicos y mixtos [15].

1.6.1 Métodos de cobro.

- Efectivo
- Tarjeta de crédito
- Convalidación de vales pre-impresos
- Convalidación de exentos
- Tarjeta de proximidad con y sin contacto
- Tarjeta multiviaje pre-pago
- Tarjeta cuenta corriente (pre o pospago)
- Tarjeta de vecinos
- Tarjeta de exento
- TAG pre-pago [15]

1.6.2 Sistema de cobro por TAGS.

El sistema de cobro por Tags es un sistema inteligente que permite agilizar el cobro del peaje y el flujo vehicular en las carreteras. Este sistema es de fácil implementación, pero la obra civil que necesita es costosa dependiendo del lugar donde se instale [17].

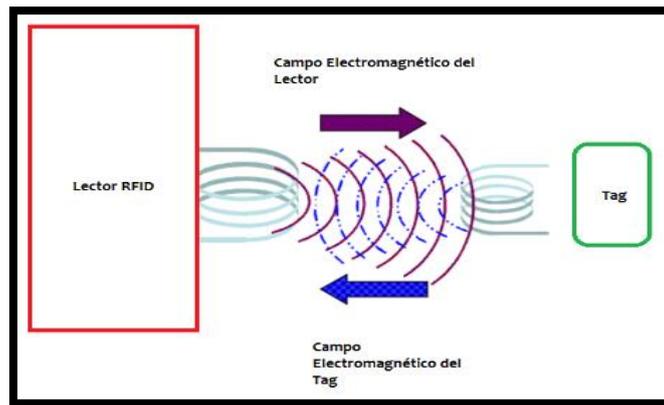
Características

- 1 a 6 m de alcance
- Frecuencia 902 a 9258 MHZ
- Interfaz de comunicación Weigand
- Tarjeta ASC – HF01 / ASC – HF02 [15]

1.7 Tecnología RFID

La tecnología RFID es un método que usa la radiofrecuencia contenida en las tarjetas o llaveros (TAGS), estos poseen una información contenida en etiquetas que al entrar al área de cobertura de un lector RFID envía una señal al receptor para que compare la información y en caso de estar almacenada en la memoria emita una señal activando cualquier actuador conectado al sistema como se muestra en la Fig. 18 [18].

Una de las ventajas de este sistema es que no es necesario tener contacto físico o visual entre las tarjetas y el lector RFID, estas son básicamente programadas según el uso que se les vaya a dar y su instalación es sencillo, además que su costo no es tan elevado [18].



*Figura 108 Muestra el funcionamiento de un sistema RFID
Fuente: [18]*

1.7.1 Componentes.

Los sistemas RFID tiene 4 elementos principales

1.7.1.1 Etiqueta RFID.

Son conocidas como tags o transpondedor, las más utilizadas en el mercado son las tarjetas y los llaveros que son las encargadas de guardar información y datos en el microchip que posteriormente será comparada en el lector, además estas poseen una pequeña antena que permite la radiofrecuencia con el lector [18].

1.7.1.2 Lector.

Es el encargado de transmitir energía y recibir los datos e información que envía la etiqueta, esta cuenta con una unidad de control y una antena de radiofrecuencia para recibir dicha información [18].

Todos estos lectores poseen interfaces estándar que facilitan enviar la información a un subsistema como una computadora o una base de datos, en algunos casos los lectores tienen programa capaz de mejorar las etiquetas [18].

1.7.1.3 Computador.

Puede ser un ordenador o controlador que permite el desarrollo de la aplicación RFID, contiene la información del receptor y emisor y es capaz de transmitir órdenes al lector [18].

1.7.1.4 Middleware

Es un sistema ERP que permiten recoger, filtrar y manejar los datos del sistema [18].

Tipos de RFID

Tabla 1
Características de los RFID

	RFID Pasivo	RFID Activo
Etiqueta de batería	No	Si
Etiqueta fuente de energía	Energía transferida desde el lector	Interno para etiquetar
Disponibilidad de Tag Power	Solo dentro del campo de un lector activado	Continuo
Intensidad de señal requerido	Alto (debe alimentar la etiqueta)	Baja (solo para llevar la información)
Intensidad de señal disponible	Bajo	Alto
Rango de comunicación	Alcance muy corto 3 mo	Largo Alcance 100 mo
Vida útil de la etiqueta	Muy largo	Limitado a la duración de la batería
Tamaño de etiqueta típico	Pequeña	Grande

Colección de etiquetas	Recopila cientos de etiquetas en menos de 3 metros. Recoge 20 etiquetas que se mueve a 8km/h	Recopila miles de etiquetas en 28000 m ² de un solo lector. Recoge 20 etiquetas a mas de 160 km/h
Capacidad del sensor	Capacidad de leer y transferir los valores del sensor solo cuando la etiqueta es alimentada por el lector	Capacidad para monitorear y registrar continuamente la entrada del sensor
Almacenamiento de datos	Pequeño para datos de lectura Bytes	Gran almacenamiento de datos de lectura y escritura Kbytes
Aplicaciones típicas	Procesos comerciales rígidos, movimiento de activos restringido, seguridad básica (detección única de eventos de manipulación), impacto sustancial en los procesos comerciales. Etiquetado individual de artículos, maletas, cajas, cartones, palés, etiquetas impresas	Proceso de negocio dinámico, movimiento de activos sin restricciones, seguridad / detección, almacenamiento / registro de datos Contenedor intermodal, monitoreo del área de vagones, portales de etiquetas múltiples de alta velocidad, aplicaciones de seguridad de carga sofisticadas (detección

		continua de manipulación, sello de fecha / hora), manifiesto electrónico
Costo	Bajo menos de \$0.50	Alto superior a los \$5

Fuente: [19]

1.8 Detección de placas

1.8.1 ANPR

La (Automatic Number Plate Recognition) o el reconocimiento automático de matrículas es un proceso basado en la visión artificial el cual detecta la placa de un vehículo de manera automática y lo transforma de manera digital para un uso determinado por el usuario, actualmente existen varias aplicaciones [14].

Se toma en cuenta que cada país tiene una estructura de placas diferentes, es por esto que cada algoritmo que se desarrolla debe ser basado en las características por zona según las letras y números que esta ocupe [14].

En Ecuador el formato establecido hoy en día es de 3 letras (X) y cuatro números (Y) “XXX-YYYY” [14]. Como se muestra en la Fig. 13



Figura 119 Muestra el formato de placa en Ecuador

Fuente: [14]

1.8.2 Placas de vehículos en Ecuador por provincias

Tabla 2
Letras de placas por provincia en el Ecuador

Nº	PROVINCIA	LETRA
1	Azuay	A
2	Bolívar	B
3	Cañar	U
4	Carchi	C
5	Cotopaxi	X
6	Chimborazo	H
7	El Oro	O
8	Esmeraldas	E
9	Galápagos	W
10	Guayas	G
11	Imbabura	I
12	Loja	L
13	Los Ríos	R
14	Manabí	M
15	Morona Santiago	V

16	Napo	N
17	Pastaza	S
18	Pichincha	P
19	Orellana	Q
20	Sucumbíos	K
21	Tungurahua	T
22	Zamora Chinchipe	Z
23	Santa Elena	Y
24	Sto. Domingo de los Tsáchilas	J

Fuente: [6]

En la tabla 2 se puede apreciar la letra que distingue a cada provincia y a su vez va en cada placa del vehículo que pertenece a Ecuador [6].

Es una norma que los vehículos que transiten por el país deben tener la placa en la parte frontal y posterior del vehículo y que en la noche deban tener luz blanca sobre la placa para facilitar su lectura, los vehículos que no se acaten a estas normas serán multados [6].

1.9 Algoritmos de programación

Los algoritmos de programación son una parte esencial del reconocimiento de placas vehiculares, otro factor importante es la cámara la misma que para optimizar los resultados debe ser infrarroja, esta permite que los pasos como segmentación y detección sean más fáciles [6].

Existen diversos lenguajes que permiten el desarrollo de algoritmos unos más eficientes que otros y otros más veloces que otros, tomando en cuenta la calidad de la imagen y la velocidad de procesamiento del ordenador donde se realiza esta lectura [6].

Un algoritmo debe ser capaz de tomar en cuenta que la luz tiene 3 propiedades básicas como lo son: propagación en línea recta, reflexión al contacto con una superficie y cambia de dirección al ponerse en contacto con la superficie [6].

Por esto la reflexión puede ser difusa o dispersa, pero la mayoría de placas tienen una característica llamada retrorreflexión, es decir un material que hace que la luz no deje la placa y permita la captación de la imagen para el proceso de visión y reconocimiento requerido [6].

1.10 Open CV

Proviene de los términos anglosajones Open Source Computer Vision Library diseñada por Intel y su comercialización es libre por lo que para el desarrollo de proyectos de aula o similares es perfecta [20].

Open CV es una librería de tratamiento de imágenes y usado en la visión por computador en tiempo real, además es compatible con GNU/Linux, Mac OS, y Windows y tiene 500 funciones que permiten el proceso de visión como reconocimiento de objetos, calibración de cámaras, visión estérea y visión robótica [20].

Open CV tiene como finalidad proporcionar un desarrollo fácil de usar y a la vez que sea eficiente, es por esto que su programación está basada en C y C++. Open CV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel [20].

Desde su aparición, se ha utilizado en infinidad de aplicaciones, desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esta librería se compone de 4 módulos: [20].

- Cv: contiene las funciones básicas de la biblioteca.
- Cvaux: contiene las funciones auxiliares (experimental).
- Cxcore: contiene las estructuras de datos y funciones de soporte para álgebra lineal.
- Highgui: funciones para el manejo de la GUI.

Open CV tiene una estructura modular, lo que significa que el conjunto incluye diferentes librerías compartidas o estáticas. Los siguientes módulos están disponibles: [20].

Core

Un módulo compacto que define las estructuras de datos básicas, incluyendo el denso array multidimensional Mat y funciones básicas usadas por otros módulos [20].

Imgproc

Un módulo de procesamiento de imágenes que incluye filtrados lineales y no lineales de imágenes, transformación geométrica de imágenes, conversión de color, histogramas, etc [20].

Video

Un módulo de análisis de vídeo que incluye estimación de movimiento, sustracción de fondos y algoritmos de seguimiento de objetos [20].

Calib3d

Algoritmos geométricos básicos multi-vistas [20].

Features2d

Detección de características más destacadas, descriptores [20].

Objdetect

Detección de objetos y ejemplos de clases predefinidas [20].

Highgui

Un interfaz de uso fácil para la captura de vídeo [20].

Gpu

Algoritmos GPU-acelerados de diferentes módulos Open CV [20].

Otros módulos de ayuda [20].

1.11 Machine Learning

Definido como el estudio de algoritmos informáticos programados, mismos que están con la capacidad de aprender y mejorar basándose en la propia experiencia anterior [8].

Comprende un entorno en donde se enfatiza en un aprendizaje automático, se toma con una introducción a la visión artificial que día a día desarrolla nuevas herramientas y procesos

capaces de mejorar a sus predecesores. Machine Learning según la RAE es adquirir un conocimiento de algo por medio del estudio o la experiencia [8].

Existen 3 tipos de aprendizaje humano

Inductivo

Se obtienen resultados a partir de ejemplos ‘aprendidos’ con anterioridad. Se trata de un aprendizaje basado en la generalización o conclusiones extraídas de un conocimiento ‘aprendido’ en un inicio [8].

Deductivo

Se parte de un conocimiento ‘aprendido’ a priori. Se obtienen resultados a partir de unas premisas ‘aprendidas’ con anterioridad [8].

Transductivo

Consiste en el aprendizaje creativo. Se trata de buscar relaciones comparativas entre ejemplos, ya sea por su forma, tamaño, color, simetría, etc. [8].

Los tipos de aprendizaje mencionados

Supervisado

Se usa los ejemplos ya aprendidos para buscar un patrón similar en todos para poder así generalizar un resultado [14]. Como se muestra en el proceso de la Fig. 20

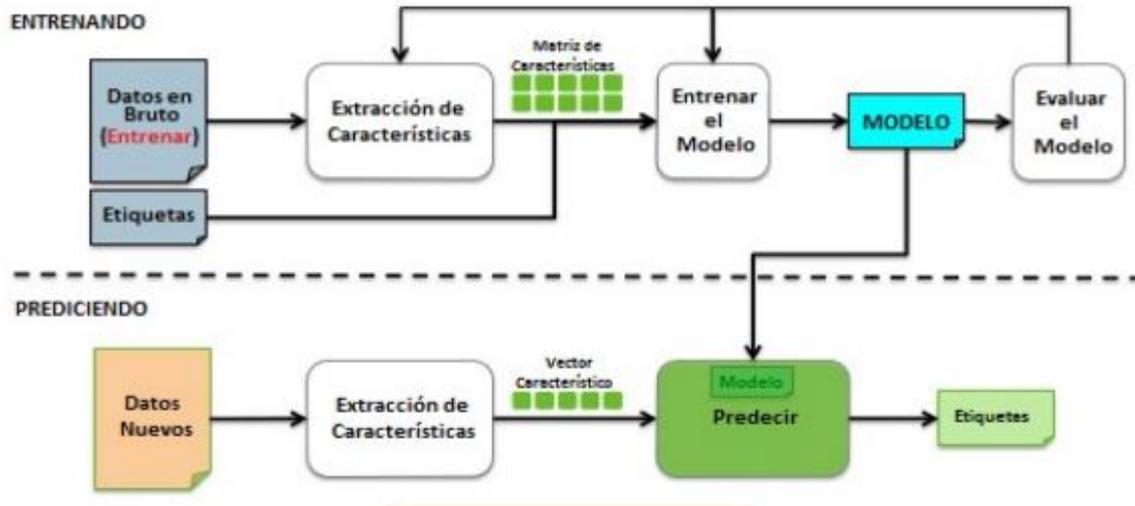


Figura 20 Muestra el proceso a seguir para entrenar un sistema autónomo y eficiente
Fuente: [21]

No supervisado

Es un aprendizaje basado en datos analizados y la estructura se forma desde cero hasta el final [14].

Refuerzo

Se optimiza el resultado con el proceso ensayo – error para optimizar y mejorar los resultados [14].

1.12 Métodos

K-Nearest Neighbour

Vecino más cercano es un algoritmo de clasificación para el aprendizaje supervisado, el cual consiste en buscar la coincidencia más cercana de los datos de prueba en el espacio de características [6].

Los ejemplos que sirven como entrenamiento son utilizados y almacenados, así cuando se requiere clasificar nuevos elementos, se extraen los elementos que cuentan con

características más parecidas y así clasificar al nuevo elemento. El algoritmo vecino más cercano utiliza la distancia Euclidiana sobre los n posibles atributos [6].

K-Means

El algoritmo K-means, creado por MacQueen en 1967 es el algoritmo de clustering más conocido sigue un procedimiento simple de clasificación de un conjunto de objetos en un determinado número K de clusters, K determinado a priori. El nombre de K-means viene porque representa cada uno de los clusters por la media (o media ponderada) de sus puntos, es decir, por su centroide [6].

Deep Learning

Actualmente es una técnica muy usada para el entrenamiento y un aprendizaje automático, está basada en varios algoritmos que buscan asociar patrones y extraer características en común para posteriormente identificar un segmento u objeto [22].

Esta secuencia se la realiza por niveles de manera no lineal [22]. Como se muestra en la Fig. 21



Figura 121 Muestra los estilos de aprendizaje dentro de las redes neuronales

Fuente: [22]

1.13 OCR

Es una técnica de reconocimiento de caracteres, que nos permite pasar un texto por algún dispositivo, este sistema tenga la capacidad de reconocer y comparar los caracteres con una base de datos o el alfabeto y almacenado posteriormente [23].

Este método analiza las letras y números y los compara con una plantilla asignada, una vez verificada compara a la base de datos del servidor y permite o no la entrada de un vehículo al parqueadero. Todos estos sistemas buscan aprender mediante una red neuronal y cada vez ser capaces de reconocer caracteres que estén en diferente ángulo, posición o la claridad de la imagen q este tenga [24]. Como se muestra el proceso en la Fig. 22

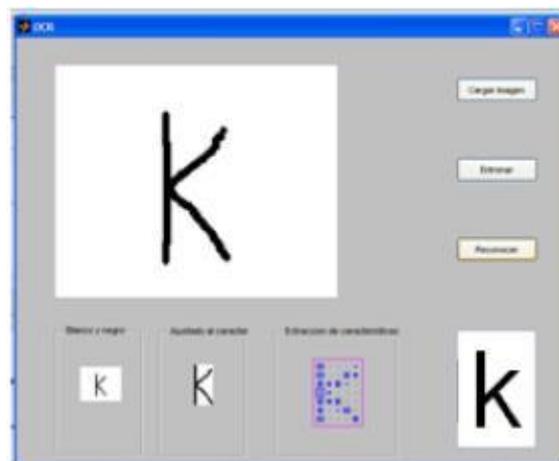


Figura 132 Implementación toolbox neuronal en network Matlab
Fuente: [24]

1.14 Base de datos

Es considerado un repositorio de datos donde se almacena datos con similares características y que posteriormente serán evaluados o analizados [25].

Existen algunas bases de datos:

- MySQL

Tiene un servidor y licencia GPL, es muy veloz, pero en un gran volumen de datos no es recomendable [25].

- **PostgreSQL y Oracle**

Son bases de datos de gran capacidad de almacenamiento y son eficaces, son usadas en Intranets y sistemas de gran calibre [25].

- **Access**

Es una base de datos sencilla y creada dentro del paquete de office para el uso del usuario [25].

- **SQL Server**

Es un sistema de base de datos muy completo, dispone el manejo de grandes cantidades de datos al mismo tiempo y es considerado una alternativa a todos los sistemas antes mencionados [25].

Este sistema soporta transacciones, escalabilidad, estabilidad, seguridad, contiene un potente gráfico de administración permitiendo el uso de comandos DDL y DM, trabaja en el modo cliente - servidor permitiendo que los clientes solo accedan a los datos de red y permite administrar la información desde otros servidores. Todo este sistema requiere una gran cantidad de memoria RAM para su correcto desempeño y funcionamiento [25].

1.15 Sistemas de video vigilancia

Desde la antigüedad ha existido la delincuencia por lo que el afán de los propietarios por buscar un método que brinde seguridad a sus pertenencias siempre ha prevalecido, desde el uso de trampas manuales hasta hoy usar la tecnología para controlar esto [26].

Un sistema de video vigilancia tiene como objetivo prevenir y dar seguridad a los usuarios sobre sus pertenencias materiales o bienes inmuebles, estos sistemas permiten la

monitorización en tiempo real y llevar un registro que se almacena para cualquier uso determinado por el usuario como por ejemplo tener el control de los vehículos que han ingresado y salido de algún lugar en específico o de igual manera de las personas que transitan dentro de una empresa [26].

Los sistemas de video vigilancia están compuestos por una serie de sensores y actuadores conectados entre sí y destinados a vigilar espacios abiertos o cerrados, estos están conectados a un ordenador para su visualización [26].

1.17 Tipos de cámaras

Cámara de red fija

Utilizada para un enfoque directo, este tipo de cámaras son visibles y una vez instalada con el ángulo requerido quedan fijas y estáticas [27]. Como se muestra en la Fig.23



*Figura 143 Cámara de red fija
Fuente: [27]*

Cámara de red fija tipo domo

Son cámaras fáciles y sencillas de instalar, además es pequeña y se lo puede direccionar en cualquier lado, aunque es un poco difícil por el lente que esta posee, otra desventaja es que su lente no es intercambiable, Generalmente se las monta en la pared o el techo [27]. Como se muestra en la Fig. 27



*Figura 154 Cámara de red fija tipo domo
Fuente: [27]*

Cámara de red PTZ

Es de una gama más alta a las anteriores por lo que se traslada y rota en 3 dimensiones, además posee zoom que le permite alejarse o acercarse del objetivo el cual puede seguir, es ideal para interiores y tiene rangos de 10X a 26X [27]. Como se muestra en la Fig. 25



*Figura 165 Cámara de red PTZ
Fuente: [27]*

Cámara de red PTZ no mecánica

Posee una visión angular de 100 a 180 grados y un gran campo de visión, su zoom se mueve de manera no mecánica, esto es una ventaja ya que evita el desgaste de las piezas móviles, son perfectas para ser ubicadas en la pared [27]. Como se muestra en la Fig. 26



Figura 176 Cámara de red PTZ no mecánica
Fuente: [27]

Cámara de red domo PTZ

Tiene un diseño mejorado que permite un giro de 360 grados y una inclinación de 180 grados, estas cámaras son ideales para entornos discretos, y su efectividad es similar a las de 10 cámaras fijas, pero solo un ángulo de visibilidad puede ser monitoreado a la vez, es decir los 9 restantes quedas ciegas [27]. Como se muestra en la Fig. 27



Figura 27 Cámara de red domo PTZ
Fuente: [27]

1.18 Características del lugar turístico

El lugar turístico se encuentra ubicado a las orillas del Lago San Pablo en la comunidad de Araque parroquia San Pablo. [28]

El área es aproximadamente 65.000 m² y se realizó un estudio del lugar con la ayuda de la directiva y profesionales en el tema dando como resultado los requerimientos y necesidades existentes. [28]

CAPÍTULO 2

METODOLOGÍA

2.1 Elaboración de planos CAD del lugar

AutoCAD es un software asistido por computadora que permite dibujar en 2D y modelar en 3D [29], para este proyecto se realizó un plano CAD en 2D con las dimensiones del lugar como se muestra en la Fig.28 para la selección de componentes y adecuaciones necesarias para la posterior implementación del proyecto.

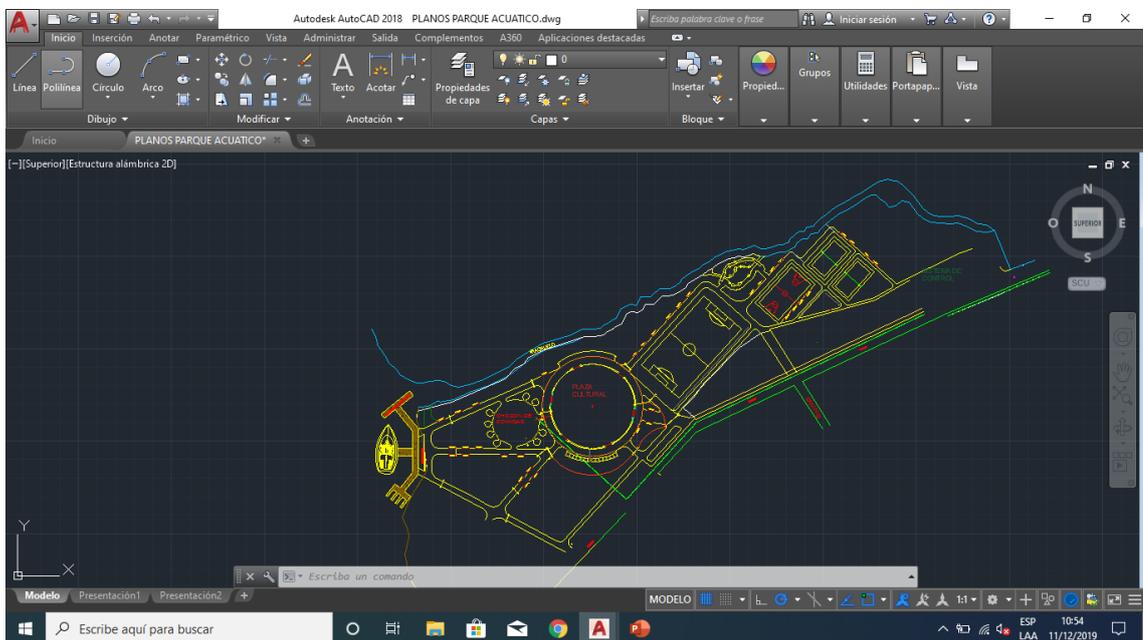


Figura 188. Planos CAD Parque Acuático Araque

2.2 Análisis y selección del sistema de cobro de peajes

Para determinar el mejor sistema de cobro de peajes se realizó un análisis tomando en cuenta las tecnologías actuales y los sistemas que más se usan hoy en día en lugares y situaciones similares a esta.

Existen 5 métodos los cuales son los más utilizados en la actualidad

- Cobro de Peaje Manual (tradicional)

La principal ventaja que posee este sistema es que no necesita mantenimiento y carece de errores al ser un usuario el operador, pero por esta misma situación es el más antiguo y ha sido reemplazado por las máquinas en su mayoría ya que no da la fiabilidad y rapidez que necesita estos sistemas. A continuación, se observa los costos del sistema en la Tabla 3.

Tabla 3
Costo en un sistema de peaje manual.

Costo por Operario	
Semanal	\$35
Mensual	\$140
Anual	\$1680
Feriados	\$400
Alimentación	\$236
Total	\$2316

Fuente: [28]

Este sistema es válido solo fines de semana y días feriados, los demás días están sujetos a la gratuidad por la no rentabilidad del pago diario de una persona.

- Cobro de Peaje por TAGS



Figura 29 Funcionamiento de las TAGS

Fuente: [30]

El sistema de cobro por TAGS es un método muy eficiente que nos permite acceder al lugar sin efectivo y de manera rápida, pero necesita ciertos parámetros para su funcionamiento como lo es las etiquetas en todos los vehículos que ingresen y un sistema de detección del mismo muy eficiente y con una amplia línea de visión directa hacia el vehículo para evitar errores y aglomeraciones. Además, permite llevar un análisis estadístico de todos los vehículos que ingresen. [30] Las principales características de estos sistemas se muestran en la Tabla 4

Tabla 4
Características de los TAGS

Características Principales	
Alcance	1 a 6 m
Frecuencia	902 a 928 MHZ
Interfaz	Weigand
Voltaje	9 a 12 DC
Tarjeta	ASC-HF01 ASC-HF02

Fuente: [30]

El costo del lector de Tags vehicular oscila entre los \$400 dólares aproximadamente.

- Cobro de Peaje por Monedero Electrónico

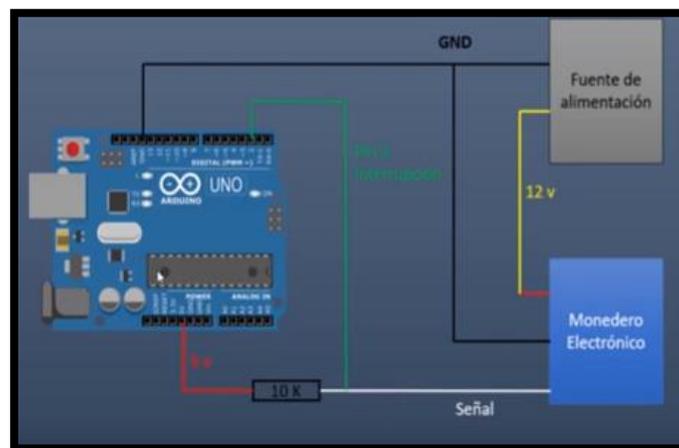


Figura 30 Diagrama de conexión de un monedero electrónico

Fuente: [31]

Es un método semi-tradicional que permite ingresar al lugar depositando una moneda equivalente al valor o costo, mismo que es tarifado por el dueño del inmueble. Este sistema permite un mejor manejo del flujo vehicular pero no es 100% fiable debido a que el ingreso de monedas falsas u objetos similares activarían el sistema, su instalación no tiene mucha complejidad y su costo es mínimo, también permite llevar un registro de la cantidad de vehículos ingresados por día, mes o año según desee el propietario [31]. A continuación, se detallan las características principales de este sistema.

Tabla 5
Características Monedero electrónico

Características Principales	
Alimentación Arduino	5 V
Fuente de Alimentación (Monedero Electrónico)	12V
Impresora de ticktes	110 V (Contador Automático, sensor de fin de papel, Matriz de puntos en serie de 9 Pines)
Voltaje	9 a 12 DC
Tarjeta	ASC-HF01 ASC-HF02
Pulsador	Manual

Fuente: [31]

Costo de implementación aproximadamente \$250 solo el sistema que incluye (Arduino, fuentes de Alimentación, Impresora de tickets y componentes de conexión)

- Cobro de Peaje por Lector QR



Figura 31 Lector código QR
Fuente: [32]

El sistema de cobro por lector QR requiere un poco más de tecnología, en este caso es necesario un teléfono con Android para acceder al Código QR e imprimir el ticket de acceso y a su vez con el mismo ticket habilitar la salida del lugar turístico, este sistema también nos permite llevar un registro y agilizar el flujo vehicular [32]. A continuación, se muestra las características principales en la Tabla 6.

Tabla 6.
Características de un lector QR

Características Principales OCBS-T2001	
Material	Plástico 228g
Dimensiones	7.3 x 8.2 x 10.2
Alimentación	5V
Lectura	200 x seg
Conexión	USB
Protocolos	UPC/EAN 128/30

Fuente: [32]

El costo aproximado del sistema es de \$250 dólares incluyendo la Impresora de tickets

- Cobro de Peaje por Lector RFID

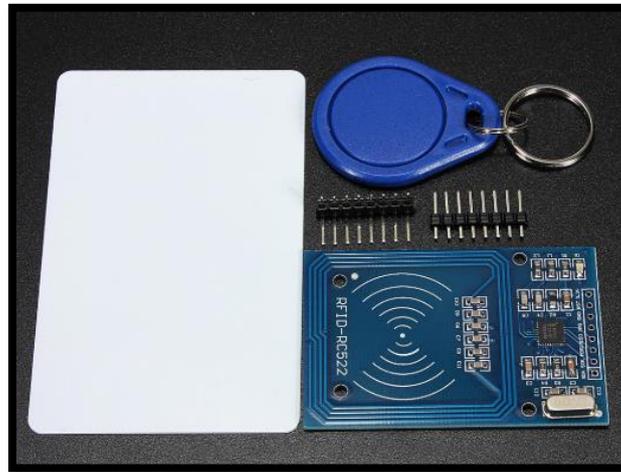


Figura 192 Componentes de un sistema RFID
Fuente: [18]

La tecnología RFID está basada en la radiofrecuencia y tiene como componentes principales un detector o lector de tarjetas y las mismas tarjetas, este sistema es muy eficiente permitiendo agilizar el cobro vehicular y su flujo además que no requiere una línea directa de visión, tiene una larga vida útil, fácil de instalar y dar mantenimiento, además permite llevar un registro de ingresos de vehículos para el uso del propietarios [18]. A continuación, se muestra las características de los componentes RFID.

Tabla 7.
Componentes RFID

Características Principales Módulo	
Alimentación	3.3V
Conexión	SPI
Frecuencia	13.56 MHZ
Características Tarjeta	
Material	PVC 40x60mm

Frecuencia	125 KHZ
Lectura	0 a 60mm

Fuente: [18]

El costo de este sistema está valorado en \$140 dólares incluido el Arduino y cables de conexión

Evaluando los factores determinantes en cada sistema de cobro de peaje se obtiene los siguientes resultados mostrados en la Tabla 8.

Tabla 8
Análisis de factores de los diferentes sistemas de cobro de peajes.

FACTORES	PESO RELATIVO (%)	MANUAL		TAGS		RFID		M.ELECTRÓNICO		LECTOR QR	
		A	B	C	D	E					
BAJO COSTO	20	1	0,2	2	0,4	5	1	4	0,8	3	0,6
FÁCIL INSTALACIÓN	25	5	1,25	2	0,5	4	1	4	1	3	0,8
FÁCIL MANTENIMIENTO	10	5	0,5	5	0,5	4	0,4	3	0,3	5	0,5
COBRO DE PEAJE RÁPIDO	25	1	0,25	5	1,3	4	1	4	1	5	1,3
TARIFA ÚNICA	10	5	0,5	1	0,1	1	0,1	5	0,5	1	0,1
TARIFA PREFERENCIAL	10	1	0,1	3	0,3	5	0,5	1	0,1	3	0,3
TOTAL				2,8	3,1	4		3,7		3,5	

La propuesta que más se ajusta a los requerimientos del cliente y el presupuesto presentado es el Sistema de ingreso con Tarjetas RFID.

2.3 Implementación del sistema

Para la implementación de este sistema es necesario adquirir varios componentes:

- Arduino nano
- Micro Servomotor
- Cables de conexión
- Módulo RFID RC522
- Tarjetas de proximidad

El sistema consta de 3 partes importantes, el generador de señales de radiofrecuencia que es el encargado de generar el campo magnético, un receptor/detector de señales que son los encargados de dar la respuesta y el microcontrolador que es el que controla los componentes y procesa la información [18].

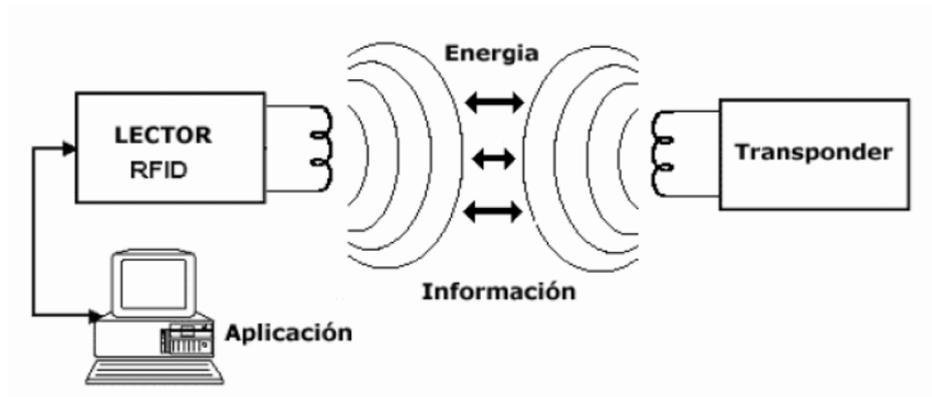


Figura 203 Funcionamiento de un sistema RFID
Fuente: [18]

Existen dos tipos de sistemas RFID, los pasivos y activos. Su principal diferencia radica en la batería interna que estos poseen permitiendo mayores rangos de distancia para su detección [33].

En esta ocasión se usará el módulo RFID RC522 que es un lector grabador con 13.56 MHz, posee comunicación SPI por lo que nos permite trabajar fácilmente con el microcontrolador Arduino nano y las tarjetas Tag RFID que permiten la conexión a menos de 10cm estas poseen un número de serie única y tiene una larga vida útil, la Frecuencia que estas manejan es de 13.56 MHz, protocolo ISO14443A a una velocidad de 106KBPS [34].

2.3.1 Esquema de conexión

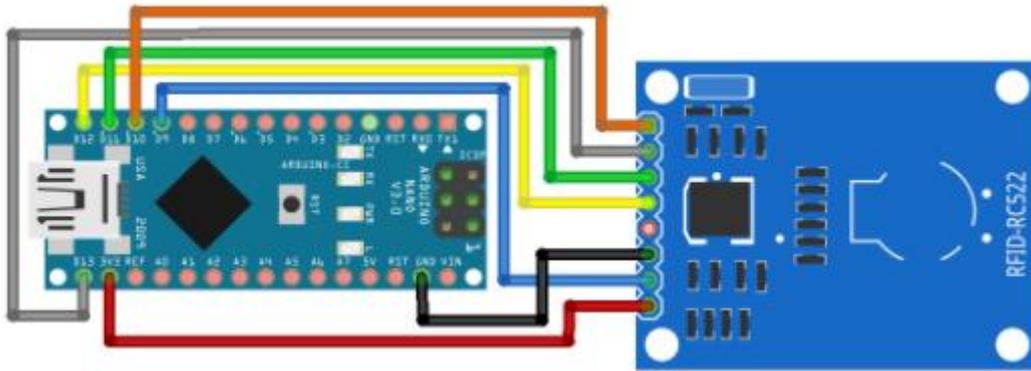
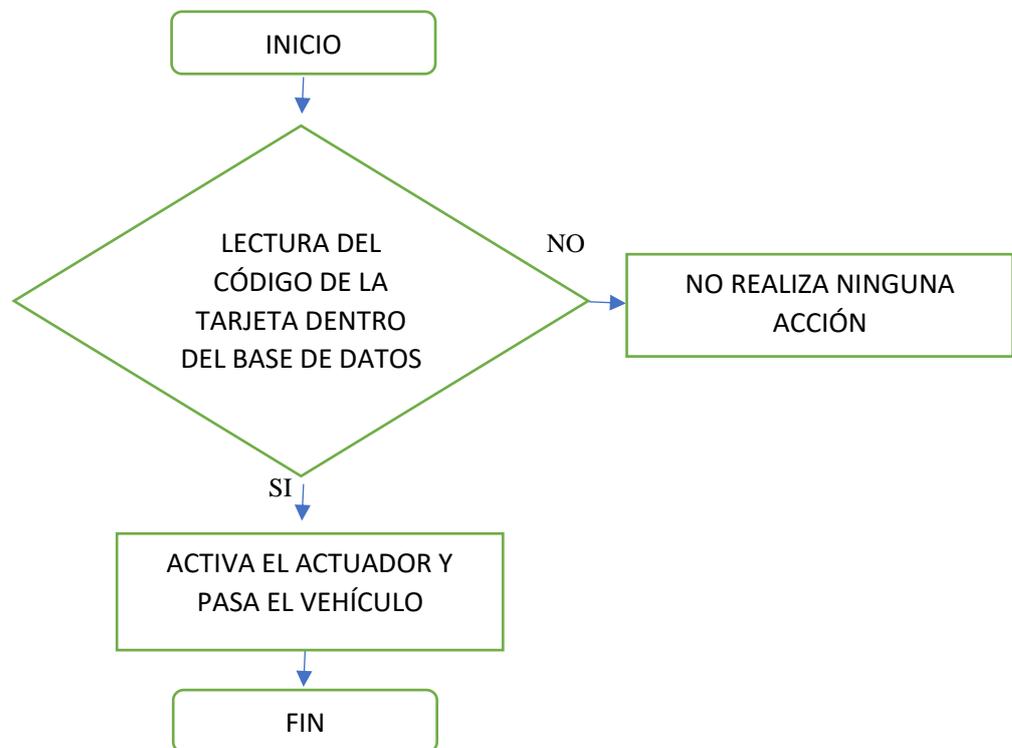


Figura 214 Diagrama de conexión RFID y Arduino NANO
Fuente: [34]

2.3.2 Diagrama de flujo del funcionamiento



2.4 Selección de los componentes para la detección de placas vehiculares

- Raspberry PI4 4gb de Ram

La elección de este componente fue por su fácil instalación, bajo costo y alta velocidad de procesamiento que tiene y requiere para el procesamiento de imágenes.

- Python (Lenguaje de programación)

Es un lenguaje abierto que nos permite utilizar librerías y herramientas que facilitan el proceso.

- Cámara Hilook THC-B110-P

Es una cámara de alta resolución y fácil de instalar junto con el sistema de seguridad que también se va a implementar en el mismo lugar.

2.4.1 Adquisición de la Imagen

Para la adquisición de la imagen se utilizará como base un lugar turístico donde la afluencia de vehículos es considerable como se muestra en la Fig. 35.



Figura 35 Paradero Turístico "Don Pablo"

Para la adquisición de la imagen se debe tomar en cuenta algunos factores importantes como:

- El color, forma y tamaño del vehículo
- El ambiente en el que se encuentra

El procedimiento para obtener el número de placas de los vehículos es el siguiente:

2.4.2 Deep Learning para Detectar el Vehículo

Este método o proceso está familiarizado a la inteligencia artificial, y está enfocado a las redes neuronales convolucionales para la detección o clasificación de objetos con el objetivo de desarrollar una secuencia de algoritmos capaces de aprender una tarea en específico [22].

Este método intenta emular la inteligencia humana asociada al reconocimiento de objetos y patrones para detectar un objeto específico, en este caso se usará para la detección de vehículos los cuales tienen patrones definidos como espejos, focos, chasis, etc [8].

Primero se procede a que el sistema reconozca que es un vehículo, esto se lo hace debido a que cualquier cosa que pase cerca de la cámara sea persona, animal, motocicleta u otro objeto puede alterar o activar el funcionamiento del sistema.

Después se realiza un tratamiento de imágenes para convertirlas según nuestro requerimiento, en este caso se lo binariza para el posterior proceso que se vaya a realizar. Como se muestra en la Fig.36

El método más eficaz es la detección de contornos para que sean comparados, entrenados y el sistema sea capaz de reconocer que es un vehículo como se muestra en la Fig. 37

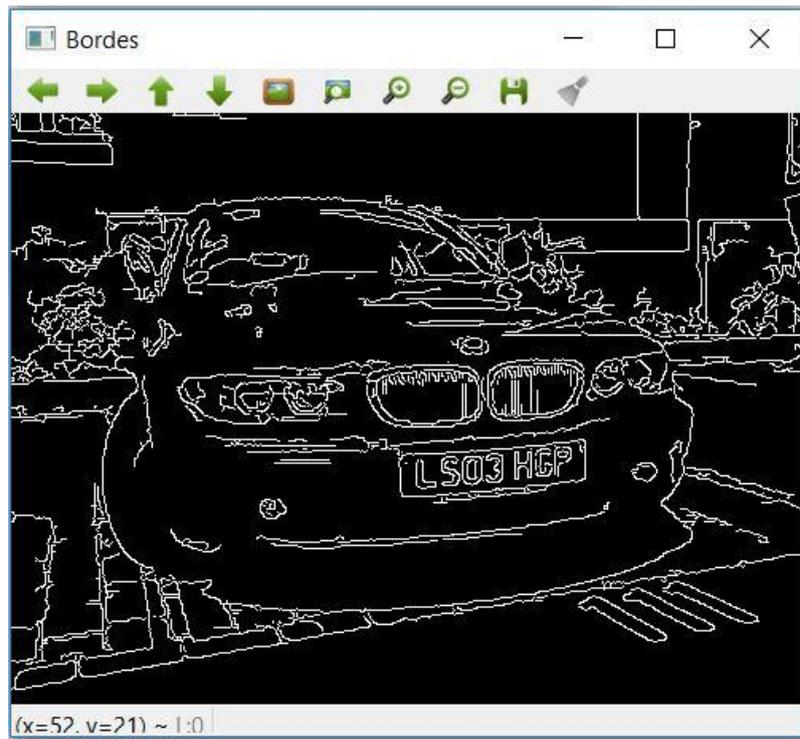


Figura 226 Tratamiento de imágenes por binarización



Figura 3723 Deep Learning para detectar vehículos

2.4.3 Deep Learning para detectar la placa

Después se procede a tratar la imagen con varios filtros hasta delimitar los requerimientos, en este caso es buscar contornos o figuras rectangulares. Como se muestra en la Fig. 39

A continuación, se extrae la placa en formato jpg para su posterior análisis. Como se muestra en la Fig.39



Figura 38 Deep Learning para detectar placas.



Figura 249 Extracción de la placa en formato JPG.

2.4.4 OCR Detección de Caracteres.

Se realiza este método o procedimiento con la única finalidad de obtener la imagen de la placa como se muestra en la Fig. 40 y mediante dicho proceso sea transformada como se muestra en la Fig.41 para posteriormente sea transformado en string para almacenarla en una base de datos para cuantificar información requeridas por los beneficiarios del proyecto.



Figura 40 Extracción imagen JPG del vehículo

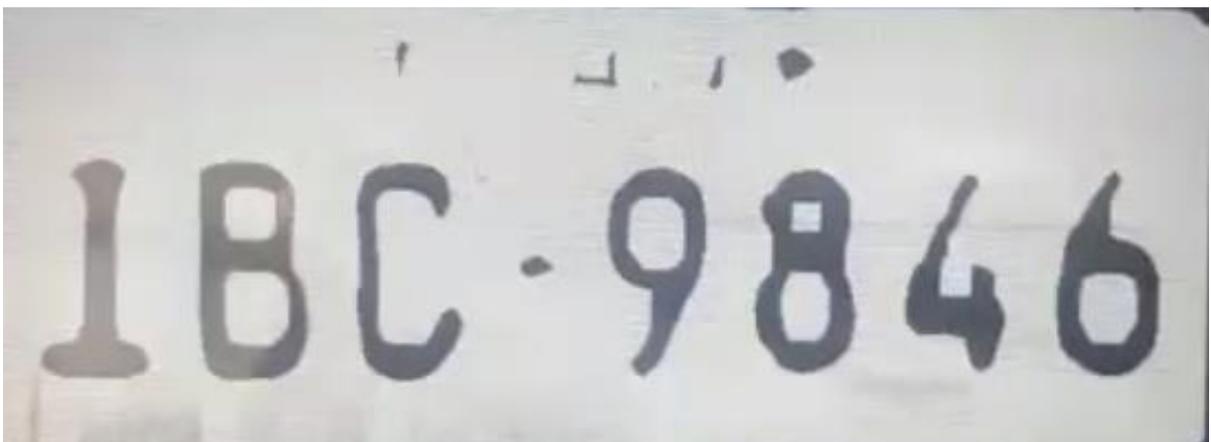


Figura 41 Placa procesada por el método OCR

2.4.5 Creación de la Base de datos.

La base de datos para este sistema es sencilla debido a que los requerimientos del cliente no son tan altos, los parámetros que ellos creen pertinente manejar son: la hora de ingreso, la hora salida y la placa del vehículo ingresado como se muestra en la Fig. 42 para llevar un registro mensual y anual de los visitantes que acuden al lugar turístico.

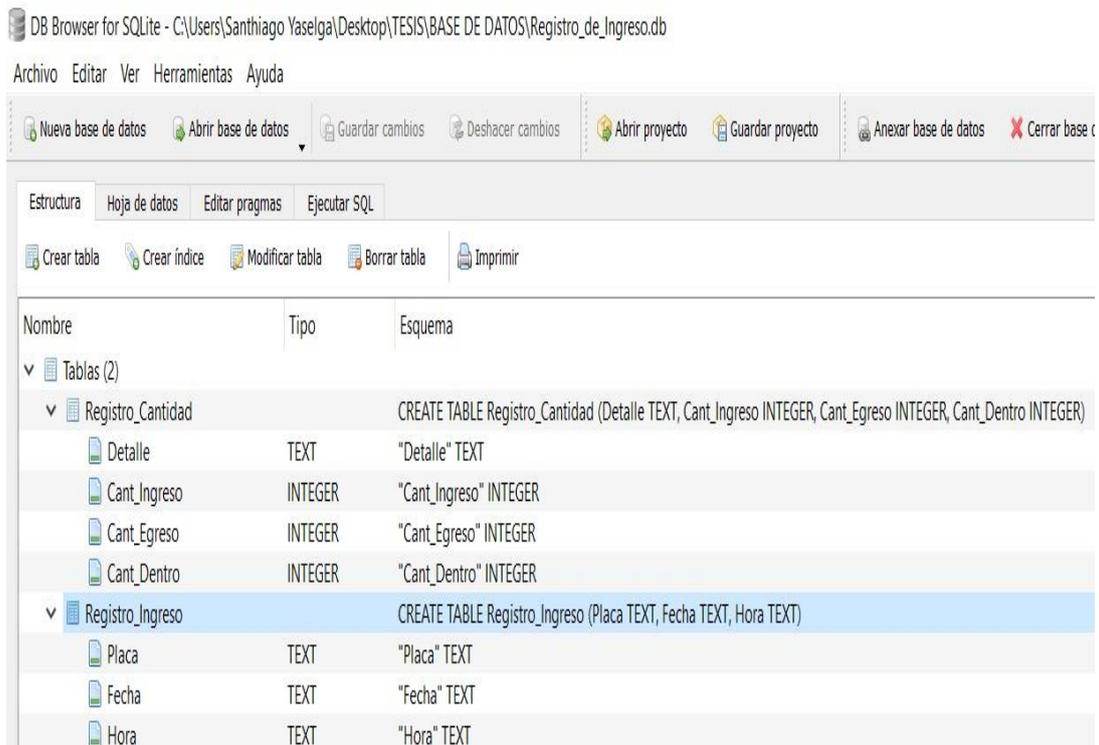


Figura 42 Parámetros de la Base de Datos

La interfaz presentada al usuario es básica para su manejo, el único ingreso que solicita la base es la placa del vehículo. Como se muestra en la Fig. 43



Figura 43. Interfaz del sistema de parqueo

Para la constancia de los vehículos que ingresen o salgan el sistema cuenta con un mensaje que valida el número de placa y valida la operación como exitosa. Como se muestra en la Fig. 44

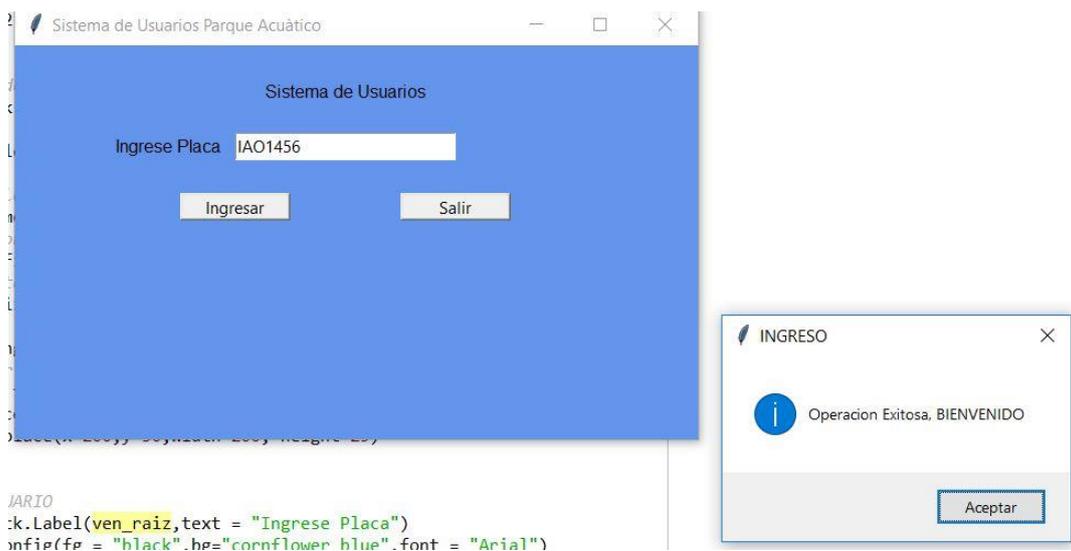


Figura 254 Verificación del sistema de ingreso vehicular

2.5 Análisis de los componentes del sistema de seguridad

La principal desventaja de realizar un sistema desde cero es que se necesita una computadora en la cual crear la cuenta para el direccionamiento y la visualización de las cámaras, además por otra parte toca crear una aplicación que permita interactuar al usuario con el sistema.

La ventaja de adquirir los componentes de un sistema en línea como es HikVision es que poseen su propio Software el cual se lo programa según los requerimientos del cliente, además que posee su propia aplicación para descargar en cualquier teléfono y usarlo, cabe recalcar que esta línea de productos ofrece garantía en cada uno de sus productos caso q no existe en el otro sistema.

El costo varía en cada sistema y su diferencia es representativa del uno con respecto al otro como se muestra en las Tablas 9 y 10.

Tabla 9.
Presupuesto para un sistema de seguridad convencional

Sistema de seguridad convencional			
Cant.	Descripción	V.Unitario	V.Total
1	Computadora	\$500	\$500
1	Disco Externo	\$70	\$70
1	Cuenta Home Camera	\$0	\$0
1	Cableado	\$50	\$50
4	Cámaras	\$25	\$200
1	Aplicación	\$0	\$0
			\$820

Tabla 10
Análisis de un Sistema de Seguridad HikVision

Sistema de seguridad HikVision			
Cant.	Descripción	V.Unitario	V.Total
1	DVR	\$100	\$150
4	Disco Externo	\$70	\$70
4	Cámaras	\$25	\$100
1	Aplicación	\$0	\$0
			\$320

Como se puede apreciar en las dos tablas el costo de las cámaras no varía mucho independientemente de la resolución y si su conexión es alámbrica o inalámbrica, además que el costo de sensores y alarmas es similar para los dos casos.

Pero en cuanto a la implementación el sistema convencional no es factible debido al presupuesto que se estima para el proyecto, además de tomar en cuenta que es importante saber utilizar los recursos ya existentes para la obtención de mejores resultados.

2.7 Instalación de los componentes de seguridad

Las cámaras seleccionadas para el montaje del sistema son:

Hi Look THC-B110-P

- 1MP high- performance CMOS
- 1280 x 720 resolución
- 2.8mm 3.6mm 6 mm fixed lens
- Day/Night switch
- Alcance 20m
- IP66

Hi Look THC-T110-P

- 1MP high- performance CMOS
- 1280 x 720 resolución
- 2.8mm 3.6mm 6 mm fixed lens
- Day/Night switch
- Alcance 20m
- Switchable TVI/AHD/CVI/CVBS

CAPITULO 3

PRUEBAS ANÁLISIS Y RESULTADOS

3.1 Pruebas de Funcionamiento del Sistema RFID

Una vez determina el proceso para el sistema de cobro de peajes y evaluando los algoritmos de programación desarrollados e implementados se procede al validar su correcto funcionamiento por dos etapas.

Prototipado y simulación de un sistema RFID con un actuador (servomotor)

El diseño fabricado mediante impresión 3D con el Material PLA mostrado en el Anexo 1 cumple con los requisitos de conexión a escala necesarios para un posterior montaje con un actuador (barrera de paso vehicular) que en la Fig. 45 esta representada por un Servomotor.

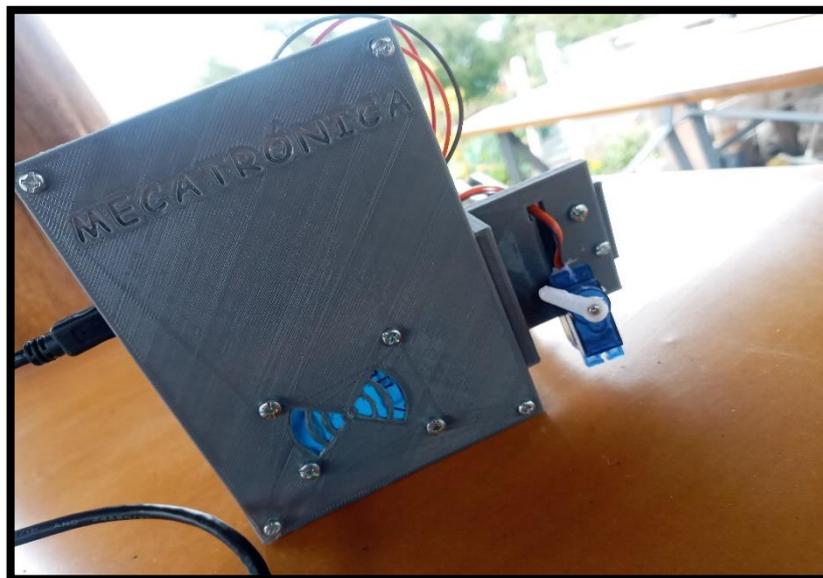


Figura 45 Conexión del sistema RFID con un Servomotor.

Activación del Servomotor mediante las TAGS programadas.

Los algoritmos de programación planteados en el sistema son respaldados y validados mediante unas tarjetas TAGS como se muestran en la FIG. 46 que activan el actuador (Servomotor) permitiendo el ingreso al lugar turístico. Como se muestra en la Fig. 48 o la vez

no activando el actuador (servomotor) como se muestra en la Fig. 47 evitando que se ingrese al lugar turístico, también se debe considerar que si la tarjeta no esta dentro de la base de datos programada tampoco activara el actuador.

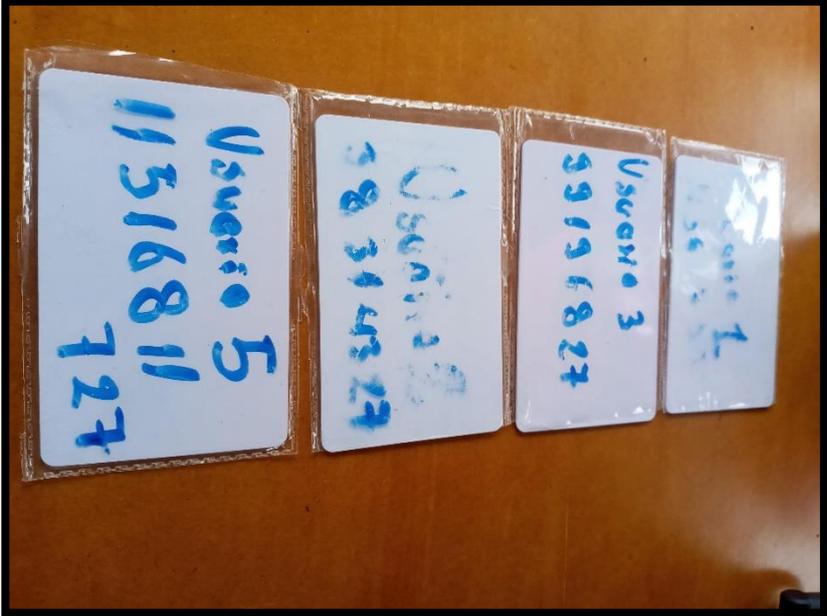


Figura 266 Tarjetas TAG para el sistema RFID



Figura 277 Servomotor Off



Figura 288 Servomotor ON activado por la Tarjeta RFID

3.2 Pruebas de Funcionamiento del Sistema de detección de placas.

Una vez determinado el proceso para la detección de placas mostrado en el Capítulo 2 y evaluando el funcionamiento los algoritmos planteados se puede considerar varios factores importantes para un adecuado funcionamiento del sistema. Para esto se ha realizado varias pruebas con vehículos ingresando al lugar turístico. Como se muestra en la Fig. 49, 50, 51,52,53,54.



Figura 299 Vehículo 1 ingresando al lugar turístico



Figura 50 Vehículo 2 ingresando al lugar turístico



Figura 51 Vehículo 3 ingresando al lugar turístico



Figura 302 Vehículo 4 ingreso al lugar turístico



Figura 53 Vehículo 5 ingreso al lugar turístico



Figura 54 Vehículo 6 ingresando al sector turístico

Previo a la creación del sistema se realizó pruebas con una base de datos de 1000 vehículos dentro del data set para entrenar el sistema. Como se muestra en la Fig.52



Figura 55 Base de datos de la Red

La evaluación del Deep Learning para detección de vehículos el sistema tiene una eficiencia del 98.3% como se muestra en la Tabla 11 debido a que dentro del entrenamiento de la red neuronal se usó 1000 vehículos con diferentes factores como:

- Ambiente de luminosidad del día entre un rango de horas de 07:00 hasta las 17:00 que es el horario que abre el lugar turístico
- El color la carrocería de los vehículos como son (rojo, negro, blanco, entre otros), tomando en cuenta que con el último se tuvo más problemas debido al color y la similitud del color de las placas que también son blancas en el Ecuador.
- El ángulo e inclinación de la cámara así como la distancia de la misma al momento de detectar el vehículo que varía entre 1 y 3 metros de distancia.

Además, se debe tomar en cuenta que el vehículo es un objeto grande por lo que la distancia del sistema alcanza hasta los 10 metros para detectarlo sin problema y en este caso el vehículo para su ingreso se coloca a máximo 3 metros de distancia.

Tabla 11 Análisis Deep Learning para detección de vehículos

Vehículo	Ambiente	Placa	Distancia	Deep Learning Automóvil	Eficiencia (98.3%)
1 (rojo)	No controlado (10 am)	IBC-9648	1 metro	Resultado Positivo	99%
2 (gris)	No controlado (7 am)	PPA-8884	2 metros	Resultado Positivo	98%
3 (blanco)	No controlado (12 am)	IAI-574	1 metro	Resultado Positivo	99%
4 (negro)	No controlado (5 pm)	IBB-1844	1 metro	Resultado Positivo	98%
5 (blanco)	No controlado (3 pm)	IBD-8758	3 metros	Resultado Positivo	98%
6 (blanco)	No controlado (3 pm)	PZZ-865	4 metros	Resultado Positivo	98%

La evaluación del Deep Learning para detección de placas tiene una eficiencia del 94.3% basado en la media aritmética tomada de todas las eficiencias de los vehículos procesados como se muestra en la Tabla 12, para esto se entrenó una red neuronal con 1000 vehículos en el data set dando como resultados los siguientes aspectos.

- La luminosidad del ambiente es importante para obtener una imagen de la placa con alta resolución para el siguiente proceso.
- La distancia de operación es más corta debido a que la placa es un objeto mas pequeño y está dentro del chasis delantero el mismo que tiene varias facciones similares en su entorno.

Tabla 12 Análisis Deep Learning para detección de placas vehiculares

Vehículo	Ambiente	Placa	Distancia	Deep Learning Placa	Eficiencia (94.3%)
1 (rojo)	No controlado (10 am)	IBC-9648	1 metro	Resultado Positivo	99%
2 (gris)	No controlado (7 am)	PPA-8884	2 metros	Resultado Positivo	98%
3(blanco)	No controlado (12 am)	IAI-574	1 metro	Resultado Negativo	85%
4(negro)	No controlado (5 pm)	IBB-1844	1 metro	Resultado Positivo	98%
5(blanco)	No controlado (3 pm)	IBD-8758	3 metros	Resultado Positivo	90%
6(blanco)	No controlado (3 pm)	PZZ-865	4 metros	Resultado Positivo	96%

El OCR para detección para vehículos tiene un rendimiento del 81.16% como se muestra en la Tabla 13. Los principales factores para estos resultados son:

- La calidad y nitidez de la imagen extraida en el proceso previo de extracción.
- La cantidad de vehículos dentro del data set para su entrenamiento

El rendimiento se puede aumentar con duplicar o triplicar el data set de la red neuronal

Tabla 13 Análisis OCR para detección de caracteres

Vehículo	Ambiente	Placa	Distancia	OCR Placa	Eficiencia (81.16%)
1 (rojo)	No controlado (10 am)	IBC-9648	1 metro	Resultado Positivo	99%
2 (gris)	No controlado (7 am)	PPA-8884	2 metros	Resultado Positivo	98%
3(blanco)	No controlado (12 am)	IAI-574	1 metro	Resultado Negativo	25%
4(negro)	No controlado (5 pm)	IBB-1844	1 metro	Resultado Positivo	92%
5(blanco)	No controlado (3 pm)	IBD-8758	3 metros	Resultado Positivo	80%
6(blanco)	No controlado (3 pm)	PZZ-865	4 metros	Resultado Positivo	93%

Es importante considerar en ángulo como se muestra en la Fig. 56 y la calidad de la imagen para un correcto análisis y extracción de la imagen como en la Fig. 58, y posterior enlace a la base de datos. Como se muestra en la Fig. 59

La eficiencia mostrada del sistema es de un 91.3% dando resultados positivos con los vehículos evaluados.

Tabla 14 Análisis del sistema completo

Vehículo	Placa	DL Automóvil	DL. Placa	OCR Placa	Eficiencia (91.3%)
1 (rojo)	IBC-9648	99%	99%	99%	99%
2 (gris)	PPA-8884	98%	98%	98%	98%
3(blanco)	IAI-574	99%	85%	25%	70%
4(negro)	IBB-1844	98%	98%	92%	96%
5(blanco)	IBD-8758	98%	90%	80%	89%
6(blanco)	PZZ-865	98%	96%	93%	96%

Dentro de la evaluación del Deep Learning para detección de vehículos el sistema tiene una eficiencia del 98.3 debido a que dentro del entrenamiento de la red neuronal se usó 1000 vehículos con diferentes ambientes de luminosidad, además se debe tomar en cuenta que el vehículo es un objeto grande por lo que la distancia del sistema alcanza hasta los 10 metros

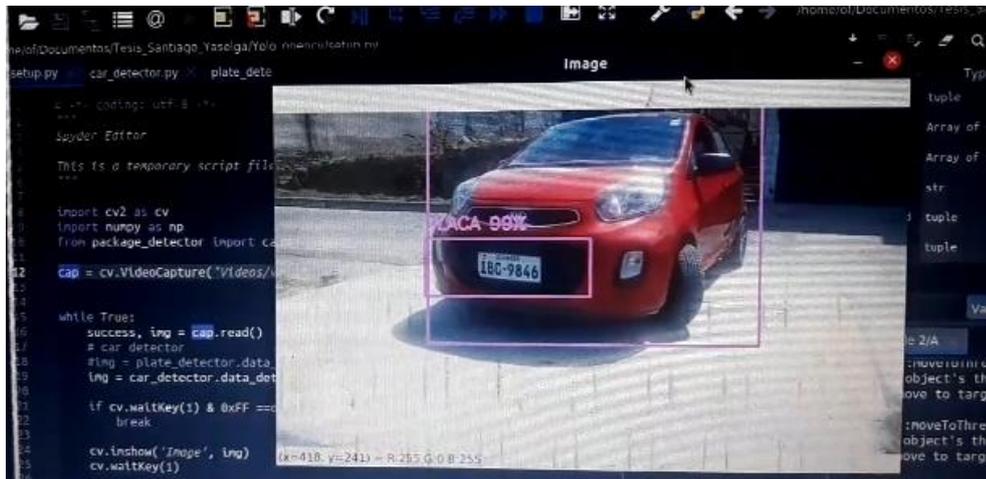


Figura 56 Imagen con el ángulo adecuado para la extracción de imagen.



Figura 317 Deep Learning Detección de placa

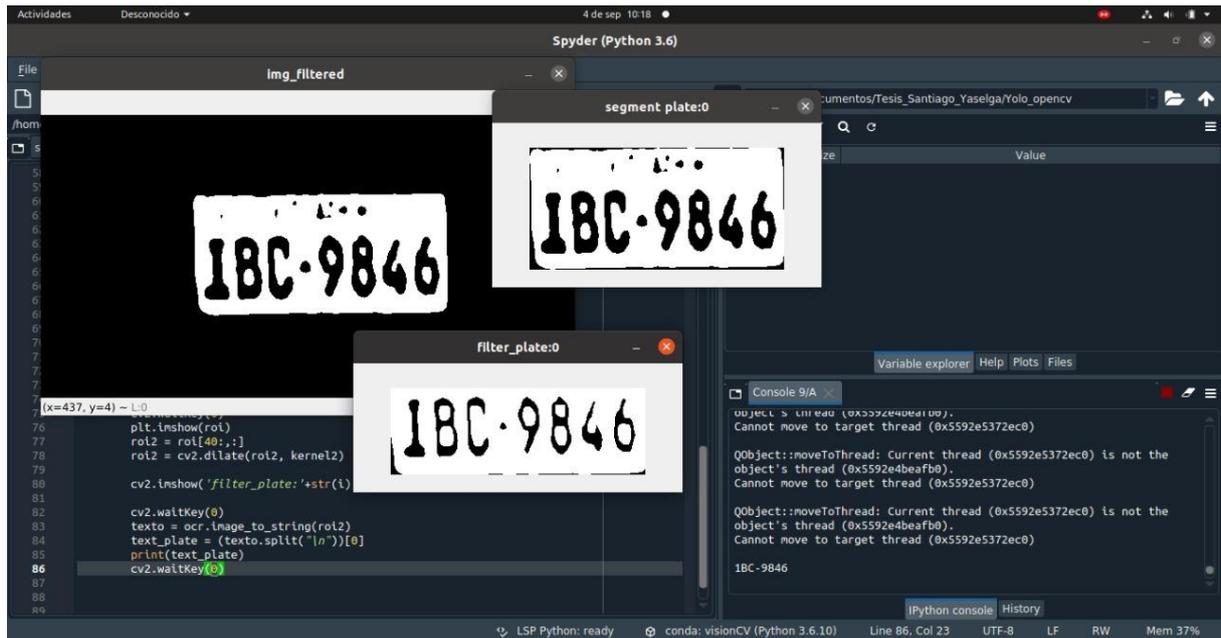


Figura 58 OCR Placas vehiculares

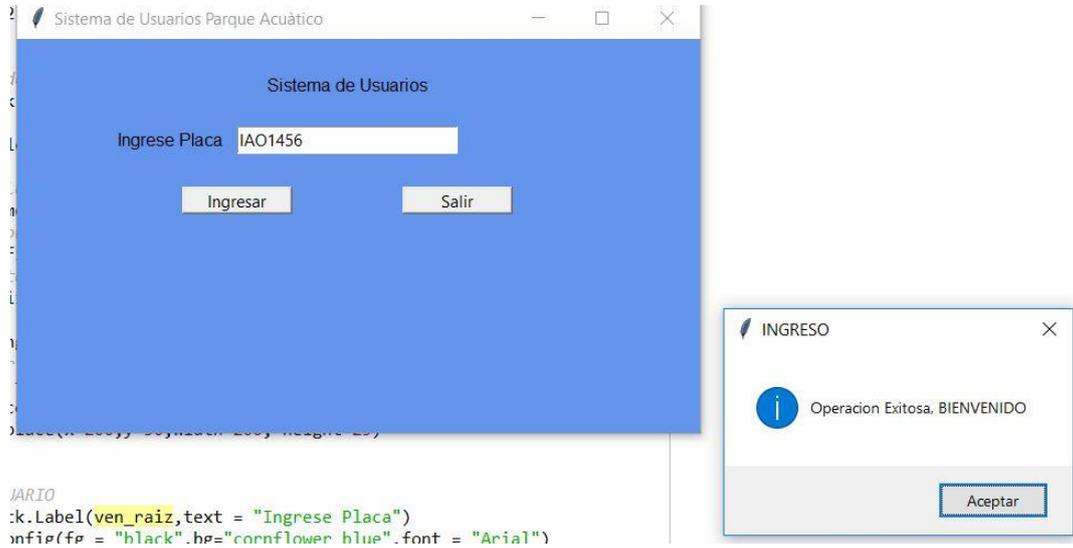


Figura 59 Enlace con la base de datos

3.3 Validación del sistema de cámaras.

Habiendo evaluado los criterios de selección de componentes en el Capítulo 2 se tiene un resultado favorable y positivo frente al sistema adquirido e instalado en el lugar turístico.



Figura 60 Componentes del sistema de seguridad.

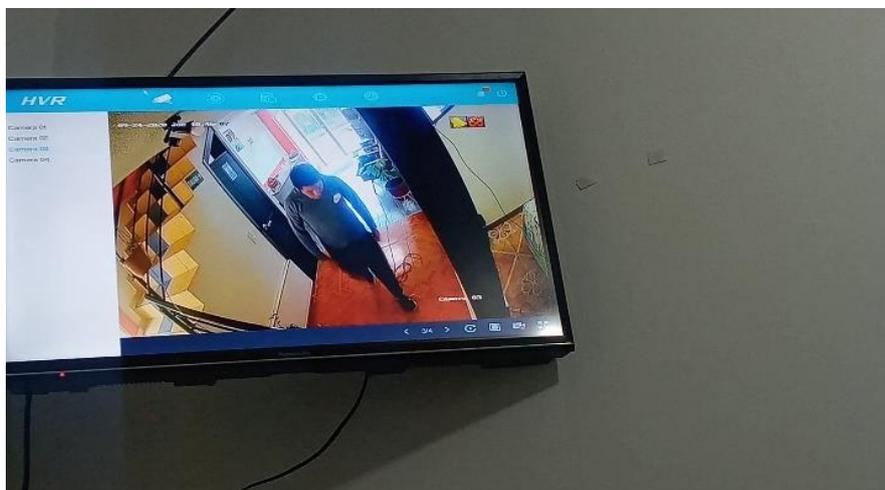


Figura 61 Validación del sistema de cámaras

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Basado en los análisis y estudios realizados se determinó los sistemas apropiados basado en los requerimientos del cliente y el alcance económico del mismo.

El mejor sistema de cobro de peajes para un lugar turístico similar al estudiado es el Sistema RFID gracias a que brinda versatilidad, eficiencia, fácil instalación y bajo costo, además que permite un mejor flujo vehicular y llevar un registro de los vehículos que han ingresado y salido, así como también un control monetario para obras en el lugar.

La metodología implementada permite desarrollar un sistema de aprendizaje continuo, actualmente la eficiencia del sistema es del 91% pero puede mejorar aumentando el número de vehículos del data set ampliando su rango de medición y velocidad para la extracción de imágenes y procesamiento de las mismas, de la misma manera con un procesador más rápido se puede obtener un OCR más eficiente. Es importante mencionar que para el lugar turístico el rendimiento es el apropiado por la localización, el horario que maneja el establecimiento, y la frecuencia y similitud de vehículos que acuden al mismo.

Se demostró que un sistema HikVision permite brindar seguridad a los lugares turísticos públicos y privados y se puede concluir que si existe ventaja sobre los sistemas tradicionales gracias a su fácil instalación y bajo costo.

4.2 Recomendaciones

Realizar una investigación exhaustiva para una mejor comprensión de los métodos existentes para entrenar el reconocimiento de caracteres, objetos, formas, así como también el correcto uso de los softwares y lenguajes de programación para este proyecto.

Realizar diferentes pruebas variando el ambiente donde se extraerá la placa y de igual manera con varios vehículos de diferente color, tamaño, ángulos de cámaras y número de caracteres en las placas.

Se sugiere realizar el entrenamiento en una máquina de alta capacidad y velocidad de procesamiento para una mejor eficiencia del sistema

Plantear planos y esquemas para un futuro proyecto mejorando al antecesor brindando mayores garantías y bajando los costos.

Bibliografía

Bibliografía

- [1] L. Grunewald, «Foro de seguridad,» *Articulos de seguridad*.
- [2] M. V. C. I. P. A. V. Ávila, "Diseño e implementación de un sistema de seguridad vehicular mediante reconocimiento facial con visión artificial", Cuenca, 2017.
- [3] Andalucía, «"Smart City",» *Andalucía Digital*, p. 1, 2018.
- [4] V. Ana, «Beneficios de las nuevas tecnologías en el sector turístico,» *Hosteltur*, p. 5, 26 09 2016.
- [5] RES, «Destino Turístico Inteligente,» *EcoInteligencia*, pp. 2-4, 2014.
- [6] J. M. Q. A. GUISELA MARITZA FERNANDEZ PAUCAR, RECONOCIMIENTO DE PLACAS VEHICULARES EN TIEMPO REAL POR MEDIO DE VISIÓN ARTIFICIAL, QUITO, 2017.
- [7] Cognex, «Una guía para la automatización de procesos y mejoras de calidad,» *Introducción a la visión artificial*.
- [8] M. Awad y R. Khanna, *Efficient Learning Machines*, 2014.
- [9] A. 21, «Centro de formación técnica para la industria,» [En línea]. Available: <https://www.cursosaula21.com/que-es-la-vision-artificial/>. [Último acceso: 2020 10 13].
- [10] M. A. Á. Durán, Análisis, Diseño e implementación de un sistema de control de ingreso de vehículos basado en visión artificial y reconocimiento de placas en el parqueadero de la universidad Politécnica Salesiana SEDE CUENCA, Cuenca, 2014.
- [11] G. G. Bonet, *Etapas y componentes de un sistema de visión artificial*.
- [12] M. F. César, *Detector Automático de Franqueamiento de señal de maniobra para el sistema de transporte colectivo metro mediante algoritmos de visión artificial*, 2015, Mexico.
- [13] M. D. T. Carrera, "Reconocimiento de placas vehiculares mediante procesamiento de imágenes para optimizar el acceso a los parqueaderos de la UTA, Campus Huachi", Cuenca, 2014.
- [14] N. Jorge, *Sistema de detección de matrícula con Open CV*, Madrid.
- [15] J. F. P. Jiménez, *Propuesta para la implementación de la estación de peaje en Quito*, Quito, 2014.
- [16] R. V. V. Moscoso, *Diseño y construcción de un prototipo para cobro de peajes con Visión Artificial*, Loja, 2018.
- [17] C. X. C. d. Hierro, *Análisis del sistema de telepeaje en la autopista Rumiñahui*, Quito, 2009.

- [18] J. I. P. García, A. B. B. Nieto y A. M. B. Barbolla, Tecnología de identificación por radiofrecuencia., Madrid: Fundación Madrid para el conocimiento.
- [19] D. RFID, «Dipole RFID,» 01 Junio 2017. [En línea]. Available: <https://www.dipolerfid.es/blog/categor-a-1/Tipos-Sistemas-RFID>. [Último acceso: 2020 10 14].
- [20] R. M. Manso, Sistema de visión artificial para la detección de lecturas de matriculas, Valladolid, 2014.
- [21] M. Estévez, «Inteñigencia Anlítica,» 03 Octubre 2016. [En línea]. Available: <https://inteligencia-analitica.com/procesamiento-lenguaje-natural/>. [Último acceso: 2020 10 14].
- [22] D. I. P. Villaseca, Herramientas Deep Learning para Predictive Analytics, Chile, 2017.
- [23] A. Darwin, C. Santiago y R. Paúl, PROCESAMIENTO DE IMÁGENES CON VISIÓN ARTIFICIAL PARA, Ambato, 2016.
- [24] J. P. O. Lopez, RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR) CON REDES NEURONALES E IMPLEMENTACION CON EL TOOLBOX DE MATLAB, 2009.
- [25] J. L. Q. Medina, "SISTEMA INFORMÁTICO QUE PERMITA LA ADMINISTRACIÓN CONTABLE DE LOS ACTIVOS FIJOS QUE POSEE LA PUCESE.", Esmeraldas, 2016.
- [26] D. Pedro Gil Jiménez, Sistemas de seguridad, Alcalá de Henares, 2009.
- [27] A. D. A. Salazar y K. L. C. Mite, Diseño e implementación de un sistema de seguridad a través de cámaras, sensores y alarma monitorizado y controlado telemétricamente para el centro de acogida "Patio mi pana" perteneciente a la fundación proyecto salesiano, Guayaquil, 2015.
- [28] J. Araque, «Lineamientos territoriales de la comunidad de Araque,» 2017.
- [29] «La educación en la era digital,» 22 Febrero 2019. [En línea]. Available: <https://ayto-torrijos.com/herramientas/que-es-autocad-y-para-que-sirve/>. [Último acceso: 2020 10 14].
- [30] D. A. M. Jhanneth y F. B. L. Ximena, Optimización del sistema de cobros de impuesto y servicios en el municipio del cantón Girón sobre sistemas distribuidos, Cuenca, 2010.
- [31] Moralla, «GitHub,» [En línea]. [Último acceso: 2020 10 14].
- [32] J. C. Estrada y J. C. Herrera, Estudio del código QR para el desarrollo de los planes de marketing y publicidad en las empresas del sector comercial de la ciudad de Guayaquil, Guayaquil, 2012.
- [33] DiyIoT. [En línea]. Available: <https://diyi0t.com/rfid-sensor-tutorial-for-arduino-and-esp8266/>. [Último acceso: 10 09 2020].
- [34] N. Mechatronics, «Naylamp Mechatronics,» [En línea]. Available: <https://naylampmechatronics.com/rfid-nfc/80-modulo-lector-rfid-1356mhz-rc522.html>. [Último acceso: 10 09 2020].
- [35] RES, «Destino Turístico Inteligente,» *EcolInteligencia*.

Anexos

1. Código Programación RFID

```
String ID_Tarjeta ;
int pos = 0;
void printArray(byte *buffer, byte bufferSize) {
  for (byte i = 0; i < bufferSize; i++) {
    //Serial.print(buffer[i] < 0x10 ? " 0" : " ");
    ID_Tarjeta += (String)buffer[i];
    //Serial.print(buffer[i]);

  }
  for (int i = 0; i < 4; i++) {
    if (ID_Tarjeta == Ids[i]) {

      Serial.println("Id existe");
      servo();
      break;
    }

  }
  //delay(250);
  //Serial.println();
}
void servo() {

  for (pos = 0; pos <= 90; pos += 1) {
    // in steps of 1 degree
    myservo.write(pos);
    delay(25);
  }
  for (pos = 90; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(25);
  }

}
void setup()
{
  myservo.attach(7); // puerto servo
  Serial.begin(9600); //Inicializa la velocidad de Serial
  SPI.begin(); //Función que inicializa SPI
  mfr522.PCD_Init(); //Función que inicializa RFID
}

void loop()
```

```

{
// Detectar tarjeta
if (mfrc522.PICC_IsNewCardPresent())
{
if (mfrc522.PICC_ReadCardSerial())
{

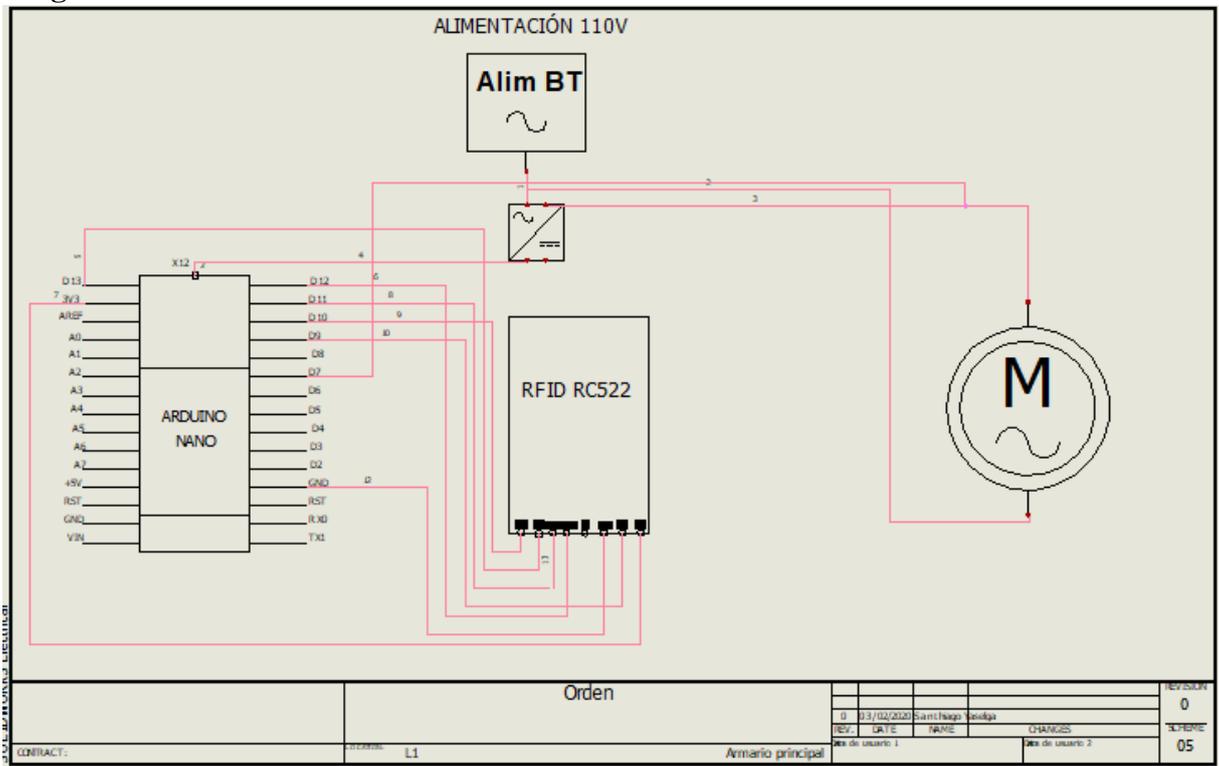
printArray(mfrc522.uid.uidByte, mfrc522.uid.size);

// Finalizar lectura actual
mfrc522.PICC_HaltA();
}
}

if (digitalRead(5)) {
servo();
}
ID_Tarjeta = "";
delay(250);
}

```

2. Diagrama de conexión RFID



3. Códigos de programación del sistema de extracción de placas

- **Set up**

```
import cv2 as cv
import numpy as np
from package_detector import car_detector#, plate_detector

#cap = cv.VideoCapture("Videos/video_1.mp4")
cap = cv.VideoCapture("Videos/video_3.mp4")

cont=0
while True:
    success, img = cap.read()
    # car detector
    #img = plate_detector.data_deteccion(img)

    img = car_detector.data_deteccion(img)

    #cv.imwrite("car" + str(cont)+".png", img)

    if cv.waitKey(1) & 0xFF ==ord('q'):
        break

    cv.imshow('Image', img)
    cv.waitKey(1)
    cont +=1
cap.release()
cv.destroyAllWindows()
```

- **Car detector**

```
import cv2 as cv
import numpy as np
from package_detector import plate_detector

whT = 320
confThreshold =0.5
nmsThreshold= 0.2

#### LOAD MODEL

classesFile = "data/car.names"
classNames = []

with open(classesFile, 'rt') as f:
```

```

classNames = f.read().rstrip('\n').split('\n')

## Model Files
modelConfiguration = "data/car.cfg"
modelWeights = "data/car.weights"
net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights)
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)

def findObjects(outputs,img):

    hT, wT, cT = img.shape
    bbox = []
    classIds = []
    confs = []
    for output in outputs:
        for det in output:

            scores = det[5:]
            classId = np.argmax(scores)

            confidence = scores[classId]
            if confidence > confThreshold:
                img = plate_detector.data_deteccion(img)
                w,h = int(det[2]*wT) , int(det[3]*hT)
                x,y = int((det[0]*wT)-w/2) , int((det[1]*hT)-h/2)
                bbox.append([x,y,w,h])
                classIds.append(classId)
                confs.append(float(confidence))

    indices = cv.dnn.NMSBoxes(bbox, confs, confThreshold, nmsThreshold)

    for i in indices:
        i = i[0]
        box = bbox[i]
        x, y, w, h = box[0], box[1], box[2], box[3]
        # print(x,y,w,h)
        cv.rectangle(img, (x, y), (x+w,y+h), (255, 0 , 255), 2)
        cv.putText(img,f'{classNames[classIds[i]].upper()} {int(confs[i]*100)}%',
            (x, y-10), cv.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 255), 2)
    return img

def data_deteccion(img):

    blob = cv.dnn.blobFromImage(img, 1 / 255, (whT, whT), [0, 0, 0], 1, crop=False)
    net.setInput(blob)
    layersNames = net.getLayerNames()

```

```

outputNames = [(layersNames[i[0] - 1]) for i in net.getUnconnectedOutLayers()]
outputs = net.forward(outputNames)

img = findObjects(outputs, img)
return img

```

- **Plate detector**

```

import cv2 as cv
import numpy as np
from package_detector import extraction_text_plate

whT = 320
confThreshold = 0.8
nmsThreshold = 0.2

#### LOAD MODEL
## Coco Names
classesFile = "data/plate.names"
classNames = []
with open(classesFile, 'rt') as f:
    classNames = f.read().rstrip('\n').split('\n')
#print(classNames)
## Model Files
modelConfiguration = "data/plate.cfg"
modelWeights = "data/plate.weights"
net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights)
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)

def findObjects(outputs, img):

    hT, wT, cT = img.shape
    bbox = []
    classIds = []
    confs = []
    for output in outputs:

        for det in output:

            scores = det[5:]
            classId = np.argmax(scores)

            confidence = scores[classId]
            if confidence > confThreshold:
                w, h = int(det[2]*wT) , int(det[3]*hT)

```

```

x,y = int((det[0]*wT)-w/2) , int((det[1]*hT)-h/2)
bbox.append([x,y,w,h])
classIds.append(classId)
confs.append(float(confidence))

```

```
indices = cv.dnn.NMSBoxes(bbox, confs, confThreshold, nmsThreshold)
```

```
for i in indices:
```

```

    i = i[0]
    box = bbox[i]
    x, y, w, h = box[0], box[1], box[2], box[3]
    xi,xd,yi,yd = x+40, x+w-40, y+5, y+h-5
    cv.rectangle(img, (xi, yi), (xd,yd), (255, 0 , 255), 2)
    #plate extraction
    img_plate = img[yi:yd,xi:xd]
    cv.imwrite("placa.png", img_plate)
    #extraction_text_plate.text_plate(img_plate)
    cv.putText(img,f'{classNames[classIds[i]].upper()} {int(confs[i]*100)}%',
               (x, y-10), cv.FONT_HERSHEY_SIMPLEX, 0.6, (255, 0, 255), 2)
return img

```

```
def data_deteccion(img):
```

```

    blob = cv.dnn.blobFromImage(img, 1 / 255, (whT, whT), [0, 0, 0], 1, crop=False)
    net.setInput(blob)
    layersNames = net.getLayerNames()
    outputNames = [(layersNames[i[0] - 1]) for i in net.getUnconnectedOutLayers()]
    outputs = net.forward(outputNames)

    img = findObjects(outputs, img)
    return img

```

- **Extraction Plate**

```

import cv2
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import pytesseract as ocr

black = (0,0,0)
white = (255,255,255)
threshold = (90,90,90)

kernel = np.ones((3,3),np.uint8)
kernel2 = np.ones((5,5),np.uint8)

```

```

## thresholding of img BGR

def umbralizacion(img):
    pixels = img.getdata()

    newPixels = []

    # Compare each pixel
    for pixel in pixels:
        if pixel < threshold:
            newPixels.append(black)
        else:
            newPixels.append(white)

    # Create and save new image.
    newImg = Image.new("RGB",img.size)
    newImg.putdata(newPixels)

    return newImg

# IMPORT IMAGE
def text_plate(img):

    img = Image.fromarray(img, 'RGB')
    img=img.resize((800,320))

    #TRESHOLDIG FUN
    img = umbralizacion(img)

    img = np.array(img)
    img = img[:,100:700]
    img = img[:,:,1]

    img_erode = cv2.erode(img, kernel, iterations = 1)
    img_dil = cv2.dilate(img_erode, kernel)

    #find contours external
    ctrs, hier = cv2.findContours(img_dil.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    #sort contours
    sorted_ctrs = sorted(ctrs, key=lambda ctr: cv2.boundingRect(ctr)[0])

    for i, ctr in enumerate(sorted_ctrs):
        # Get bounding box
        x, y, w, h = cv2.boundingRect(ctr)

```

```

area = int(cv2.contourArea(ctr))
#
if area>36500 & area<38000:
    # Getting
    roi = img[y:y+h, x:x+w]
    # show ROI

    roi2 = roi[40:,:]
    roi2 = cv2.dilate(roi2, kernel2)

    texto = ocr.image_to_string(roi2)
    text_plate = (texto.split("\n"))[0]

    # number chars
    if len(text_plate)==8:
        print(text_plate)

```

- **Creation Dataset**

```

# libreria a Utilizar
import sqlite3

des = "Cantidad"
# creacion de la base de datos
conexion = sqlite3.connect("Registro_de_Ingreso.db")
# creacion de un objeto referencia a la base de datos
cursor = conexion.cursor()
# Creacion de la tabla REgistro Usuarios
cursor.execute("CREATE TABLE Registro_Ingreso (Placa TEXT, Fecha TEXT, Hora TEXT)")
# Creacion de la tabla Cant_Ingreso
cursor.execute("CREATE TABLE Registro_Cantidad (Detalle TEXT, Cant_Ingreso INTEGER, Cant_Egreso INTEGER, Cant_Dentro INTEGER)")
# Asignacion de parametros iniciales en la tabla Cant_Ingreso
cursor.execute("INSERT INTO Registro_Cantidad (Detalle, Cant_Ingreso, Cant_Egreso, Cant_Dentro) VALUES (?, ?, ?, ?)",(des, 0, 0, 0))
# cerramos la base de datos
conexion.commit()
conexion.close()

```

- **Data set managment**

```

import sqlite3 as sq3
from datetime import date
from datetime import datetime
from tkinter import messagebox
import time

```

```

def insertar_datos(placa):

    #Fecha actual
    Now = datetime.now()
    fecha = (str)(Now.date())
    hora = (str)(Now.time())
    conexion = sq3.connect("Registro_de_Ingreso.db")
    cursor_1 = conexion.cursor()
    cursor_1.execute("INSERT INTO Registro_Ingreso (Placa, Fecha, Hora) VALUES
    (?, ?, ?)",(placa, fecha, hora))

    conexion.commit()
    conexion.close()

    Actualizar_Ingreso()
    Actualizar_Cantidad_Dentro()
    messagebox.showinfo(message="Operacion Exitosa, BIENVENIDO",
    title="INGRESO")

```

```

def Actualizar_Ingreso():

    conexion = sq3.connect('Registro_de_Ingreso.db')
    cursor = conexion.cursor()
    cursor.execute("SELECT * FROM Registro_Cantidad")
    usu = cursor.fetchall()
    conexion.commit()
    conexion.close()

    # Actualizacion Cantidad Ingreso
    det =usu[0]

    cant_ingreso = det[1]
    cant_ingreso+=1

    detalle = "Cantidad"
    conexion2 = sq3.connect('Registro_de_Ingreso.db')
    cursor2 = conexion2.cursor()
    cursor2.execute("SELECT * FROM Registro_Cantidad")

    cursor2.execute("UPDATE Registro_Cantidad SET Cant_Ingreso = ? WHERE
    Detalle = ?",(cant_ingreso, detalle))

    conexion2.commit()
    conexion2.close()

```

```

def Actualizar_Egreso(placa):

    conec = sq3.connect("Registro_de_Ingreso.db")
    cursor_1 = conec.cursor()
    cursor_1.execute("SELECT * FROM Registro_Ingreso")
    usuarios = cursor_1.fetchall()

    for usuario in usuarios:
        reg = usuario
        reg_Id = str(usuario[0])

        if (reg_Id == placa):

            conec.commit()
            conec.close()

            ## Actualizacion
            conexion = sq3.connect('Registro_de_Ingreso.db')
            cursor = conexion.cursor()
            cursor.execute("SELECT * FROM Registro_Cantidad")
            usu = cursor.fetchall()
            conexion.commit()
            conexion.close()

            # Actualizacion Cantidad Ingreso
            det =usu[0]

            detalle = "Cantidad"
            conexion2 = sq3.connect('Registro_de_Ingreso.db')
            cursor2 = conexion2.cursor()
            cursor2.execute("SELECT * FROM Registro_Cantidad")

            cant_egreso = det[2]
            cant_egreso+=1
            cursor2.execute("UPDATE Registro_Cantidad SET Cant_Egreso = ? WHERE
Detalle = ?",(cant_egreso, detalle))

            conexion2.commit()
            conexion2.close()
            Actualizar_Cantidad_Dentro()
            messagebox.showinfo(message="Operacion Exitosa Adios, UN BUEN DIA",
title="SALIDA")

def Actualizar_Cantidad_Dentro():

```

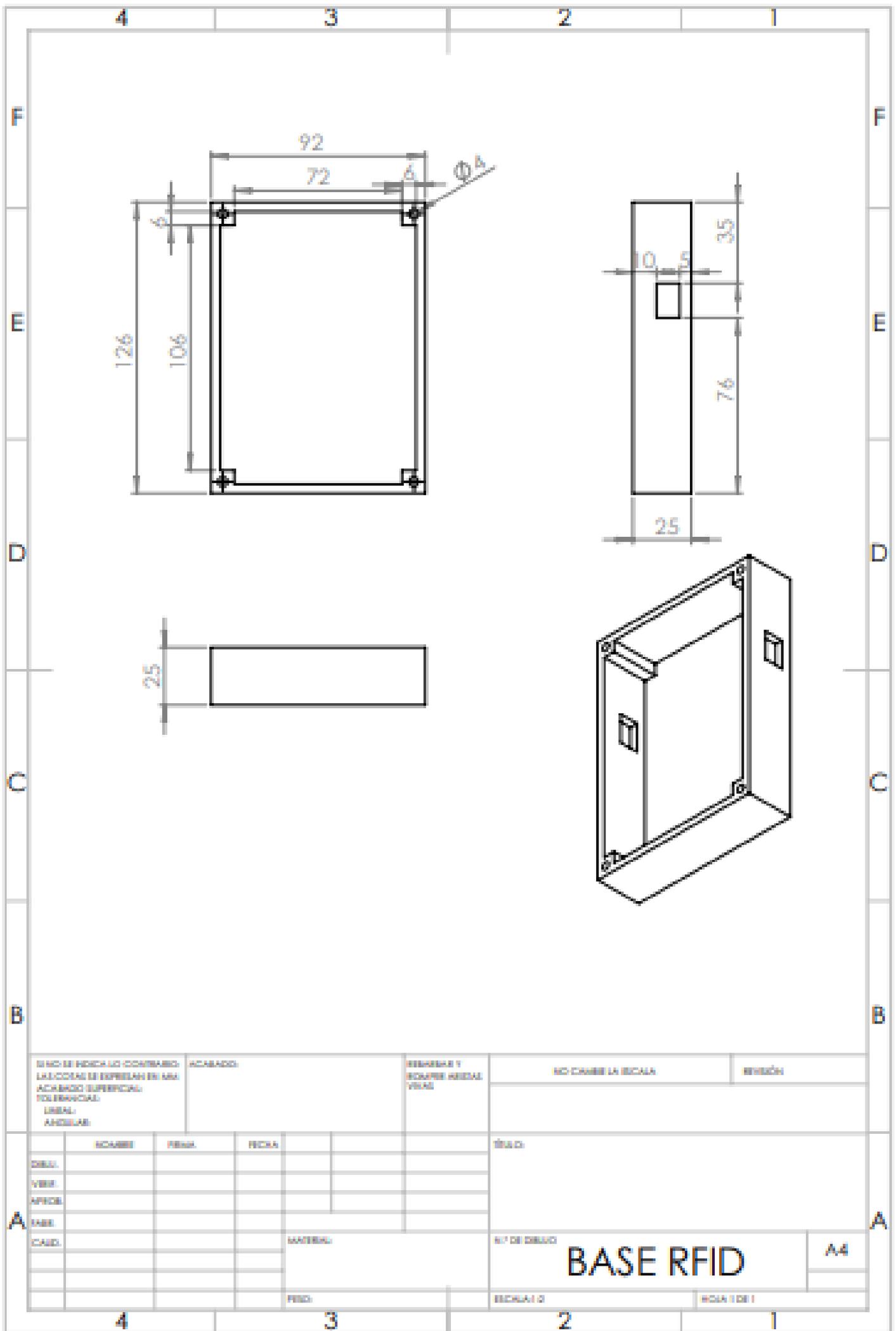
```

conexion = sq3.connect('Registro_de_Ingreso.db')
cursor = conexion.cursor()
cursor.execute("SELECT * FROM Registro_Cantidad")
usu = cursor.fetchall()
conexion.commit()
conexion.close()

# Actualizacion Cantidad Ingreso
det =usu[0]
valor = det[1]-det[2]

detalle = "Cantidad"
conexion2 = sq3.connect('Registro_de_Ingreso.db')
cursor2 = conexion2.cursor()
cursor2.execute("SELECT * FROM Registro_Cantidad")
cursor2.execute("UPDATE Registro_Cantidad SET Cant_Dentro = ? WHERE
Detalle = ?",(valor, detalle))
conexion2.commit()
conexion2.close()

```

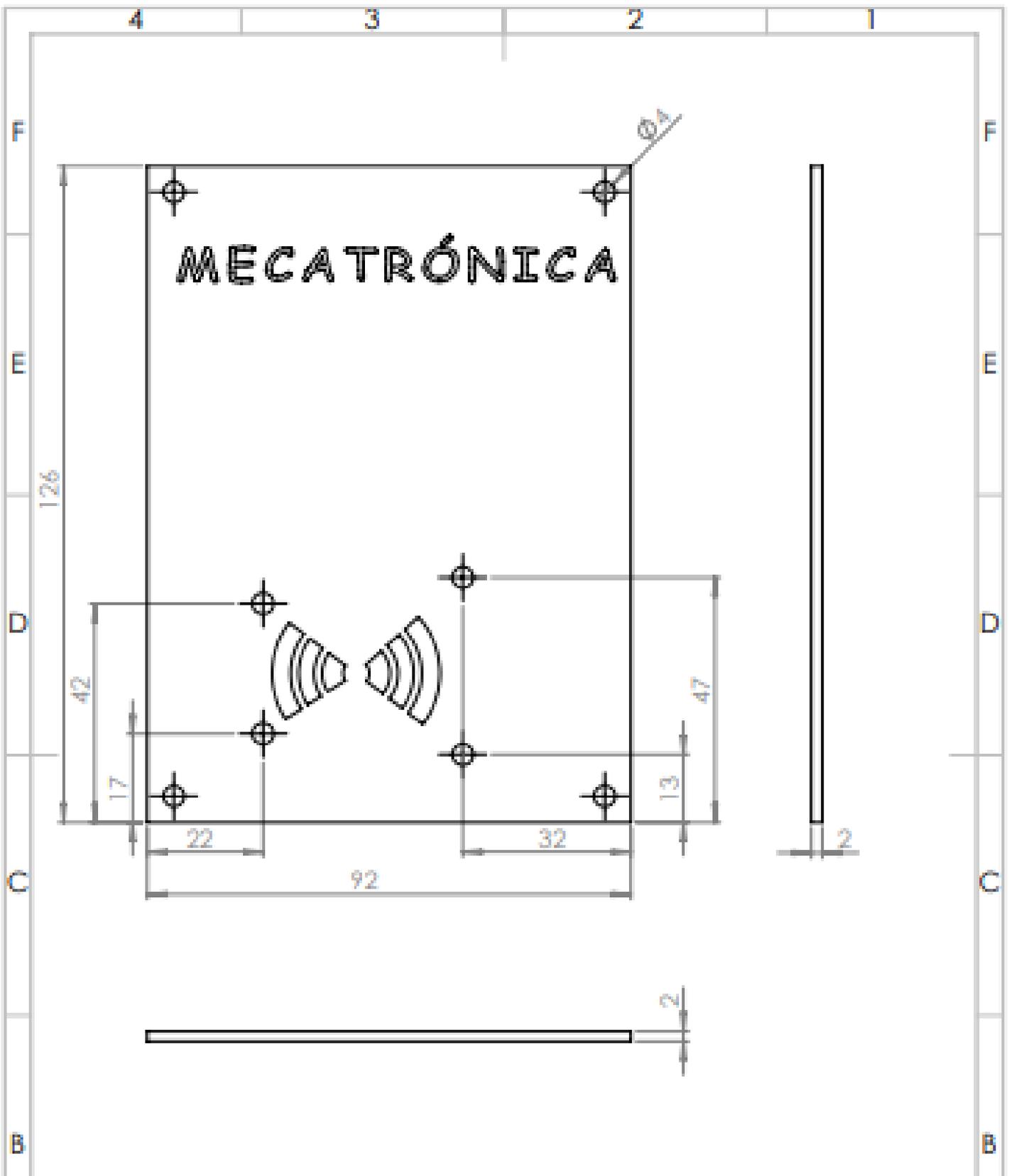


SI NO SE INDICA LO CONTRARIO LAS COTAS SE ESPRESAN EN MM. ACABADO SUPERFICIAL: TOLERANCIA: LINEAL: ANGULAR:	ACABADO:			REFINIRSE Y BOMBEAR ANGULAR VISUAL	NO CAMBIE LA ESCALA	REVISIÓN

DISEÑ. VERIF. APROB.	NOCAMBIE FECHA	FECHA	FECHA	FECHA	FECHA	DISEÑO
FABR. CALIF.						

MATERIAL:				Nº DE DISEÑO		BASE RFID	A4
FECH:				FECHA 1 0			

4 3 2 1



SI NO SE INDICA LO CONTRARIO LAS COTAS SE ESPRESAN EN MM. ACABADO SUPERFICIAL: TELURICO: LIBRE: ANILINAR:		ACABADO:		REVISIONES Y COMPROBACIONES VICIAS		NO CAMBIE LA ESCALA	REVISION
DESE.	NO CAMBIE	FECHA	FECHA			TITULO	
VISI.						FRONTAL RFID A4	
APROB.							
FABR.							
CADEL.							
				MATERIAL:		Nº DE DISEÑO	
				FECH:		ESCALA: 1:1	
						HOJA 1 DE 1	

A **A**

4 3 2 1