



UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

ESCUELA DE INGENIERÍA DE SISTEMAS Y TELEMÁTICA

“Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares”

**Tesis de grado previo a la obtención del título de
Ingeniero de Sistemas y Telemática**

Autores

Darwin Darío Espinoza Saquicela
Christian Antonio Salinas Escobar

Director

Ing. Kenneth S. Palacio Baus, MSc

Cuenca - Ecuador
2015

DEDICATORIA

La elaboración de este proyecto de titulación la dedico a mi padre, a mi madre, a mi hermana y hermanos que con su apoyo y confianza incondicional estuvieron presentes de varias formas en el transcurso de esta etapa, inculcando valores y compartiendo experiencias que han sido importantes para sortear los obstáculos que he encontrado durante mi formación.

Darwin Darío Espinoza Saquicela.

Dedico la elaboración de este proyecto de titulación a mis padres, por ser el pilar fundamental del hogar, ya que con su apoyo, nunca dudaron de mi capacidad para enfrentar los retos que se presentaron en el trayecto para el cumplimiento de mis logros y mi formación como profesional.

A mi esposa e hija, que día a día son mi orgullo, motivación, y mi fortaleza para seguir adelante y alcanzar nuevos objetivos en el camino de la vida.

Christian Salinas Escobar.

AGRADECIMIENTOS

A mi familia, mis amigos y aquellos anónimos que en el transcurso de mi vida marcaron etapas importantes para formarme como persona.

A todos los docentes que formaron parte de mi proceso educativo y de manera muy especial al Ing. Kenneth Palacio por habernos compartido su conocimiento, experiencias y ayuda durante el desarrollo de este proyecto.

Darwin Darío Espinoza Saquicela.

A Dios por la vida y la salud. A mi familia y amigos por sus palabras de aliento y las experiencias vividas que quedarán marcadas a lo largo de mi vida. A los profesores de la Universidad del Azuay, especialmente al Ing. Kenneth Palacio, por ser parte de nuestro proceso de formación, dirigiendo nuestra tesis y también compartiendo sus conocimientos.

Finalmente a mi estimado amigo Darío Espinoza por participar en el presente trabajo de titulación y por compartir varios años de estudios.

Christian Salinas Escobar.

ÍNDICE DE CONTENIDOS

Índice de contenidos	III
Índice de figuras.	VII
Índice de tablas.	XII
Resumen	XIV
Abstract	XV
1. Introducción y Descripción del Proyecto	1
1.1. Introducción	1
1.2. Antecedentes	2
1.3. Planteamiento del Problema	2
1.4. Justificación	3
1.5. Definición de Objetivos	4
1.5.1. Objetivo General	4
1.5.2. Objetivos Específicos	4
1.6. Metodología	4
2. Análisis Vehicular a través de Imágenes	7
2.1. Introducción	7
2.2. Visión por computadora en el análisis del tráfico vehicular	8
2.2.1. Aplicación de conteo de vehículos.	9
2.2.2. Reconocimiento Automático de Placas Vehiculares	9
2.2.3. Detección de Incidentes	10
2.3. Detección de placas vehiculares	11
2.3.1. Procesamiento de Imágenes Binarias	11
2.3.2. Procesamiento de Nivel de Grises	14
2.3.3. Procesamiento de Color	16

2.4.	Algoritmos y técnicas de reconocimiento de caracteres	16
2.4.1.	Comparación con Plantillas	19
2.4.2.	Algoritmo K-Vecinos más Próximos (K-NN)	20
2.4.3.	Reconocimiento en base a características	22
2.4.4.	Árboles de decisión	23
2.4.5.	Redes Neuronales	24
2.4.6.	Tesseract	28
2.5.	Conclusiones	29
3.	Diseño e implementación del sistema	30
3.1.	Introducción	30
3.2.	Análisis y planificación de requisitos	32
3.2.1.	Ámbito del sistema	32
3.2.2.	Descripción general	32
3.2.3.	Requisitos específicos	35
3.2.4.	Interfaces externas	39
3.3.	Diseño del sistema	40
3.4.	Especificaciones Hardware del Sistema	50
3.4.1.	Cámara IP	50
3.4.2.	Servidor de procesamiento de imágenes	58
3.4.3.	Arduino UNO	58
3.4.4.	Dispositivos de red	59
3.5.	Especificaciones software del sistema	61
3.5.1.	Lenguaje de programación	61
3.5.2.	Interfaz gráfica Qt	61
3.5.3.	Librería de visión por computador	62
3.5.4.	Base de datos	63
3.5.5.	Ambiente para desarrollo WEB	63
3.6.	Implementación del módulo del detección de placas	64
3.6.1.	Pre-procesamiento de imágenes	65
3.6.2.	Procesamiento de Imágenes	66
3.7.	Implementación del módulo de reconocimiento	69
3.8.	Descripción del Prototipo	72
4.	Pruebas y Evaluación del Sistema	74
4.1.	Introducción	74
4.2.	Contexto de evaluación del sistema	74
4.2.1.	Otros factores influyentes	75

4.3.	Modelo de evaluación del sistema	77
4.3.1.	Recolección de Datos	77
4.3.2.	Procesamiento de datos	79
4.4.	Evaluación Cuantitativa	81
4.4.1.	Detección de placas vehiculares	82
4.4.2.	Reconocimiento de caracteres	85
4.5.	Evaluación Cualitativa	89
4.5.1.	Rendimiento Óptimo	90
4.5.2.	Rendimiento Bueno	90
4.5.3.	Rendimiento Malo	90
4.5.4.	Rendimiento Deficiente	91
4.6.	Conclusiones	91
5.	Conclusiones y Trabajo Futuro	92
5.1.	Conclusiones	92
5.2.	Trabajo Futuro	96
	Referencias	98
	Anexos	103
	Anexo A. Descripción de Casos de Uso	104
	Anexo B. Diccionario de Datos	115
	Anexo C. Manual de usuario - LPR UDA	121
C.1.	Pestaña de detección de placas vehiculares - License Plate Recognition	121
C.1.1.	Cargar video	122
C.1.2.	Mostrar análisis	122
C.1.3.	Ajustar área de placa vehicular	123
C.1.4.	Avanzar y Retrasar video	123
C.1.5.	Etiquetas de información	124
C.1.6.	Mostrar, mover y ajustar región de interés (ROI)	125
C.1.7.	Mostrar, mover y ajustar sub región de interés (ROI)	125
C.1.8.	Activar y desactivar sensor láser	126
C.1.9.	Activar y desactivar sensor imagen	126
C.2.	Pestaña de pruebas	127
C.2.1.	Cargar imagen y obtener ROI	127

C.2.2. Preprocesado y búsqueda de contornos	128
C.2.3. Reconocimiento	129
C.3. Pestaña de control Arduino UNO	129
Anexo D. Manual de Estilos - LPR UDA	131
D.1. Introducción	131
D.2. Wireframes	131
D.3. Mockups	137
D.4. Cabecera	142
D.5. Iconos	142
D.6. Color	142



ÍNDICE DE FIGURAS.

2.1. Radar indicador de velocidad usado en Ecuador	8
2.2. Peaje o acceso automatizado con un sistema ALPR	10
2.3. La región con marco verde muestra la detección exitosa de la placa vehicular.	11
2.4. Imagen Binarizada.	12
2.5. Análisis de componentes conectados	13
2.6. Efecto de la aplicación del operador de Sobel en los bordes en una imagen de entrada.	14
2.7. Placa vehicular (subimagen) obtenida tras usar técnicas de detección.	16
2.8. Caracteres segmentados que luego serán reconocidos con OCR. . .	17
2.9. Convirtiendo una imagen a una cadena de “1” y “0”	18
2.10. Comparación con plantillas	19
2.11. Búsqueda de la clase correspondiente a “X” en base a sus “K” vecinos más próximos, tomando inicialmente los estereotipos E1, E2 y E3.	21
2.12. Reconocimiento usando Características Geométricas y Topológicas (strokes and bays)	23
2.13. Recorrido de un Árbol de Decisión para OCR.	24
2.14. Reconocimiento Óptico de Caracteres usando una Red Neuronal .	25
2.15. Función de activación <i>Symmetrical Sigmoid</i>	25
2.16. Caracter que será reconocido.	26
2.17. Red neuronal.	26
2.18. Placa recortada que será reconocida por Tesseract.	28
3.1. Procesos a seguir para el reconocimiento de una placa vehicular. .	31

3.2. Diagrama de casos de uso esencial	34
3.3. Diagrama de casos de uso extendido	35
3.4. Diagrama de clases	41
3.5. Diagrama de actividad	46
3.6. Diagrama de estados de usuario	47
3.7. Diagrama de estados de administrador	47
3.8. Modelo entidad-relación	48
3.9. Esquema del circuito electrónico	49
3.10. Implementación práctica del circuito electrónico.	50
3.11. Cámara IP foscam FI8921W-V2	50
3.12. Descripción física cámara IP	52
3.13. Conexión cámara IP	52
3.14. CD de instalación cámara IP	53
3.15. Dirección de la cámara IP	53
3.16. Plugin cámara IP	54
3.17. Login cámara IP	54
3.18. Ingreso de usuario y password para la cámara IP	54
3.19. Interfaz cámara IP	55
3.20. Botón WPS cámara IP	55
3.21. Búsqueda de redes WiFi cámara IP	56
3.22. Configuración de red WiFi cámara IP	56
3.23. Arduino UNO	58
3.24. Diagrama de red	60
3.25. Router TP-LINK TD-8961N	60
3.26. Switch D-LINK DES-1008A	61
3.27. Logo de software Qt	62
3.28. Logo de librería OpenCV	62
3.29. Logo MySQL	63
3.30. Logo de WampServer	64
3.31. Imagen capturada con OpenCV	65
3.32. Pre-procesamiento: región de interes	65
3.33. Pre-procesamiento: escala de grises	66
3.34. Pre-procesamiento: ecualización del histograma	66
3.35. Procesamiento: adaptive threshold	67
3.36. Procesamiento: Otsu threshold	67
3.37. Procesamiento: operador Sobel para detección de bordes verticales.	68
3.38. Procesamiento: operaciones morfológicas.	68

3.39. Procesamiento: placa detectada.	69
3.40. Imagen proveniente de la detección de contornos.	70
3.41. Contornos correspondientes a los caracteres de identificación	70
3.42. Placa binarizada, sin bordes y sin detalles innecesarios para el re- conocimiento.	70
3.43. Prototipo de acceso a un parqueadero.	72
4.1. Ejemplos de placas vehiculares en buen estado.	75
4.2. Ejemplos de placas vehiculares en mal estado.	76
4.3. Ejemplos de placas personalizadas.	76
4.4. Ejemplos de caracteres encontrados en placas vehiculares de Ecua- dor y sus distintas fuentes. (1) letra “A”, (2) letra “D”, (3) número “cero”, (4) número “seis” y (5) número “siete”.	77
4.5. Distancia y ángulo a los que se han capturado las imágenes de prueba.	78
4.6. Enfoque de la placa vehicular dentro de la “Región de Interés”. . .	78
4.7. A la izquierda, la imagen binarizada. A la derecha, el contorno identificado potencialmente como placa vehicular.	79
4.8. A la izquierda, la imagen binarizada. A la derecha, el contorno identificado potencialmente como placa vehicular.	80
4.9. A la izquierda, la imagen binarizada. A la derecha, el contorno identificado potencialmente como placa vehicular.	80
4.10. Contorno de placa vehicular.	82
4.11. Detección de placas usando Otsu Threshold.	82
4.12. Detección de placas usando Adaptive Mean Threshold.	83
4.13. Detección de placas usando Operaciones Morfológicas.	83
4.14. Reconocimiento de óptico caracteres usando una red neuronal. . .	86
4.15. Reconocimiento óptico de caracteres usando Tesseract	86
4.16. Placa vehicular en mal estado, a color y binarizada.	87
4.17. Reconocimiento óptico de caracteres usando algoritmo de vecinos más próximos.	87
4.18. Ejemplo de una placa en mal estado, binarizada y reformada. . . .	89
4.19. Ejemplo de una placa vehicular detectada parcialmente.	91
C.1. Pestañas del sistema de reconocimiento de placas vehiculares. . .	121
C.2. Pestaña de detección de placas vehiculares.	122
C.3. Cargar o detener el video a analizar	122
C.4. Análisis en tiempo real del video.	123

C.5. Ajustes de área mínima y máxima de placas vehiculares.	123
C.6. Slider para avanzar o retrasar video.	124
C.7. Información y resultados del análisis.	124
C.8. Visualización y ajuste de ROI.	125
C.9. Visualización y ajuste de Sub ROI.	126
C.10. Casilla para activar y desactivar sensor láser.	126
C.11. Casilla para activar y desactivar sensor de imagen en movimiento	127
C.12. Pestaña de pruebas de detección y reconocimiento.	127
C.13. Componentes necesarios para cargar una imagen de prueba. . . .	128
C.14. Botón para pre procesamiento y búsqueda de contornos.	128
C.15. Imágenes resultantes del análisis.	129
C.16. Botones para indicar si el reconocimiento fue exitoso o no.	129
C.17. Botones para controlar el prototipo de ingreso al parqueadero. . .	130
D.1. Wireframe inicio de sesión LPR - UDA.	132
D.2. Wireframe inicio de sesión desde dispositivos móviles.	132
D.3. Wireframe home	133
D.4. Wireframe de qué se trata? LPR - UDA.	133
D.5. Wireframe servicios.	133
D.6. Wireframe proyecto LPR - UDA.	134
D.7. Wireframe contáctanos.	134
D.8. Wireframe Ingreso de Usuarios.	135
D.9. Wireframe registrar vehículo	135
D.10. Wireframe reporte de vehículos	136
D.11. Wireframe video vigilancia	136
D.12. Mockup inicio de sesión LPR - UDA.	137
D.13. Mockup inicio de sesión desde dispositivos móviles.	137
D.14. Mockup home	138
D.15. Mockup de qué se trata? LPR - UDA.	138
D.16. Mockup servicios.	138
D.17. Mockup proyecto LPR - UDA.	139
D.18. Mockup contáctanos.	139
D.19. Mockup Ingreso de Usuarios.	140
D.20. Mockup registrar vehículo	140
D.21. Mockup reporte de vehículos	141
D.22. Mockup video vigilancia	141
D.23. Menu 1 LPR - UDA	142
D.24. Menu 2 LPR - UDA	142



D.25.Iconos	142
D.26.Colores	143



ÍNDICE DE TABLAS.

2.1. Estereotipos representantes de cada clase.	21
2.2. Muestras pertenecientes a las clases filtradas con los estereotipos.	22
2.3. $2K$ muestras más cercanas a la entrada "X".	22
3.1. Especificaciones del servidor de imágenes	58
3.2. Especificaciones técnicas Arduino UNO	59
3.3. Arreglo de imágenes, correspondientes a caracteres, producto de la segmentación.	71
4.1. Votación caracter por caracter para determinar una única placa vehicular.	81
4.2. Número de placas encontradas y no encontradas para cada método.	82
4.3. Sub contornos encontrados para identificar potenciales placas vehi- culares.	85
4.4. Aciertos y errores producidos en el reconocimiento de las 211 placas.	85
4.5. Ejemplos de placas vehiculares reconocidas y no reconocidas. . . .	88
A.1. Caso de uso 1: identificarse(usuario)	104
A.2. Caso de uso 2: identificarse(administrador)	105
A.3. Caso de uso 3: solicitar ingreso	105
A.4. Caso de uso 4: detectar placa	106
A.5. Caso de uso 5: reconocer placa	107
A.6. Caso de uso 6: registrar ingreso	108
A.7. Caso de uso 7: iniciar parante	109
A.8. Caso de uso 8: abrir parante	110
A.9. Caso de uso 9: cerrar parante	111



A.10.Caso de uso 10: registrar usuario	112
A.11.Caso de uso 11: modificar usuario	113
A.12.Caso de uso 12: consultar usuarios	114



En el presente trabajo de titulación, se describen los aspectos teóricos, técnicos y prácticos involucrados en el desarrollo e implementación de un sistema de Reconocimiento de Placas Vehiculares – LPR basado en el análisis de imágenes digitales, mediante la construcción de un prototipo que demuestre la automatización del acceso a un parqueadero. Adicionalmente se propone el desarrollo de una interfaz gráfica WEB para la administración de usuarios.

El sistema desarrollado ha sido evaluado en un conjunto de más de 200 imágenes reales, obtenidas en diferentes horarios del día y bajo distintas condiciones de iluminación. Obteniendo así una tasa de reconocimiento de 91 % aproximadamente, dentro del cual se considera aquellas placas que incluso serían difíciles de leer por un ser humano, sea por deterioro, mala ubicación, pintura discordante a la normativa vigente, etc.

Espinoza Dario, Salinas Christian.

ABSTRACT

ABSTRACT

This graduation paper describes the theoretical, technical and practical aspects involved in the development and implementation of a Vehicle License Plate Recognition system - LPR based on digital image analysis, by building a prototype to demonstrate the automation of access to a parking lot. Additionally, we propose the development of a web-based graphical interface for users' administration.

The developed system has been evaluated in a set of more than 200 real images obtained at different times of the day and under various lighting conditions. As a result, we obtained a recognition rate of approximately 91%, within which we considered those plates that would be difficult to be read even by a human being, whether because they are worn out, poor location, or painting discordant with current regulations, etc.




Translated by,
Lic. Lourdes Crespo

CAPÍTULO 1

INTRODUCCIÓN Y DESCRIPCIÓN DEL PROYECTO

1.1. Introducción

El desarrollo de sistemas informáticos, es una tarea en donde se combinan componentes lógicos, que hacen posible la realización de tareas específicas, automatizando procedimientos con la finalidad de optimizar recursos (costos, tiempo) y reducir la carga de trabajo. Por tal motivo se propone el desarrollo de un sistema que automatice el acceso de vehículos a diferentes ambientes y facilite su administración mediante el reconocimiento de placas vehiculares.

Los sistemas informáticos que utilizan visión por computador, hacen el uso de técnicas de tratamiento digital de imágenes con el propósito de emular el sistema de visión humano incluso llegando a superarlo (Klinger, 2003). La aplicación de técnicas de procesamiento digital de imágenes en muchas actividades de la vida contemporánea, se ha visto incrementada notablemente.

El sistema planteado no constituye únicamente un reconocimiento de los caracteres que contiene una placa vehicular. En realidad, propone una manera ágil de ingreso a lugares donde se requieran ciertas medidas de seguridad. Se hace uso de una cámara de vigilancia IP como sensor principal para el reconocimiento de vehículos el mismo que se conecta a un servidor dedicado para el procesamiento de las imágenes capturadas. Este proceso pretende eliminar errores humanos y brindar información relacionada a los vehículos en un ambiente para su administración.

1.2. Antecedentes

Actualmente, en el mercado, podemos observar una gran cantidad de aplicaciones y servicios que utilizan tecnología de cámaras y también técnicas de tratamiento de imágenes para resolver problemas diarios como el reconocimiento de un patrón en particular. En todas las ciudades del mundo, se han presentado problemas de tráfico vehicular, accidentes y conductores que infringen la ley (Hsieh, Chang, Hung, y Huang, 2009). Este tipo de problemas han impulsado la creación de sistemas de identificación y seguimiento de vehículos y su uso se ha extendido incluso a la identificación de vehículos robados (Abdullah, 2006).

Muchas investigaciones dirigidas a la identificación de vehículos se han enfocado en la extracción y reconocimiento de caracteres de placas vehiculares (Sirithinaphong y Chamnongthai, 1999). Lotufo, Morgan y Jhonson proponen un reconocimiento de placas vehiculares usando reconocimiento óptico de caracteres (Lotufo, Morgan, y Johnson, 1990). Jhonson y Bird proponen un aprendizaje guiado para seguimiento de bordes y emparejamiento con plantillas para identificación automática de vehículos (Johnson y Bird, 1990). Fahmy propuso una red neuronal BAM (Bidirectional Associative Memory) para el reconocimiento de placas (Fahmy, 1994) mientras que Ter Brugge y Helmholf proponen un sistema de reconocimiento de placas que usa lógica difusa y redes neuronales (Nijhuis y cols., 1995).

De las investigaciones mencionadas se puede observar que entre los requisitos primordiales de los algoritmos empleados en el reconocimiento de Placas Vehiculares es que deben tener la capacidad de operar lo suficientemente rápido, con el fin de cubrir las necesidades de los Sistemas Inteligentes de Transporte (ITS). Sin embargo, con el crecimiento exponencial de la capacidad de procesamiento computacional, las últimas tecnologías en reconocimiento de placas vehiculares operan fácilmente en diversos escenarios.

1.3. Planteamiento del Problema

En general, los sistemas destinados a gestionar el acceso de vehículos como a parqueaderos tarifados, carecen de un mecanismo de identificación automático de sus clientes, y normalmente requieren la presencia de un operario humano de manera constante. Adicionalmente, la mayoría de sistemas de gestión existentes no poseen herramientas de manejo de la información relacionada a los vehículos

y sus usuarios. Un ejemplo concreto se presenta en el campus de la Universidad del Azuay, cuyo acceso al parqueadero para profesores es gestionado por una persona encargada de verificar que el vehículo posea una etiqueta que le autoriza el ingreso, lo cual está sujeto al error humano.

En este contexto, es importante mencionar también que el uso de la información recopilada por las videocámaras de vigilancia en las ciudades es muy limitado, ya que principalmente están destinadas a tareas de monitoreo y registro de eventos. Muchos de estos sistemas únicamente almacenan las grabaciones sin analizar la información asociada al contenido del video. Aprovechando las infraestructuras de video vigilancia implementadas en la actualidad, es posible integrar sistemas de Procesamiento Digital de Imágenes que permitan obtener datos en tiempo real para la extracción de información requerida.

1.4. Justificación

La reducción de los costos de producción de dispositivos de adquisición como cámaras digitales, y el mejoramiento paulatino de sus características han atraído la atención de los consumidores y al mismo tiempo, de investigadores en el área de procesamiento y análisis digital de imágenes (Klinger, 2003), quienes aprovechan las robustas plataformas de hardware existentes hoy en día, y que permiten el desarrollo de novedosas aplicaciones, tanto en terminales fijos como móviles. Los sistemas de tratamiento digital de imágenes poseen típicamente una cámara como “sensor universal” (Klinger, 2003), el cual define ventajas como:

- Costos de hardware reducidos: actualmente es fácil conseguir cámaras digitales de resoluciones aceptables de 2 megapíxeles en adelante.
- Interoperabilidad: para trabajar con diferentes dispositivos y sistemas operativos.
- Facilidad de configuración y programación: para desarrolladores de aplicaciones de visión por computador.
- Disponibilidad de librerías de código abierto como OpenCV¹.

En el contexto de este proyecto, se plantea el uso de herramientas de visión por computador para el reconocimiento de placas vehiculares. Esta aplicación ha sido ampliamente utilizada en sistemas de control y análisis de tráfico vehicular a nivel mundial (Draghici, 1997). Es por ello que con el presente proyecto, se pretende

¹<http://www.opencv.org/>

impulsar el uso de herramientas de procesamiento de imágenes en el entorno de un sistema automatizado de control, monitoreo y gestión de parqueaderos, el cual constituirá un sólido punto de partida que sustentará el desarrollo de aplicaciones más complejas en el futuro.

1.5. Definición de Objetivos

1.5.1. Objetivo General

Implementar un sistema de gestión y automatización del acceso a un parqueadero, mediante el reconocimiento de placas vehiculares a través de una cámara como mecanismo de identificación de clientes.

1.5.2. Objetivos Específicos

- Desarrollar un sistema capaz de reconocer placas vehiculares.
- Desarrollar un sistema de manejo de parqueaderos integrando técnicas de reconocimiento de placas vehiculares para la identificación de usuarios.
- Diseñar e implementar un prototipo que muestre la automatización del acceso a un parqueadero utilizando una cámara IP y una barrera móvil controlada por un servo motor activado mediante una tarjeta programable Arduino² UNO.
- Generar una interfaz de administración de usuarios autorizados a acceder al parqueadero.
- Generar una interfaz web para el acceso remoto a la administración de usuarios del parqueadero.

1.6. Metodología

La secuencia de actividades que se considera para seguir un proceso continuo, gradual y progresivo para el cumplimiento de los objetivos de este trabajo de tesis es la siguiente:

1. *Indagación y adquisición de bases conceptuales.* En esta fase se debe adquirir los conocimientos necesarios para emprender el camino a la realización del proyecto propuesto, ya que se verán involucrados temas principales como: pre procesamiento de la imagen, algoritmos y las técnicas para detección de bordes, utilizadas para la definir el área de una placa vehicular.

²<http://www.arduino.cc/>

2. *Establecer ajustes*, en base al tamaño y posición de la placa, para la corrección de la perspectiva y como proceso siguiente realizar la segmentación de caracteres.
3. *Evaluar diferentes técnicas y aplicaciones para el reconocimiento de caracteres*: Entre las técnicas se pretende el estudio de métodos basados en redes neuronales y reconocimiento óptico de patrones. Entre las aplicaciones se pretende el estudio de:
 - API OCR mediante servicios WEB.
 - Uso de OCR denominado “Tesseract”³.
 - Uso de OCR empleando los métodos basados en redes neuronales.
4. *Definir los elementos que integrarán la Plataforma para la Administración de un Parqueadero*. Previa a un análisis entre las diferentes alternativas tecnológicas, se debe seleccionar la opción más adecuada para implementar el sistema en cuanto a:

Hardware

- Cámara IP
- Servidor de Procesamiento de Imágenes
- Dispositivos de Red (router, switch)

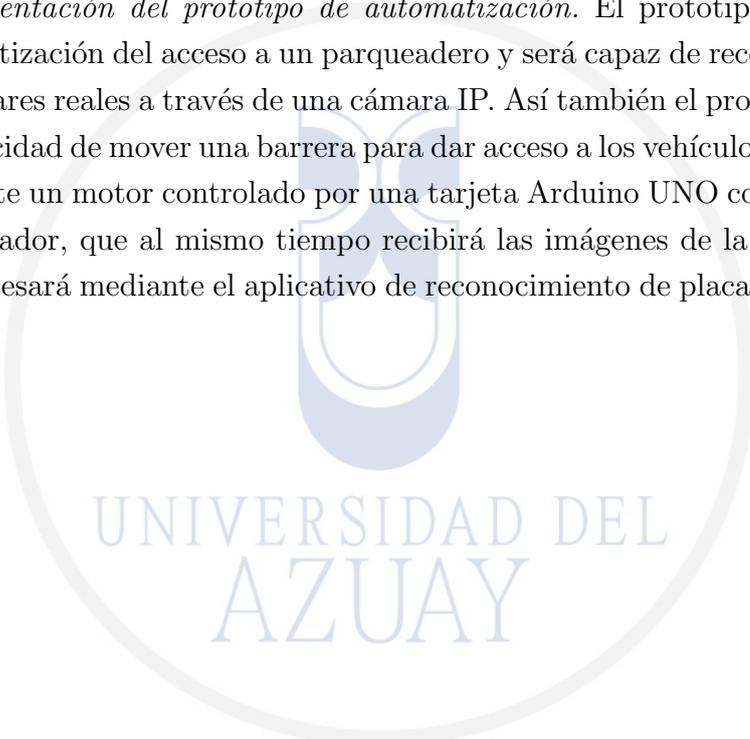
Software

- Red de datos
 - Base de datos
 - Lenguaje de Programación para ambientes de Escritorio y Web
 - Librerías de visión por computador
 - Servidor de Aplicaciones Web
5. *Desarrollo del Sistema que sea capaz de Reconocer Placas Vehiculares*, previo a un análisis y establecidos los elementos más adecuados que conformarán la plataforma del sistema, se procederá a la creación del software, en base a las técnicas y algoritmos estudiados anteriormente.
 6. *Pruebas de laboratorio*. A medida que se desarrolla el sistema, en paralelo se realizarán experimentos y pruebas de verificación, para lo cual se recopilará un número considerable de imágenes de placas de vehículos diferentes, bajo

³<https://code.google.com/p/tesseract-ocr/>

condiciones de iluminación y ubicación de la cámara, totalmente controladas. En base a los resultados obtenidos se podrá determinar si establecer el sistema con la mejor de las técnicas evaluadas, o parametrizar un mecanismo de votación en donde simultáneamente operarán más de un algoritmo, o también existe la alternativa de realizar una fusión de estas técnicas de manera que el sistema final sea más robusto.

7. *Implementación del prototipo de automatización.* El prototipo exhibirá la automatización del acceso a un parqueadero y será capaz de reconocer placas vehiculares reales a través de una cámara IP. Así también el prototipo tendrá la capacidad de mover una barrera para dar acceso a los vehículos autorizados mediante un motor controlado por una tarjeta Arduino UNO conectada a un computador, que al mismo tiempo recibirá las imágenes de la cámara IP y las procesará mediante el aplicativo de reconocimiento de placas vehiculares.



UNIVERSIDAD DEL
AZUAY

CAPÍTULO 2

ANÁLISIS VEHICULAR A TRAVÉS DE IMÁGENES

2.1. Introducción

En los últimos años, un incremento notable del número de vehículos presente en las principales ciudades del mundo ha originado la necesidad de implementar sistemas que permitan, por ejemplo, la gestión y el análisis automático del flujo vehicular a través de técnicas de procesamiento digital de imágenes. Entre estas aplicaciones, se encuentra el reconocimiento de una Placa Vehicular - LPR (License Plate Recognition) en una imagen o secuencia de imágenes (video). La demanda de estos sistemas incrementa cada día, consolidándose diversas aplicaciones que incluso hoy en día forman parte de la vida común de las personas, como por ejemplo (Liaqat, 2011):

- Análisis de tráfico en carreteras.
- Cobro automático de peaje en autopistas.
- Detección de vehículos robados o con infracciones y/o multas.
- Estacionamiento de vehículos.
- Multas para vehículos con exceso de velocidad.

Los sistemas para el análisis vehicular son cada vez más robustos, debido al uso de múltiples cámaras y otros sensores que complementan su infraestructura. Además, debe considerarse el avance en cuanto a las técnicas para el análisis de video e imágenes ligado al crecimiento en el procesamiento computacional de los últimos años (Buch, Velastin, y Orwell, 2011).

Los sistemas LPR operan en diferentes ambientes y en general, requieren incorporar mecanismos de ajuste para los diferentes factores climáticos que afectan las condiciones de iluminación de manera dinámica causando que el sistema deba adaptarse al ambiente y ser capaz de cumplir su objetivo al mismo tiempo (Khalil y Kurniawan, 2014).

El análisis vehicular a través del reconocimiento de placas vehiculares requiere del área de procesamiento de imágenes, en donde se utilizan diferentes técnicas y algoritmos para la obtención e inferencia de cualquier tipo de información. La mayoría de veces, el tratamiento de imágenes está enfocado a mejorar o resaltar detalles que faciliten la obtención de características que posibiliten la extracción de información útil para tareas de clasificación, comparación, dimensionamiento geométrico, etc.

2.2. Visión por computadora en el análisis del tráfico vehicular

El uso de las cámaras instaladas para monitorear carreteras en conjunto con técnicas de visión por computador ofrecen una alternativa atractiva para el análisis del tráfico vehicular. En particular, los sensores basados en tratamiento digital de imágenes tienen una potencial ventaja que incluye el poder medir una gran variedad de parámetros en comparación con los sensores tradicionales (Setchell, 1997). En Ecuador por ejemplo, es muy común encontrar dispositivos medidores de velocidad en sus principales carreteras y autopistas, como el que se observa en la Figura 2.1. Los sensores más usados actualmente para medir eventos de tráfico vehicular son los radares, detectores de microondas y detectores de bucle magnético o inducción (Huang y Yen, 2004).



Figura 2.1: Radar indicador de velocidad usado en Ecuador. Fuente: (LaHora, 2013).

Tanto en el tráfico vehicular de carretera como en el de ambientes urbanos, los objetivos de analizar y aplicar técnicas de visión por computador pueden ser varios, entre ellos se destaca la detección de infractores de leyes de tránsito, el conteo de vehículos, estudios de flujo de tráfico, la identificación de tipos de vehículos (sedanes, camiones o camionetas) y la identificación del registro vehicular (identificación del dueño de un vehículo). Para esta última tarea, el enfoque más usado y confiable se centra en el reconocimiento automático de las placas vehiculares, también conocido como ANPR (Automatic Number Plate Recognition) o ALPR (Automatic License Plate Recognition), según se establece en (Buch y cols., 2011). En poco tiempo países como Reino Unido, Singapur, Japón, Canadá, Alemania, Italia, Estados Unidos y Francia han desarrollado e implementado exitosamente sistemas de reconocimiento de placas vehiculares para ciertas actividades de la gestión de tráfico (Jin y cols., 2012).

2.2.1. Aplicación de conteo de vehículos.

Comúnmente, el problema de registrar el número de vehículos circulando suele ser resuelto con el uso de sensores de inducción (Buch y cols., 2011), los cuales proveen resultados muy confiables pero a la vez bastante vulnerables a intrusiones por el pavimento o asfalto de los caminos y carreteras, además de ser un método bastante costoso en términos de mantenimiento. Un sensor de inducción o sensor inductivo es capaz de detectar cuerpos metálicos, como un automóvil. Se basa en un oscilador que reacciona a las variaciones de corriente inducidas por 2 bobinas cercanas o lejanas de objetos metálicos. La distancia de detección puede variar por factores como las características metálicas de la pieza a detectar, la temperatura o el alcance nominal del sensor (Monzó, 2014). Otro enfoque se basa en implementar sistemas de visión por computador, que utilizan cámaras ubicadas en postes muy altos de tal forma que las imágenes adquiridas reflejen la escena de análisis en su totalidad. Los datos del número de vehículos en una carretera o camino en particular, se utilizan para tomar acciones como habilitar carriles adicionales para evitar una posible congestión del flujo vehicular (Setchell, 1997).

2.2.2. Reconocimiento Automático de Placas Vehiculares

Involucra un campo muy amplio de investigación y aplicación en el análisis de video. Existen varias compañías como 3M¹ dedicadas a brindar soluciones para peajes, manejo de congestionamiento vehicular, identificación de vehículos o cobro de tarifas (Buch y cols., 2011). Para este fin es común el uso de cámaras con

¹<http://www.3m.com/>

una alta resolución con el fin de garantizar el nivel de detalle necesario para extraer características de las imágenes e inferir información, además estos sistemas hacen uso de heurísticas relacionadas al formato de las placas vehiculares establecido por cada país. Así, existen estaciones de peaje que usan carriles dedicados con cámaras para permitir acceso a vehículos autorizados sin necesidad de que el mismo se detenga, puesto que es automáticamente reconocido y tarifado por un sistema e incluso con posibilidad de vincular estos pagos a una cuenta bancaria (Setchell, 1997).



Figura 2.2: Peaje o acceso automatizado con un sistema ALPR. Fuente: (BitWindows, 2015).

En este contexto, es importante establecer que el proceso de reconocimiento automático de placas vehiculares se compone fundamentalmente de dos etapas: i) Detección de la placa, y ii) Reconocimiento de sus caracteres. Estos dos procesos se analizan con detalle en las secciones 2.3 y 2.4 respectivamente.

2.2.3. Detección de Incidentes

La detección de incidentes se enfoca mayormente en el entendimiento de una escena, así en ambientes tales como autopistas es posible detectar accidentes de tránsito o vehículos detenidos. También existen proyectos piloto (Setchell, 1997) en los que el análisis va enfocado a detectar vehículos compitiendo *hombro a hombro*. En túneles por ejemplo, se dedica el análisis a la detección de incendios o humo.

En ambientes urbanos el análisis se torna mucho más complejo que en una autopista y requiere un nivel más alto de entendimiento de la escena. Existen proyectos (Setchell, 1997) en los que se analiza congestionamientos vehiculares haciendo uso de cámaras de CCTV (Closed Circuit Television). También siste-

mas que requieren un alto nivel de seguimiento de vehículos para poder detectar maniobras ilegales (Ej. Un giro en U), el uso no autorizado de lugares de estacionamiento, monitoreo de carriles dedicados a buses que incluye además clasificación de vehículos (Buch y cols., 2011) para saber de dichas infracciones.

2.3. Detección de placas vehiculares

Las técnicas que conforman la detección de placas vehiculares deben ser elegidas de acuerdo a las necesidades del ambiente en particular en donde operará el Sistema, por lo que no se puede determinar el/(los) método(s) con más alto desempeño de manera general. De igual manera para cada método se debe tomar en cuenta los factores como el rendimiento, tiempo de ejecución y la plataforma, ya que afectarán directamente en el desempeño del sistema (Anagnostopoulos, Anagnostopoulos, Psoroulas, Loumos, y Kayafas, 2008).



Figura 2.3: La región con marco verde muestra la detección exitosa de la placa vehicular.

En cuanto se refiere a la extracción de la región de la placa, a continuación se presenta una categorización de los métodos más comunes junto con la descripción del procesamiento asociado a cada uno.

2.3.1. Procesamiento de Imágenes Binarias

Esta técnica se basa en combinaciones de métodos estadísticos de análisis y morfología matemática. (Utilizada para extraer componentes útiles de imágenes, para la representación y descripción de forma de regiones), como se documenta en (Radha, 2012). El método, se fundamenta en el principio de que el cambio de brillo en la región de la placa, es más remarcable y más frecuente que en cualquier otro lugar de la imagen. Una desventaja de este método, es que puede llegar a ser demasiado sensible a los bordes no deseados, mostrándolos con una elevada magnitud (por ejemplo, en la región del radiador en la vista frontal del vehículo). A pesar de esta desventaja, si este método se combina con una técnica para

eliminar los bordes no deseados de las imágenes procesadas, la tasa de extracción de la región de la placa es relativamente alta y además, la técnica es muy rápida en comparación con otros métodos, como se detalla en (Anagnostopoulos y cols., 2008).



Figura 2.4: Imagen Binarizada.

En (Martin, García, y Alba, 2002), el modelo conceptual del algoritmo propuesto, se basa en una operación morfológica llamada *top-hat*, que es capaz de localizar objetos pequeños con diferencias de brillo. Este algoritmo, con una tasa de detección de 80 %, es altamente dependiente de la distancia entre la cámara y el vehículo, ya que las operaciones morfológicas se refieren a la dimensiones de los objetos binarios. Una aplicación similar se describe en (C. Wu, On, Weng, Kuan, y Ng, 2005) obteniendo un nivel de precisión del 93 %.

Por otra parte existen también algoritmos híbridos de extracción de Placas Vehiculares como los que implementan los autores de (Hongliang y Changping, 2004), basándose en *estadísticas de borde y la morfología* para los sistemas de emisión de tickets para vigilancia de carreteras. Este enfoque se puede dividir en las siguientes cuatro secciones:

- La detección de bordes verticales.
- Análisis estadístico de bordes.
- Ubicación jerárquica basada en placas vehiculares.
- Extracción de placas basado en la morfología.

En la precisión promedio de localización de una placa vehicular, empleando estos algoritmos híbridos, se obtiene una impresionante tasa del 99,6 %, ya que sus autores mediante diferentes pruebas obtuvieron 9786 aciertos de las 9825 imágenes procesadas.

El **análisis de componentes conectados** o por sus siglas del inglés CCA (Connected Component Analysis), es una técnica fundamental en el procesamiento de imágenes binarias ya que escanea una imagen binarizada y las etiquetas de sus píxeles en componentes en función de la conectividad de los mismos (ya sea de 4 puntos conectados, o por lo general de 8 puntos) (Jonker, 1992). Los componentes conectados en el análisis de imágenes se definen como cualquier píxel conectado por sus caras (4 puntos), o por sus caras y aristas (8 puntos), según se muestra en la Figura 2.5.

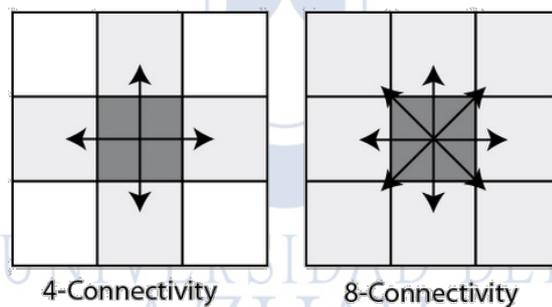


Figura 2.5: Análisis de componentes conectados. Fuente: (Dewar, 2015).

Una vez que todos los grupos de píxeles se han determinado, cada píxel está etiquetado con un valor de acuerdo con el componente al que fue asignado. El proceso de extracción y etiquetado de los diversos componentes conectados y desconectados en una imagen es básico para muchas aplicaciones de análisis de imágenes automatizado. **Mediciones Espaciales** tales como el área, la orientación y la relación de aspecto son sólo algunas de las características que se integran con frecuencia en los algoritmos de procesamiento de imágenes para la detección de placas vehiculares. Adicionalmente, usando técnicas de filtrado simples, los objetos binarios con mediciones que superan los límites deseados pueden eliminarse en los pasos algorítmicos subsecuentes.

El trabajo propuesto en (H.-H. Wu, Chen, Wu, y Shen, 2006) propone la extracción de placas vehiculares implementando mediciones espaciales, pero trabajando con videos de baja resolución (320x240 px) o degradados por la compresión del video. En algunos sistemas LPR, para ubicar la zona de la placa vehicular, se emplea la detección de los bordes de la imagen empleando el operador de **Sobel**, seguido de la eliminación de la mayor parte de los bordes de fondo y de ruido y la

búsqueda de la región placa utilizando una ventana rectangular. El operador de *Sobel* es considerado como un detector de borde clásico que convoluciona la imagen con un filtro en direcciones horizontal y vertical utilizando una máscara de 3x3 para cada dirección. La máscara es aplicada sobre toda la imagen, procesando un grupo de píxeles a la vez, lo que da énfasis a las regiones de alta frecuencia correspondientes a los bordes (Musoromy, Ramalingam, y Bekooy, 2010), como se muestra en la Figura 2.6.



Figura 2.6: Efecto de la aplicación del operador de Sobel en los bordes en una imagen de entrada.

2.3.2. Procesamiento de Nivel de Grises

1. Procesamiento de Imagen Global

En (Comelli, Ferragina, Granieri, y Stabile, 1995) se presenta un sistema denominado (RITA), *Riconoscitore di Targhe Automobilistiche* (Reconocedor de Placas de Automóviles), para el reconocimiento de placas de automóviles. En términos de procesamiento de imágenes para la identificación de la placa, RITA estaba compuesto de un módulo de localización del área de la placa y un módulo de procesamiento. Los pasos para la localización de la placa fueron adaptados para la estructura de placas italianas, cuya forma es rectangular y contienen caracteres negros sobre un fondo blanco. Por lo tanto, el algoritmo selecciona el área que presenta el máximo contraste local que posiblemente corresponde a la zona que contiene la placa.

Para la ubicación de la placa, varios sistemas utilizan un enfoque como el presentado en (Draghici, 1997), que consiste en escanear horizontalmente la imagen, en busca de repetidos cambios de contraste en una escala de 15 píxeles o más.

Draghici establece los supuestos que el contraste entre los caracteres y el

fondo de la placa es suficientemente buena y que hay al menos de tres a cuatro caracteres en una placa cuyo mínimo tamaño vertical es de 15 píxeles. Cabe recalcar que el valor particular de 15 píxeles se determina en base a la resolución de la cámara utilizada, la distancia promedio del vehículo desde la cámara y el tamaño real de los caracteres.

2. Análisis Parcial de la Imagen

En este método se escanea la imagen del vehículo parcialmente, realizando varias repeticiones del análisis en diferentes secciones de la imagen tomando en cuenta una distancia *N-fila* como límite de la sección, para luego contar los bordes existentes en dicha sección. Si es que el número de bordes es mayor a un promedio, se puede suponer que en esa área se localiza la placa. El tiempo de ejecución de este método es muy rápido ya que escanea solo ciertas filas de la imagen. El nivel de eficacia es muy efectivo y se puede implementar en varios escenarios. En este método no existen dependencias entre la distancia de la cámara y el tamaño de la placa vehicular, como se detalla en (Cano y Pérez-Cortés, 2003).

3. Mediciones Estadísticas

Al igual que en los métodos anteriores, se identifica las posibles regiones de la placa analizando las secciones con una alta magnitud o varianza de borde. Sin embargo no todos los bloques detectados en una imagen son realmente la región de la placa ya que son similares al coincidir con ciertos criterios geométricos como el área. En (Lee, Chen, y Wang, 2004) se analiza 180 pares de imágenes que incluyen principalmente placas de motocicletas obteniendo una tasa de precisión de 92.5 %.

4. Representaciones Jerárquicas

En (Zunino y Rovetta, 2000) se presenta un método basado en la cuantificación vectorial (CV) para procesar las imágenes de los vehículos. La CV puede dar algunas pistas sobre el contenido de las regiones de la imagen, proporcionando información adicional que puede ser aprovechada para aumentar el rendimiento de la localización de una placa vehicular. La CV adopta un mecanismo para la extracción de líneas. Las regiones de la imagen con mayor contraste y más detalle se asignan a bloques más pequeños. Las placas vehiculares pertenecen a esta clase de regiones debido al alto contraste entre los caracteres y el fondo. Además, la información del promedio del fondo podría ser de gran ayuda a la identificación de la placa, ya que normalmente tiene

un fondo brillante u oscuro. Por lo tanto, el conjunto de regiones de mayor importancia de la imagen, pueden adoptarse fácilmente por lo siguiente:

- a) La exploración de la estructura de la imagen para las áreas adyacentes asignadas por bloques de tamaño pequeño
- b) Explorar el correspondiente conjunto de bloques, con valores de brillo elevados (o de brillo bajos).

2.3.3. Procesamiento de Color

En la literatura, se proponen muchos métodos de tratamiento basados en el color para la localización automática de la placa vehicular. Entre ellos, técnicas que hacen uso de la apariencia de la placa como el fondo y el color del texto especificado en cada país. Por otro lado, las soluciones actualmente disponibles no proporcionan un alto grado de precisión en escenarios naturales, ya que el color no es estable cuando las condiciones de iluminación cambian. Además, como estos métodos se basan en el color, servirían de manera específica para ciertos países lo que los hace menos atractivos en el contexto de este proyecto.

2.4. Algoritmos y técnicas de reconocimiento de caracteres

Una vez conocida la ubicación de la placa vehicular dentro de la imagen provista, sin importar el método o técnica que se haya usado para ello, se procede al reconocimiento de los caracteres contenidos dentro de dicha matrícula, puesto que lo obtenido no representa datos utilizables en sí.

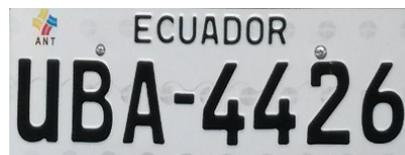


Figura 2.7: Placa vehicular (subimagen) obtenida tras usar técnicas de detección.

El reconocimiento de caracteres no es una tarea sencilla, existen varios algoritmos y técnicas para ello. Antes de reconocer un carácter o un texto, es necesario localizarlo en el espacio con el que se trabaja, es decir segmentar las áreas que son de interés. La segmentación suele ser difícil ya que en ciertos casos no es fácil definir dónde comienza y dónde termina un carácter, o si un carácter tiene los suficientes detalles para distinguirlo como tal, además se debe considerar el orden en el que se realizará el reconocimiento.

En las matrículas o placas vehiculares de Ecuador se pueden distinguir 2 categorías, muy notables por sus dimensiones y número de caracteres en función del año en el que fueron otorgadas a los vehículos. Según esta distinción, se ha establecido las siguientes denominaciones:

- *Placas modernas*: con dimensiones de 15cm por 40cm y 7 caracteres de identificación.
- *Placas tradicionales*: con dimensiones de 30cm por 15cm y 6 caracteres de identificación.
- *Placas de motocicletas*: con dimensiones de 19cm por 15cm y 6 caracteres de identificación.

Considerando la geometría normalizada en cada país, la segmentación en las placas vehiculares se simplifica en cierta medida al hacer uso de aspectos naturales como:

1. Los caracteres de identificación son de dimensiones similares.
2. Los caracteres pertenecen a una misma fuente o estilo de letra.
3. El orden de lectura es de izquierda a derecha.
4. Los caracteres de identificación se ubican en un solo renglón (aunque en el caso de las motos se usan dos).

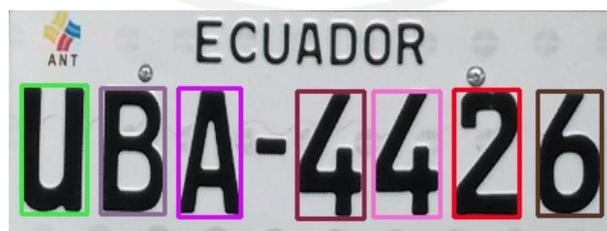


Figura 2.8: Caracteres segmentados que luego serán reconocidos con OCR.

Antes de realizar el reconocimiento de los caracteres, es necesario realizar un procesamiento adicional que permitirá la operación de las técnicas de reconocimiento que se analizan más adelante, como se puede observar en la Figura 2.9.

1. Obtener la imagen a reconocer, es decir una sub imagen de la placa vehicular detectada para los caracteres.
2. Encontrar un rectángulo que envuelva a esa imagen, es decir sin espacios que no contienen detalles relevantes de dicha imagen.

3. Reducir la escala, en caso del sistema propuesto a 16x16 píxeles por cada caracter.
4. Convertir en una cadena (*string*) los píxeles representados en la imagen como una matriz, haciendo un recorrido de filas y columnas.

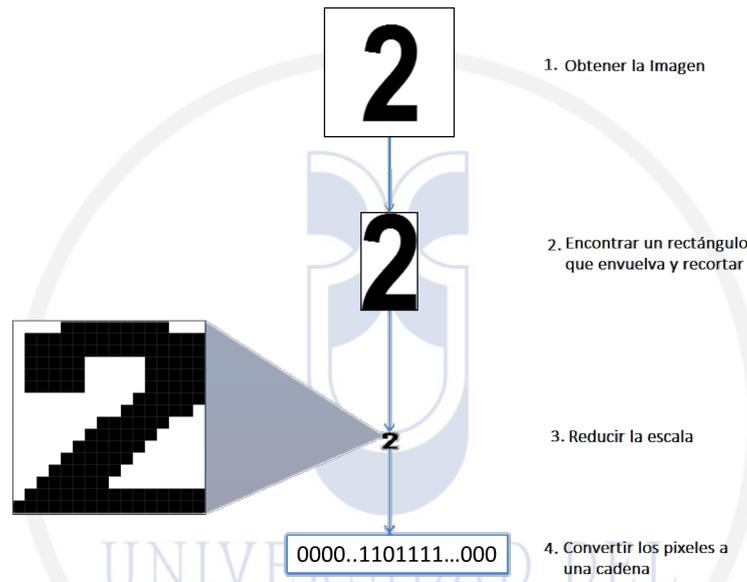


Figura 2.9: Convirtiendo una imagen a una cadena de “1” y “0”. Fuente: (*OCR using Artificial Neural Network (OpenCV) – Part 2 – Preprocessing*, 2015).

Una vez realizado este procesamiento, el sistema propuesto podrá operar con los métodos de reconocimientos presentados en las secciones 2.4.2 y 2.4.5. Las cadenas generadas son almacenadas en archivos de texto para ser recuperadas después por dichas técnicas implementadas. En el campo de la visión por computador, existe un gran número de investigaciones relacionadas al Reconocimiento Óptico de Caracteres (OCR), área que engloba a un grupo de técnicas basadas en estadísticas, formas, transformadas matemáticas y/o comparaciones, que usadas en conjunto o individualmente, permiten distinguir de una manera automática a un caracter alfanumérico de otro.

El Reconocimiento Óptico de Caracteres no trata de reconocer exactamente los caracteres de un solo alfabeto en particular, si no en general de un conjunto de símbolos establecido (Fernández y Consuegra, 2005). Por ello, es notable en la práctica, que el reconocimiento no es siempre exacto puesto que entre mayor sea el conjunto de símbolos es mayor la posibilidad de que se produzca un fallo, como por ejemplo, confundir una *T* con una *I*.

2.4.1. Comparación con Plantillas

Esta técnica, involucra el uso de una base de datos en la que se disponga de plantillas de los caracteres. Debe existir una plantilla separada para cada posible caracter que se ingrese. El reconocimiento se logra comparando un caracter de entrada con cada una de las plantillas siguiendo un orden para encontrar la mejor coincidencia.

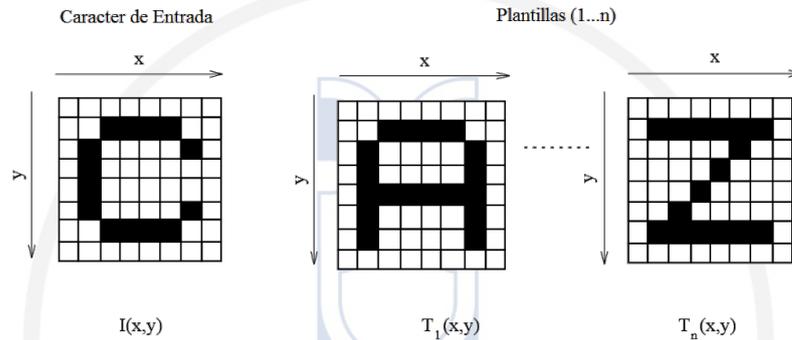


Figura 2.10: Comparación con plantillas. Fuente: (Setchell, 1997).

Si $I(x, y)$ es un caracter de entrada y $T_n(x, y)$ es una plantilla n , entonces la función de coincidencia $s(I, T_n)$ tendrá como resultado un valor que indique cuan bien la plantilla n coincide con el caracter de entrada (Setchell, 1997). Existen varias funciones de comparación de uso común, entre las que se destacan:

City Block

$$s(I, T_n) = \sum_{i=0}^w \sum_{j=0}^h |I(i, j) - T_n(i, j)| \quad (2.1)$$

Euclideano

$$s(I, T_n) = \sum_{i=0}^w \sum_{j=0}^h (I(i, j) - T_n(i, j))^2 \quad (2.2)$$

Correlación Cruzada

$$s(I, T_n) = \sum_{i=0}^w \sum_{j=0}^h I(i, j)T_n(i, j) \quad (2.3)$$

Correlación Normalizada

$$s(I, T_n) = \frac{\sum_{i=0}^w \sum_{j=0}^h (I(i, j) - I)(T_n(i, j) - T)}{\sqrt{\sum_{i=0}^w \sum_{j=0}^h (I(i, j) - I)^2} \sqrt{\sum_{i=0}^w \sum_{j=0}^h (T_n(i, j) - T)^2}} \quad (2.4)$$

El reconocimiento de caracteres se logra tras identificar cual es la plantilla T_n

que da el mejor resultado de coincidencia para una entrada $s(I, T_n)$. El método de *Comparación de Plantillas* es exitoso si la entrada y las plantillas almacenadas son iguales o muy parecidas. El método puede ser realizado sobre entradas en binario o en escala de grises. Para trabajar en escala de grises se usan funciones como la correlación normalizada ya que son inmunes a las variaciones de brillo.

2.4.2. Algoritmo K-Vecinos más Próximos (K-NN)

Se trata de un método o algoritmo muy conveniente, sin parámetros y de aprendizaje supervisado, que es capaz de proporcionar resultados bastante precisos. El algoritmo K-NN es muy conocido debido a la sencillez y otras propiedades que le proporcionan comportamientos aceptables al enfrentarse a problemas de clasificación, no solo en el contexto del reconocimiento óptico de caracteres (Fernández y Consuegra, 2005).

Para el uso de este método es necesario poseer un conjunto de prototipos o muestras de los objetos ya conocidos. Es decir, dado un conjunto de caracteres de muestra y teniendo una entrada se realiza una búsqueda de K prototipos con el objetivo de encontrar el más parecido a la entrada. Al finalizarse la búsqueda se le asigna la clase del vecino(s) más cercano(s).

El algoritmo se basa en calcular la distancia euclídeana de una entrada “X” hacia sus vecinos para tomar los K más cercanos en un espacio de características multidimensional, con el objetivo de vincularlo con una clase correspondiente. Al usar un enfoque basado en estereotipos, el algoritmo se vuelve más preciso y menos pesado en términos de procesamiento computacional. Un estereotipo es la muestra promedio de una clase. Teniendo las coordenadas de la entrada $X(x_1, x_2, \dots, x_n)$ y de sus vecinos $V(v_1, v_2, \dots, v_n)$ es posible calcular la distancia euclídeana con la Ecuación 2.5, que se presenta de forma generalizada en la Ecuación 2.6.

$$d_E(X, V) = \sqrt{(x_1 - v_1)^2 + (x_2 - v_2)^2 + \dots + (x_n - v_n)^2} \quad (2.5)$$

$$d_E(X, V) = \sqrt{\sum_{i=1}^n (x_i - v_i)^2} \quad (2.6)$$

En la Figura 2.11 se puede apreciar las distancias d_1 , d_2 y d_3 , medidas desde la entrada “X” a cada uno de los estereotipos de las clases. En caso de ser $K = 1$, se tomará la menor distancia y se le asigna a “X” la clase correspondiente al estereotipo seleccionado. En caso de ser $K \geq 2$, se calcula distancias a todos los estereotipos y tras obtener los K más próximos se calcula la distancia a todas

las muestras pertenecientes a las clases de dichos estereotipos. Realizado esto es necesario filtrar una vez más a las más próximas usando un número como $2K$. Terminada la selección de las $2K$ muestras, se cuenta las ocurrencias de cada clase y la que tenga el mayor número será asignada a la entrada.

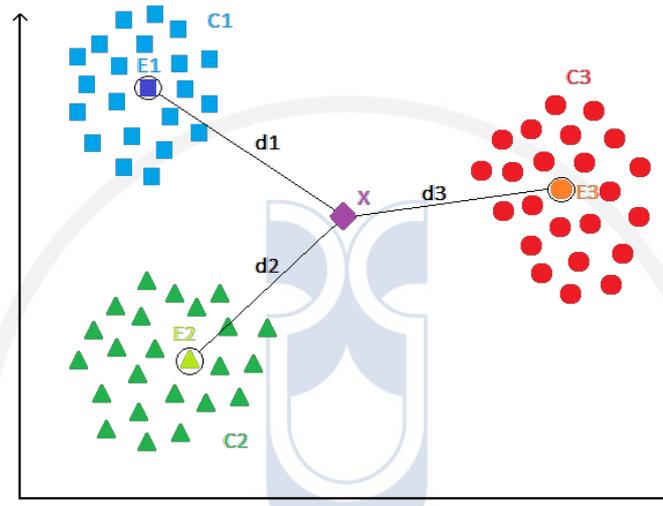


Figura 2.11: Búsqueda de la clase correspondiente a “X” en base a sus “K” vecinos más próximos, tomando inicialmente los estereotipos E_1 , E_2 y E_3 .

Funcionamiento del algoritmo K-NN

En este ejemplo se busca una clase a la que corresponda una entrada “X”. Tomando los estereotipos de la Tabla 2.1, en la que se encuentra las distancias de cada uno hacia la entrada y la clase a la que pertenecen, y al establecer $K = 3$, se obtiene que los vecinos más cercanos a “X” son E_1 , E_3 y E_4 .

Clase	Estereotipo	Distancia hacia “X”
C1	E_1	1.88
C2	E_2	1.98
C3	E_3	1.74
C4	E_4	1.90
C5	E_5	2.65

Tabla 2.1: Estereotipos representantes de cada clase.

Con los estereotipos más cercanos se procede a calcular nuevamente distancias euclidianas, pero esta vez solamente a las muestras pertenecientes a las clases de los estereotipos. Ahora se escogen las $2K$ muestras más cercanas de la Tabla 2.2, donde cada n -ésima muestra se denota por Mc_n .

Clase	Muestra	Distancia hacia "X"
C_1	Mc11	1.88
C_1	Mc12	1.87
C_1	Mc13	1.70
C_3	Mc31	1.74
C_3	Mc32	1.76
C_3	Mc33	1.61
C_4	Mc41	1.90
C_4	Mc42	1.79
C_4	Mc43	1.92

Tabla 2.2: Muestras pertenecientes a las clases filtradas con los estereotipos.

La Tabla 2.3 muestra los $2K$ vecinos más próximos a "X" filtrados de la Tabla 2.2. Se puede observar que existen 3 muestras pertenecientes a la clase C_3 , 2 pertenecientes a C_1 y 1 perteneciente a C_4 . En esta última parte del algoritmo K-NN se aplica un sistema de votación en base a las clases. Ya que la clase con mayor número de muestras cercanas a "X" es C_3 , es la que se asigna a dicha entrada.

Clase	Muestra	Distancia hacia "X"
C_1	Mc12	1.87
C_1	Mc13	1.70
C_3	Mc31	1.74
C_3	Mc32	1.76
C_3	Mc33	1.61
C_4	Mc42	1.79

Tabla 2.3: $2K$ muestras más cercanas a la entrada "X".

2.4.3. Reconocimiento en base a características

Esta técnica se realiza extrayendo *características* significativas de un caracter de entrada. Dichas características son compradas con una base de datos contenedora de descripciones de características para todas las posibles entradas y pueden ser de varios tipos, como se menciona en (Setchell, 1997):

- *Caraterísticas producidas por transformadas globales o series de expansión.*
Reducen la dimensionalidad del vector de características y no varían demasiado en traducción, escala o rotación. Son usadas las series de Fourier, Haar o Transformadas de Hough.
- *Características provenientes de la distribución estadística de puntos.*

Incluyen entre algunos ejemplos momentos, *zoning* (extracción de características por zonas), *n-tuplas*, etc. Las cuales no varían mucho en traducción o rotación incluso con fuentes distintas.

- *Características geométricas y topológicas.*

Son altamente no variantes con diferentes fuentes y no toman en cuenta rotaciones o traslaciones. Las más comunes usan carreras y bahías (strokes and bays), intersección de líneas, relación de ángulos entre líneas.

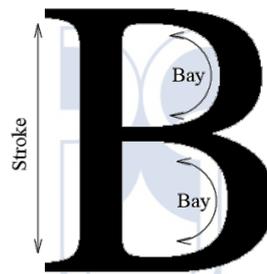


Figura 2.12: Reconocimiento usando Características Geométricas y Topológicas (strokes and bays). Fuente: (Setchell, 1997).

2.4.4. Árboles de decisión

Al igual que K-NN, es una técnica de minería de datos y su aprendizaje es inductivo y no supervisado. Los patrones o atributos a evaluar de un carácter de entrada son los nodos del árbol, mientras que los resultados son las hojas. Terminada la construcción del árbol y su estructura, y si estas son eficientes, la técnica puede llegar a ser más efectiva en costo computacional que K-NN, según se explica en (Fernández y Consuegra, 2005).

Se debe recalcar que es necesario detallar la manera en la que se recorrerá y también que se pueden implementar algoritmos de poda con el objetivo de reducir la complejidad y realizar una búsqueda más rápida. De igual manera se pueden agregar nuevas *hojas* para incrementar la precisión del reconocimiento.

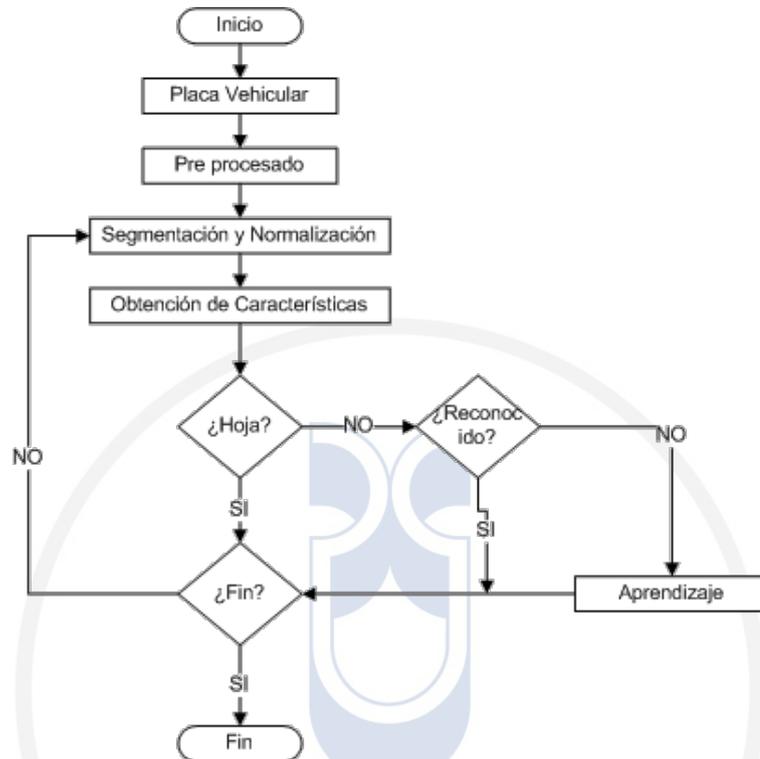


Figura 2.13: Recorrido de un Árbol de Decisión para OCR.

2.4.5. Redes Neuronales

Las redes neuronales son esquemas que pretenden imitar la arquitectura del cerebro. Se componen de unidades básicas llamadas *neuronas* que básicamente reciben una entrada vectorial, donde cada componente se multiplica por un factor correspondiente conocido como peso, con el objetivo de activar una neurona que procesa la suma ponderada de las entradas mediante una función de activación para entregar una salida. Sirven para representar eficazmente cualquier función que resultara muy compleja en términos algebraicos, así como en tareas de clasificación de patrones (Fernández y Consuegra, 2005). Los pesos asociados a cada neurona se pueden ajustar durante la fase de entrenamiento por medio de algoritmos de retropropagación o *back propagation* los cuales usan métodos basados en el descenso del gradiente de una función de error para minimizarlo. En general, bajo el esquema de un sistema de aprendizaje supervisado, una red neuronal se compone de una capa de entrada (*input layer*), una capa de salida (*output layer*) y una o más capas ocultas (*hidden layers*). La capa de entrada tendrá una neurona para cada característica del vector de entrada. Para el reconocimiento óptico de caracteres la capa de salida tendrá una neurona asignada a cada posible carácter (Setchell, 1997). Cuando un vector de entrada, producto

de la segmentación de caracteres previa, es ingresado a la capa de entrada, la neurona de salida que proporciona la respuesta más acertada es la que identifica al carácter, es decir a la clase que corresponde. Cada neurona o perceptrón posee una función de activación que hace uso de los pesos antes mencionados.

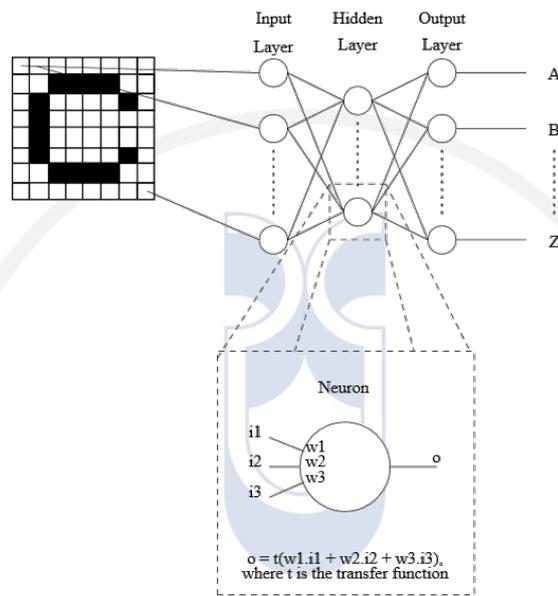


Figura 2.14: Reconocimiento óptico de caracteres usando una red neuronal. Fuente: (Setchell, 1997).

Cada neurona o perceptrón posee una función de activación que hace uso de los pesos antes mencionados. Típicamente la función de activación usada en una red neuronal suele ser sigmoide. *OpenCV*² implementa las funciones de activación *Identity*, *Symmetrical Sigmoid* y *Gaussian*.

$$f(x) = \beta \cdot \frac{1 - e^{\alpha x}}{1 + e^{\alpha x}} \quad (2.7)$$

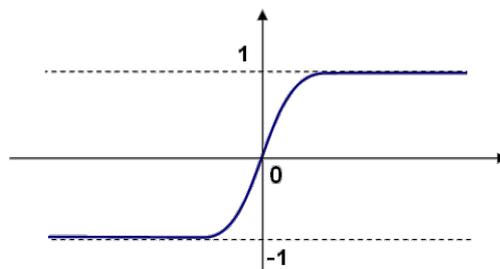


Figura 2.15: Función de activación *Symmetrical sigmoid* Fuente: (OpenCV Dev Team, 2015).

²www.opencv.org

Si se toma como entrada la Figura 2.16 para una red neuronal dedicada al reconocimiento de caracteres alfanuméricos, es decir números y letras, determinar la clase a la que pertenece sería poco preciso puesto que dicha entrada podría pertenecer a la clase numérica “cero” o a la clase alfabética “O”. Por tanto, para incrementar la precisión del reconocimiento, se puede usar una red neuronal dedicada al reconocimiento de caracteres letras y otra al reconocimiento de caracteres números.



Figura 2.16: Caracter que será reconocido.

Funcionamiento de una red neuronal

En la Figura 2.17 se puede observar un ejemplo de una red neuronal muy básica en la que su capa de entrada *input layer* posee 2 neuronas, su capa oculta *hidden layer* posee 2 neuronas y su capa de salida *output layer* tiene 1 neurona.

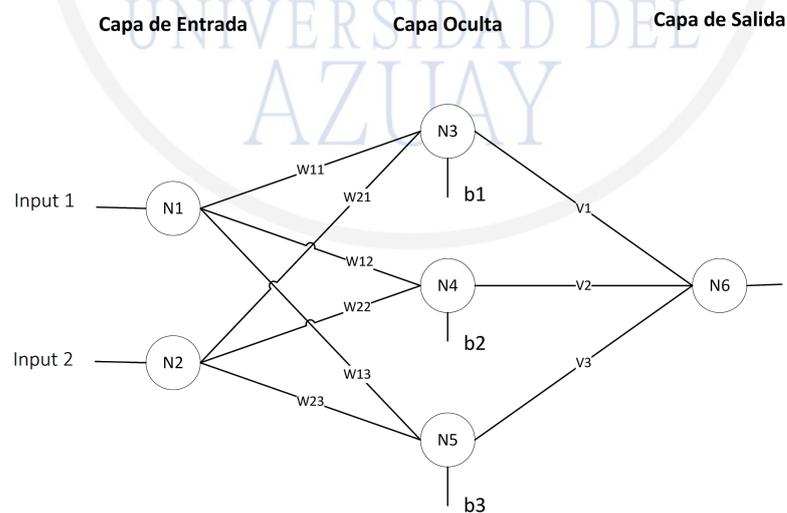


Figura 2.17: Red neuronal.

Cada conexión tiene sus pesos W_n y V_n . Esta red neuronal posee una función de activación sigmoide representada por la fórmula:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

La función de activación hace uso de los pesos especificados en las sinapsis de los perceptrones, están dados por las ecuaciones:

$$N1 = f(\text{Input } 1) \quad (2.9)$$

$$N2 = f(\text{Input } 2) \quad (2.10)$$

$$N3 = f(N1 * W_{11} + N2 * W_{21} + b1) \quad (2.11)$$

$$N4 = f(N1 * W_{12} + N2 * W_{22} + b2) \quad (2.12)$$

$$N5 = f(N1 * W_{13} + N2 * W_{23} + b3) \quad (2.13)$$

$$N6 = f(N4 * V_1 + N5 * V_2 + N6 * V_3) \quad (2.14)$$

Se puede apreciar la existencia de un peso adicional b_n en las neuronas $N3$, $N4$ y $N5$. Los llamados *bias* que son usados como una propensión, es decir una tendencia hacia determinado comportamiento, de los perceptrones para activarse o no ante determinadas entradas.

Los pesos W_n y V_n suelen ser inicialmente iguales a cero o algún valor aleatorio, que posteriormente en las fases de entrenamiento cambian usando una constante de ajuste C . Además, es necesario establecer el número de iteraciones que serán usadas para entrenar a la red neuronal.

El entrenamiento de la red neuronal es realizado con:

- Un **conjunto de entrenamiento**, *training set*, que posee las entradas y las correspondientes salidas que serán aplicadas en la red neuronal. Este conjunto es usado para ajustar los pesos W_n y V_n , de manera que se obtengan las respuestas correctas en la predicción.
- Un **conjunto de validación**, *validation set*, usado para evitar el sobreentrenamiento y garantizar la capacidad de generalización de la red neuronal. Este conjunto no se usa para ajustar los pesos, si no para verificar si existe algún incremento en la precisión, tras terminar el entrenamiento, al mostrar a la red neuronal un conjunto antes no conocido. Si la precisión con el *training set* incrementa, pero disminuye o se mantiene con el *validation set*, entonces la red se está sobreentrenando y no se debe seguir con el entrenamiento, es decir ya se ha encontrado un modelo apropiado.

- Un **conjunto de prueba**, *test set*, usado únicamente para probar la precisión predictiva de la red neuronal. Es decir para poder medir los resultados de un modelo obtenido.

Los conjuntos de entrenamiento y validación son disjuntos, y usados en secuencia hasta encontrar el modelo más preciso posible para la red, respetando el número de iteraciones establecido. El modelo con menor error de validación, es entonces utilizado para los datos del conjunto de prueba.

2.4.6. Tesseract

*Tesseract*³ es un motor OCR open-source desarrollado por HP⁴ entre los años 1984 y 1994. Nació como un proyecto de tesis doctoral de investigación en los laboratorios HP, en Bristol, y ganó fama al ser un posible “add-on” para la línea HP de scanners. Existía una gran motivación en su desarrollo por el hecho de que en aquellos días los motores OCR estaban en plena infancia. Después de trabajar en un proyecto conjunto los laboratorios HP en Bristol y la división de scanners HP en Colorado, *Tesseract* tuvo un incremento significativo en precisión frente a OCR’s comerciales. En 1995 fue enviado a UNLV (University of Nevada, Las Vegas) para el concurso “Annual Test of OCR Accuracy” en el que probó su eficacia frente a los motores OCR de aquella época (Smith, 2007). En 2005 HP liberó Tesseract como código abierto (*open source*), por lo que hoy en día se constituye como una solución ampliamente usada en diversos servicios de OCR en la web y muchas otras aplicaciones.



Figura 2.18: Placa recortada que será reconocida por Tesseract.

En el sistema propuesto en este proyecto, se utiliza Tesseract como uno de los métodos de reconocimiento de caracteres. La Figura 2.18 presenta un ejemplo de placa que será reconocida, en la que se puede observar la ausencia de bordes de la placa, y otras características no necesarias como la palabra “Ecuador” (usualmente presente en las placas vehiculares ecuatorianas). Se procura enviar la imagen al API de Tesseract lo más limpia posible, es decir sin detalles no necesarios, ya que pueden causar resultados menos precisos en el reconocimiento.

³<https://github.com/tesseract-ocr>

⁴www.hp.com

2.5. Conclusiones

El análisis vehicular mediante visión artificial ha incrementado su popularidad y su uso a medida que el número de vehículos en ciudades grandes ha crecido notablemente. Se ha encomendado a sistemas de tratamiento de imágenes el cumplimiento de varias tareas como la detección de incidentes y reconocimiento de vehículos, obteniendo resultados favorables para sus usuarios además de extender el uso de dichos sistemas.

Sin duda uno de los usos de estos sistemas es el reconocimiento de placas vehiculares (License Plate Recognition). Existen diversas técnicas para llevar a cabo esta tarea, tomando en cuenta que la misma involucra sub tareas como la detección de una placa vehicular en una imagen dada y el reconocimiento de los caracteres que contiene. El sistema propuesto pretende aplicar los métodos y técnicas más favorables para su correcta operación en un ambiente controlado como se esperaría en una implementación real. Se ha tomado en cuenta que las condiciones en las que el sistema operará no siempre serán favorables, por tanto el sistema propuesto pretende ser lo más robusto posible.

3.1. Introducción

Con el objetivo de realizar una implementación modular, el sistema propuesto puede ser dividido en bloques o procesos, como se puede observar en la Figura 3.1. La secuencia que sigue el sistema es un enfoque basado en un sistema de votación sobre los resultados provistos por varios subsistemas de reconocimiento y localización de placas vehiculares.

En cuanto al proceso *Captura de Imagen* se usa una cámara IP cuyas características y detalles de implementación se pueden encontrar en la sección de 3.4.

El bloque de *Detección de Placa* involucra el preprocesamiento de la imagen capturada, convirtiéndola en escala de grises y realizando la ecualización del histograma para obtener una distribución uniforme en el nivel de grises de la imagen, posterior a ello se aplican variaciones de umbral (threshold) u operaciones morfológicas.

Una vez preprocesada la imagen, para la tarea de detección de la placa se utiliza un conjunto de técnicas principalmente relacionadas con la detección de bordes, con el objetivo de determinar los posibles candidatos a ser identificados como una placa en la imagen, mediante métodos de búsqueda de contornos y otros como se detalla en la sección 2.3.

Detectar la placa vehicular no termina el proceso de identificar su posición en el plano de la imagen digital, sino que es necesario practicar la *Segmentación de Caracteres* o de Regiones que son de interés para el reconocimiento, lo cual involucra un preprocesamiento similar al usado para detectar la placa vehicular.

Detalles de este proceso se pueden encontrar en la sección 2.4.



Figura 3.1: Procesos a seguir para el reconocimiento de una placa vehicular.

Para el *Reconocimiento de Caracteres*, más conocido como OCR (Optical Character Recognition), existen varios enfoques y técnicas que se detallan en la sección 2.4, las más usadas suelen ser las redes neuronales (ANN) y algoritmos de los K-vecinos más cercanos (KNN).

El presente capítulo corresponde a la etapa de análisis y especificación de los requisitos, es decir, se describirán las funcionalidades con las que el sistema debe contar en base a los requerimientos, funcionales y no funcionales, especificados por el usuario. Una vez especificados los requisitos del usuario se presenta la etapa de Diseño del Sistema de Reconocimiento de Placas Vehiculares, mediante el uso de diagramas UML y a su vez se detallan las especificaciones de hardware y software bajo las que opera el sistema.

3.2. Análisis y planificación de requisitos

En esta sección se plasma el análisis del sistema propuesto, tomando como base la especificación de sus requisitos funcionales y no funcionales. Los requisitos funcionales se agrupan y especifican en casos de uso, y los no funcionales son detallados brevemente en función del estándar ISO/IEC 9126 ¹. También se describe de manera muy breve las interfaces de comunicación del sistema. Este análisis es el primer paso para realizar el diseño del sistema, el cual se detalla más adelante en la sección 3.3

3.2.1. Ámbito del sistema

Este proyecto comprende tareas básicas como el análisis y obtención de la imágenes capturadas, hasta el desarrollo de las interfaces de administración y de ajustes para detección de placas vehiculares. El sistema contempla lo siguiente:

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

Con el sistema propuesto se pretende disminuir los tiempos de ingreso a un parqueadero tras automatizar el proceso de autenticación de los usuarios registrados, incrementar la seguridad mediante el reconocimiento de las placas vehiculares y también proveer al usuario una sensación de personalización al identificarlo de manera única.

3.2.2. Descripción general

Perspectiva del producto

El sistema deberá permitir mejorar la administración de usuarios de un parqueadero, identificándolos mediante las placas de sus vehículos, lo que reducirá considerablemente los posibles errores originados usualmente por un operador humano.

¹<http://www.cse.unsw.edu.au/cs3710/PMmaterials/Resources/9126-1%20Standard.pdf>

1. Dependencias del producto

- Dependerá de una base de datos relacional para poder identificar a usuarios registrados y no registrados.
- El sistema depende de una red local LAN para comunicarse con la cámara IP.

2. Interfaces hardware

El sistema constará de un ordenador con los siguientes componentes.

- Cámara IP.
- Control de parante, implementado en el prototipo con una placa de desarrollo electrónico Arduino UNO² basada en el microcontrolador ATMe-ga328 de Atmel³.

3. Interfaces de Software

El sistema deberá presentar una interfaz de administración de usuario y una de ajustes/calibración para la detección de placas vehiculares, que sean intuitivas y fáciles de usar.

Funciones del producto

El sistema propuesto permite el ingreso a un parqueadero mediante el reconocimiento de placas vehiculares, asegurando que dicha identificación sea la correspondiente a la de cada usuario. Un diagrama de casos de uso esencial se muestra en la Figura 3.2.

Características de los usuarios

Los usuarios del sistema son identificados en primera instancia con una placa vehicular, por lo que se asume que siempre ingresarán al parqueadero con vehículo.

Para la parte de administración, que se realizará en un ambiente web, el administrador deberá tener un conocimiento básico del manejo de un computador, sin embargo el manejo de dicha interfaz deberá ser muy intuitivo.

Restricciones

El sistema será capaz de reconocer las placas vehiculares que obedecen a los formatos establecidos por la Agencia Nacional de Tránsito del Ecuador - ANT⁴,

²<https://www.arduino.cc/en/Main/arduinoBoardUno>

³www.atmel.com

⁴<http://www.ant.gob.ec/>

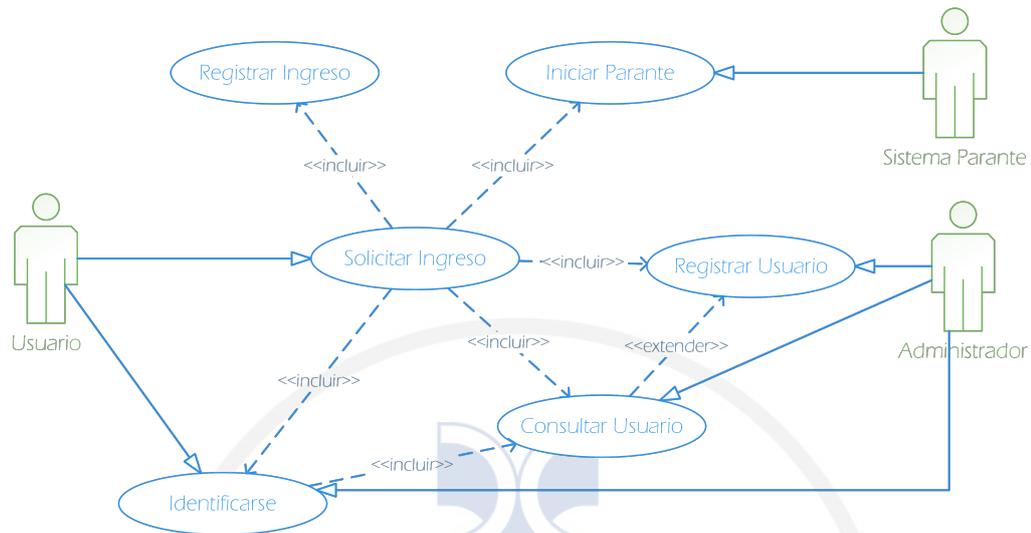


Figura 3.2: Diagrama de casos de uso esencial

sin incluir a motocicletas ya que la ubicación de la placa en estas no es favorable para el sistema propuesto.

Además, el sistema buscará identificar aquellas matrículas vehiculares que no cumplen con dichos formatos o se encuentran alteradas o deterioradas por agentes externos.

Suposiciones y dependencias

Para que el sistema sea completamente funcional, se asume que:

- Las placas vehiculares se encuentran en buen estado y que obedecen a los formatos establecidos por la ANT del Ecuador. Sin embargo, el sistema buscará de identificar aquellas que no cumplan dichas características.
- La cámara IP cumple con los requisitos especificados en la sección 3.4, y que la misma se encuentra ubicada en un lugar propicio y adecuado para obtener las imágenes que serán procesadas.
- La red local estará disponible y sin interrupciones para establecer conexión entre la cámara IP y el sistema.

Requisitos futuros

En un futuro, el sistema podría realizar tareas adicionales como:

- Identificar otros formatos de placas vehiculares.
- Identificar vehículos y registrar su salida del parqueadero.

- Identificar y reconocer placas de motocicletas.
- Integrar el uso de bases de datos externas como la de la ANT del Ecuador, en la que se encuentra información detallada de los vehículos.
- Usar cámaras IP de mejores prestaciones, capaces de entregar imágenes de mayor resolución.

3.2.3. Requisitos específicos

Requerimientos funcionales

Los requisitos funcionales son agrupados en los casos de uso que se pueden observar en la Figura 3.3, los cuales permiten tener una perspectiva general de la interacción de los usuarios y del ámbito del producto.

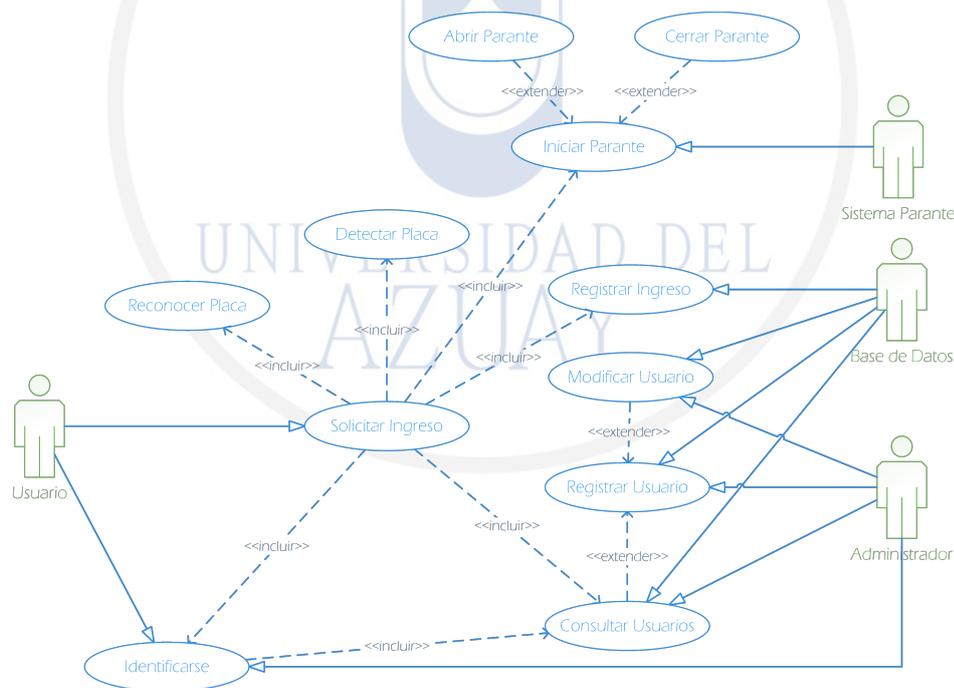


Figura 3.3: Diagrama de casos de uso extendido

1. Descripción de los actores

- **Usuario:** Persona que solicita el ingreso al parqueadero, puede estar registrado en el sistema o no.
- **Administrador:** Persona que maneja y gestiona las conFIGuraciones en el sistema LPR. Administra también el sistema para registro de usuarios

y vehículos.

- **Sistema parante:** Mecanismo controlado por Arduino UNO, que permite abrir y cerrar el parante ubicado en el ingreso del parqueadero.
- **Base de datos:** Colección de información referente a los usuarios registrados en el sistema y a los administradores que pueden utilizarlo.

2. Descripción de los Casos de Uso

La descripción de los Casos de Uso, se encuentra disponible en la sección de Anexos A.

Requerimientos no Funcionales

Los requerimientos no funcionales del sistema se han especificado en base a las seis características del *Modelo de Calidad de Software* definidos en el estándar ISO/IEC 9126.

1. Funcionalidad

- *Adecuación:* Se desea que todos los requerimientos funcionales sean cumplidos y se encuentren operativos.
- *Precisión:* Las placas vehiculares reconocidas deben corresponder a las pertenecientes a los vehículos.
- *Interoperabilidad:* El sistema debe interactuar fluidamente con una cámara IP que proveerá las imágenes a procesar y una base de datos relacional en la que se almacenan datos relacionados a las placas vehiculares.
- *Cumplimiento:* El sistema contará con una interfaz de usuario (GUI) amigable e intuitiva para los usuarios, que muestre la información necesaria para el usuario de manera adecuada.
- *Seguridad:* El sistema permitirá la autenticación de usuarios con reconocimiento de placas vehiculares o el número de cédula de identidad en caso de no poder reconocer la placa. Además de acceder a la base de datos relacional con un nombre de usuario y contraseña únicos.

2. Fiabilidad

- *Madurez:* Los errores en el sistema deben mantenerse al mínimo. Además que los errores no deben afectar la ejecución del sistema.

- *Tolerancia a Fallos:* El sistema deberá permitir que los usuarios ingresen al parqueadero incluso en los casos que sea difícil reconocer una placa vehicular (por ejemplo, por una placa sucia debido a las condiciones climáticas).
- *Recuperabilidad:* El sistema deberá poder reiniciar inmediatamente en caso de producirse un error que detenga la ejecución normal del mismo.
- *Cumplimiento de Fiabilidad:* El sistema deberá poder detectar y reconocer las placas vehiculares que cumplen con el formato establecido por la Agencia Nacional de Tránsito del Ecuador, además de la posibilidad de detectar las que no cumplen dicho formato.

3. Usabilidad

- *Comprensión:* El sistema deberá presentar la localización de la placa vehicular, los caracteres que contiene y los datos relacionados (Nombre del Propietario, Marca del Vehículo, etc). Y en su interfaz de administración los formularios y menús deberán ser comprensibles y mostrar los detalles suficientes.
- *Aprendizaje:* El sistema deberá ser altamente intuitivo para todos los usuarios, incluso aquellos que nunca hubieran usado el sistema, facilitando un rápido aprendizaje.
- *Operatividad:* El sistema contará con botones, menús, cajas de texto y otros elementos que permitan una correcta operación.
- *Atractividad:* El sistema mostrará una interfaz con una paleta de colores relacionada y con secciones de imágenes relacionadas.

4. Eficiencia

- *Tiempo:* Una placa vehicular deberá ser reconocida en un tiempo máximo de 2 segundos.
- *Uso de Recursos:* El sistema se ejecutará en un computador con sistema operativo Windows⁵ 7 u 8 y un procesador de varios núcleos y al menos 2.4GHz de velocidad.
Usará una base de datos relacional MySQL⁶ con un servidor Apache⁷ incluido en el paquete WampServer⁸.

⁵<http://www.microsoft.com/es-es/windows>

⁶<https://www.mysql.com/>

⁷<http://www.apache.org/>

⁸<http://www.wampserver.com/en/>

5. Mantenibilidad

- *Estabilidad:* El sistema será probado exhaustivamente después de cada modificación importante con el objetivo de mitigar efectos inesperados.
- *Facilidad de Análisis:* El sistema deberá indicar de manera clara y oportuna los posibles errores que se dieran en ejecución.
- *Facilidad de Cambio:* Se proveerá de la documentación requerida del sistema para facilitar cambios futuros.
El sistema contará con métodos y funciones genéricas que permitan breves modificaciones para adaptarse a otros formatos de placas vehiculares.
- *Facilidad de Prueba:* Se contará con un plan de pruebas para asegurar que el sistema cumple los requerimientos planteados.

6. Portabilidad

- *Capacidad de instalación:* Las especificaciones de hardware y software serán detalladas para que el sistema pueda ser usado en nuevas implementaciones.
- *Adaptabilidad:* El sistema usa las librerías de visión por computador OpenCV⁹, lo que permitiría migrar el sistema a otros lenguajes de programación como Java o Python. Al usar una cámara IP se elimina la necesidad de usar manejadores y programas específicos para un fabricante o modelo específico puesto que esta hace uso del protocolo TCP/IP para la transmisión del video.
- *Intercambiabilidad:* El sistema usa Qt¹⁰ Creator para generar su interfaz de usuario (GUI), sin embargo, es posible el uso de otros generadores de interfaces gráficas en otros sistemas e integrarlos a la parte operativa ya que ésta usa las librerías de OpenCV. La cámara IP podría ser reemplazada por una equivalente de otros fabricantes de similares características..
- *Co-Existencia:* El sistema es desarrollado en el lenguaje C++, pero usa las librerías de OpenCV lo que lo hace multiplataforma, por su disponibilidad en otros lenguajes como Java¹¹.

⁹<http://opencv.org/>

¹⁰<http://www.qt.io/developers/>

¹¹<https://www.java.com/en/about/>

3.2.4. Interfaces externas

Interfaces del software

El sistema debe establecer comunicaciones con una base de datos relacional en la que se almacenan datos relacionados a placas vehiculares. Además de comunicarse con una cámara IP la cual provee las imágenes para el procesamiento en tiempo real.

Interfaces del usuario

El sistema posee una **interfaz de escritorio**, para ajustar y calibrar la detección de las placas vehiculares. En esta interfaz el administrador podrá:

- Ajustar el tamaño y posición de la Región de Interés *ROI*.
- Ajustar el área de la placa vehicular.
- Calibrar el área y la posición del sistema sensor de imagen.
- Visualizar los métodos (*Otsu Threshold*, *Adaptive Threshold*, Operaciones Morfológicas) empleados para la detección de la placa vehicular.
- Utilizar los comandos para abrir y cerrar el parante directamente.

El sistema posee una **interfaz WEB**, para la administración de usuarios. En esta interfaz el administrador podrá:

- Ingresar, Modificar información de los usuarios.
- Ingresar, Modificar información de los vehículos.
- Consultar el reporte de usuarios - vehículos registrados.
- Acceder en tiempo real a la cámara IP para realizar video-vigilancia.

En esta interfaz el usuario podrá:

- Visualizar el mensaje de bienvenida con su nombre y número de placa
- Escuchar un mensaje de bienvenida.
- En caso de que el sistema lo requiera, el usuario podrá ingresar su número de cédula y una contraseña de 4 dígitos.

Intefaces de comunicaciones

La comunicación con la cámara se realiza mediante una red que usa el protocolo TCP/IP, mientras que la comunicación con la base de datos se realiza con un conector provisto por MySQL.

3.3. Diseño del sistema

En esta sección se toma como base el análisis de requisitos de los usuarios realizado en la sección 3.2 para presentar el diseño del sistema propuesto mediante diagramas UML. El sistema propuesto es detallado en los niveles de modelado estructural y modelado de comportamiento.

Modelado estructural

Este modelado sirve para describir los diferentes tipos y relaciones estáticas existentes entre los diferentes objetos de un sistema. Para entender este modelo realizamos una analogía de un arquitecto que crea los planos de un edificio antes de su construcción, con un ingeniero a la hora de desarrollar software, se debe modelar correctamente el sistema previamente al desarrollo, pudiendo incorporar el patrón o los patrones de diseño que ayuda a optimizar la solución a un determinado problema. La principal forma de emplear el modelado estructural es mediante el uso de diagramas de clase.

1. Diagrama de clases

Con los conocimientos adquiridos mediante la revisión presentada en el capítulo 2 se implementa el funcionamiento del sistema de reconocimiento de placas vehiculares. Este diagrama representa las clases involucradas en el funcionamiento del sistema, las cuales serán usadas para generar el código ejecutable de la aplicación, con sus respectivos atributos, métodos y relaciones; como se observa en la Figura 3.4.

La clase `rectangleTracking` posee los métodos y atributos para realizar todo el análisis requerido para detectar una placa vehicular, la misma hace uso de las librerías provistas en el paquete de visión por computador OpenCV. Además depende de las librerías de comunicación serial para poder comunicarse con el prototipo implementado con Arduino UNO.

- `procesarImagenAuto` es un método que permite el tratamiento previo a las imágenes, como binarización mediante `threshold` u operaciones

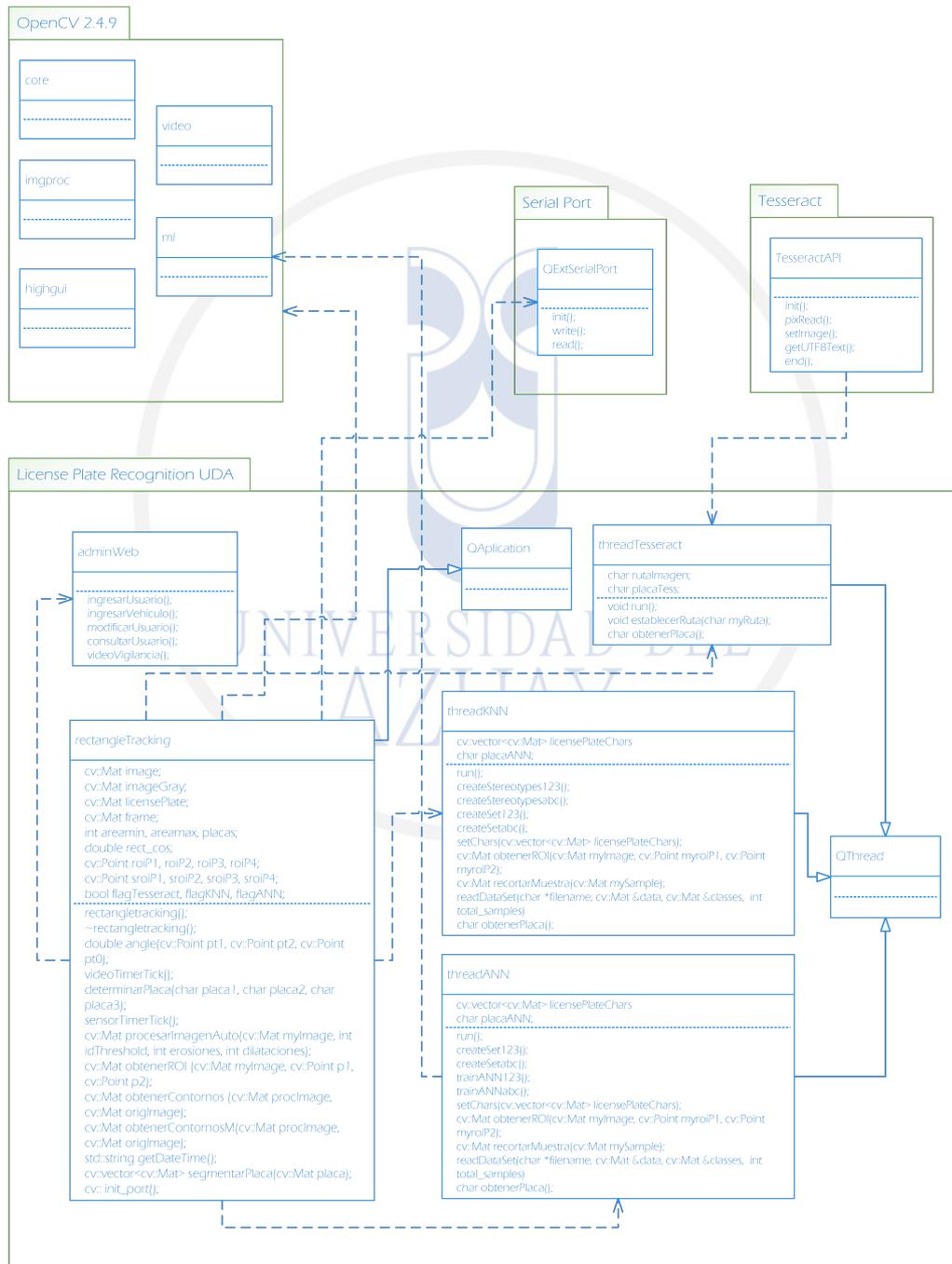


Figura 3.4: Diagrama de clases

morfológicas cuyos detalles se pueden encontrar en el Capítulo 2.

- `obtenerROI`, en base a 2 puntos de referencia permite extraer una región de interés de una imagen.
- `obtenerContornos` y `obtenerContornosM` hacen uso de métodos que realizan operaciones y transformaciones para buscar contornos y Figuras similares a placas. Dichos métodos necesitan que las imágenes sean procesadas por el método `procesarImagenAuto` y son usados para poder localizar las placas vehiculares contenidas en una imagen.
- `segmentarPlaca` realiza la segmentación de los caracteres contenidos en una placa. Se supone que para usar este método ya se debe haber localizado una placa vehicular en las imágenes procesadas.
- `videoTimerTick` se utiliza para obtener las imágenes generadas por la cámara IP de manera iterativa. En este método se hace uso de los métodos antes mencionados para hacer el procesamiento requerido para el reconocimiento de placas vehiculares en tiempo real.
- `sensorTimerTick` es usado para leer el estado del puerto serial, es decir para saber si existe un vehículo cerca del sensor láser de seguridad.
- `determinarPlaca` es un método para determinar una sola respuesta correcta en base a tres entradas que son provistas por los métodos de reconocimiento de caracteres que se encuentran en las clases `threadTesseract`, `threadANN` y `threadKNN`. Es un método de votación.
- `initport` se encarga de iniciar la comunicación con el puerto serial, por el cual se comunica el sistema con el prototipo de ingreso al parqueadero.

La clase `threadANN` corresponde a un *thread* o hilo, lo que le permite ser ejecutado en segundo plano y no en el mismo que el programa principal. Esta clase implementa una red neuronal para el reconocimiento de caracteres y hace uso de la librerías de aprendizaje de máquina *Machine Learning* de OpenCV. Se usa una red neuronal para reconocimiento de números y otra para reconocimiento de letras con el objetivo de aumentar la precisión del reconocimiento.

- `createSet123` y `createSetabc` son métodos usados para tomar imágenes binarizadas de números y letras respectivamente, normalizarlas con el método de procesamiento `recortarMuestra` y `obtenerROI`

para crear archivos contenedores de muestras que serán usados para entrenar a la red neuronal. Hace uso del método `readDataSet` para leer archivos de texto contenedores de las muestras.

- `readDataSet` lee un archivo de texto contenedor de muestras de caracteres letras y números, y las almacena en memoria usando una matriz. Cada muestra es representada como una sucesión de unos y ceros, y una etiqueta identificadora. Cada muestra tiene una longitud de 256 caracteres, ya que provienen de imágenes con tamaño de 16x16 píxeles. Algunos ejemplos de muestras son:

Muestras de números:

0,0,0,1,1,1,1,1,1,1,0,0,0,0 ... 0,0,0,1,1,1,1,1,1,0,0

0,0,1,1,1,1,1,0,1,0,1,0,1,1 ... 0,0,0,0,0,0,0,1,1,1,1

Muestras de letras:

0,0,0,1,1,1,1,1,1,0,0,0,0,0 ... 1,1,1,1,1,1,0,0,0,24

1,1,1,1,0,0,0,0,1,1,1,1,1,0,0 ... 1,1,1,0,0,0,1,1,1,1,18

- El archivo de texto contenedor de muestras para entrenamiento de letras, *training set*, contiene 780 muestras. 30 por cada letra.
 - El archivo contenedor de muestras para entrenamiento de números, contiene 200 muestras. 20 por cada número.
 - Los archivos de prueba, *test set*, contienen 390 y 100 muestras de letras y números, respectivamente.
- `trainANN123` y `trainANNabc` usan las matrices contenedoras de muestras generadas con el método anterior para crear, entrenar y guardar estructura de redes neuronales de números y letras respectivamente. Ambos métodos guardan la estructura de las redes neuronales de números y letras respectivamente, en archivos XML que serán usados posteriormente. Las redes neuronales han sido creadas con una función de activación *symetric sigmoid*, un número máximo de 50 y 90 iteraciones para números y letras respectivamente, y con una constante de ajuste de 0,000001. El entrenamiento se realiza mediante el algoritmo de retropropagación del error o *back propagation*.

La estructura de la red neuronal empleada se define con:

- Una capa de entrada (input layer), con 256 neuronas.

- Una capa oculta (hidden layer), con 16 neuronas.
- Una capa de salida (output layer), con 26 neuronas en el caso de la red neuronal de reconocimiento de letras, y 10 en el caso de la red neuronal de reconocimiento de números.

Tras terminar el entrenamiento y usar el *test set*, se obtuvo un 98 % de aciertos en la predicción de letras y un 99 % de aciertos para la predicción de números.

- `setChars` guarda un arreglo de imágenes con los caracteres que posteriormente serán reconocidos.
- `run` es el método que permite a esta clase trabajar en segundo plano como un *thread* o hilo. Aquí se usa los caracteres establecidos en el método anterior y se usa la estructura de las redes neuronales, almacenadas en archivos XML, para reconocerlos.
- `obtenerPlaca` es un método que permite obtener la respuesta generada.

La clase `threadKNN` también es un *thread* o hilo. Implementa un algoritmo K-NN (K Vecinos más Cercanos) basado en estereotipos para reconocer caracteres.

- `createStereotypes123` y `createStereotypesabc` son usados para crear los estereotipos y vecinos que serán usados para determinar las clases de una muestra particular. Se ha implementado un algoritmo KNN para números y letras con la intención de incrementar la precisión del reconocimiento, para ello se crean matrices en memoria, con el método `readDataSet`, con 200 muestras y 10 estereotipos, para el algoritmo de reconocimiento números. En el caso del algoritmo de reconocimiento de letras, se obtienen 780 muestras y 26 estereotipos almacenadas en matrices en memoria. Luego de obtener las muestras y estereotipos de números, se usaron 100 muestras para pruebas de reconocimiento de números, obteniendo un 81 % de respuestas correctas. Tras obtener las muestras y estereotipos de letras, se usaron 360 muestras para pruebas de reconocimiento de números, obteniendo un 79 % de respuestas correctas.
- `createSet123` y `createSetabc` son métodos usados para tomar imágenes binarizadas números y letras respectivamente, para luego normalizarlas usando el método `recortarMuestra` y `obtenerROI` para

crear archivos contenedores de muestras que serán usados para entrenar a la red neuronal. Hace uso del método `readDataSet`.

- `readDataSet` lee un archivo de texto contenedor de muestras de caracteres letras y números, y las almacena en memoria usando una matriz. Cada muestra es representada como una sucesión de 256 unos y ceros, y una etiqueta identificadora.
- `setChars` guarda un arreglo de imágenes que serán luego reconocidas.
- `run` es el método que permite a esta clase trabajar en segundo plano como un thread o hilo. Aquí se usa los caracteres establecidos en el método anterior y se usa el algoritmo K-NN para reconocerlos.
- `obtenerPlaca` es un método que permite obtener la respuesta generada.

La clase `threadTesseract` hace uso del API Tesseract para reconocer una imagen en el método `establecerRuta`. Devuelve una respuesta a través el método `obtenerPlaca`. Posee un método `run`, heredado de la clase `QThread` que lo convierte en un hilo.

Modelado de comportamiento

Este modelado describe el comportamiento del sistema cuando está en funcionamiento además de representar las interacciones en el sistema. Para el sistema propuesto se ha especificado los diagramas de actividad, estados y casos de uso.

1. Diagrama de actividad.

El diagrama de actividad recopila todo el proceso de ingreso a un parqueadero vehicular sin tomar en cuenta demasiados detalles, como los algoritmos para detección de placas por ejemplo. Para el sistema propuesto este diagrama representa de mejor manera la secuencia de actividades que realiza el sistema.

El diagrama describe las actividades del sistema y el flujo general desde el inicio de sesión hasta la operación del prototipo de parante de ingreso. El diagrama permite también conocer los actores o sistemas involucrados en las actividades, los cuales se detallan en los carriles usuario, administrador, sistema LPR, base de datos y sistema parante.

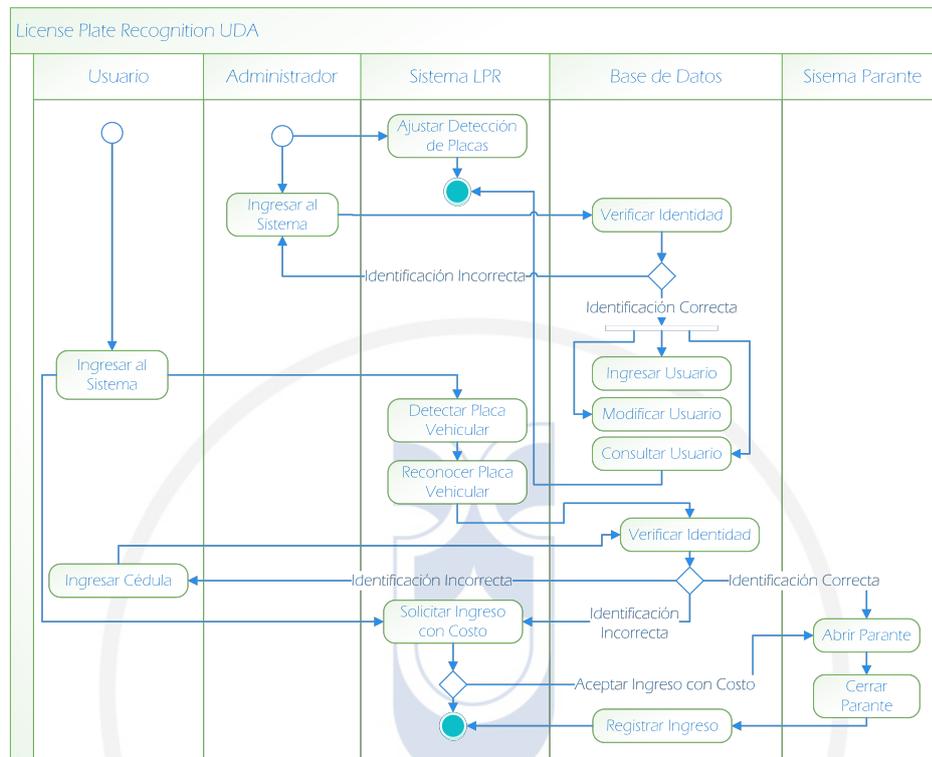


Figura 3.5: Diagrama de actividad

2. Casos de uso

Representa el comportamiento dinámico del sistema y se utiliza para reunir los requerimientos funcionales del sistema, identificar los actores involucrados en las funciones y la interacción entre actores y funciones.

En primera instancia se definió el diagrama de usos esencial, Figura 3.2, el cual muestra las funciones y actores principales del sistema propuesto de una manera sintetizada. Teniendo como caso de uso principal la *solicitud de ingreso* al parqueadero.

El diagrama de casos de uso, especificado en la sección anterior, Figura 3.3, muestra el refinamiento del caso de uso esencial. Aquí aparecen de forma más detallada otros casos de uso que en primera instancia no se describen. Un ejemplo puntual es la aparición del caso de uso de *abrir parante* el cual es usado tras identificar a un usuario del sistema.

3. Diagrama de estados

Describe una máquina de estados, que define los diferentes estados en los que puede encontrarse el sistema. El flujo de un estado hacia otro es controlado por los eventos internos y externos que se den mientras el sistema se ejecuta.

Los diagramas de estado han sido descritos desde el punto de vista del usuario y del administrador, como se observa en las Figuras 3.6 y 3.7.

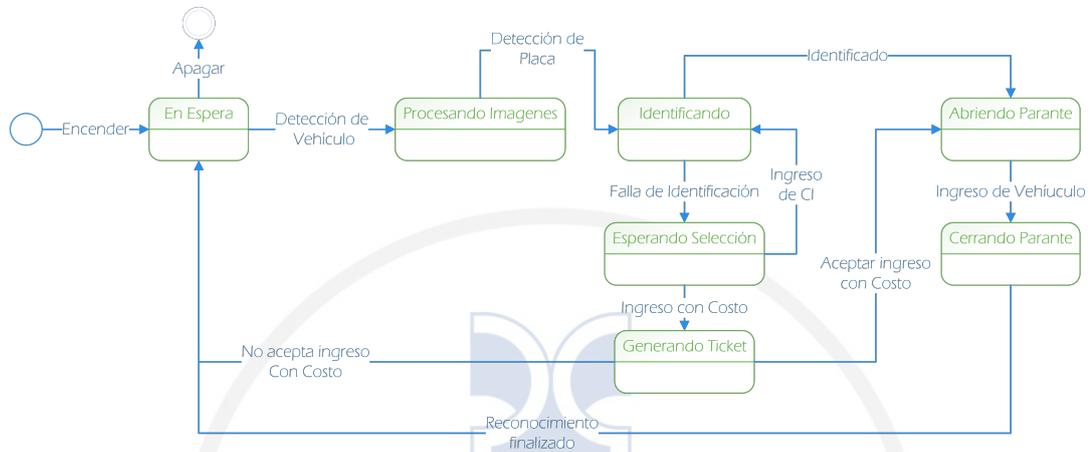


Figura 3.6: Diagrama de estados de usuario

En la Figura 3.6, el estado *En Espera* representa cuando la aplicación se encuentra en espera de detectar un vehículo para pasar al estado de *Procesando Imagen*. Una vez que la placa vehicular es detectada y reconocida se pasa al estado *Identificando* en el cual se busca que dicha placa conste en la base de datos de usuarios. En caso de que no se encuentre la placa vehicular se pasa a un estado *Esperando Selección* en la que el usuario puede optar por ingresar digitando su número de cédula o con un costo adicional *Generando Ticket*. Al finalizarse esto, el sistema vuelve al estado de espera.

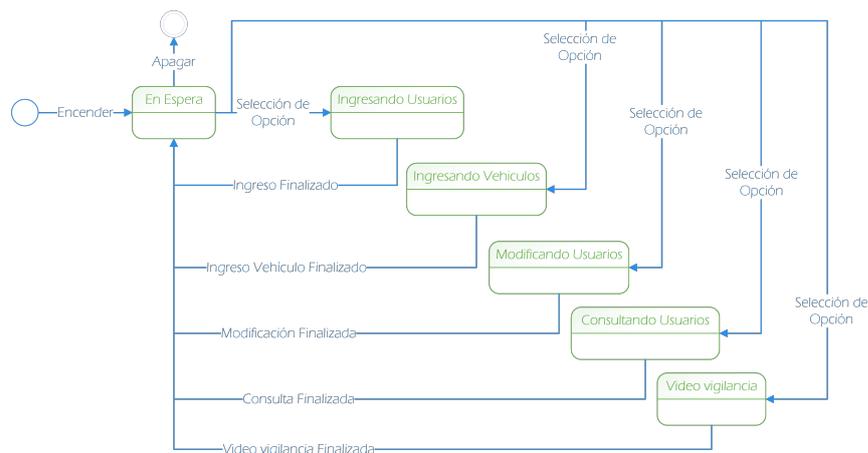


Figura 3.7: Diagrama de estados de administrador

El diagrama de la Figura 3.7 muestra los estados desde el punto de vista del administrador, los cuales siguen una secuencia sencilla. El administra-

El usuario puede elegir una opción y pasar a los estados *Ingresando Usuarios*, *Ingresando Vehículos*, *Consultando Vehículos*, *Modificando Usuarios* y *Video vigilancia*. Finalmente el sistema vuelve al estado de espera.

Modelado de la base de datos

Este modelo, presentado en la Figura 3.8, describe la estructura de tablas que permiten almacenar los datos referente a: usuarios, vehículos, direcciones y la bitácora para registrar las fechas de ingreso de los vehículos.

El diccionario de datos generado a partir del Modelo Entidad-Relación, se encuentra disponible en la sección de Anexos B.

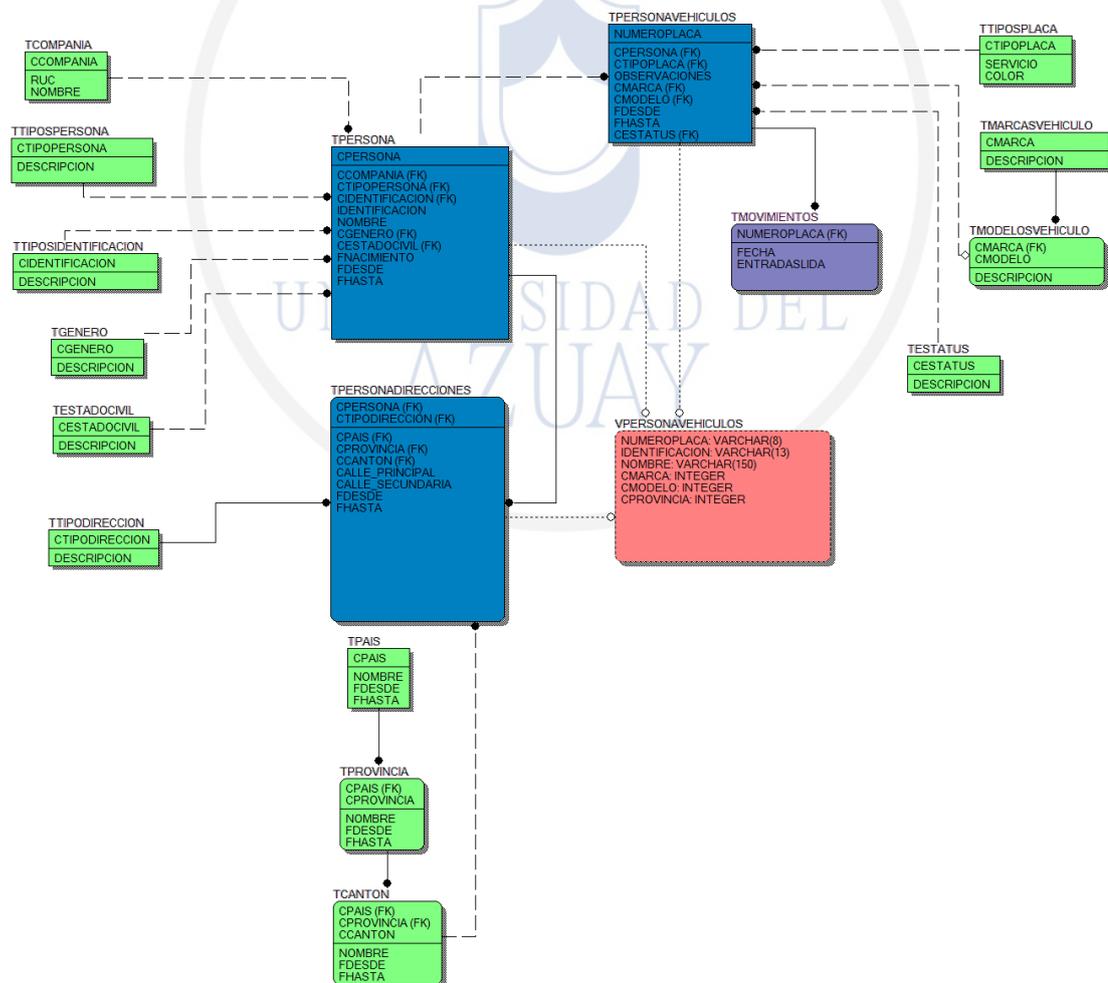


Figura 3.8: Modelo entidad-relación

Diseño del circuito electrónico

El circuito electrónico tiene varias funciones: Mediante un puntero láser y una foto resistencia se puede verificar cuando un vehículo se encuentra en la zona de ingreso - salida, cerca del parante, de esta forma se evita que el parante se cierre aun cuando el vehículo no ha ingresado-salido totalmente del parqueadero. Este circuito permite agregar un mecanismo de seguridad adicional para la detección del vehículo y el control del parante cuyo movimiento debe precautelar la seguridad de las personas y sus vehículos.

En la Figura 3.9 se presenta el Esquema Eléctrico del Circuito.

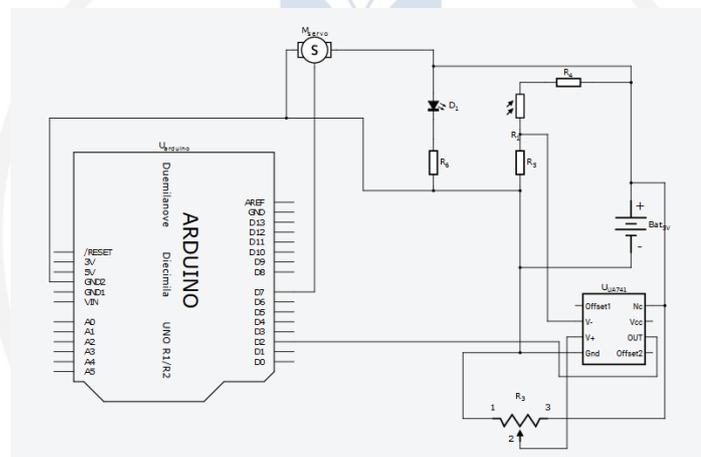


Figura 3.9: Esquema del circuito electrónico

Dentro del prototipo implementado, otra de las funciones del circuito electrónico es la de girar el mini servo motor de manera que se pueda levantar y cerrar el parante acoplado a su eje, para posibilitar el acceso a los vehículos.

Los componentes que integran el circuito electrónico son:

- 1 Tarjeta de desarrollo Arduino UNO
- 1 Mini servo motor
- 1 Amplificador operacional LM-358
- 1 Potenciómetro 10 K Ω
- 1 FotoResistencia LDR
- 2 Resistencias 1 K Ω
- 1 Resistencia 10 Ω

- 1 Puntero Láser
- 1 Cargador-Fuente 5V

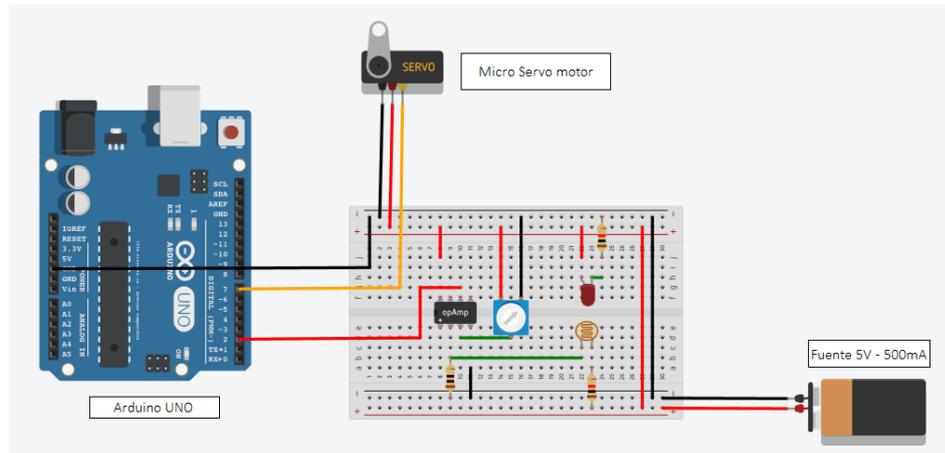


Figura 3.10: Implementación práctica del circuito electrónico.

3.4. Especificaciones Hardware del Sistema

3.4.1. Cámara IP

El presente trabajo, utiliza una cámara IP marca **FOSCAM**, modelo **FI9821W-V2**¹², que permite la captura de imágenes en **alta resolución (High Definition - HD)**, una característica fundamental ya que para obtener resultados aceptables en el procesamiento de imágenes, es recomendable contar con una alta resolución de la imagen capturada, a pesar del incremento en la carga computacional que ello conlleva.



Figura 3.11: Cámara IP foscam FI8921W V2. Fuente: (Foscam, 2015).

¹²<http://foscam.us/>

Esta cámara, también tiene la capacidad de operar en **modo nocturno**, ideal para la captura de imágenes en parqueaderos de ambiente cerrado. La falta de iluminación es un gran problema al momento del procesamiento de la imagen, porque se pierden objetos de interés y la visualización de una placa vehicular podría ser casi nula. Gracias a sus luces de infrarrojo con alcance de hasta 8m, se evitan estos inconvenientes.

Características

A continuación se describe las características más relevantes de la Cámara IP del sistema, según lo indica su fabricante¹³:

- Resolución HD 720, (1280x720 compresión H.264)
- LED Infrarrojo.
- Compatible con redes WiFi b,g,N y cable UTP
- Ranura para tarjeta microSD
- Sonido Bidireccional I/O
- Detección de movimiento con alarmas email y FTP
- Movimiento Robotizado
- DDNS gratis
- Compatible con S.O IOS y Android

Descripción Física

Entre las características físicas que integran el modelo FI9821W V2, se puede resaltar su *chipset* ya que es más rápido que modelos anteriores, ofreciendo un mejor rendimiento. También incorpora un micrófono y un altavoz para proporcionar audio de forma bidireccional.

¹³<http://foscam.us/foscam-fi9821w-megapixel-wireless-ip-camera.html>

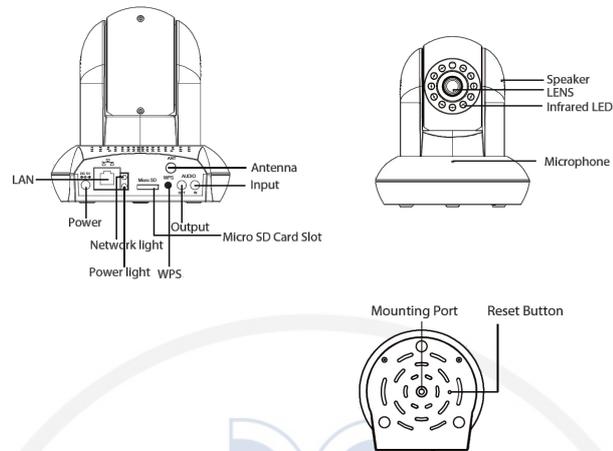


Figura 3.12: Descripción física cámara IP. Fuente: (Foscam, 2015).

Conexión Hardware

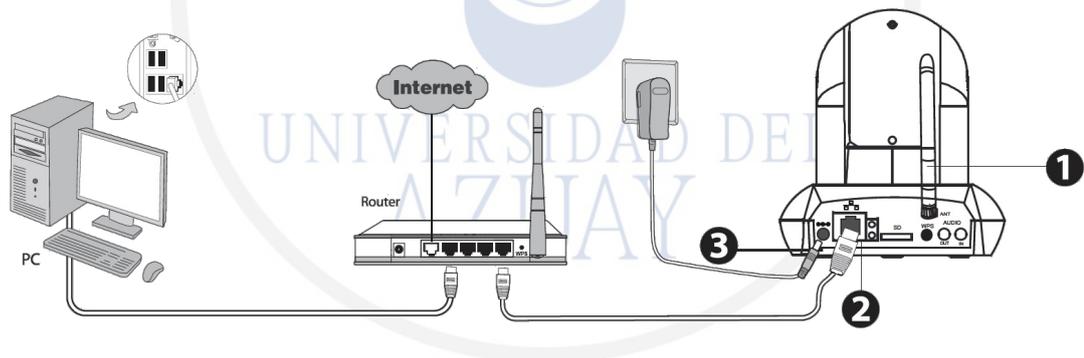


Figura 3.13: Conexión cámara IP. Fuente: (Foscam, 2015).

La conexión de la cámara requiere de tres simples pasos, según se ilustra en la Figura 3.13.

1. Montar la antena y colocarla de manera vertical.
2. Conectar la cámara a la red LAN (router o switch), vía cable de red.
3. Conectar el adaptador de corriente a la cámara.

Configuración inicial

En esta etapa, se procederá a configurar por primera vez la cámara IP, de manera que sea posible visualizar el video en tiempo real a través de su interfaz de usuario. Para la configuración de la cámara IP, se ha seguido los procedimientos

que el fabricante FOSCAM recomienda, en su Guía de Instalación Rápida¹⁴:

1. Insertar el CD en la unidad de CD del computador.
2. Buscar la carpeta “IP Camera Search Tool” y dentro de ella buscar la carpeta “For Windows OS” o “For Mac OS” (dependiendo del sistema operativo), copiar y pegar el archivo “IPCamera” en la computadora.

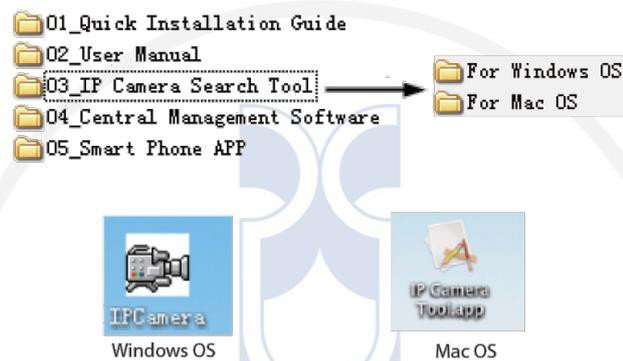


Figura 3.14: CD de instalación cámara IP

3. Habilitar la opción DHCP en las configuraciones del router (Normalmente por defecto) y abrir el software “IPCamera”, en esta pantalla se muestra la dirección IP de la cámara que le asignó la red LAN.



Figura 3.15: Dirección de la cámara IP

A continuación doble click, en la cámara de la lista, y se procede a abrir su navegador por defecto. La primera vez aparece la página de login, se requiere descargar e instalar un *plugin* compatible con los navegadores de Internet más conocidos.

¹⁴http://foscam.us/media/mconnect_uploadfiles/f/i/fi9821w%20v2%20quick%20installation%20guide.pdf



Figura 3.16: Plugin cámara IP

- Después de instalar el plugin, refrescar el navegador y efectuar un click en login, con los valores por defecto.

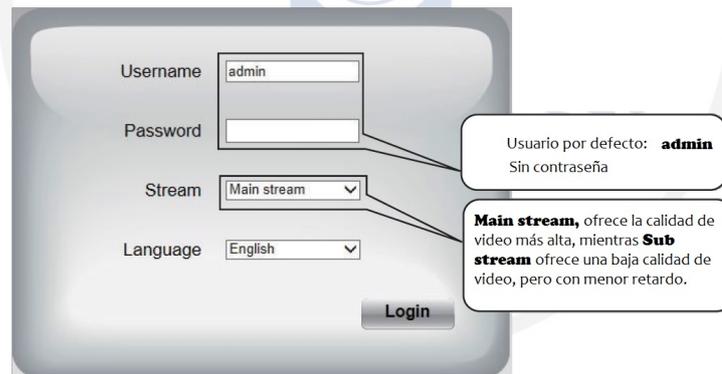


Figura 3.17: Login cámara IP

- Al iniciar sesión por primera vez, se requiere que se modifique el usuario y contraseña. Ingresar el **nuevo usuario** y la **nueva contraseña** y click en modificar.

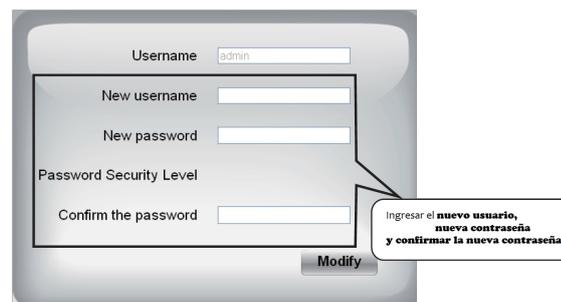


Figura 3.18: Ingreso de usuario y password para la cámara IP

6. Una vez iniciada la sesión se puede observar el video en vivo a través de la interfaz.

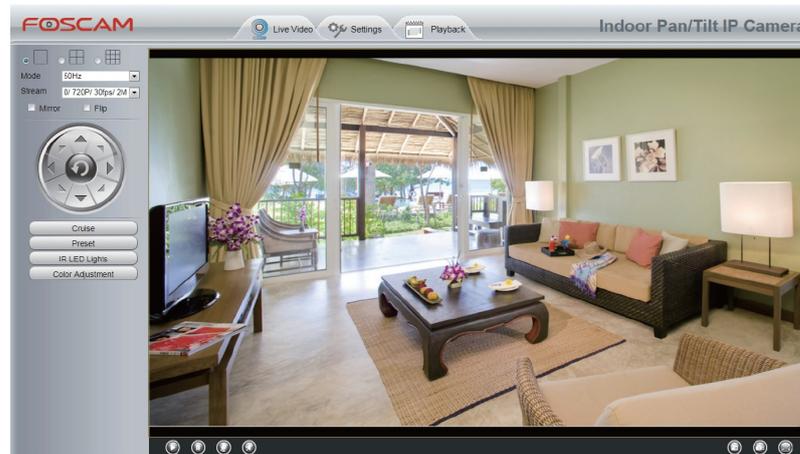


Figura 3.19: Interfaz cámara IP

Conexión WiFi

Una vez realizadas las configuraciones iniciales, la conexión WiFi es opcional debido a que en la mayoría de ocasiones se utiliza la conexión cableada, por su velocidad de transferencia de datos. Para la conexión WiFi de la cámara IP existen dos métodos:

- Conexión vía WPS (Wi-Fi Protected Set-up)

Este método de configuración es el más sencillo, ya que no se requiere ingresar a la interfaz de la Cámara IP, para ello se utiliza el botón WPS ubicado en la parte posterior de la cámara, por 2 segundos.

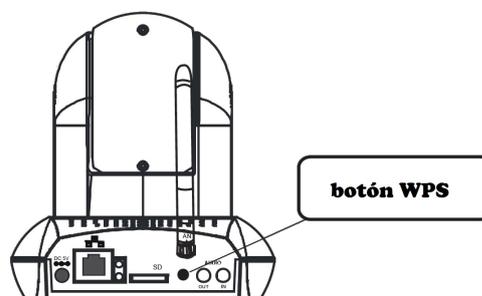


Figura 3.20: Botón WPS cámara IP. Fuente: (Foscam, 2015).

A continuación debe presionarse el botón WPS del router, generalmente ubicado en la parte posterior, por 60 segundos.

La cámara IP creará de manera automática la conexión al router, al presionar el botón WPS durante 60 segundos.

- Conexión desde la interfaz FOSCAM

Si el router del usuario no soporta la función WPS, se debe seguir los siguientes pasos:

En la interfaz de la cámara seleccionar la pestaña “Settings” a continuación “Network-Wireless Settings”, y presionar el boton “Scan”



Figura 3.21: Búsqueda de redes WiFi cámara IP

Seleccionar el nombre de la red WiFi y digitar la contraseña y presionar en el boton *Save* y desconectar los cable de red conectado a la cámara IP.

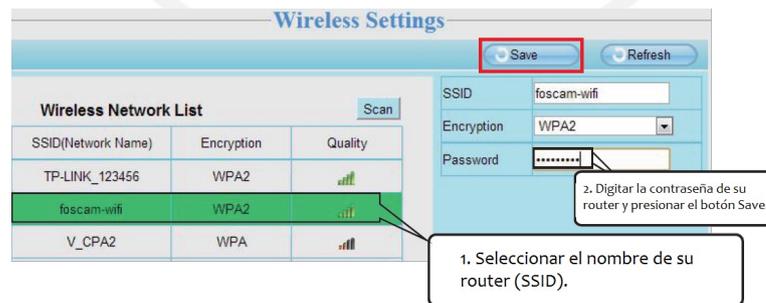


Figura 3.22: Configuración de red WiFi cámara IP

Conexión con interfaces externas

- Cámara IP y OpenCV

Para poder visualizar la cámara IP y que la librería **OpenCV** pueda interactuar con el contenido multimedia, existe un protocolo de transmisión en tiempo real, llamado *RTSP* por sus siglas en inglés: Real Time Streaming Protocol. RTSP establece y controla uno o varios flujos sincronizados

de datos, ya sean de audio o de video. El RTSP actúa como un mando a distancia mediante la red para servidores multimedia. RTPS es compatible con la mayoría de cámaras IP y proporciona múltiples ventajas como:

- Permite recuperar contenido multimedia del servidor, ya que el cliente puede solicitar la transmisión de datos de audio y video.
- Invitación de un servidor multimedia a una conferencia: Un servidor puede ser invitado a unirse a una conferencia existente en lugar de reproducir la presentación o grabar todo o una parte del contenido. Este modo es útil para aplicaciones de enseñanza distribuida donde diferentes partes de la conferencia van tomando parte en la discusión.
- Adición multimedia a una presentación existente: Es utilizado particularmente para presentaciones en vivo, útil si el servidor puede avisar al cliente sobre los nuevos contenidos disponibles.

Para empezar con la transmisión y captura de video se debe especificar la cadena de conexión con los siguientes parámetros:

```
rtsp://[USUARIO]:[PASSWORD]@[DIRECCIONIP]:[PUERTO]/videoMain
```

Una vez establecidos los parámetros se pueden realizar pruebas mediante el software VLC¹⁵, para verificar que se está efectuando la transmisión de video.

■ Cámara IP y HTML

En la sección 3.2.4, se definió una interfaz WEB con HTML5¹⁶ para la administración y video vigilancia en tiempo real a través de la cámara IP, para ello se hace el uso de la tecnología CGI (Common Gateway Interface), Interfaz de Entrada Común. CGI especifica un estándar para transferir datos entre un cliente y un programa y actúa de la siguiente forma:

- En primera instancia, el servidor recibe una petición del cliente mediante una URL que contiene el CGI, y comprueba si se trata de una invocación de un CGI.
- Posteriormente, el servidor prepara el entorno para ejecutar la aplicación. Esta información procede mayoritariamente del cliente.
- Seguidamente, el servidor ejecuta la aplicación y envía la información producida al cliente.

De este modo utilizando la tecnología CGI, y la secuencia de comandos que FOSCAM proporciona para el entorno de desarrollo, se debe realizar peticiones sucesivas a la cámara IP, para que retorne una secuencia de capturas cada cierto tiempo, simulando transmitir un video en tiempo real. Para ello se hace el uso del siguiente comando.

¹⁵<http://www.videolan.org/vlc/>

¹⁶HTML, constituye un estándar que hace referencia al lenguaje de marcado para la elaboración de páginas web, y que define una estructura básica y un código para la definición de contenido de una página web, como: texto, imágenes, videos, entre otros.

```
https://[DIRECCIONIP]:[PUERTO]/cgi-bin/CGIProxy.fcgi?cmd=
snapPicture2&usr=[USUARIO]&pwd=[PASSWORD]
```

3.4.2. Servidor de procesamiento de imágenes

El servidor es un computador que está dedicado al procesamiento de las imágenes capturadas por la cámara IP. Para el sistema propuesto el servidor puede ser un computador personal que posea los requerimientos mínimos especificados a continuación:

PC de Escritorio o Laptop	
Procesador	Intel Core i3
Memoria RAM	4GB
Almacenamiento	250GB
Tarjeta de Red	802.11 b,g,n ó 802.11 b,g
Sistema Operativo	Windows 7, Windows 8 Windows 8.1 Todas las versiones 32 y 64 bits.

Tabla 3.1: Especificaciones del servidor de imágenes

3.4.3. Arduino UNO

Arduino UNO es una placa electrónica con un microcontrolador del fabricante Atmel ¹⁷ que incluye reguladores de tensión, un puerto USB conectado a un módulo USB-serie que permite programar el microcontrolador desde cualquier PC y realizar pruebas de comunicación con el propio chip.



Figura 3.23: Arduino UNO. Fuente: (Arduino, 2015).

¹⁷<http://www.atmel.com/Images/doc8161.pdf>

Microcontrolador	Atmega328
Voltaje de Operación	5 V
Voltaje de Entrada (Recomendado)	7-12 V
Voltaje de Entrada (Limite)	6-20 V
Pines de Entrada-Salida digital	14 (6 pines para PWM)
Pines de Entrada Analógica	6
Corriente continua por pin I/O	40mA
Corriente continua en el pin 3.3v	50mA
Memoria FLASH	32KB (0.5 bootloader)
SRAM	2KB
EEPROM	1KB
Frecuencia de Reloj	16Mhz

Tabla 3.2: Especificaciones técnicas Arduino UNO

Arduino UNO dispone de 14 pines, que pueden configurarse como entrada o salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5v. También dispone de entradas y salidas analógicas con las que podemos obtener datos de sensores en forma de variaciones continuas de un voltaje. Para el caso de las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM (Pulse-width Modulation). Esta tarjeta puede alimentarse mediante el cable USB o mediante una fuente externa como puede ser un adaptador AC-DC o una batería de 9v. Más especificaciones se muestran en la Tabla 3.2.

3.4.4. Dispositivos de red

Es conveniente mencionar los dispositivos que conforman la red del Sistema de Reconocimiento de Placas Vehiculares, para tener una idea más clara de que como interactúan entre sí para la transmisión de datos.



Figura 3.24: Diagrama de red

En esta sección se encuentran involucrados todos los dispositivos que se conectan a un segmento de red y sus funciones pueden ser varias:

- Compartir datos e información.
- Compartir otros dispositivos de hardware y software
- Centralizar la administración y soporte.

Los dispositivos se dividen en dos grupos, el primero está enfocado al usuario final como una PC, impresora, scanner, etc. El segundo grupo está enfocado a la conexión entre los dispositivos de usuario final.

Router

Este dispositivo proporciona la comunicación a nivel de red. La función principal es la de enviar o encaminar paquetes de una red a otra, es decir interconectar subredes.



Figura 3.25: Router TP-LINK TD-8961N. Fuente: (TPLINK, 2015).

Para el presente proyecto, se utiliza un router **TPLINK**, modelo *TD-W8961ND*¹⁸, debido a sus buenas prestaciones para streaming de video en alta definición - HD. Este modelo es multifunción ya que trabaja como un módem DSL de alta velocidad para la conexión a Internet, un router de 4 puertos de 10/100 Mbps y a la vez proporciona el acceso inalámbrico entre los dispositivos.

Switch

Un *switch* interconecta dos más segmentos de red, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red. Estos dispositivos se utilizan cuando se desea conectar múltiples tramos de una red, fusionándolos como una sola red. Para el sistema propuesto, se utiliza un switch **D-LINK**, modelo *DES-1008A*.



Figura 3.26: Switch D-LINK DES-1008A Fuente: (DLINK, 2015).

3.5. Especificaciones software del sistema

3.5.1. Lenguaje de programación

En un principio para el desarrollo del sistema se propuso el uso del lenguaje **C++**, ya que es compatible con la librería **OpenCV** para el tratamiento de imágenes, como se muestra en la sección: 3.5.3. **C++** es un lenguaje orientado a objetos y es una extensión del Lenguaje C.

3.5.2. Interfaz gráfica Qt

Qt¹⁹ es una biblioteca multiplataforma, utilizada principalmente para desarrollar aplicaciones con interfaz gráfica de usuario. También se utiliza para trabajar

¹⁸<http://www.tp-link.ec/products/details/?model=TD-W8961ND>

¹⁹www.qt.io

con herramientas como línea de comandos y consolas para servidores.

Se ha optado el uso de Qt, ya que es un software de distribución gratuita y código abierto, además existen versiones para sistemas operativos como: Windows, Linux, Unix, Mac OS y es 100 % compatible con la librería *OpenCV* y el lenguaje *C++*.



Figura 3.27: Logo de software Qt. Fuente: (Qt, 2015).

3.5.3. Librería de visión por computador

OpenCV

OpenCV²⁰ u *Open Computer Vision* es una librería de visión artificial de código abierto, desarrollada en principios por Intel. Desde el lanzamiento de su primera versión se han desarrollado una infinidad de aplicaciones dedicadas a diferentes áreas como es la seguridad con aplicaciones para la detección de movimientos. También se han desarrollado aplicaciones en el área de control de procesos para el reconocimiento de objetos.

OpenCV es una librería multiplataforma ya que opera en Windows, Mac OS, GNU/Linux y contiene más de 600 funciones dedicadas al procesamiento de imágenes, reconocimiento de objetos, calibración de cámaras, visión robótica. La programación se la realiza en código C, C++ y python, por lo que se aprovecha al máximo las capacidades que ofrecen los procesadores multinúcleo.



Figura 3.28: Logo de librería OpenCV. Fuente: (Opencv, 2015).

²⁰www.opencv.org

3.5.4. Base de datos

La base de datos es una colección de información que contiene datos relativos pertenecientes a un mismo contexto, compartiendo entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto. Realizando una analogía una biblioteca puede considerarse una base de datos, compuesta por libros y documentos impresos indexados para su consulta. Existen sistemas dedicados a la gestión de base de datos (DBMS), que permiten almacenar datos de manera rápida y estructurada. Entre estos sistemas DBMS, los más conocidos son:

- Oracle
- DB2 de IBM
- Sql Server de Microsoft
- MySQL (Oracle)
- Microsoft Access

Para el sistema propuesto, se ha optado el uso de **MySQL**²¹ ya que es un gestor de código abierto y no tiene costos de licenciamiento, además que este gestor de base de datos es el más utilizado a nivel mundial en aplicaciones web para comercio electrónico y aplicaciones dinámicas. Adicionalmente MySQL es compatible con el lenguaje PHP para la implementación de páginas WEB.



Figura 3.29: Logo MySQL. Fuente: (Mysql, 2015).

MySQL tiene muchas ventajas por lo que es utilizado incluso con fines comerciales y muy versátil para trabajar en ambientes donde se requiera un término medio de transaccionabilidad. De igual forma existe documentación, artículos y foros en donde se puede dar soporte a los problemas o consultas que se tenga.

3.5.5. Ambiente para desarrollo WEB

Para el sistema de administración de usuarios basado en WEB propuesto inicialmente, se ha optado por el uso de **WampServer**²² - WINDOWS APACHE

²¹<https://www.mysql.com/products/community/>

²²<http://www.wampserver.com/en/>

MYSQL PHP, ya que es ideal para creación y administración de aplicaciones con el servidor HTTP²³ de Apache 2, PHP y base de datos MySQL.



Figura 3.30: Logo de WampServer

3.6. Implementación del módulo del detección de placas

Para empezar con la detección de placas vehiculares, es necesario definir la ruta de video o de la cámara IP como se detalla en la sección 3.4.1, desde la cual se obtendrán las imágenes para ser procesadas.

```
char* rutaCamaraLocal=  
``rtsp://admin:ua048989@192.168.1.150:88/videoMain``;
```

Una vez definida la ruta, se utiliza la función `cv::VideoCapture cap(rutaVideo);` de OpenCV para iniciar la captura. Los videos son procesados como una secuencia de imágenes, por lo tanto se debe establecer un temporizador a manera de bucle para obtener cada uno de los cuadros o *frames*. Cada *frame* es asignado a una variable previamente declarada de tipo *Mat* que es un tipo de dato que maneja OpenCV para almacenar un cuadro de imagen mediante la sentencia:

```
cap >> imagen;
```

Para visualizar la imagen capturada se utiliza la función `imshow`, estableciendo como parámetros el nombre de la ventana que se visualizará y la variable *Mat*, que contiene la matriz de la imagen.

²³HTTP: acrónimo de Hypertext Transfer protocol



Figura 3.31: Imagen capturada con OpenCV

3.6.1. Pre-procesamiento de imágenes

A partir de la imagen original, el pre-procesamiento comienza determinando la región de interés ROI, en donde posiblemente se ubicará la zona de la placa vehicular. Para ello se define el área a recortar de tipo Rect y se crea una nueva variable `imgRecortada` pasando la imagen original y el área como parámetros:

```
cv::Rect myROI (width/4, height/2, width/2, height/2);  
cv::Mat imgRecortada (imagen, myROI);
```

Al trabajar con una ROI se optimiza el tiempo de procesamiento ya que se evita el análisis de regiones que no son importantes en la imagen real para un objetivo específico a cumplir.



Figura 3.32: Pre-procesamiento: región de interes

En el siguiente proceso, se convierte la imagen recortada en escala de grises, puesto que existe una reducción en la complejidad computacional de los algoritmos cuando se trabaja en un solo canal, en lugar de tres (modelo en color).



Figura 3.33: Pre-procesamiento: escala de grises

A continuación se aplica una técnica para suavizar o desenfocar la imagen, mediante la función `blur`, que se basa en los principios de un filtro de suavizado tras la convolución de la imagen con un operador kernel específico (Rosado-Muñoz, Bataller-Mompeán, Soria-Olivas, Scarante, y Martínez, 2011). Esta es una operación sencilla y de uso frecuente en el procesamiento de imágenes con la finalidad de reducir el ruido. La última técnica en el pre-procesamiento se utiliza para nivelar los niveles de grises en la imagen se utiliza la función `cv::equalizeHist`. El objetivo de utilizar la ecualización del histograma (Rosado-Muñoz y cols., 2011) en algunos casos, es la de distinguir de mejor manera los bordes y objetos pequeños.



Figura 3.34: Pre-procesamiento: ecualización del histograma

3.6.2. Procesamiento de Imágenes

En el procesamiento de la imagen se establece un mecanismo de votación, que permita elegir el mejor de los métodos implementados para la **detección de bordes** sobre la imagen pre-procesada, en este caso son:

- Adaptive Threshold: Umbral adaptativo
- Otsu Threshold: Umbral de Otsu
- Operaciones Morfológicas

El método *Adaptive Threshold* (Rosado-Muñoz y cols., 2011), es utilizado con el fin de separar objetos de interés en una imagen, mediante el cálculo de valor umbral para cada píxel. De esta manera se puede determinar qué píxeles conforman los objetos que se desea reconocer y cuáles son los contornos de estos objetos.



Figura 3.35: Procesamiento: adaptive threshold

El método *Otsu Threshold* (Rosado-Muñoz y cols., 2011), es utilizado para determinar automáticamente el valor umbral óptimo para cada píxel. Este algoritmo asume que la imagen se divide en dos conjuntos de píxeles, el primer conjunto determina el fondo y el segundo los objetos que resaltan.



Figura 3.36: Procesamiento: Otsu threshold

El tercer método, *Operaciones morfológicas* (Agam, 2006), ha sido implementado con el objetivo de detectar las placas de los vehículos que **no tienen bien definida su área** debido a malformaciones, o por las condiciones en las que se

captura la imagen no se puede determinar los contornos de su región. Este método combina una técnica de detección de bordes verticales denominada Operador de Sobel (Figura 3.37) como se detalla en la sección 2.3.1 y operaciones morfológicas (Figura 3.38) para identificar y reconstruir la forma de una placa a partir de sus caracteres distorsionados.



Figura 3.37: Procesamiento: operador Sobel para detección de bordes verticales.

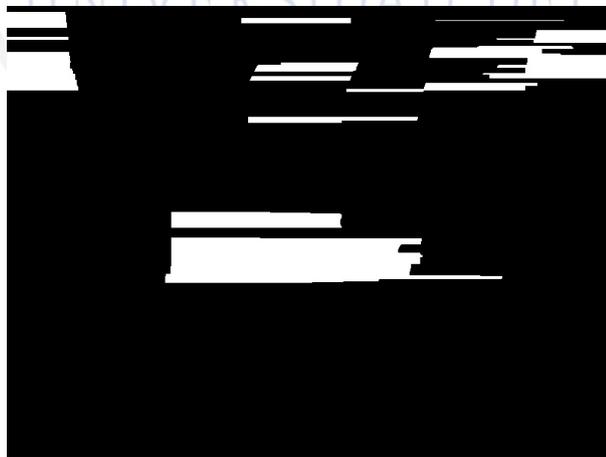


Figura 3.38: Procesamiento: operaciones morfológicas.

El paso siguiente que realiza el sistema es buscar todos los contornos con la función de OpenCV `cv::findContours` (Bradski y Kaehler, 2011), a partir de la imagen obtenida en la fase de detección de bordes. En el filtrado de contornos se buscan obtener el/las áreas con las posibles regiones de la placa, para esto se establecen varios parámetros:

- Rango del tamaño del contorno $3500 < \text{areaContorno} < 30000$

- Número de lados del contorno $3 < \text{contorno.size}() < 6$, por lo general el area de la placa = 4 lados
- Proporción entre el ancho y alto de una placa en escala real $2 < \text{ancho} / \text{alto} < 3$, para el formato de las placas nuevas que establece la Comisión de Tránsito del Ecuador ancho=404 cm, alto=154 cm. Por lo tanto la proporción es 2.62

Una vez filtrados los contornos, en esta etapa entra en juego el mecanismo de votación, ya que si el **primer método de detección de bordes** no detectó ningún contorno con los parámetros establecidos, se continúa el proceso con el **segundo método** y de ser necesario con el **tercero** de manera secuencial.

Una vez detectado el contorno se dibuja el área de la placa vehicular en la imagen original con la función `cv::drawContours` y enviarla al módulo para reconocimiento de sus caracteres.



Figura 3.39: Procesamiento: placa detectada.

3.7. Implementación del módulo de reconocimiento

Una vez localizada una placa vehicular como una subimagen de la señal de video adquirida por la cámara, como se observa en la Figura 3.40, se realiza el reconocimiento de sus caracteres utilizando técnicas de clasificación de patrones. Para esto es necesario un pre procesamiento adicional, puesto que las imágenes son recibidas a color y contienen detalles que no son de interés para este objetivo, como por ejemplo la presencia de la palabra “Ecuador”.

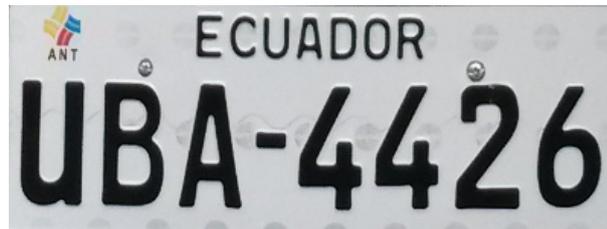


Figura 3.40: Imagen proveniente de la detección de contornos.

Para ello, en el sistema se implementó un método denominado `quitarBordesPlaca`, que recibe una imagen a color y realiza una búsqueda de contornos de potenciales caracteres, como en la Figura 3.41, y que recorta la imagen manteniendo solo los detalles de interés. La búsqueda de contornos se realiza con la función `findContours` (Bradski y Kaehler, 2011) de *OpenCV*. No todo contorno encontrado es un potencial carácter, se debe tener un área mínima para ser considerado como tal. El recorte de la placa se realiza tomando las coordenadas del primer y último contorno, de izquierda a derecha. Se debe realizar una mínima ampliación de la zona de recorte, ya que en caso de no hacerlo se podría realizar recortes muy ajustados que dificultarían el reconocimiento de los caracteres posteriormente.

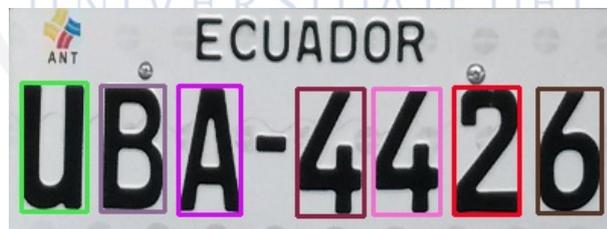


Figura 3.41: Contornos correspondientes a los caracteres de identificación

El resultado de usar este método es una imagen binarizada, contenedora de los caracteres de identificación de la placa vehicular, como se ve en la Figura 3.42, y sin detalles innecesarios. Si se mira cuidadosamente la imagen se notarán pequeñas manchas sobre la letra “B” y el número “2”, las cuales son producidas por 2 tornillos sujetadores. Detalles como estos pueden causar fallas en el reconocimiento, como se explica en el siguiente capítulo.

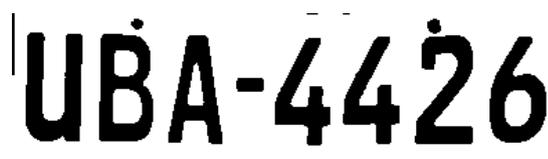


Figura 3.42: Placa binarizada, sin bordes y sin detalles innecesarios para el reconocimiento.

La imagen es guardada como un archivo y su ruta es pasada como parámetro al método `establecerRuta` de la clase de reconocimiento `threadTesseract` que usa el API de *Tesseract* al ejecutar el método `run`. Para el reconocimiento en base a redes neuronales y algoritmo de vecinos más cercanos es necesario usar el método `segmentarCaracteres`, el cual toma una imagen binarizada como entrada y busca contornos que potencialmente son caracteres usando el método de *OpenCV* `findContours`.

Indice	Imagen	Imagen Escalada
0	U	U
1	B	B
2	A	A
3	4	4
4	4	4
5	2	2
6	6	6

Tabla 3.3: Arreglo de imágenes, correspondientes a caracteres, producto de la segmentación.

La Tabla 3.3 muestra un vector resultante de la segmentación de caracteres de la imagen 3.42. El método devuelve las imágenes reducidas a una escala de 16x16 píxeles, se nota en el vector la ausencia del guión separador de números y demás detalles no importantes ya que estos fueron ignorados en la búsqueda de contornos por no cumplir el área mínima. El cálculo de las áreas de los contornos se realiza con la función `contourArea`.

El vector es pasado como parámetro a la función `setChars` de las clases de reconocimiento `threadANN` y `threadKNN`. Para realizar el reconocimiento en base a redes neuronales y algoritmos de vecinos más cercanos, se debe ejecutar el método `run` de ambas clases.

Tras finalizarse el reconocimiento de caracteres se obtienen las cadenas de caracteres resultantes usando los métodos `obtenerPlaca` de cada clase. Las 3 respuestas del reconocimiento son usadas por el método `determinarPlaca`, que implementa una votación caracter a caracter para determinar una única placa vehicular. Esta placa vehicular pasa luego a ser buscada en una base de datos para verificar la identidad de un usuario.

Los procesos de detección y reconocimiento, al igual que otros detalles como las razones por la cual una placa vehicular no puede ser correctamente detectada y reconocida, se indican en el capítulo siguiente.

3.8. Descripción del Prototipo

El prototipo construido permite el ingreso y/o salida de un vehículo tras haberse identificado por el sistema de reconocimiento de placas vehiculares propuesto. Es claro que solo los usuarios autenticados serán quienes activen la funcionalidad del prototipo. En un aplicación real el prototipo sería replicado en la salida del parqueadero.

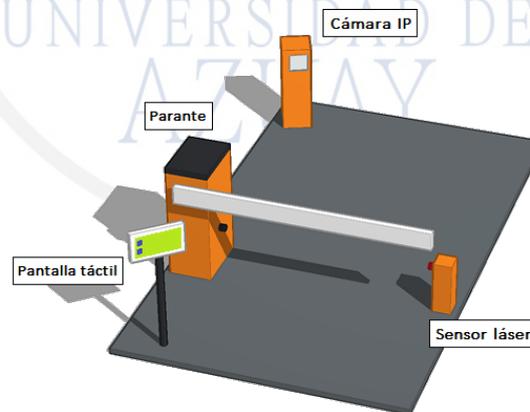


Figura 3.43: Prototipo de acceso a un parqueadero.

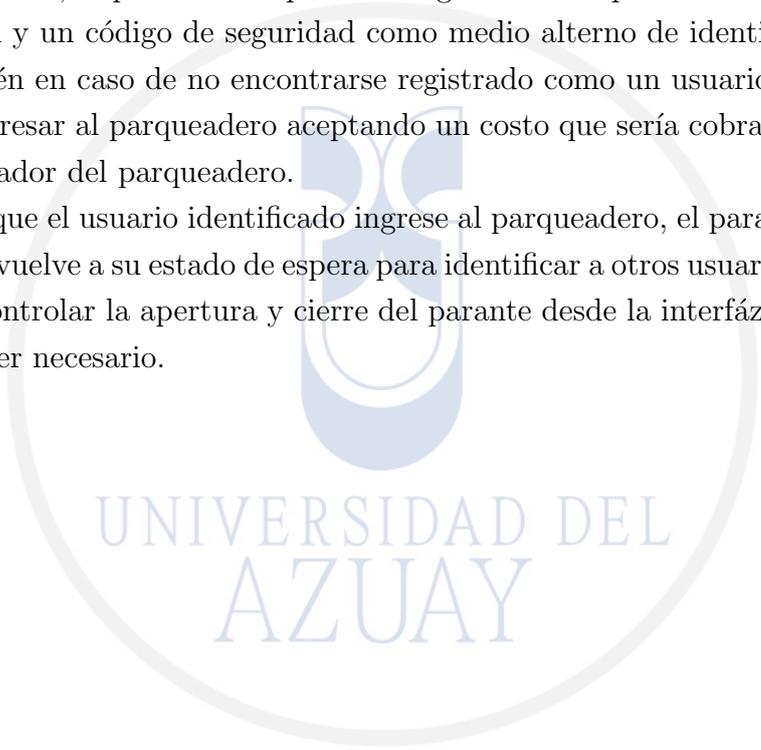
El prototipo usa los componentes especificados en la sección 3.4 para su funcionamiento, los cuales se encuentran dentro de la estructura del parante. Una vez que el vehículo se coloca cerca del área de detección determinado por la *ROI* en las imágenes capturadas por la cámara IP se inicia el proceso de detección y reconocimiento de la placa vehicular. El reconocimiento es activado por un operador *Background Subtractor*, que sirve como detector de objetos en movimiento.

Una vez reconocida la placa vehicular, el sistema propuesto se comunica con el

módulo Arduino UNO a través de una comunicación serial usando el puerto *USB* del computador. Entonces, se abrirá el parante que es controlado con un servo motor y se escuchará un mensaje de bienvenida, “Bienvenido al parqueadero de la Universidad del Azuay”.

En caso de producirse un error de autenticación al no reconocer la placa vehicular del usuario, se presenta la opción de digitar en una pantalla táctil la cédula de identidad y un código de seguridad como medio alternativo de identificación. Así como también en caso de no encontrarse registrado como un usuario del sistema se puede ingresar al parqueadero aceptando un costo que sería cobrado luego por el administrador del parqueadero.

Una vez que el usuario identificado ingrese al parqueadero, el parante se cierra y el sistema vuelve a su estado de espera para identificar a otros usuarios. También es posible controlar la apertura y cierre del parante desde la interfáz del sistema, en caso de ser necesario.



UNIVERSIDAD DEL
AZUAY

4.1. Introducción

En este capítulo se detallan los resultados y el procedimiento usado para evaluar el desempeño del sistema propuesto. Es importante el mencionar que las pruebas se han realizado con placas vehiculares encontradas en un ambiente controlado, similar al que se esperaría para una implementación real del sistema.

4.2. Contexto de evaluación del sistema

Cabe mencionar que las tareas involucradas en un sistema LPR son de complejidad considerable, lo que se hace evidente en ambientes reales donde las placas vehiculares no siempre están en condiciones favorables para su localización y reconocimiento (Duan, Du, Phuoc, y Hoang, 2005). A continuación se detallan algunas razones para que una placa pueda no ser localizada y/o reconocida correctamente:

- Inexistencia de la placa vehicular.
- Placa vehicular fuera del rango de visión de la cámara.
- Placa dañada o alterada.
 - Porciones faltantes, placas rotas o despintadas.
 - Placas dobladas o modificadas de manera que la forma de los caracteres no corresponde a las regulaciones locales.
 - Placas cubiertas con micas o materiales que no permitan su lectura.

- Placa ubicada en un lugar no establecido para ello. Por ejemplo, ubicadas en el habitáculo o en una ventana.
- Placas obstruidas:
 - Por objetos que no permitan tener una vista clara de la placa, como un motor eléctrico de remolque (*winch*) o un parachoques.
 - Por marcos o bordes que no permitan distinguir los caracteres que contienen.
 - Por sujetadores, tornillos o pernos que interfieran en los detalles de la placa.
 - Por agentes temporales como lodo, polvo o similares.

4.2.1. Otros factores influyentes

El ángulo y distancia de captura, y las condiciones del ambiente podrían también definir si una placa vehicular es localizada y reconocida correctamente o no. Por eso, en sistemas e implementaciones reales, estos factores son controlados al establecer condiciones bajo las que el sistema debe trabajar para conseguir resultados de manera óptima (Duan y cols., 2005). El estado de las placas vehiculares juega un papel importante en su reconocimiento. Una placa vehicular en buen estado posee sus caracteres, tamaño, colores y fuente, bien definidos y visibles, es decir no están modificados o alterados. Además las placas en buen estado deben ser legales, es decir ser emitidas por la Agencia Nacional de Tránsito del Ecuador, como se puede observar en la Figura 4.1, y cumplir con los reglamentos de forma, dimensiones, color y ubicación establecidos por dicha entidad gubernamental o de cualquier país en particular.



Figura 4.1: Ejemplos de placas vehiculares en buen estado.

Existen también placas que a pesar de haber sido emitidas por la Agencia Nacional de Tránsito, se encuentran en mal estado, deterioradas o alteradas de manera que se dificulta su correcto análisis. Como se puede observar en la Figura

4.2, existen vehículos con placas alteradas, las mismas que lucen así por falta de cuidado o por desgaste natural debido al medio ambiente.



Figura 4.2: Ejemplos de placas vehiculares en mal estado.

Hay otros formatos de placas que no son emitidos por la ANT, son elaboradas por terceros y/o son vendidas incluso en portales web de anuncios clasificados como lo son Nexolocal¹ o MercadoLibre². Estas placas no se asemejan a las emitidas por la ANT, en su totalidad o parcialmente, y no se tiene conocimiento si sus productores están autorizados para la venta de ellas. Algunos ejemplos de placas denominadas “personalizadas” se aprecian en la Figura 4.3, como se puede observar estas tienen variedad de formas, fuentes y colores.



Figura 4.3: Ejemplos de placas personalizadas.

Como se explica en el capítulo 3, el sistema propuesto puede detectar y reconocer a las placas vehiculares que son emitidas por la ANT. Sin embargo, el sistema tratará de reconocer todas aquellas que se le presenten.

Al realizar la recolección de los datos, los autores notaron la existencia de 2 tipos de fuente diferentes usadas en las placas emitidas por la ANT. En la Figura 4.4 se pueden apreciar ejemplos un mismo caracter en diferentes tipos de fuente.

¹<http://www.nexolocal.com.ec/>

²<http://www.mercadolibre.com.ec/>



Figura 4.4: Ejemplos de caracteres encontrados en placas vehiculares de Ecuador y sus distintas fuentes. (1) letra “A”, (2) letra “D”, (3) número “cero”, (4) número “seis” y (5) número “siete”.

4.3. Modelo de evaluación del sistema

4.3.1. Recolección de Datos

Son necesarias placas vehiculares reales y usadas en Ecuador para poder medir la precisión del sistema en cuanto a localización y reconocimiento. Para ello se ha recolectado 214 placas vehiculares en los parqueaderos de estudiantes y de profesores de la Universidad del Azuay, las mismas han sido recolectadas en diferentes horas del día , entre las 08H00 y 17H00, bajo condiciones climáticas distintas, cielos despejados y nublados, en días distintos de la semana.

A continuación se detalla el proceso de captura de imágenes, las cuales han sido tomadas con la cámara de un teléfono celular inteligente con una resolución de 2448x3264 píxeles, con el objetivo de contar con una base de imágenes que pueda ser utilizada para pruebas, entrenamiento y verificaciones.

Procedimiento de captura o recolección de datos

La recolección de imágenes se realizó manualmente siguiendo el siguiente procedimiento:

1. Seleccionar un vehículo y enfocar su parte frontal o posterior.
2. Ubicarse a una distancia de 100cm a 200cm, aproximadamente, y enfocar la placa vehicular en un ángulo descendente no mayor a 45 grados de manera que se tenga un enfoque tal como se puede apreciar en la Figura 4.6.

$$100cm < Distancia \leq 200cm \quad (4.1)$$

$$0^\circ < \text{Ángulo} \leq 45^\circ \quad (4.2)$$

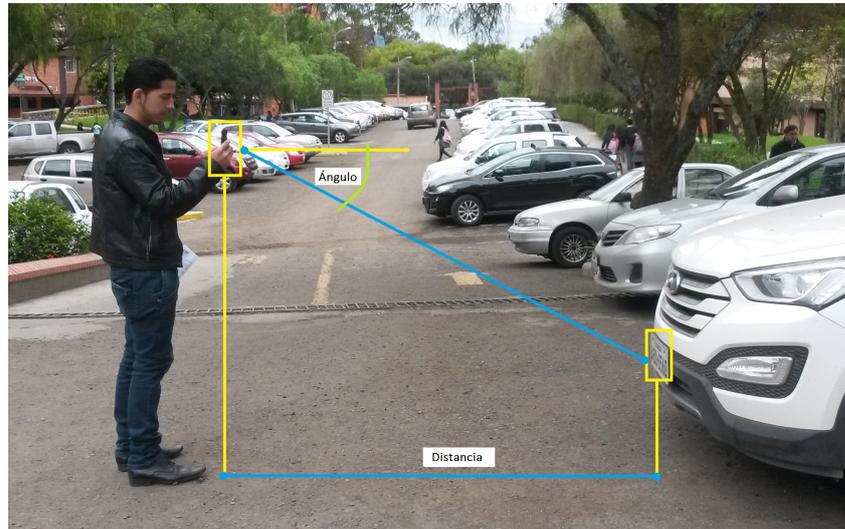


Figura 4.5: Distancia y ángulo a los que se han capturado las imágenes de prueba.

3. Enfocar la imagen con un ángulo no mayor a 30° a la derecha o izquierda, de manera que la placa vehicular quede dentro de la “Región de Interés” y sea visible en su totalidad para la cámara.

$$0^\circ < \text{Ángulo}1 \leq 30^\circ \quad (4.3)$$

$$0^\circ < \text{Ángulo}2 \leq 30^\circ \quad (4.4)$$

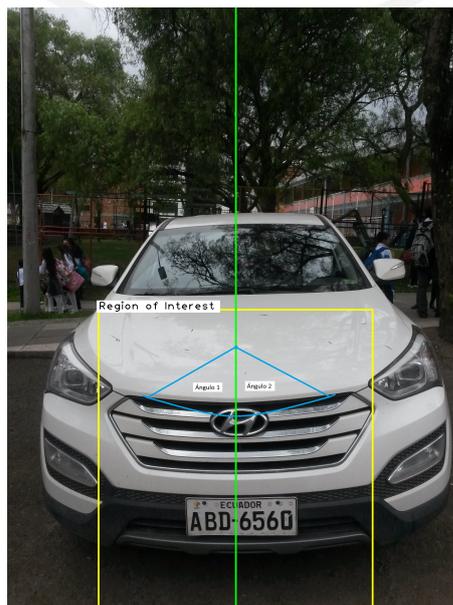


Figura 4.6: Enfoque de la placa vehicular dentro de la “Región de Interés”.

4.3.2. Procesamiento de datos

Los datos obtenidos son procesados de manera secuencial, imagen por imagen, para obtener los resultados de detección y reconocimiento que serán analizados en la siguiente sección. El proceso se detalla en los pasos:

1. Cargar una imagen y obtener la región de interés donde se encuentra ubicada la placa vehicular.
2. Obtener las potenciales placas vehiculares usando los 3 métodos implementados en el sistema propuesto, mismos que se detallan en la sección 2.3. Los 3 métodos de detección y sus resultados son presentados en este capítulo con fines analíticos.

a) Detección de contornos en base a *Otsu Threshold*, ver Figura 4.7.



Figura 4.7: A la izquierda, la imagen binarizada. A la derecha, el contorno identificado potencialmente como placa vehicular.

b) Detección de contornos en base a *Adaptive Mean Threshold*, ver Figura 4.8.



Figura 4.8: A la izquierda, la imagen binarizada. A la derecha, el contorno identificado potencialmente como placa vehicular.

- c) Detección de contornos en base a *Operaciones Morfológicas*, ver Figura 4.9.



Figura 4.9: A la izquierda, la imagen binarizada. A la derecha, el contorno identificado potencialmente como placa vehicular.

3. Reconocimiento de caracteres en la placa vehicular detectada: utilizando los tres métodos OCR implementados en el sistema propuesto. Los mismos que son usados para definir candidatos que participan en un proceso de votación para definir una única placa vehicular caracter por caracter. Detalles del reconocimiento de caracteres se pueden encontrar en la sección 2.4.

Tomando como ejemplo la placa vehicular mostrada en la Figura 4.7, se obtienen los siguientes resultados de reconocimiento:

- Tesseract: *ABD6560*
- ANN: *ABD6560*
- KNN: *AUU8580*
- Resultado en votación: *ABD6560*

El proceso de votación se realiza tomando las respuestas de los tres candidatos, como se puede observar en la Tabla 4.1. En caso de existir tres candidatos distintos, es decir tres caracteres distintos, se da prioridad al método de reconocimiento en base a red neuronal (ANN) puesto que se ha visto, al realizar las pruebas y por experiencia, que es el método con menos errores.

Método	C1	C2	C3	C4	C5	C6	C7
Tesseract	<i>A</i>	<i>B</i>	<i>D</i>	<i>6</i>	<i>5</i>	<i>6</i>	<i>0</i>
ANN	<i>A</i>	<i>B</i>	<i>D</i>	<i>6</i>	<i>5</i>	<i>6</i>	<i>0</i>
KNN	<i>A</i>	<i>U</i>	<i>U</i>	<i>6</i>	<i>5</i>	<i>6</i>	<i>0</i>
Votación	A	B	D	6	5	6	0

Tabla 4.1: Votación caracter por caracter para determinar una única placa vehicular.

4.4. Evaluación Cuantitativa

La evaluación cuantitativa del sistema propuesto se basa en los siguientes puntos:

- Número de placas vehiculares detectadas, definido como el nivel de **Detección** y expresado como:

$$Detección = \frac{N. \text{ Placas Detectadas}}{N. \text{ Total de Placas Analizadas}} \quad (4.5)$$

- Número de placas vehiculares reconocidas, definido como nivel de **Reconocimiento** y expresado como:

$$Reconocimiento = \frac{N. \text{ Placas Exitosamente Reconocidas}}{N. \text{ Placas Detectadas}} \quad (4.6)$$

4.4.1. Detección de placas vehiculares

A continuación, la Tabla 4.2 y las Figuras 4.11, 4.12 y 4.13 muestran el desempeño de cada método de detección de placas vehiculares utilizado, tomando en cuenta que se ha logrado detectar exitosamente 211 de 214 disponibles sometidas a prueba.

	Placas Detectadas	Placas No Detectadas	Porcentaje Aciertos
Otsu Threshold	111	103	52%
Adaptive Mean Threshold	153	61	71%
Operaciones Morfológicas	211	3	99%

Tabla 4.2: Número de placas encontradas y no encontradas para cada método.

El número de placas vehiculares detectadas usando Otsu Threshold es superior al 50%. Este método se basa en la búsqueda de contornos rectangulares, similares a placas, como se ve en la Figura 4.10.



Figura 4.10: Contorno de placa vehicular.

Pero existen vehículos en los que, para el sistema, es difícil identificar un borde rectangular correspondiente a la placa y es aquí donde se producen los fallos de no detección. Dichos fallos ocurren por factores como, color del vehículo, iluminación del ambiente, estado de la placa o enfoque de la cámara.

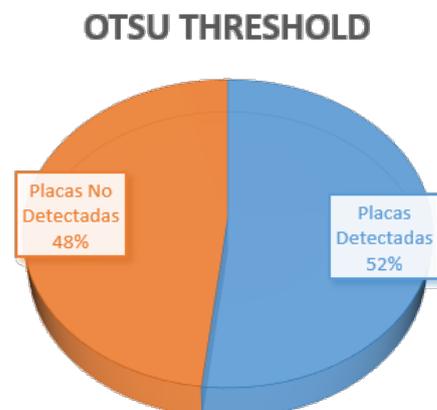


Figura 4.11: Detección de placas usando Otsu Threshold.

El número de placas vehiculares detectadas usando Adaptive Mean Threshold es mayor al método anterior y logra más de un 70 % del total. La búsqueda de contornos es similar que en el método anterior, pero se incrementa el porcentaje de detección ya que el pre procesamiento es diferente y da una vista mejor de los contornos que contiene una imagen.

ADAPTIVE MEAN THRESHOLD

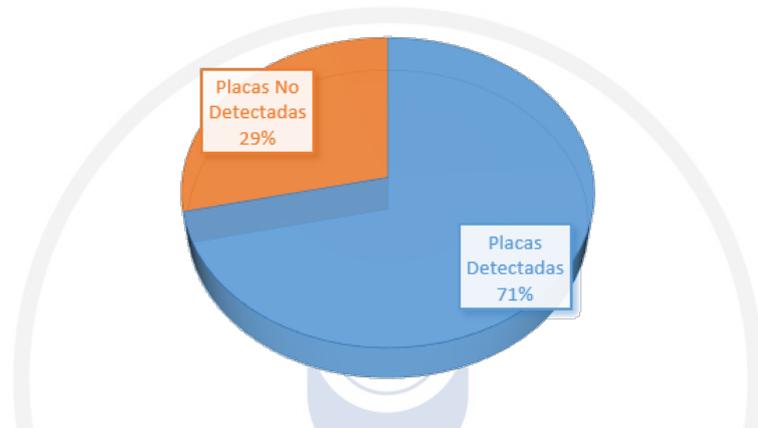


Figura 4.12: Detección de placas usando Adaptive Mean Threshold.

Realizar operaciones morfológicas para encontrar placas vehiculares en una imagen da un notable incremento en la detección. Este método no busca los contornos de la placa vehicular, en su totalidad, si no solo los bordes verticales en base a un filtro *sobel*, dilataciones y erosiones.

OPERACIONES MORFOLÓGICAS



Figura 4.13: Detección de placas usando Operaciones Morfológicas.

Con estos resultados, se puede notar un elevado nivel de detección de placas vehiculares para el sistema propuesto.

$$\text{Detección} = \frac{211}{214} * 100 = 98,60 \%$$

Para comprender la razón por la cual ciertas placas no son detectadas como tal en las imágenes, es necesario entender el proceso que realiza el sistema para detectar una potencial placa. El mismo se puede resumir, a breves rasgos, en los pasos:

1. Binarización de la imagen.
2. Búsqueda de contornos similares a placas vehiculares.
3. Búsqueda de sub contornos dentro de una potencial placa vehicular.
4. Si la potencial placa vehicular contiene al menos 5 sub contornos, es tratada como tal. Caso contrario, es descartada y se continúa la búsqueda en el paso 2 hasta terminar de examinar todos los contornos.

En la Tabla 4.3 se puede observar ejemplos de las imágenes binarizadas y el número de sub contornos detectados. Se ha considerado que una placa vehicular debe tener un mínimo de 5 sub contornos que son los correspondientes a los caracteres de identificación. Esta condición permite descartar objetos que no son placas vehiculares.

Un sub contorno es identificado como tal si este está ubicado en un espacio limpio y sin ruidos. Se puede observar que las imágenes no detectadas como potenciales placas, no cubren el número mínimo de subcontornos. Los caracteres que no son identificados como subcontornos no se encuentran en un espacio limpio. Las placas no detectadas están alteradas, particularmente estas se encuentran dobladas o con un marco que se confunde con los caracteres e imposibilita su correcta detección.

Imagen Binarizada	Número de Potenciales Caracteres	Potencial Placa
	0	No
	4	No
	2	No
	7	Si
	7	Si

Tabla 4.3: Sub contornos encontrados para identificar potenciales placas vehiculares.

4.4.2. Reconocimiento de caracteres

La Tabla 4.4 y las Figuras 4.14, 4.15 y 4.17, muestran el desempeño del sistema en cuanto al reconocimiento para cada uno de los métodos de reconocimiento óptico de caracteres implementado en el sistema propuesto. Se toman como datos de entrada las placas vehiculares, detectadas anteriormente, que suman en total 211.

Método	Correcto	1 Error	2 Errores	3 o más Errores
ANN	189	11	2	9
Tesseract	107	26	4	74
KNN	4	38	63	106

Tabla 4.4: Aciertos y errores producidos en el reconocimiento de las 211 placas.

El número de placas correctamente reconocidas usando la red neuronal para reconocimiento de caracteres, es aproximadamente del 90 %, sin embargo existen errores de reconocimiento encontrados en placas vehiculares en mal estado o que tienen diferentes tipos de letra que aquellas establecidas en la normativa legal.

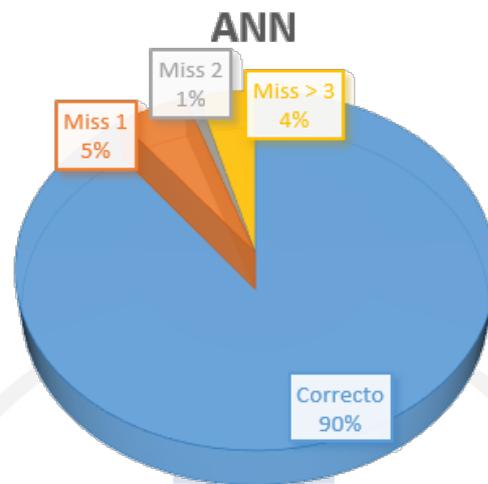


Figura 4.14: Reconocimiento de óptico caracteres usando una red neuronal.

El sistema Tesseract por su parte, tiene un porcentaje de acierto del 50 % aproximadamente. Este valor se presenta posiblemente debido a que Tesseract es en general un OCR muy preciso, siempre y cuando la imagen sea lo más limpia posible, sin ruidos, es decir sin detalles no necesarios. Tesseract falla en muchas ocasiones en el sistema propuesto, especialmente al tratar de reconocer placas en mal estado.

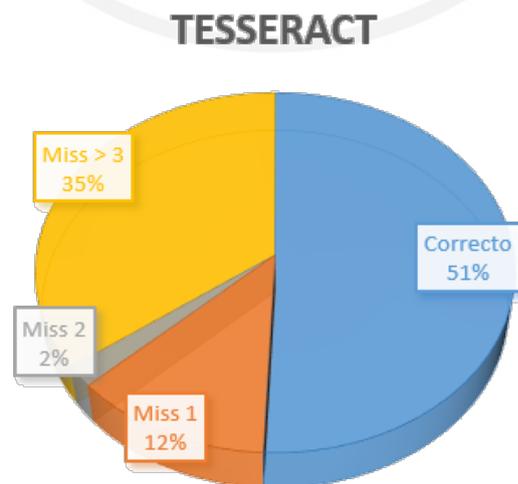


Figura 4.15: Reconocimiento óptico de caracteres usando Tesseract

Existen escasas ocasiones en las que la respuesta OCR de Tesseract es una cadena vacía o que carezca de caracteres suficientes para identificar una placa,

esto se debe a que se realiza un filtrado que conserva únicamente números y letras en dicha respuesta, puesto que esta suele contener en muchas ocasiones caracteres que no son de interés. Los caracteres no deseados suelen ser típicamente comillas simples, comillas dobles, corchetes, etc. que son producidas por ruido en las imágenes de placas vehiculares en mal estado como se observa en la Figura 4.16.



Figura 4.16: Placa vehicular en mal estado, a color y binarizada.

El algoritmo de vecinos más próximos implementado resulta ser el menos preciso, pero ya que el sistema propuesto utiliza un mecanismo de votación *caracter a caracter*, su reconocimiento llega a ser de utilidad en casos en que los otros métodos fallan.

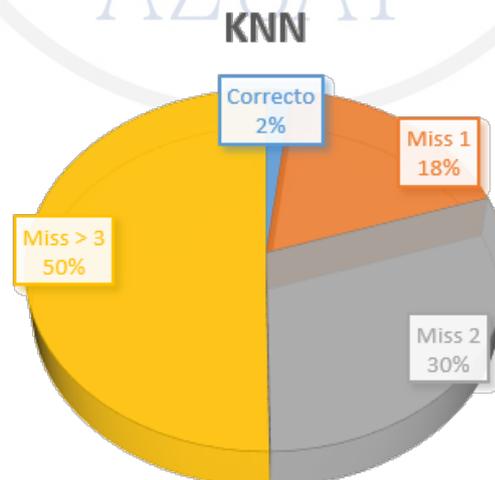


Figura 4.17: Reconocimiento óptico de caracteres usando algoritmo de vecinos más próximos.

En la Tabla 4.5 se muestran ejemplos de los resultados de reconocimiento para diferentes placas, los cuales han sido exitosos y no exitosos, así como su porcentaje de acierto. Como se puede observar, los resultados dependen mucho de la calidad de las imágenes de las placas vehiculares. Existen placas vehiculares en muy buen

estado sobre las cuales el reconocimiento de caracteres se ejecuta sin mayores inconvenientes y se asegura hasta tasa de acierto del 100%. Así también, existen placas vehiculares en mal estado en las que el reconocimiento de caracteres se dificulta y se suele tener porcentajes bajos de acierto.

Tras haber reconocido exitosamente 191 de las 211 placas detectadas, el nivel de reconocimiento del sistema propuesto se ha calculado como sigue:

$$\text{Reconocimiento} = \frac{191}{211} * 100 = 90,52\%$$

Imagen	ANN	Tesseract	KNN	Votación	Placa Real	Reconocida	Caracteres Correctos	Número Caracteres	Porcentaje Aciertos	Caracteres Errados
	PPM544	PH54	CEM944	PPM544	PPW544	No	5	6	83.33 %	W = M
	AXW	AJLI	AZQ	AXW	ADL215	No	1	6	16.66 %	D = X L = W 2 = Nada 1 = Nada 5 = Nada
	AAV083	AAV0383	AAY333	AAV033	AVV0383	No	5	7	71.42 %	8 = 3 3 = Nada
	HIM19	IH91	WIU19	HIM19	ADH919	No	0	6	0 %	A = H D = I H = M 9 = 1 1 = 9 9 = Nada
	UIL04		UIL30	UIL04	GRE896	No	0	6	0 %	G = U R = I E = L 8 = 0 9 = 4 6 = Nada
	AFN708	AFN708	AFN798	AFN708	AFN708	Si	6	6	100 %	
	PQG592	PUG5924	FUU892	PUG592	PQG592	No	5	6	83.33 %	Q = U
	ABD8795	ABD8795	AQO8795	ABD8795	ABD8795	Si	7	7	100 %	
	ABD7244	ABD7234	AUU7234	ABD7234	ABD7234	Si	7	7	100 %	
	UBA2724	UBA2724	UUA2724	UBA2724	UBA2724	Si	7	7	100 %	
	AFY299	AFYZ99	ACY299	AFY299	AFY299	Si	7	7	100 %	
	ABA2870	ABA2870	ADA2873	ABA2870	ABA2870	Si	7	7	100 %	

Tabla 4.5: Ejemplos de placas vehiculares reconocidas y no reconocidas.

4.5. Evaluación Cualitativa

Tomando como base los resultados de la evaluación cuantitativa, se puede establecer los niveles de rendimiento en los que el sistema propuesto puede trabajar. El rendimiento del sistema en este aspecto, ha sido evaluado bajo el criterio “Si tú la puedes ver, la podemos leer” el cual es usado por Perceptics³, compañía con trayectoria en sistemas reconocimiento de placas vehiculares.



Figura 4.18: Ejemplo de una placa en mal estado, binarizada y reformada.

Para el análisis cualitativo es necesario el comprender el punto de vista con el que se trabaja para realizar las tareas de detección y reconocimiento de placas. En la Figura 4.18 se consideran 2 perspectivas, la visión humana y la visión por computador.

- a Una placa vehicular a color, como sería vista por un ser humano; la cual a pesar de estar en mal estado puede ser reconocida por una persona.
- b La misma placa vehicular binarizada, en la que se puede apreciar segmentos separados que pertenecen a un mismo caracter. Para el sistema propuesto, determinar con precisión los caracteres contenidos, tal y como lo haría un humano, resulta difícil y requiere tratamientos adicionales.
- c Placa vehicular binarizada y reformada, en la que los segmentos separados de un mismo caracter se han unido.

³<http://www.perceptics.com/insights/9/measuring-license-plate-recognition-accuracy/>

De hecho, reconocer los caracteres de la imagen (b), para un ser humano puede resultar difícil, sin haber visto antes la imagen (a). Sin embargo un humano es capaz de reconocer esos caracteres por el hecho de serlo y tener la capacidad de unir o completar las secciones faltantes gracias al aprendizaje y razonamiento del cerebro humano. Para una computadora, esto no es intuitivo.

Con este antecedente, se ha considerado los niveles de rendimiento óptimo, bueno, malo y deficiente, los mismos que son detallados a continuación:

4.5.1. Rendimiento Óptimo

Este nivel de rendimiento se consigue en el sistema propuesto, cuando:

- La placa vehicular es detectada sin inconvenientes por cualquiera de los métodos implementados.
- Los caracteres de identificación de la placa son reconocidos con un 100 % de certeza, sin caracteres sobrantes o faltantes.

Es claro que para obtener este nivel de rendimiento las condiciones también han de ser óptimas. La placa vehicular ha de ser detectada sin perder detalles necesarios para el reconocimiento, la placa ha de estar en buenas condiciones, la placa ha de respetar los formatos establecidos por la ANT y la placa ha de estar en un lugar visible para el sistema.

4.5.2. Rendimiento Bueno

Para conseguir este nivel de rendimiento, será necesario que:

- La placa vehicular sea detectada, por alguno de los métodos implementados.
- El reconocimiento es realizado con un 70 % de certeza o más. Permitiendo pérdidas de hasta un caracter, o detección de un caracter adicional.

Se considera que este escenario se tendrá cuando una placa vehicular esté en buen estado pero incluya detalles ornamentales o funcionales que no permitan su correcto reconocimiento. También estarán aquí aquellas placas vehiculares en buen estado y con sus caracteres correctamente segmentados, pero en las que el sistema no ha logrado un correcto reconocimiento de caracteres.

4.5.3. Rendimiento Malo

Este nivel de rendimiento será conseguido cuando:

- La placa vehicular se detecte, total o parcialmente, con cualquiera de los métodos implementados en el sistema propuesto.
- Los caracteres de la placa son reconocidos con un nivel de certeza del 60 % o menos, con pérdidas o incrementos de varios caracteres.

Se considera que este rendimiento se dará al detectar placas en mal estado, lo cual no asegurará un correcto reconocimiento de sus caracteres. Además, se toma en cuenta ocasiones en las que las placas vehiculares son detectadas parcialmente como se muestra en la Figura 4.19.

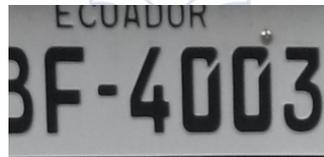


Figura 4.19: Ejemplo de una placa vehicular detectada parcialmente.

4.5.4. Rendimiento Deficiente

Este nivel de reconocimiento resulta ser el menos favorable, puesto que se estima que aquí no se detectará una placa vehicular por tanto el reconocimiento no será posible. Como se muestra en la Tabla 4.3, existen escenarios con placas vehiculares que no pueden ser detectadas por el sistema propuesto.

4.6. Conclusiones

El modelo de evaluación pretende evaluar el rendimiento y la precisión del sistema propuesto en las tareas de detección y reconocimiento de caracteres. Se evaluaron cada uno de los métodos de detección y reconocimiento de caracteres con el propósito de obtener una visión clara de la manera en que el sistema trabaja con cada una de las muestras obtenidas.

Los resultados obtenidos reflejan que el sistema propuesto supera los resultados planteados en principio, tomando en cuenta que el reconocimiento de placas vehiculares es una tarea de complejidad considerable y que depende de factores que en varios casos están fuera del alcance del mismo, como las condiciones de iluminación y el estado de las placas vehiculares.

5.1. Conclusiones

Una vez realizadas las tareas en cada una de las etapas de análisis, diseño, desarrollo y pruebas del presente proyecto de titulación: *Sistema de Reconocimiento de Placas Vehiculares*, se logró cumplir con los objetivos propuestos. Así mismo se verificó la funcionalidad del sistema de acuerdo a los resultados esperados que se plantearon al inicio de este trabajo de tesis, logrando alcanzar niveles de detección y reconocimiento de placas vehiculares acorde al estado del arte.

Adicionalmente, los autores concluyen que:

En el área de Visión por Computador, existe una gran cantidad de librerías que realizan tareas específicas que ayudan a simular el sistema de visión humano para la detección, segmentación, localización y reconocimiento de objetos de interés en una imagen, mediante técnicas de procesamiento digital de imágenes. Gracias a las librerías de acceso libre *OpenCV* y la documentación que presenta, se lograron solucionar muchos inconvenientes en cuanto a la detección de las placas vehiculares.

Al inicio, para la detección de una placa vehicular, se estimó que la mejor estrategia debía basarse en encontrar contornos de cuatro lados con una relación r , entre el alto y ancho de una placa vehicular convencional. Pero a medida que se realizaron las pruebas de detección se identificó que no todas las placas tienen marcados sus contornos debido a las condiciones en las que se captura la imagen, o debido al estado en el estas se encuentran, por lo que se investigó e implementó un

método que ayude a solucionar este problema. Se optó por el uso de operaciones morfológicas como alternativa para la detección de la zona de la placa vehicular. Lo que realiza este método es buscar contornos verticales y unirlos como si fueran un solo conjunto de píxeles, ayudando a remarcar la zona en donde se localizan los caracteres de la placa. Sin embargo, esto no fue suficiente ya que en ciertos casos existían segmentos de las imágenes identificados como placas puesto que poseían líneas verticales o contornos similares a los de una placa vehicular. En estas detecciones erróneas se encontraban segmentos de la imagen que contenían los faros de los vehículos, nombres de sus fabricantes, radiadores, etc. por lo que se decidió que un segmento debería contener al menos 5 sub contornos, pertenecientes a los caracteres de identificación, para ser considerado como una potencial placa vehicular.

En principio el análisis para detección de placas vehiculares se realizó usando el tamaño real de la imagen capturada, ya que los algoritmos de pre procesamiento y detección de placas, no requerían un esfuerzo computacional significativo. Sin embargo, a medida que se iba desarrollando el sistema, el rendimiento y los tiempos de respuesta se degradaron paulatinamente, por lo que se optó por empezar a trabajar con una **Región de Interés (ROI)** específica y de menores dimensiones, es decir un segmento de la imagen en donde posiblemente se ubicara la zona de la placa al momento de ingresar o salir un vehículo.

Así mismo, se encontró pertinente incluir una etapa de pre procesamiento sobre las imágenes capturadas, ya que se busca eliminar el ruido y mejorar principalmente las diferencias de brillo causadas por el color del vehículo y la placa. En esta etapa se realiza una conversión de la imagen capturada a escala de grises ya que la mayoría de funciones de los sistemas de visión por computador trabajan únicamente sobre un canal de la imagen. También se busca resaltar los bordes de la placa vehicular aplicando técnicas como la ecualización del histograma para la mejora de contrastes, basada en la normalización de los valores en la escala de grises para cada píxel.

En la tarea de Reconocimiento de Placas Vehiculares, las variables impuestas por el ambiente en donde se implementa el sistema juegan un papel importante, entre las más influyentes se tiene: los niveles de iluminación, niveles de contraste y niveles de brillo, ya que la experiencia ha demostrado que constituye una tarea compleja el controlar los efectos que generan al momento de capturar las

imágenes. Aún así con estos efectos indeseables, el sistema desarrollado es lo suficientemente robusto gracias al pre procesamiento mencionado.

En Ecuador no existe un estándar único definido para las placas vehiculares, incluso las que emite la Agencia Nacional de Tránsito tienen diferencias de formatos entre unas y otras en cuanto a fuente y tamaño de letra, tamaño de la placa vehicular y la posición de los elementos de la misma. De igual manera no existe un control por parte de las comisiones o agencias de tránsito pertinentes para el correcto uso de la placa vehicular. Estos y otros factores dificultan el correcto análisis debido a los diversos casos que se presentaron y que se mencionan a continuación:

- Placas deformadas por el resultado de un estrellamiento o choque.
- Placas desgastadas por el tiempo o factores externos.
- Placas modificadas intencionalmente.
- Placas con caracteres ilegibles.
- Placas con láminas publicitarias del fabricante del vehículo o del lugar en donde fue adquirido.
- Placas cuyos caracteres se presentan con el mismo color del fondo (ej. placa y letras de color blanco).
- Placas escritas o repintadas a mano.

La red neuronal para reconocimiento óptico de caracteres ha sido entrenada con 20 diferentes tipos de números y 30 diferentes tipos de letras, por lo que está optimizada para trabajar con caracteres muy semejantes a los que impone la ANT (Agencia Nacional de Tránsito) en la normativa legal vigente, de igual forma la red neuronal tratará de obtener los mejores resultados posibles al reconocer caracteres de fuentes desconocidas.

Cuando se comenzó a crear las redes neuronales para el reconocimiento de caracteres se probaron varias configuraciones como las siguientes:

1. Variar el número de neuronas en la capa de entrada, el cual está dado por las dimensiones de los caracteres a reconocer. Los tamaños de los caracteres variaron desde dimensiones de 32x48 píxeles, 16x32 píxeles hasta 16x16 píxeles. El proceso de entrenamiento y posterior reconocimiento de caracteres se volvía pesado en términos de procesamiento computacional al usar

tamaños grandes, por ello se optó por usar un tamaño relativamente pequeño de 16x16 píxeles, es decir 256 neuronas en la capa de entrada.

2. Variar el número de perceptrones en la capa oculta, pasando por 512, 128, 64, 32 hasta 16. Durante el proceso de entrenamiento y posterior reconocimiento, el usar un número grande de neuronas en la capa oculta incrementaba significativamente el tiempo de respuesta pero los resultados no eran los esperados en el reconocimiento. Por ello, tras varias pruebas, se optó por usar un número de neuronas pequeño en la capa oculta, es decir 16 neuronas.
3. Variar el número de neuronas en la capa de salida, que originalmente fue 8 con el objetivo de formar una salida codificada en binario. Se optó por usar 10 neuronas en la capa de salida para la red neuronal de reconocimiento de números y 26 para la red de reconocimiento de letras. Esto simplificó el entendimiento de las respuestas que producían las redes neuronales.
4. Variar el número de capas dentro de la capa oculta. Se realizaron pruebas hasta 3 con capas ocultas, lo cual generaba tiempos de respuesta rápidos pero sin precisión en reconocimiento. Por ello se optó por usar una sola capa dentro de la capa oculta la cual entregaba la mejor tasa de aciertos de reconocimiento.
5. El número de iteraciones para el entrenamiento de las redes neuronales también cambió desde 10 hasta 150. Si el número de iteraciones era muy alto, el porcentaje de aciertos en el reconocimiento era bastante bajo al igual que cuando el número de iteraciones era muy pequeño. Se observó que el porcentaje de aciertos y el número de iteraciones tenían una relación semejante a una curva Gaussiana, produciéndose sub entrenamientos al usar pocas iteraciones y sobre entrenamientos al usar muchas iteraciones. Por ello, y al realizar varias pruebas, se optó por usar 50 iteraciones para el entrenamiento de la red neuronal de números y 90 para el entrenamiento de la red neuronal de letras, consiguiendo tasas de aciertos superiores al 90%.

A medida que se probó cada uno de las opciones mencionadas y se realizaba el proceso de entrenamiento se notó que el rendimiento disminuía o aumentaba. De manera que se eligieron los parámetros que daban el mejor porcentaje de aciertos y fueron usados para generar las estructuras de las redes neuronales implementadas.

5.2. Trabajo Futuro

Para obtener mayor robustez en el Sistema propuesto, los autores encuentran oportuno recomendar que:

Se debe investigar sobre cámaras o dispositivos de tecnología equivalente que brinden mayores prestaciones a las que se menciona en la sección 3.4.1. Dependiendo del tipo de cámara se puede aumentar la velocidad con la que se adecúen los niveles de brillo en escenas con exceso o carencia de luz, como también mejorar el nivel de detalle en las imágenes que se analicen.

En una implementación real se debe sustituir el uso del láser y la fotoresistencia (circuito implementado a nivel de prototipo), para la detección del vehículo en la zona de ingreso o salida cerca del parante. Por lo que se recomienda el uso de sensores de bucle de inducción, ya que el láser puede ser interrumpido por cualquier persona u objeto que no sea un vehículo. Los sensores de inducción son utilizados para detectar vehículos que pasan o llegan a un punto o altura determinada, por ejemplo delante de un semáforo, o en este caso en el ingreso o salida de un parqueadero. En general, se instala un sensor de inducción electromagnética bajo la calzada, generándose una tensión eléctrica cada vez que un cuerpo con masa de hierro o acero pasa cerca de los conductores o del sensor.

Para la etapa de pruebas, se capturaron 214 imágenes de vehículos particulares. Se recomienda realizar una etapa de pruebas con placas de vehículos comerciales, gubernamentales, de gobiernos autónomos descentralizados, ya que varían de los vehículos particulares, por el color de la franja que denota la palabra “ECUADOR” y se podrían obtener variaciones en los resultados.

De igual manera los algoritmos y técnicas empleadas para el reconocimiento de caracteres han sido optimizados para placas de automóviles, segmentando los caracteres de manera lineal. Por lo que se recomienda adaptar el reconocimiento para placas de **motos**. Este trabajo a futuro es un reto ya que se tendrá que analizar varios aspectos como: la localización de la cámara enfocando a la parte posterior de una motocicleta. A diferencia de la segmentación de las placas de automóviles, otro de los aspectos a revisar, es la forma de segmentar la placa de una motocicleta ya que se tendrá que analizar los dos renglones que contienen los caracteres de identificación.

Para obtener resultados aún más precisos en cuanto a la detección de la placa es conveniente el establecer restricciones bajo las que el sistema debe operar

para asegurar resultados óptimos. Como por ejemplo, colocar la cámara en una ubicación en la que siempre se tenga el mejor enfoque de un vehículo y su placa. Así como también considerar que la cámara se encuentre en un lugar en el que factores externos no puedan alterar el enfoque, por ejemplo usando una caja metálica. O en su defecto usar cámaras diseñadas para soportar condiciones climáticas adversas sin necesidad de un cobertor.

El reconocimiento podría llegar a ser mucho más robusto y casi infalible si se implementan técnicas y algoritmos que permitan el recuperar los caracteres de placas vehiculares en mal estado o aquellas en las que no sea fácil distinguir fuentes conocidas o caracteres con segmentos separados. Se pueden probar otras técnicas de reconocimiento óptico de caracteres, como se indica en el capítulo 2, diferentes a las implementadas en el sistema propuesto. También se puede probar usando otros motores OCR de terceros u otros modos de operación del motor *Tesseract*.

Es importante el resaltar que un sistema de reconocimiento de placas vehiculares puede incrementar notablemente la precisión de reconocimiento si se cuenta con normas legales que unifiquen los formatos de las matrículas. No sería suficiente que existan las regulaciones necesarias para placas vehiculares sin que las autoridades respectivas, en Ecuador la ANT, ejerzan un control estricto verificando que estas cumplan los formatos establecidos y estén en lugares visibles. Ya que una placa es un indicador unívoco de un vehículo es importante que se tomen estas recomendaciones, no solo en caso de implementar un LPR.

REFERENCIAS

- Abdullah, S. (2006, June). License plate recognition using multilayer neural networks. En *Computing informatics, 2006. icoci '06. international conference on* (p. 1-7). doi: 10.1109/ICOCI.2006.5276525
- Agam, G. (2006). Introduction to programming with OpenCV. *Online Document, 27*.
- Anagnostopoulos, C.-N., Anagnostopoulos, I., Psoroulas, I., Loumos, V., y Kayafas, E. (2008). License plate recognition from still images and video sequences: A survey. *IEEE Transactions on Intelligent Transportation Systems, 9*(3), 377-391. doi: 10.1109/TITS.2008.922938
- Arduino. (2015). *Arduino - arduinoarduino*. Descargado 2015-06-15, de <http://www.arduino.cc/en/Main/ArduinoBoardUno>
- BitWindows. (2015). *Componente pentru automatizari*. Descargado 2015-05-03, de <http://www.bitwindows.ro/ro/categorii/151-Componente-pentru-automatizari>
- Bradski, G. R., y Kaehler, A. (2011). *Learning OpenCV: [computer vision with the OpenCV library]* (1. ed., [Nachdr.] ed.). O'Reilly.
- Buch, N., Velastin, S., y Orwell, J. (2011, Sept). A review of computer vision techniques for the analysis of urban traffic. *Intelligent Transportation Systems, IEEE Transactions on, 12*(3), 920-939. doi: 10.1109/TITS.2011.2119372
- Cano, J., y Pérez-Cortés, J.-C. (2003). Vehicle license plate segmentation in natural images. En *Pattern recognition and image analysis* (pp. 142-149). Springer.

- Comelli, P., Ferragina, P., Granieri, M., y Stabile, F. (1995, Nov). Optical recognition of motor vehicle license plates. *Vehicular Technology, IEEE Transactions on*, 44(4), 790-799. doi: 10.1109/25.467963
- Dewar, M. (2015). *Connectivity cca*. Descargado 2015-05-22, de <http://www.ualberta.ca/~ccwj/teaching/image/morph/>
- DLINK. (2015). *DES-1008d switch d-link*. Descargado 2015-07-27, de <http://www.dlink.com/es/es/home-solutions/connect/switches/des-1008d-8-port-10-100mbps-desktop-switch>
- Draghici, S. (1997). A neural network based artificial vision system for licence plate recognition. *Int. J. Neural Syst.*, 8(1), 113-126.
- Duan, T. D., Du, T. H., Phuoc, T. V., y Hoang, N. V. (2005). Building an automatic vehicle license plate recognition system. En *Proc. Int. Conf. Comput. Sci. RIVF* (p. 59-63). Citeseer.
- Fahmy, M. (1994, Aug). Automatic number-plate recognition: neural network approach. En *Vehicle navigation and information systems conference, 1994. proceedings., 1994* (p. 99-101). doi: 10.1109/VNIS.1994.396858
- Fernández, C. J. S., y Consuegra, V. S. (2005). *Reconocimiento Óptico de Caracteres (OCR)*. *Phys. Rev. Lett.*
- Foscam. (2015). *Foscam quick installation guide*. Descargado 2015-07-03, de http://foscam.us/media/mconnect_uploadfiles/f/i/fi9821w%20v2%20quick%20installation%20guide.pdf
- Hongliang, B., y Changping, L. (2004, Aug). A hybrid license plate extraction method based on edge statistics and morphology. En *Pattern recognition, 2004. icpr 2004. proceedings of the 17th international conference on* (Vol. 2, p. 831-834 Vol.2). doi: 10.1109/ICPR.2004.1334387
- Hsieh, C.-T., Chang, L.-C., Hung, K.-M., y Huang, H.-C. (2009, Dec). A real-time mobile vehicle license plate detection and recognition for vehicle monitoring and management. En *Pervasive computing (jcpc), 2009 joint conferences on* (p. 197-202). doi: 10.1109/JCPC.2009.5420189
- Huang, M.-C., y Yen, S.-H. (2004). A real-time and color-based computer vision for traffic monitoring system. En *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on* (Vol. 3, p. 2119-2122). IEEE.

- Jin, L., Xian, H., Bie, J., Sun, Y., Hou, H., y Niu, Q. (2012, junio). License Plate Recognition Algorithm for Passenger Cars in Chinese Residential Areas. *Sensors*, 12(12), 8355-8370. doi: 10.3390/s120608355
- Johnson, A., y Bird, B. (1990, May). Number-plate matching for automatic vehicle identification. En *Electronic images and image processing in security and forensic science, iee colloquium on* (p. 4/1-4/8).
- Jonker, P. P. (1992). *Morphological image processing: Architecture and VLSI design: Architecture and vlsi design*. Springer Science & Business Media.
- Khalil, M., y Kurniawan, F. (2014, April). License plate detection method for real-time video of low-cost webcam based on hybrid svm-heuristic approach. En *Information technology: New generations (itng), 2014 11th international conference on* (p. 321-326). doi: 10.1109/ITNG.2014.21
- Klinger, T. (2003). *Image processing with labview and imaq vision*. Pearson Education.
- LaHora. (2013). *Más sensores para zona norte*. Descargado 2013-05-20, de http://www.lahora.com.ec/index.php/noticias/show/1101509412/-1/M%C3%A1s_sensores_para_v%C3%ADas_de_zona_norte.html#.VUZ5AJ01Vha
- Lee, H.-J., Chen, S.-Y., y Wang, S.-Z. (2004, Aug). Extraction and recognition of license plates of motorcycles and vehicles on highways. En *Pattern recognition, 2004. icpr 2004. proceedings of the 17th international conference on* (Vol. 4, p. 356-359 Vol.4). doi: 10.1109/ICPR.2004.1333776
- Liaqat, A. G. (2011). *Mobile real-time license plate recognition* (Tesis de Master no publicada). Linnaeus University, School of Computer Science, Physics and Mathematics.
- Lotufo, R., Morgan, A., y Johnson, A. (1990, Feb). Automatic number-plate recognition. En *Image analysis for transport applications, iee colloquium on*.
- Martin, F., García, M., y Alba, J. L. (2002). New methods for automatic reading of VLP's (vehicle license plates). *Proc. IASTED Int. Conf. SPPRA*, 126-131.
- Monzó, R. S. i. (2014). *Automatismos Industriales. Conceptos y procedimientos*. Nau Llibres.
- Musoromy, Z., Ramalingam, S., y Bekooy, N. (2010). Edge detection comparison for license plate detection. *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, 1133-1138.

- Mysql. (2015). *My sql community edition*. Descargado 2015-06/15, de <https://www.mysql.com/products/community/>
- Nijhuis, J., ter Brugge, M., Helmholt, K., Pluim, J., Spaanenburg, L., Venema, R., y Westenberg, M. (1995, Nov). Car license plate recognition with neural networks and fuzzy logic. En *Neural networks, 1995. proceedings., ieee international conference on* (Vol. 5, p. 2232-2236 vol.5). doi: 10.1109/ICNN.1995.487708
- OCR using Artificial Neural Network (OpenCV) - Part 2 - Preprocessing*. (2015). Descargado de <http://www.nithinrajs.in/ocr-using-artificial-neural-network-opencv-part-2-preprocessing/>
- Opencv. (2015). *Open cv documentation*. Descargado 2015-06/15, de <http://docs.opencv.org/>
- OpenCV Dev Team, S. L. (2015). *Neural networks*. Descargado de http://docs.opencv.org/modules/ml/doc/neural_networks.html
- Qt. (2015). *Qt gui*. Descargado 2015-06/15, de <https://forum.qt.io/>
- Radha, R. (2012). A novel approach to extract text from license plate of vehicles. *Signal & Image Processing : An International Journal*, 3(4), 181-192. doi: 10.5121/sipij.2012.3415
- Rosado-Muñoz, A., Bataller-Mompeán, M., Soria-Olivas, E., Scarante, C., y Martínez, J. G. (2011). FPGA implementation of an adaptive filter robust to impulsive noise: two approaches. *Industrial Electronics, IEEE Transactions on*, 58(3), 860-870. Descargado 2015-07-21, de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5067312
- Setchell, C. J. (1997). *Applications of Computer Vision to Road-traffic Monitoring*. (Tesis Doctoral no publicada). University of Bristol.
- Sirithinaphong, T., y Chamnongthai, K. (1999). The recognition of car license plate for automatic parking system. En *Signal processing and its applications, 1999. isspa '99. proceedings of the fifth international symposium on* (Vol. 1, p. 455-457 vol.1). doi: 10.1109/ISSPA.1999.818210
- Smith, R. (2007). An Overview of the Tesseract OCR Engine. En *ICDAR* (Vol. 7, p. 629-633).
- TPLINK. (2015). *Modem router adsl inalámbrico td-w8961nd*. Descargado 2015-07-27, de <http://www.tp-link.ec/products/details/?model=TD-W8961ND>

Wu, C., On, L. C., Weng, C. H., Kuan, T. S., y Ng, K. (2005, Aug). A macao license plate recognition system. En *Machine learning and cybernetics, 2005. proceedings of 2005 international conference on* (Vol. 7, p. 4506-4510 Vol. 7). doi: 10.1109/ICMLC.2005.1527732

Wu, H.-H., Chen, H.-H., Wu, R.-J., y Shen, D.-F. (2006). License plate extraction in low resolution video. En *Pattern recognition, 2006. icpr 2006. 18th international conference on* (Vol. 1, p. 824-827). doi: 10.1109/ICPR.2006.761

Zunino, R., y Rovetta, S. (2000, Feb). Vector quantization for license-plate location and image coding. *Industrial Electronics, IEEE Transactions on*, 47(1), 159-167. doi: 10.1109/41.824138



UNIVERSIDAD DEL
AZUAY



ANEXO A

DESCRIPCIÓN DE CASOS DE USO

En este anexo se presenta la descripción de cada uno de los casos de uso definidos en el Capítulo 3. Para esto se ha tomado como punto de partida el diagrama de casos de uso extendido que se encuentra en la Figura 3.3, en el cual se detallan las interacciones de los usuarios y el sistema.

Caso de Uso 1	Identificarse (Usuario)
Actores	-Usuario
Descripción	El usuario se autentica mediante la placa vehicular o el ingreso del número de identificación en la interfaz WEB .
Flujo de Eventos Principal	1. El usuario ingresa con su vehiculo al parqueadero 2. El sistema reconoce la placa vehicular del usuario.
Flujo de Eventos Alternativo	2a. El sistema no reconoce la placa vehicular del usuario. 2b. El sistema presenta el campo para ingresar la identificación del usuario 2c. El usuario ingresa el campo con su numero de identificación 3. El sistema consulta si se encuentra registrada la placa del vehículo o identificación del usuario 4. El sistema identifica si el usuario es una persona registrada o ajena al parqueadero
Pre-condición	El sistema está en espera al ingreso de un vehículo
Post-condición	El usuario se encontrará identificado
Caso de éxito	El usuario ingresa al parqueadero
Triggers	El usuario ingresa con su vehículo para empezar la detección.
Requerimientos Funcionales	R.1 El sistema verificará la identidad del usuario R.2 El sistema indicará si el usuario está o no registrado R.3 El sistema notificará si el usuario no registrado desea ingresar con una tarifa.
Requerimientos Especiales	

Tabla A.1: Caso de uso 1: identificarse(usuario)

Caso de Uso 2	Identificarse (Administrador)
Actores	-Administrador
Descripción	El administrador se autentica mediante su usuario y contraseña.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El administrador ingresa a la interfaz WEB 2. El sistema presenta los campos para que el usuario pueda logearse 3. El administrador rellena los campos con el usuario y contraseña
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 3a. El administrador no completa la información. 3b. El administrador ingreso un usuario o contraseña invalido. 4. El sistema indica error en la autenticación. 5. El administrador regresa al paso 2.
Pre-condición	El servicio con la pagina WEB esta iniciado
Post-condición	El administrador se encontrará autenticado
Caso de éxito	El administrador podrá registrar nuevos usuarios y acceder a la video vigilancia.
Triggers	El administrador ingresa a la pagina WEB para logearse.
Requerimientos Funcionales	<p>R.1 El sistema verificará la identidad del administrador.</p> <p>R.2 El sistema indicará si el administrador ha ingresado usuario y/o contraseña incorrectos.</p>
Requerimientos Especiales	

Tabla A.2: Caso de uso 2: identificarse(administrador)

Caso de Uso 3	Solicitar Ingreso
Actores	-Usuario
Descripción	Se procesara la solicitud de ingreso al parqueadero de un usuario, previo a la verificación de su identidad.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El usuario solicita el ingreso al parqueadero 2. El sistema consulta los usuario registrados en la Base de Datos 3. El sistema activa el mecanismo para abrir el parante
Flujo de Eventos Alternativo	<p>2a Se produce un error al consultar los datos</p> <ol style="list-style-type: none"> 1. El sistema presenta una pantalla indicando que el usuario no esta registrado y el ingreso tiene cierta tarifa. 2. El usuario acepta la tarifa. 3. El sistema activa el mecanismo para abrir el parante
Pre-condición	El usuario debe identificarse mediante la placa del vehículo
Post-condición	El sistema regresa a la espera del ingreso de otros vehículos
Caso de éxito	El usuario podrá ingresar al parqueadero
Triggers	Al acercarse al parante para ingreso al parqueadero.
Requerimientos Funcionales	<p>R.1 El sistema reconocerá la placa del vehiculo que solicita el ingreso.</p> <p>R.2 El sistema verificará la identidad del usuario.</p> <p>R.3 El sistema indicará si el usuario esta registrado o no en el sistema</p> <p>R.4 El sistema preguntará al usuario si desea ingresar por una tarifa.</p> <p>R.5 El sistema activará el mecanismo para abrir el parante.</p>
Requerimientos Especiales	

Tabla A.3: Caso de uso 3: solicitar ingreso

Caso de Uso 4	Detectar Placa
Actores	-Sistema
Descripción	Se procederá a la detección de la placa vehicular previo a la solicitud de ingreso al parqueadero
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema realiza una secuencia de capturas del vehículo 2. El sistema realiza el preprocesamiento de la imagen 3. El sistema procede a la detección de bordes. 4. El sistema verifica todos los contornos de la imagen. 5. El sistema filtra los contornos de la imagen de acuerdo al área y el ratio (ancho/alto) de una placa vehicular otorgada por la comisión de Tránsito del Ecuador. 6. El sistema elige un posible candidato de la region de la placa entre los contornos filtrados 7. El sistema dibuja el contorno de la placa vehicular en tiempo real.
Flujo de Eventos Alternativo	<p>6a. El sistema falla en el reconocimiento de los candidatos de la region de la placa vehicular.</p> <ol style="list-style-type: none"> 1. El sistema detecta contornos mas pequeños que pueden contener las letras de la placa vehicular 2. El sistema sistema verifica que sean entre 6 y 8 los contornos detectados, de acuerdo a los caracteres de una placa vehicular. 3. El sistema enmarca en un rectangulo toda el área de los contornos de los caracteres 4. El sistema dibuja el contorno de la placa obteniendo mayor precisión que el primer método
Pre-condición	El usuario debe acercarse con su vehículo al parante para que la cámara empice con la detección placa del vehículo
Post-condición	El contorno de la placa se encuentra detectado
Caso de éxito	El sistema podrá realizar el reconocimiento de caracteres
Triggers	Al visualizar una placa vehicular
Requerimientos Funcionales	<p>R.1 El sistema detectará la placa del vehículo que solicita el ingreso.</p> <p>R.2 El sistema activará las funciones para corrección de la perspectiva de la imagen.</p> <p>R.3 El sistema activará las funciones para el reconocimiento de caracteres.</p>
Requerimientos Especiales	

Tabla A.4: Caso de uso 4: detectar placa

Caso de Uso 5	Reconocer Placa
Actores	-Sistema
Descripción	Se procederá al reconocimiento de los caracteres de la placa, previo a la detección del contorno que contiene la región con las letras y números.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema recorta un segmento de la imagen con la región de la placa. 2. El sistema realiza la binarización de la imagen recortada, para mejorar el reconocimiento de caracteres. 3. El sistema envía la imagen recortada y sin segmentar al software OCR denominado TESSERACT. 4. El software TESSERACT, retorna una cadena con los caracteres reconocidos en la imagen. 5. El sistema procede a segmentar la imagen binarizada previamente, mediante la búsqueda de contornos pequeños, para obtener el área de cada carácter de la placa vehicular. 6. El sistema recorta la imagen con la región de la placa, en segmentos más pequeños que contienen las letras y números de la placa vehicular. 7. El sistema convierte cada imagen en un arreglo, con los valores binarios correspondientes a cada píxel: 0 Blanco, 1 Negro. 8. El sistema envía como parámetros de entrada el arreglo de cada imagen, a las redes neuronales ANN y KNN. 9. Las redes neuronales retornan como salidas a cada uno de los arreglos, las letras o números reconocido. 10. El sistema inicia el mecanismo de votación por cada letra, para determinar los mejores resultados entre los 3 tipos de OCR. 11. El sistema obtiene la cadena de caracteres de la placa vehicular.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 4a. El software TESSERACT, falla en el reconocimiento de los caracteres 4b. El sistema continúa con el paso 5.
Pre-condición	El sistema debe haber detectado la región de la placa
Post-condición	Los caracteres se encuentran reconocidos
Caso de éxito	El sistema podrá consultar si la placa está registrada en el parqueadero
Triggers	Al detectar la región de una placa vehicular
Requerimientos Funcionales	<ol style="list-style-type: none"> R.1 El sistema reconocerá los caracteres de placa del vehículo que solicita el ingreso. R.2 El sistema activará las funciones del software TESSERACT R.3 El sistema activará las funciones de las redes neuronales implementadas para el reconocimiento de caracteres R.4 El sistema activará el mecanismo de votación para determinar el mejor resultado entre los métodos implementados para el reconocimiento de caracteres.
Requerimientos Especiales	

Tabla A.5: Caso de uso 5: reconocer placa

Caso de Uso 6	Registrar Ingreso
Actores	-Base de Datos
Descripción	La Base de Datos registra el Ingreso de un usuario al parqueadero ya sea una persona ajena o no al parqueadero.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema consulta la placa reconocida en la Base de Datos. 2. La Base de Datos retorna el nombre de usuario que está registrado de acuerdo a la placa del vehículo. 3. El sistema envía la información fecha, hora, placa para ingresar en la Base de Datos. 4. La Base de Datos almacena el/los datos del ingreso del usuario. 5. El sistema presenta en la página web el mensaje de que se ha registrado el/los datos del ingreso del usuario al parqueadero.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 2a. La Base de Datos retorna valores nulos, ya que la placa no se encuentra registrada. 1. El sistema presenta en pantalla el campo para el ingreso de la cédula del usuario no registrado. 2. El sistema indica si desea ingresar al parqueadero por una tarifa. 3. El usuario acepta la tarifa. 2b. Retorna al flujo desde el paso 4.
Pre-condición	Haber reconocido los caracteres de una placa vehicular o ingresar la identificación de un usuario.
Post-condición	
Caso de éxito	El ingreso del usuario quedará registrado en la Base de Datos
Triggers	Solicitar el ingreso al parqueadero
Requerimientos Funcionales	<ol style="list-style-type: none"> R.1 El sistema indicará al usuario que se encuentra registrado en el parqueadero. R.2 El sistema informará cuando el usuario este no registrado en el parqueadero. R.3 Para los nuevos usuarios el sistema pedirá la identificación del usuario. R.4 Para los nuevos usuarios, el sistema indicará si desea ingresar con una tarifa. R.5 El sistema registrará el ingreso del usuario/vehículo al parqueadero.
Requerimientos Especiales	

Tabla A.6: Caso de uso 6: registrar ingreso

Caso de Uso 7	Iniciar Parante
Actores	-Sistema Parante
Descripción	El sistema parante es el mecanismo que activará el circuito para abrir o cerrar el parante ubicado al ingreso del parqueadero.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sistema LPR envía una señal por puerto serial al sistema parante 2. El sistema parante recibe y ejecuta los comandos mediante la tarjeta ARDUINO UNO. 3. Se activa un servomotor, para girar un parante acoplado al ingreso del parqueadero.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 2a. El sistema parante no recibe ningún comando mediante la tarjeta ARDUINO UNO. 3b. Existe conflicto para activar el servomotor. <ol style="list-style-type: none"> 1. Se visualiza un mensaje de error en el sistema LPR
Pre-condición	Solicitar el Ingreso al Parqueadero
Post-condición	Abrir/Cerrar Parante
Caso de éxito	El sistema LPR realiza la comunicación serial correctamente con el sistema parante.
Triggers	Solicitar el ingreso al parqueadero
Requerimientos Funcionales	<ol style="list-style-type: none"> R.1 El sistema parante recibirá los comandos mediante el puerto serial del sistema LPR. R.2 El sistema parante notificará que está activado al sistema LPR
Requerimientos Especiales	

Tabla A.7: Caso de uso 7: iniciar parante

Caso de Uso 8	Abrir Parante
Actores	-Sistema Parante
Descripción	Se procede activar el Sistema Parante implementado en una placa perforada, mediante la tarjeta ARDUINO UNO, para abrir el parante al ingreso del parqueadero
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. La tarjeta ARDUINO UNO, envía la señal con el número de grados que tiene que rotar el servomotor ubicado en la placa perforada. 2. El motor gira el número de grados que recibió desde ARDUINO UNO 3. El parante acoplado, se levanta por la fuerza del servomotor.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. La tarjeta ARDUINO UNO, no envía la señal a la placa perforada 3a. El parante acoplado no se levanta.
Pre-condición	Establecer la comunicación entre la tarjeta ARDUINO UNO y la para perforada
Post-condición	
Caso de éxito	Se abre el parante para dejar ingresar al vehículo.
Triggers	Solicitar el ingreso al parqueadero
Requerimientos Funcionales	R.1 El motor girará el numero de grados especificados por el sistema LPR
Requerimientos Especiales	

Tabla A.8: Caso de uso 8: abrir parante

Caso de Uso 9	Cerrar Parante
Actores	-Sistema Parante
Descripción	Se procede activar el Sistema Parante implementado en una placa perforada, mediante la tarjeta ARDUINO UNO, para cerrar el parante al ingreso del parqueadero
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El sensor LDR implementado en la placa perforada verifica las señales de luz enviadas por un láser, una vez que un vehículo ingresa totalmente al parqueadero y envía la señal a la tarjeta ARDUINO UNO. 2. La tarjeta ARDUINO UNO, envía la señal con el número de grados que tiene que rotar el servomotor ubicado en la placa perforada. 2. El motor gira el número de grados que recibió desde ARDUINO UNO 3. El parante acoplado, baja por la fuerza del servomotor.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 1a. La tarjeta ARDUINO UNO, no envía la señal a la placa perforada 3a. El parante acoplado no se levanta.
Pre-condición	<p>Recibir la señal de luz enviada desde el láser hacia el sensor LDR.</p> <p>Establecer la comunicación entre la tarjeta ARDUINO UNO y la para perforada</p>
Post-condición	
Caso de éxito	Se cierra el parante para bloquear el paso a nuevos vehículos.
Triggers	<p>Recibir la señal de luz desde el láser</p> <p>Verificar mediante técnicas de procesamiento de imágenes, que no existan objetos en la escena del ingreso del parqueadero.</p>
Requerimientos Funcionales	<p>R.1 El motor girará el numero de grados especificados por el sistema LPR</p> <p>R.2 El laser emitirá la luz de manera directa al sensor cuando no existan bloqueos, es decir vehículos en el ingreso del parqueadero.</p>
Requerimientos Especiales	

Tabla A.9: Caso de uso 9: cerrar parante

Caso de Uso 10	Registrar Usuario
Actores	-Administrador
Descripción	El administrador ingresa los usuarios con sus respectivos vehículos para que el ingreso sea automatizado y el sistema no le pida identificación al usuario.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El administrador ingresa en la sección de registro usuarios en la pagina WEB. 2. La pagina WEB despliega los campos requeridos para el registro del usuario y su vehículo. 3. El administrador rellena los campos en la pagina WEB. 4. La pagina WEB se conecta con la Base de Datos. 5. La Base de Datos consulta si el usuario fue registrado en ocasiones anteriores 6. La Base de Datos almacena los datos enviados mediante la pagina WEB. 7. La pagina WEB despliega una pantalla indicando que se ha registrado el usuario y su vehículo.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 6a. La Base de Datos no almacena los datos enviados mediante la pagina WEB. ya que el usuario a registrarse ya existe. 6b. Se presenta un mensaje de error en la Pagina WEB, indicando que el usuario ya está registrado
Pre-condición	El administrador debe autenticarse en la pagina WEB
Post-condición	
Caso de éxito	El acceso de los usuarios registrados es automatizado y no se requiere el ingreso de la identificación.
Triggers	La petición desde la Página WEB hacia la Base de Datos.
Requerimientos Funcionales	<ol style="list-style-type: none"> R.1 El sistema verificará si el usuario a registrar existe en la Base de Datos R.2 El sistema indicará el caso en el que el usuario a ingresar ya esté registrado. R.3 El sistema registrará en la Base de Datos, los datos nuevos usuarios y sus vehículos R.3 El sistema indicará el caso cuando se registre un nuevo usuario
Requerimientos Especiales	

Tabla A.10: Caso de uso 10: registrar usuario

Caso de Uso 11	Modificar Usuario
Actores	-Administrador
Descripción	El administrador modifica los datos de los usuarios previamente registrados.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El administrador ingresa en la sección de modificación de usuarios en la pagina WEB. 2. La pagina WEB consulta a la Base de Datos los Usuarios que están, registrados. 3. La Base de Datos retorna la información de los usuarios registrados. 4. Se despliega en la pagina WEB todos los usuarios. 5. El administrador elige cual usuario desea modificar. 6. Se despliega en la pagina WEB la información del usuario que eligió, el administrador bloqueando los campos que están establecidos como llave primaria en la Base de Datos. 7. El administrador realiza los cambios en los campos necesarios que están permitidos modificar. 8. La pagina WEB envía la información con los nuevos Datos a la Base. 9. La Base de Datos registra los nuevos datos del usuario 10. La pagina WEB despliega un mensaje indicando que se efectuaron los cambios.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 9a. La Base de Datos no almacena los datos enviados mediante la pagina WEB. 10a. Se presenta un mensaje de error en la Pagina WEB, indicando que los datos son inválidos
Pre-condición	El administrador debe autenticarse en la pagina WEB
Post-condición	
Caso de éxito	El usuario podrá registrar los nuevos datos del usuario.
Triggers	La petición desde la Página WEB hacia la Base de Datos.
Requerimientos Funcionales	<ol style="list-style-type: none"> R.1 El sistema verificará los usuario registrados en la Base de Datos. R.2 El sistema registrará en la Base de Datos, los datos nuevos usuarios y sus vehículos R.3 El sistema indicará el caso cuando no se registre los nuevos datos del usuario.
Requerimientos Especiales	

Tabla A.11: Caso de uso 11: modificar usuario

Caso de Uso 12	Consultar Usuarios
Actores	-Administrador -Base de Datos
Descripción	El administrador consulta los usuarios que están registrados en la Base de Datos.
Flujo de Eventos Principal	<ol style="list-style-type: none"> 1. El administrador ingresa en la sección de reportes en la página WEB. 2. La página WEB consulta a la Base de Datos el listado de usuarios registrados con sus respectivos vehículos. 3. La Base de Datos retorna la información de usuarios 4. Se visualiza el listado de usuarios en una nueva pantalla de la página WEB.
Flujo de Eventos Alternativo	<ol style="list-style-type: none"> 3a. La Base de Datos retorna un error al consultar los datos. 4a. Se visualiza un error en una nueva pantalla de la página WEB.
Pre-condición	Haberse autenticado en la pagina WEB.
Post-condición	
Caso de éxito	El administrador podrá visualizar e imprimir el listado de usuarios registrados en el parqueadero.
Triggers	Solicitud a la Base de Datos para retornar el listado de usuarios.
Requerimientos Funcionales	R.1 El sistema indicará un listado con los usuarios registrados en un parqueadero.
Requerimientos Especiales	

Tabla A.12: Caso de uso 12: consultar usuarios

ANEXO B

DICCIONARIO DE DATOS

ttiposplaca

Columna	Tipo	Nulo	Predeterminado
CTIPOPLACA	varchar(1)	No	
SERVICIO	varchar(100)	No	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CTIPOPLACA	7	A	No	
XPKTTIPOSPLACA	BTREE	Sí	No	CTIPOPLACA	7	A	No	

usuarios

Columna	Tipo	Nulo	Predeterminado	Comentarios
username	varchar(50)	No		
password	varchar(16)	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	username	0	A	No	

vpersonavehiculos

Comentarios de la tabla: VIEW

Columna	Tipo	Nulo	Predeterminado	Comentarios
NUMEROPLACA	varchar(8)	No		
IDENTIFICACION	varchar(13)	No		
NOMBRE	varchar(150)	No		
CMARCA	int(11)	Sí	NULL	
CMODELO	int(11)	Sí	NULL	
CPROVINCIA	int(11)	No		

tcanton

Columna	Tipo	Nulo	Predeterminado	Comentarios
CPROVINCIA	int(11)	No		
CCANTON	int(11)	No		
NOMBRE	varchar(150)	No		
FDESDE	datetime	No		
FHASTA	datetime	No	2999-12-31 00:00:00	
CPAIS	int(11)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CPAIS	2	A	No	
				CPROVINCIA	2	A	No	
				CCANTON	2	A	No	
XPKTCIUDAD	BTREE	Sí	No	CPAIS	2	A	No	
				CPROVINCIA	2	A	No	
				CCANTON	2	A	No	
XIF1TCIUDAD	BTREE	No	No	CPROVINCIA	2	A	No	
				CPAIS	2	A	No	

tcompania

Columna	Tipo	Nulo	Predeterminado	Comentarios
CCOMPANIA	varchar(3)	No		
NOMBRE	varchar(150)	Sí	NULL	
RUC	varchar(13)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CCOMPANIA	3	A	No	
XPKCOMPANIA	BTREE	Sí	No	CCOMPANIA	3	A	No	

tcomunicacion

Columna	Tipo	Nulo	Predeterminado	Comentarios
CCOMUNICACION	varchar(1)	No		
ACCION	varchar(20)	No		
PLACA	varchar(10)	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CCOMUNICACION	1	A	No	

testadocivil

Columna	Tipo	Nulo	Predeterminado	Comentarios
CESTADOCIVIL	varchar(1)	No		
DESCRIPCION	varchar(50)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CESTADOCIVIL	4	A	No	
XPKTTIPOSESTADOCIVIL	BTREE	Sí	No	CESTADOCIVIL	4	A	No	

testatus

Columna	Tipo	Nulo	Predeterminado	Comentarios
CESTATUS	varchar(1)	No		
DESCRIPCION	varchar(15)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CESTATUS	2	A	No	
XPKTSTATUS	BTREE	Sí	No	CESTATUS	2	A	No	

tgenero

Columna	Tipo	Nulo	Predeterminado	Comentarios
CGENERO	varchar(1)	No		
DESCRIPCION	varchar(50)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CGENERO	2	A	No	
XPKTTIPOSESGENERO	BTREE	Sí	No	CGENERO	2	A	No	

tmarcasvehiculo

Columna	Tipo	Nulo	Predeterminado	Comentarios
CMARCA	int(11)	No		
DESCRIPCION	varchar(50)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CMARCA	40	A	No	
XPKTMARCASVEHICULO	BTREE	Sí	No	CMARCA	40	A	No	

tmodelosvehiculo

Columna	Tipo	Nulo	Predeterminado	Comentarios
CMARCA	int(11)	No		
CMODELO	int(11)	No		
DESCRIPCION	varchar(50)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CMARCA	4	A	No	
				CMODELO	8	A	No	
XPKTMODELOSVEHICULO	BTREE	Sí	No	CMARCA	4	A	No	
				CMODELO	8	A	No	
XIFITMODELOSVEHICULO	BTREE	No	No	CMARCA	4	A	No	

tmovimientos

Columna	Tipo	Nulo	Predeterminado	Comentarios
FECHA	datetime	Sí	NULL	
CPERSONA	int(11)	No		
NUMEROPLACA	varchar(8)	No		
ENTRADASLIDA	varchar(1)	Sí	E	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CPERSONA	0	A	No	
				NUMEROPLACA	0	A	No	
XPKTMOVIMIENTOS	BTREE	Sí	No	CPERSONA	0	A	No	
				NUMEROPLACA	0	A	No	
XIFITMOVIMIENTOS	BTREE	Sí	No	CPERSONA	0	A	No	
				NUMEROPLACA	0	A	No	

tpaís

Columna	Tipo	Nulo	Predeterminado	Comentarios
CPAIS	int(11)	No		
NOMBRE	varchar(150)	No		
FDESDE	datetime	No		
FHASTA	datetime	No	2999-12-31 00:00:00	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CPAIS	0	A	No	
XPKTPAIS	BTREE	Sí	No	CPAIS	0	A	No	

tparante

Columna	Tipo	Nulo	Predeterminado	Comentarios
CPARANTE	varchar(1)	No		
ESTADO	varchar(10)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CPARANTE	0	A	No	

tpersona

Columna	Tipo	Nulo	Predeterminado	Comentarios
NOMBRE	varchar(150)	No		
FNACIMIENTO	date	Sí	NULL	
CIDENTIFICACION	varchar(1)	No		
CTIPOPERSONA	varchar(3)	No		
CCOMPANIA	varchar(3)	No		
IDENTIFICACION	varchar(13)	No		
FDESDE	datetime	No		
FHASTA	datetime	No	2999-12-31 00:00:00	
CGENERO	varchar(1)	No		
CESTADOCIVIL	varchar(1)	No		
CPERSONA	int(11)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CPERSONA	6	A	No	
XPKTPERSONA	BTREE	Sí	No	CPERSONA	6	A	No	
XAK1TPERSONA	BTREE	Sí	No	CIDENTIFICACION	2	A	No	
				CTIPOPERSONA	2	A	No	
				IDENTIFICACION	6	A	No	
XIF1TPERSONA	BTREE	No	No	CIDENTIFICACION	2	A	No	
XIF2TPERSONA	BTREE	No	No	CTIPOPERSONA	2	A	No	
XIF3TPERSONA	BTREE	No	No	CCOMPANIA	2	A	No	
XIF4TPERSONA	BTREE	No	No	CGENERO	6	A	No	
XIF5TPERSONA	BTREE	No	No	CESTADOCIVIL	6	A	No	

tpersonadirecciones

Columna	Tipo	Nulo	Predeterminado	Comentarios
CALLE_PRINCIPAL	varchar(30)	Sí	NULL	
CALLE_SECUNDARIA	varchar(30)	Sí	NULL	
FDESDE	datetime	No		
FHASTA	datetime	No	2999-12-31 00:00:00	
CPROVINCIA	int(11)	No		
CCANTON	int(11)	No		
CPAIS	int(11)	No		
CPERSONA	int(11)	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
XIF2TPERSONADIRECCION	BTREE	No	No	CPROVINCIA	2	A	No	
				CCANTON	2	A	No	
				CPAIS	2	A	No	
XIF3TPERSONADIRECCION	BTREE	No	No	CPERSONA	2	A	Sí	
R_18	BTREE	No	No	CPAIS	2	A	No	
				CPROVINCIA	2	A	No	
				CCANTON	2	A	No	

tpersonavehiculos

Columna	Tipo	Nulo	Predeterminado	Comentarios
CPERSONA	int(11)	No		
NUMEROPLACA	varchar(8)	No		
OBSERVACIONES	varchar(150)	Sí	NULL	
CTIPOPLACA	int(11)	Sí	5	
CMARCA	int(11)	Sí	NULL	
CMODELO	int(11)	Sí	NULL	
FDESDE	datetime	No		
FHASTA	datetime	No	2999-12-31 00:00:00	
CESTATUS	varchar(1)	No	1	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo
PRIMARY	BTREE	Sí	No	CPERSONA	7	A	No
				NUMEROPLACA	7	A	No
XPKTPERSONAVEHICULOS	BTREE	Sí	No	CPERSONA	7	A	No
				NUMEROPLACA	7	A	No
NUMEROPLACA	BTREE	Sí	No	NUMEROPLACA	7	A	No
XIF1TPERSONAVEHICULOS	BTREE	No	No	CPERSONA	7	A	No
XIF2TPERSONAVEHICULOS	BTREE	No	No	CTIPOPLACA	7	A	Sí
XIF3TPERSONAVEHICULOS	BTREE	No	No	CMARCA	2	A	Sí
				CMODELO	7	A	Sí
XIF4TPERSONAVEHICULOS	BTREE	No	No	CESTATUS	2	A	No

tprovincia

Columna	Tipo	Nulo	Predeterminado	Comentarios
CPROVINCIA	int(11)	No		
FDESDE	datetime	No		
FHASTA	datetime	No	2999-12-31 00:00:00	
NOMBRE	varchar(150)	No		
CPAIS	int(11)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CPAIS	2	A	No	
				CPROVINCIA	4	A	No	
XPKTP	BTREE	Sí	No	CPAIS	2	A	No	
				CPROVINCIA	4	A	No	
XIF1TP	BTREE	No	No	CPAIS	2	A	No	

ttiposidentificacion

Columna	Tipo	Nulo	Predeterminado	Comentarios
CIDENTIFICACION	varchar(1)	No		
DESCRIPCION	varchar(50)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CIDENTIFICACION	3	A	No	
XPKTTIPOSIDENTIFICACION	BTREE	Sí	No	CIDENTIFICACION	3	A	No	

ttipospersona

Columna	Tipo	Nulo	Predeterminado	Comentarios
CTIPOPERSONA	varchar(3)	No		
DESCRIPCION	varchar(20)	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	CTIPOPERSONA	2	A	No	
XPKTTIPOPERSONA	BTREE	Sí	No	CTIPOPERSONA	2	A	No	

Este anexo explica el uso de la interfáz de usuario que maneja los ajustes de detección de placas vehiculares. Se supone que estos ajustes serán realizados por un usuario administrador. El sistema posee 3 pestañas para mostrar y configurar la detección de placas vehiculares, para el ambiente de pruebas usado para medir la precisión del sistema y para el manejo del prototipo de ingreso al parqueadero.

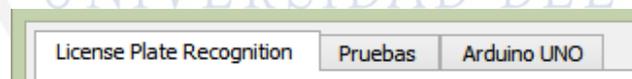


Figura C.1: Pestañas del sistema de reconocimiento de placas vehiculares.

C.1. Pestaña de detección de placas vehiculares - License Plate Recognition

En esta pestaña se pueden ajustar los parámetros para la detección de las placas vehiculares. Los mismos son explicados a continuación.

Los ajustes se deberán realizar de manera adecuada y procurando que los resultados obtenidos sean lo más precisos posibles. Se debe tomar en cuenta que los ajustes realizados aquí solo son a nivel de software; por tanto las imágenes deberán provenir de una cámara que se encuentre en un ambiente controlado y con condiciones aceptables para su captura.

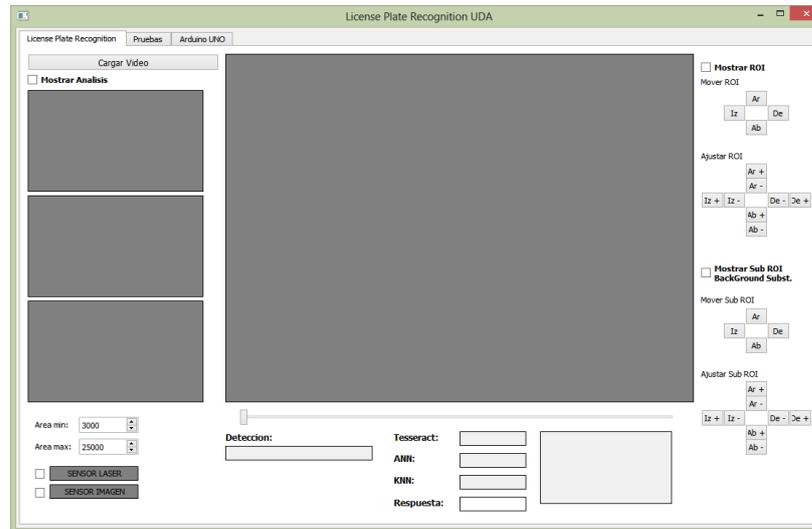


Figura C.2: Pestaña de detección de placas vehiculares.

C.1.1. Cargar video

Este botón permite cargar el video a analizar, el mismo que proviene de la cámara IP. Este botón cambia su estado de “Cargar Video” a “Detener Video” al ser pulsado, y el video resultante se puede observar cerca del mismo en una pequeña ventana.

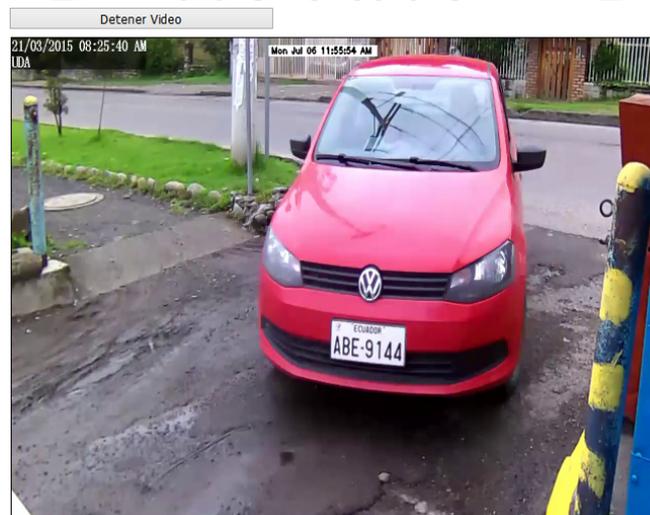


Figura C.3: Cargar o detener el video a analizar

C.1.2. Mostrar análisis

Esta casilla de selección permite observar el pre procesamiento que realiza el sistema mientras analiza las imágenes provenientes del video. Se observará que el

análisis no corresponde a la imagen en su totalidad sino solamente a la región de interés, la misma que se puede ajustar como se muestra más adelante.

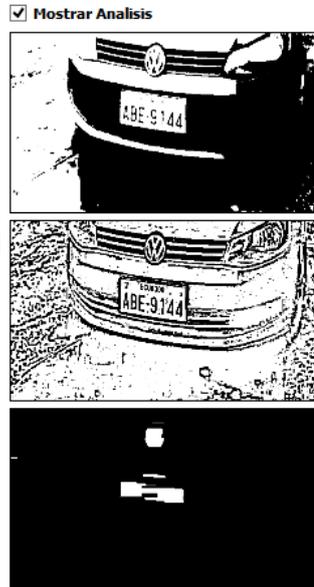


Figura C.4: Análisis en tiempo real del video.

C.1.3. Ajustar área de placa vehicular

Estas casillas permiten ajustar el área de las placas vehiculares que serán detectadas, la misma que debe ser ajustada dependiendo de la cámara que provea las imágenes. Los valores que contiene por defecto han sido establecidos para funcionar con la resolución de la cámara utilizada en el sistema propuesto. En caso de usar otra cámara o imágenes de diferente resolución, se necesitarán ajustes de estos parámetros.

Area min:	<input type="text" value="3000"/>
Area max:	<input type="text" value="25000"/>

Figura C.5: Ajustes de área mínima y máxima de placas vehiculares.

C.1.4. Avanzar y Retrasar video

Esta opción se podrá usar si se está analizando un video que no provea de imágenes en tiempo real. Este *slider* ha sido implementado con el propósito de realizar los análisis necesarios de videos capturados para y durante el desarrollo de este sistema.



Figura C.6: Slider para avanzar o retrasar video.

C.1.5. Etiquetas de información

En esta pestaña se encuentran varias etiquetas informativas que pretenden dar un mejor entendimiento del análisis que se realiza.

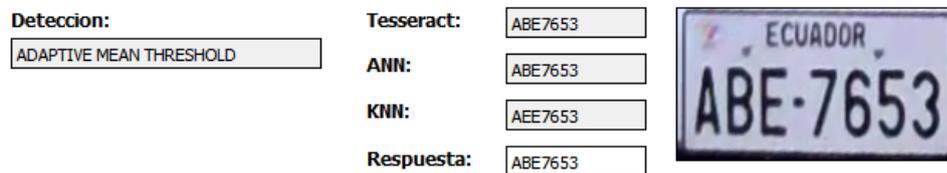


Figura C.7: Información y resultados del análisis.

- **Detección:** etiqueta informativa muestra el método de detección con el que se ha localizado la placa vehicular en la imagen analizada.
- **Tesseract:** etiqueta informativa que muestra la respuesta de reconocimiento óptico de caracteres generada por el motor OCR *Tesseract* sobre la placa vehicular detectada.
- **ANN:** esta etiqueta informativa muestra la respuesta de reconocimiento óptico de caracteres generada por la red neuronal implementada, sobre la placa vehicular que se detectó.
- **KNN:** etiqueta informativa que muestra la respuesta OCR generada por el algoritmo de vecinos más próximos implementado, sobre la placa vehicular detectada.

- **Respuesta:** esta etiqueta informativa muestra la respuesta del método de votación que recibe las 3 respuestas generadas anteriormente. Esta respuesta es la única tomada en cuenta para las operaciones que sean requeridas.
- **Placa:** es una imagen que muestra la placa vehicular que se detectó.

C.1.6. Mostrar, mover y ajustar región de interés (ROI)

La casilla “Mostrar ROI” permite que la región de interés a analizar sea visualizada en el video cargado. La región de interés debe ser ajustada y ubicada usando los botones que se encuentran debajo de la casilla, de manera que se enfoque el área donde se encuentren las placas vehiculares a reconocer.



Figura C.8: Visualización y ajuste de ROI.

C.1.7. Mostrar, mover y ajustar sub región de interés (ROI)

La casilla “Mostrar Sub ROI Background Subst.” permite visualizar la sub región de interés que utiliza un operador *Background Subtractor* para detectar objetos en movimiento en el video analizado. Esta región debe ser movida y ajustada, usando los botones que se encuentran debajo, en un punto de la imagen que varíe únicamente cuando se acerque un vehículo que requiera ser detectado.



Figura C.9: Visualización y ajuste de Sub ROI.

C.1.8. Activar y desactivar sensor láser

Esta casilla permite activar y desactivar el sensor láser usado para detectar vehículos cerca del parante en el protipo. En caso de no ser seleccionado, el parante es cerrado automáticamente al transcurrir 10 segundos. La etiqueta informativa “Sensor Laser” cambia de color al detectar un vehículo o no, se mantiene gris en caso de no detectar nada y rojo al detectar objetos.

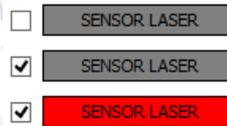


Figura C.10: Casilla para activar y desactivar sensor láser.

C.1.9. Activar y desactivar sensor imagen

Esta casilla permite activar y desactivar el sensor de detección de objetos en movimiento en el video. En caso de no ser activado, no se podrá ejecutar la detección de placas vehiculares. La etiqueta informativa “Sensor Imagen” cambia de color gris a azul al detectar un objeto en movimiento y se mantiene en gris al no detectar alguno. Este sensor trabaja en conjunto con el control de mostrar, mover y ajustar sub región de interés (ROI).

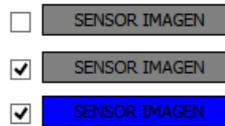


Figura C.11: Casilla para activar y desactivar sensor de imagen en movimiento

C.2. Pestaña de pruebas

Esta parte del programa ha sido implementada con el propósito de realizar las pruebas de detección y reconocimiento del sistema propuesto. Posee botones para cargar imágenes, analizarlas y reconocerlas, además de dar una salida de consola que se usa para generar un archivo `.csv` el mismo que es usado en una hoja de cálculo para obtener resultados. Es importante mencionar que las pruebas requieren de un operador humano que verifique si los resultados del reconocimiento y detección son verdaderos.

Aquí se implementa los mismos métodos usados en el procesamiento de video, ya que se pretende obtener resultados para medir la precisión de los mismos. Cabe el recalcar que las pruebas obedecen a el procedimiento señalado en el Capítulo 4. Sus elementos son detallados brevemente a continuación:

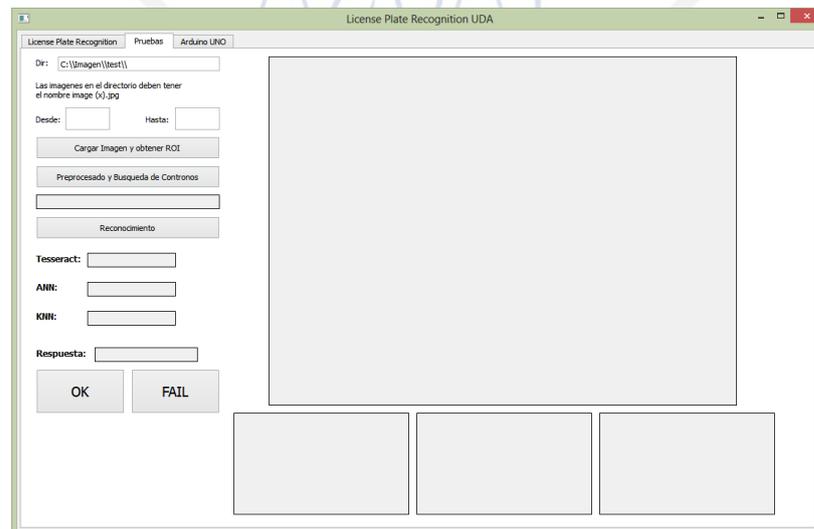


Figura C.12: Pestaña de pruebas de detección y reconocimiento.

C.2.1. Cargar imagen y obtener ROI

Para usar este botón se debe especificar en “Dir” la ruta donde se encuentran las imágenes de prueba. También, el número inicial de las imágenes y el número

final. Cada vez que se pulse el botón “Cargar imagen y obtener ROI” se cargará una nueva imagen que se puede visualizar al igual que la obtención de la región de interés.

Es importante, para el correcto funcionamiento de este ambiente, que todas las imágenes tengan el mismo nombre “image (x).jpg” donde “x” sea un número secuencial perteneciente a los límites especificados.

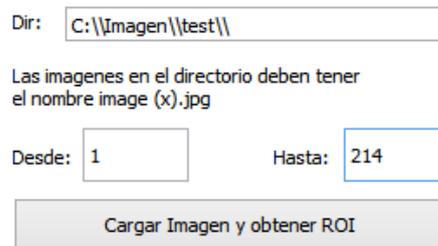


Figura C.13: Componentes necesarios para cargar una imagen de prueba.

C.2.2. Preprocesado y búsqueda de contornos

Al usar este botón se realiza el pre procesamiento para búsqueda de potenciales placas vehiculares en la imagen cargada anteriormente. Se pueden usar aquí los 3 métodos disponibles de detección de placas, es decir *Otsu Threshold*, *Adaptive Mean Threshold* y *Operaciones morfológicas*.

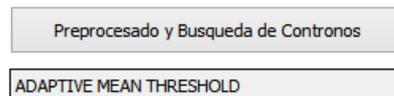


Figura C.14: Botón para pre procesamiento y búsqueda de contornos.

Se puede observar en las imágenes de la figura C.15 la placa vehicular resaltada con líneas verdes y el pre procesamiento realizado sobre la misma.

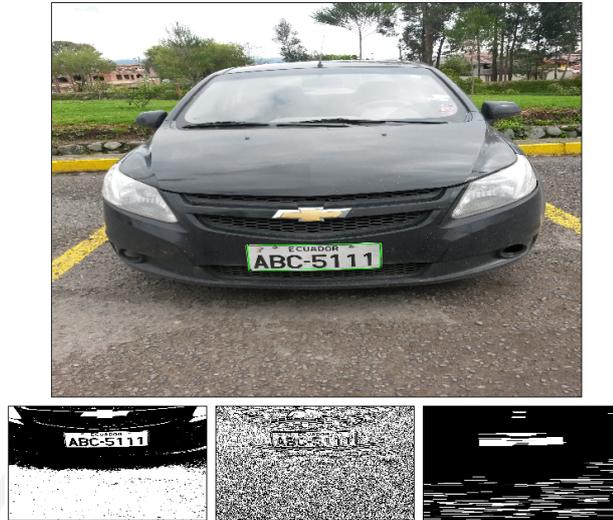


Figura C.15: Imágenes resultantes del análisis.

C.2.3. Reconocimiento

Pulsando este botón se inicia el reconocimiento óptico de caracteres sobre la placa vehicular detectada. Se puede observar las respuestas de los métodos *ANN*, *KNN* y del motor OCR *Tesseract*, además de la respuesta del método de votación. Aquí es donde el operador humano hace uso de los botones “Ok” y “Fail” para decidir si el reconocimiento ha sido exitoso o no.

Tesseract:

ANN:

KNN:

Respuesta:

Figura C.16: Botones para indicar si el reconocimiento fue exitoso o no.

C.3. Pestaña de control Arduino UNO

Esta pestaña contiene 2 botones que sirven para el control del prototipo de ingreso a parqueadero implementado con Arduino UNO. Estos controles permiten la apertura y cierre del parante.



Figura C.17: Botones para controlar el prototipo de ingreso al parqueadero.



ANEXO D

MANUAL DE ESTILOS - LPR UDA

En este anexo se especifica la estructura, diseño y presentación de la aplicación web para administración de usuarios del sistema LPR - UDA. Aquí se detalla la estructura de cada página que conforma el sitio web, con sus colores, tipografía, posición y tamaño de los elementos.

D.1. Introducción

En la actualidad con el incremento de dispositivos móviles como *smartphone's*, *notebook's*, *tablet's*, se ha definido una filosofía para la presentación y diseño de sitios web, de manera que la experiencia del usuario en el uso y la navegación, sea intuitiva y fácil. Con el concepto de **RWD**, *Responsive Web Design*, se puede lograr que la apariencia de una página web sea adaptable a cualquier dispositivo, el cual los usuarios hacen uso para visualizar el contenido.

D.2. Wireframes

Los wireframes son un tipo de guía visual en donde se representa la estructura o el esqueleto de un sitio web. Esta sección está enfocada en mostrar gráficamente la ubicación y tamaño de los elementos que conforman la página web LPR - UDA.



Figura D.1: Wireframe inicio de sesión LPR - UDA.

Inicio de sesión desde dispositivos móviles



Figura D.2: Wireframe inicio de sesión desde dispositivos móviles.

Página Home



Figura D.3: Wireframe home

Página acerca del Proyecto



Figura D.4: Wireframe de qué se trata? LPR - UDA.

Página de Servicios



Figura D.5: Wireframe servicios.

Página del proceso de reconocimiento

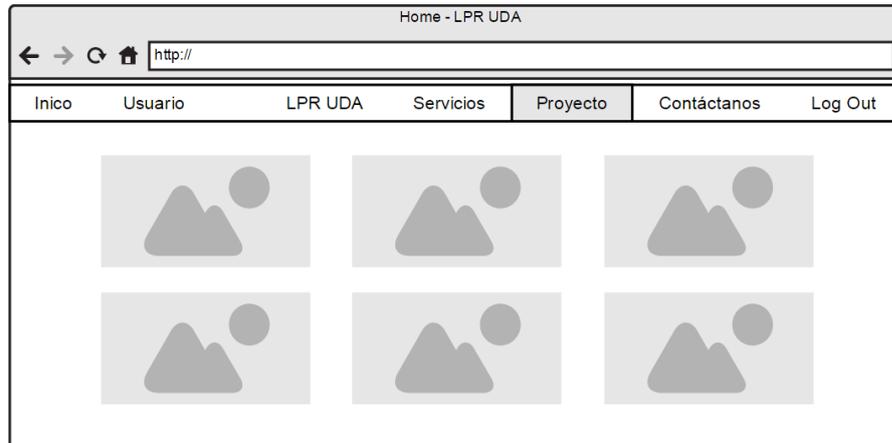


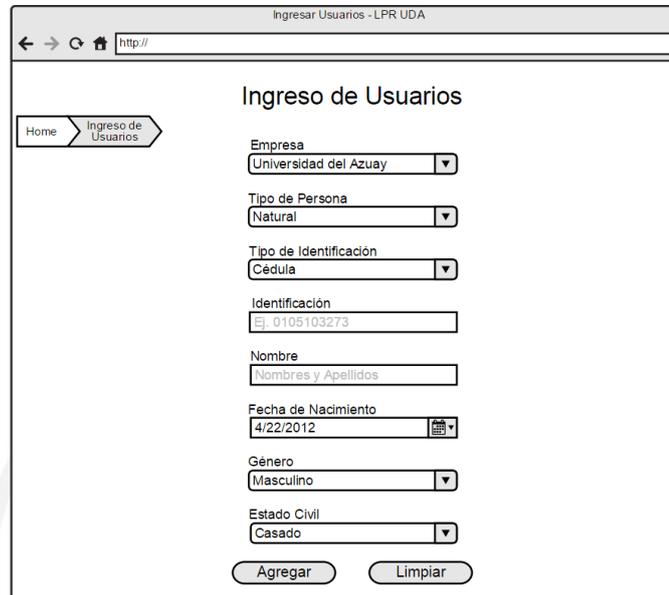
Figura D.6: Wireframe proyecto LPR - UDA.

Página Contáctanos



Figura D.7: Wireframe contáctanos.

Página Ingreso de Usuarios



Ingresar Usuarios - LPR UDA

Home > Ingreso de Usuarios

Ingreso de Usuarios

Empresa: Universidad del Azuay

Tipo de Persona: Natural

Tipo de Identificación: Cédula

Identificación: Ej. 0105103273

Nombre: Nombres y Apellidos

Fecha de Nacimiento: 4/22/2012

Género: Masculino

Estado Civil: Casado

Agregar Limpiar

Figura D.8: Wireframe Ingreso de Usuarios.

Página Registrar Vehículos



Registrar Vehículos - LPR UDA

Home > Registro de Vehículos

Registro de Vehículos

Usuario: Juan Carlos Quintana Reyes

Número de Placa: Ej. 0105103273

Tipo de Placa: Vehículo Particular

Marca Vehículo: Chevrolet

Modelo Vehículo: Aveo Family

Estatus: Activo

Observaciones: Vehículo Chocado, Vehículo Rojo,

Registrar Limpiar

Figura D.9: Wireframe registrar vehículo

Reporte de Usuarios - Vehículos

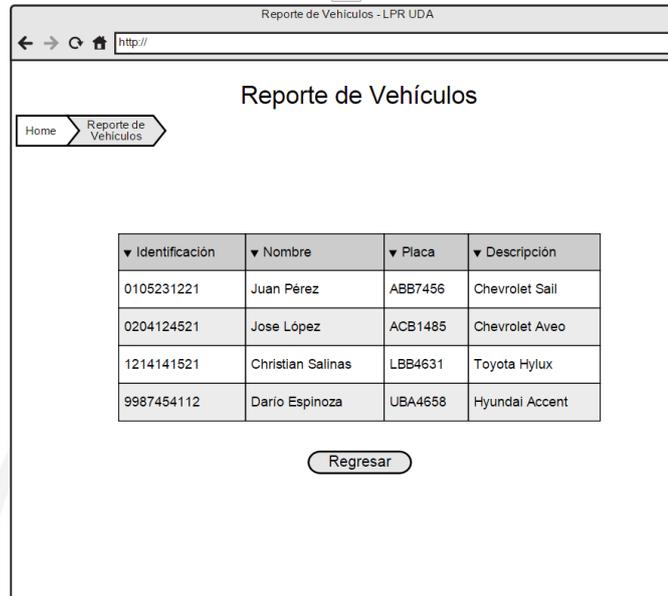


Figura D.10: Wireframe reporte de vehículos

Página para Video Vigilancia

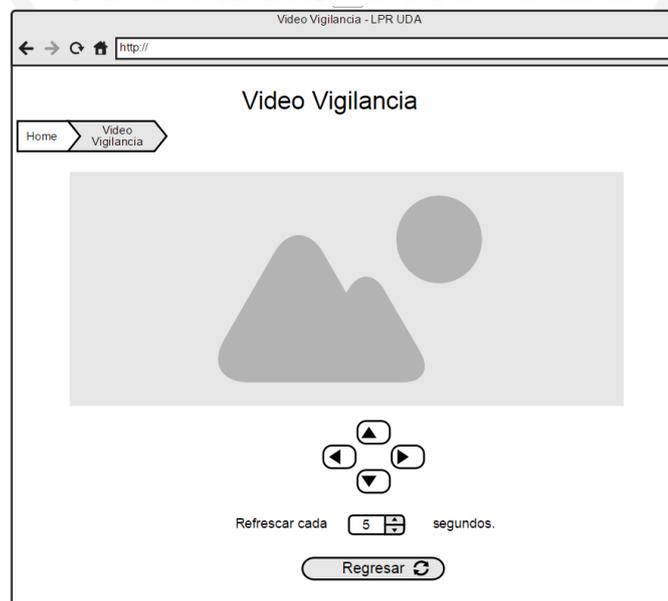


Figura D.11: Wireframe video vigilancia

D.3. Mockups

Los mockups son prototipos del diseño real del sitio web. A diferencia de los wireframes, en los mockups se adjunta el diseño y los colores de cada página.

Inicio de Sesión



Figura D.12: Mockup inicio de sesión LPR - UDA.

Inicio de Sesión desde dispositivos móviles



Figura D.13: Mockup inicio de sesión desde dispositivos móviles.

Página Home



Figura D.14: Mockup home

Página acerca del Proyecto



Figura D.15: Mockup de qué se trata? LPR - UDA.

Página de Servicios



Figura D.16: Mockup servicios.

Página del proceso de reconocimiento

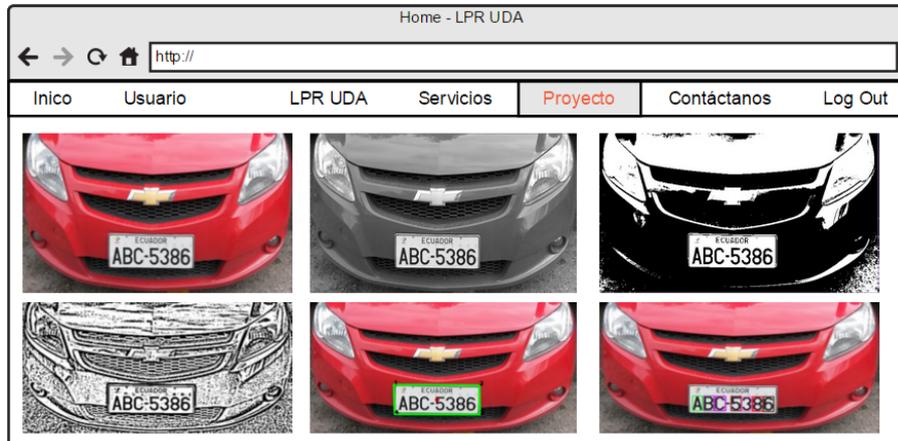


Figura D.17: Mockup proyecto LPR - UDA.

Página Contáctanos



Figura D.18: Mockup contáctanos.

Página Ingreso de Usuarios

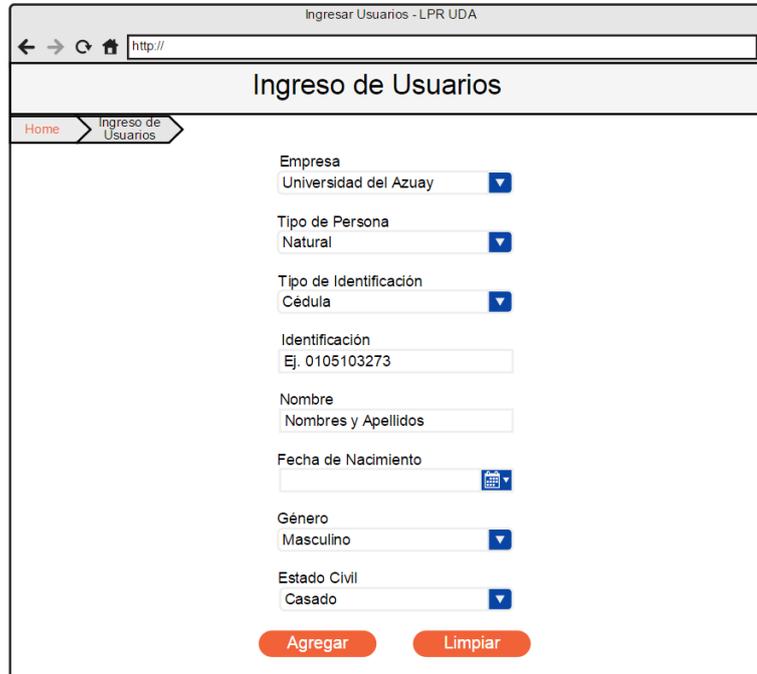


Figura D.19: Mockup Ingreso de Usuarios.

Página Registrar Vehículos

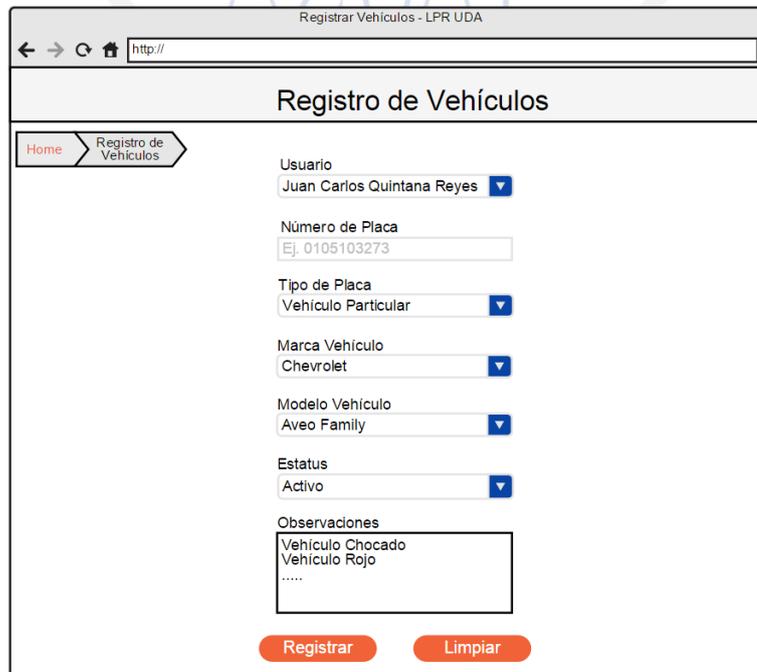


Figura D.20: Mockup registrar vehículo

Reporte de Usuarios - Vehículos

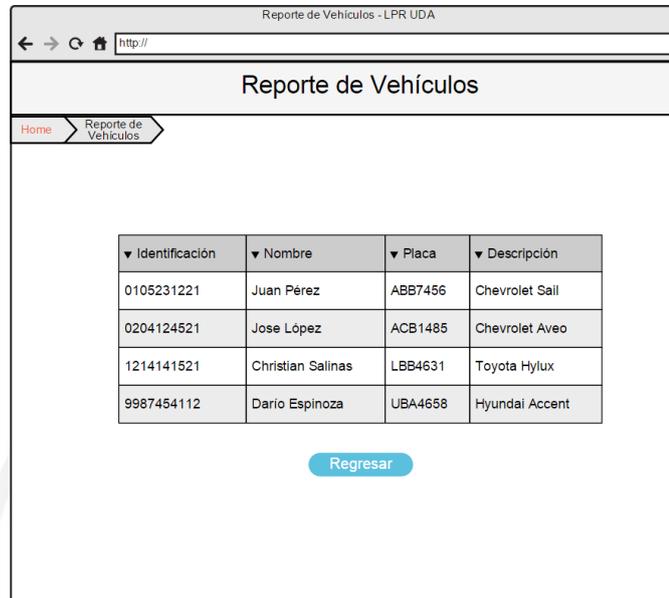


Figura D.21: Mockup reporte de vehículos

Página para Video Vigilancia

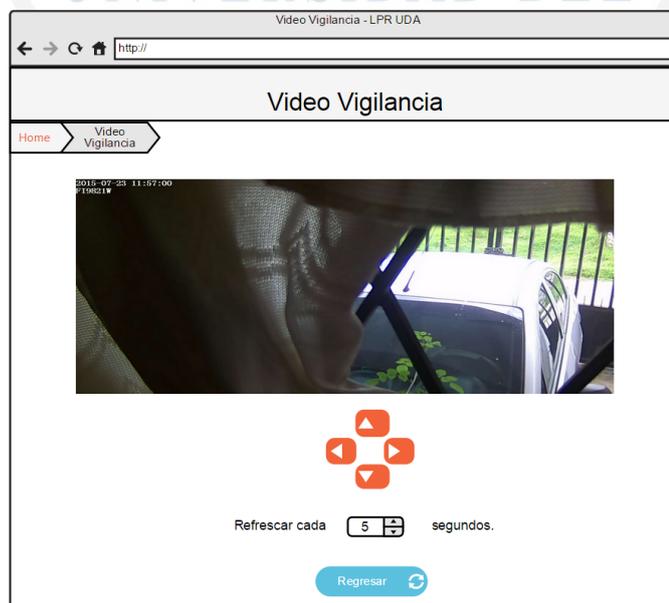


Figura D.22: Mockup video vigilancia

D.4. Cabecera

El encabezado superior de la web se compone de un menú desplegable que se compone de 7 opciones. El diseño del menú es variable dependiendo de la ubicación en nuestro sitio web. En la zona superior la página se establece un estilo principal y a medida que navegamos con el *scroll* hasta la parte inferior se establece un estilo secundario.

Menú superior - diseño principal



Figura D.23: Menu 1 LPR - UDA

Menú superior - diseño secundario



Figura D.24: Menu 2 LPR - UDA

D.5. Iconos

A continuación se detallan los iconos utilizados en varias partes del sitio web.

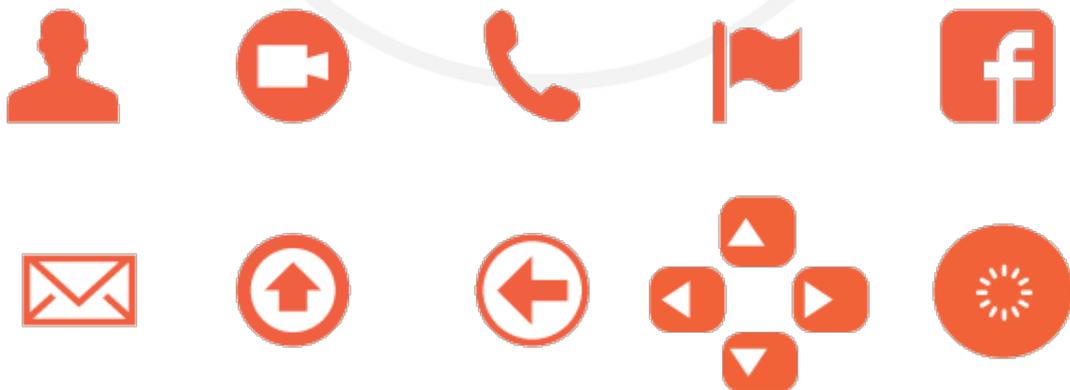


Figura D.25: Iconos

D.6. Color

Los colores utilizados en el sitio web son:

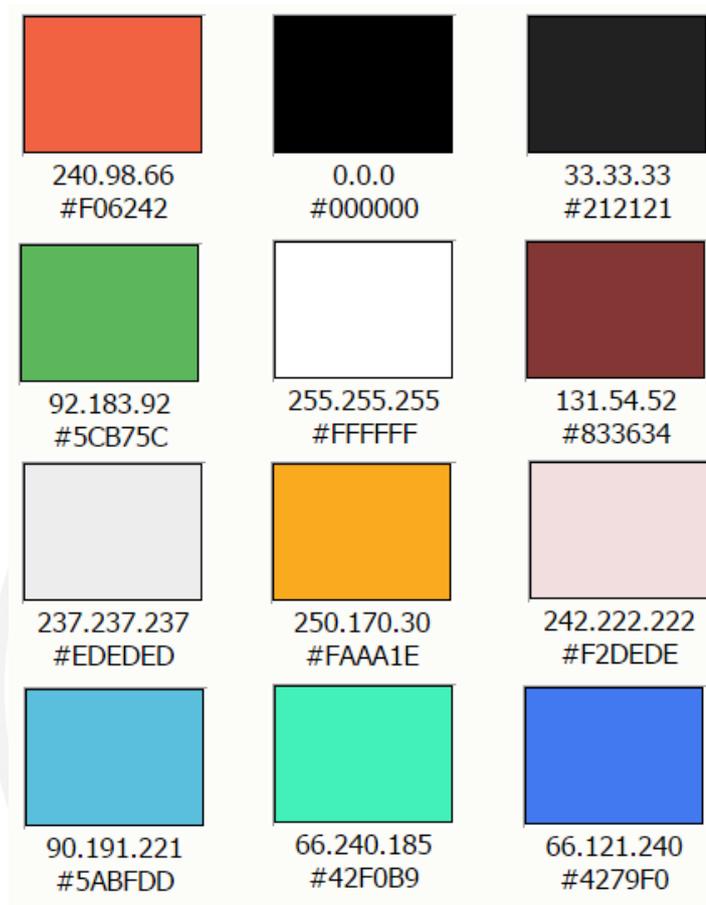


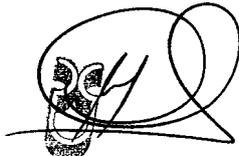
Figura D.26: Colores

Doctora Jenny Ríos Coello, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay,

CERTIFICA:

Que, el H. Consejo de Facultad en sesión realizada el 30 de enero del 2015, conocía la petición de los estudiantes **CHRISTIAN ANTONIO SALINAS ESCOBAR** con código 48989 y **DARWIN DARIO ESPINOZA SAQUICELA** con código 49883 que denuncian su trabajo de titulación denominado: **“DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE PLACAS VEHICULARES”**, previa a la obtención del Grado de Ingeniero en Sistemas y Telemática. El Consejo de Facultad acoge el informe de la Junta Académica que considera que la documentación cumple con las normas legales y reglamentarias y avala la aprobación por parte del tribunal designado para la sustentación del diseño, y aprueba la denuncia de tesis. Designa como **Director al Ing. Kenneth Palacio Baus** y como miembros del Tribunal Examinador a los ingenieros Rubén Ortega López y Bolívar Méndez Rengel. El peticionario tiene un plazo equivalente a dos períodos académicos (semestres) para desarrollar y terminar su trabajo de titulación, a partir de la fecha de la finalización de sus estudios.

Cuenca, febrero 3 de 2015



UNIVERSIDAD DEL
AZUAY
FACULTAD DE
ADMINISTRACION
SECRETARIA

CONVOCATORIA

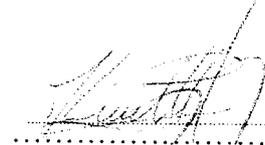
Por disposición de la Junta Académica de Ingeniería de Sistemas y Telemática, se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación : **"Desarrollo de un sistema de reconocimiento de placas vehiculares"**, presentado por los estudiantes Christian Antonio Salinas Escobar con código 48989 y Darwin Darío Espinoza Saquicela con código 49883, previa a la obtención del grado de Ingeniera en Ingeniero de Sistemas, para el día **VIERNES 9 DE ENERO DE 2015 A LAS 18H30.**

Cuenca, 6 de enero de 2015

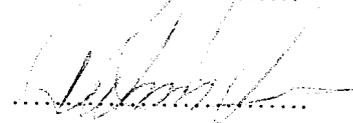


Dra. Jenny Rios Coello
Secretaria de la Facultad

Ing. Kenneth Palacio Baus



Ing. Rubén Ortega López



Eco. Bolívar Méndez Rengel



Comunicado

ESCUELA DE INGENIERIA DE SISTEMAS

Diseños de Tesis

Escuela de Ingeniería de Sistemas

Estudiante: Christian Antonio Salinas Escobar con código 48989 y Darwin Darío Espinoza Saquicela con código 49883.

Tema: "DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO DE PLACAS VEHICULARES"

Para: La obtención del título de Ingenieros de Sistemas

Director: Ing. Kenneth Palacio.

Tribunal: Ing. Rúben Ortega López.

Tribunal: Ing. Bolívar Méndez Rengel.

DIA: *VIERNES*

FECHA: *9 DE ENERO DE 2015*

HORA: *18h30*

Oficio Nro. 156-2014-DIST-UDA

Cuenca, 23 de Diciembre de 2014

Señor Ingeniero
Xavier Ortega Vázquez
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 23 de Diciembre del 2014, recibió el proyecto de tesis titulado "Desarrollo de un sistema de reconocimiento de placas vehiculares", presentada por los estudiantes Christian Salinas y Darío Espinoza, estudiantes de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Kenneth Palacio previo a la obtención del título de Ingeniero de Sistemas y Telemática.

La Junta solicita por su digno intermedio notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Kenneth Palacio y como miembros del Tribunal al Ing. Rubén Ortega e Ing. Bolívar Méndez.

Atentamente,



Ing. Marcos Orellana Cordero
Director Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay



ACTA

SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

1.1 Nombre del estudiante: Christian Antonio Salinas Escobar y Darwin Darío Espinoza Saquicela

1.1.1 Código 48989 y 49883

1.2 Director sugerido: : Ing. Kenneth Palacios Baus

1.3 Codirector (opcional): _____

1.4 Tribunal: Ing. Rubén Ortega López / Ing. Bolívar Méndez Rengel

1.5 Título propuesto: "Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares"

1.6 Resolución:

1.6.1 Aceptado sin modificaciones _____

1.6.2 Aceptado con las siguientes modificaciones:

En el contenido: Cambiar el punto
3.3 por: Implementación de Sistema
2.9: Eliminar los riesgos 2 y 3.

- Responsable de dar seguimiento a las modificaciones: : Ing. Kenneth Palacios Baus

1.6.3 No aceptado

- Justificación:

Ing. Kenneth Palacios Baus

Tribunal

Ing. Rubén Ortega López

Ing. Bolívar Méndez Rengel

Sr. Christian Salinas Escobar

Sr. Darwin Espinoza Saquicela

Dra. Jenny Ríos Coello
Secretario de Facultad

Fecha de sustentación: Viernes 9 de enero de 2015 a las 11h00



RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

1.1 Nombre del estudiante: Christian Antonio Salinas Escobar y Darwin Darío Espinoza Saquicela

1.1.1 **Código** 48989 y 49883

1.2 Director sugerido: Ing. Kenneth Palacio & Baus

1.3 Codirector (opcional):

1.4 Título propuesto: "Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares"

1.5 Revisores (tribunal): Ing. Rubén Ortega López / Ing. Bolívar Méndez Rengel

1.6 Recomendaciones generales de la revisión:

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
Línea de investigación				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	✓			
Título Propuesto				
2. ¿Es informativo?	✓			
3. ¿Es conciso?	✓			
Estado del arte				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?		✓		
5. ¿Describe la teoría en la que se enmarca el trabajo	✓			
6. ¿Describe los trabajos relacionados más relevantes?	✓			
7. ¿Utiliza citas bibliográficas?	✓			
Problemática y/o pregunta de investigación				
8. ¿Presenta una descripción precisa y clara?	✓			
9. ¿Tiene relevancia profesional y social?	✓			
Hipótesis (opcional)				
10. ¿Se expresa de forma clara?				
11. ¿Es factible de verificación?				
Objetivo general				
12. ¿Concuerda con el problema formulado?	✓			
13. ¿Se encuentra redactado en tiempo verbal infinitivo?	✓			
Objetivos específicos				



14. ¿Concuerdan con el objetivo general?	✓			
15. ¿Son comprobables cualitativa o cuantitativamente?	✓			
Metodología				
16. ¿Se encuentran disponibles los datos y materiales mencionados?	✓			
17. ¿Las actividades se presentan siguiendo una secuencia lógica?	✓			
18. ¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓			
19. ¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	✓			
Resultados esperados				
20. ¿Son relevantes para resolver o contribuir con el problema formulado?	✓			
21. ¿Concuerdan con los objetivos específicos?	✓			
22. ¿Se detalla la forma de presentación de los resultados?	✓			
23. ¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	✓			
Supuestos y riesgos				
24. ¿Se mencionan los supuestos y riesgos más relevantes?		✓		
25. ¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?	✓			
Presupuesto				
26. ¿El presupuesto es razonable?	✓			
27. ¿Se consideran los rubros más relevantes?	✓			
Cronograma				
28. ¿Los plazos para las actividades son realistas?	✓			
Referencias				
29. ¿Se siguen las recomendaciones de normas internacionales para citar?	✓			
Expresión escrita				
30. ¿La redacción es clara y fácilmente comprensible?	✓			
31. ¿El texto se encuentra libre de faltas ortográficas?	✓			

(*) Breve justificación, explicación o recomendación.



- Opcional cuando cumple totalmente,
- Obligatorio cuando cumple parcialmente y NO cumple.

Seguir las sugerencias planteadas sobre los riesgos,
el contenido. Los citas del documento de tesis
Real, los estudiantes deben referirse a la normativa de
la Universidad.

Ing. Kenneth Palacios Baus

Ing. Rubén Ortega López

Ing. Bolívar Méndez Rengel

Cuenca, Enero 22 de 2015

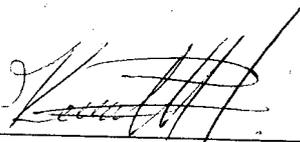
Señor Ingeniero
Xavier Ortega Vásquez, MBA
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION
UNIVERSIDAD DEL AZUAY

De mis consideraciones.

Reciba un cordial y atento saludo. El motivo de la presente es comunicarle que tras haber revisado conjuntamente con el tribunal designado para el efecto, la propuesta de titulación: *Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares*, realizado por Salinas Escobar Christian Antonio y Espinoza Saquicela Darwin Darío, egresados de la Escuela de Ingeniería de Sistemas y Telemática, se han sugerido cambios menores que serán presentados en la versión final del documento por parte de sus autores y que por lo tanto se recomienda su aprobación como proyecto de tesis.

Sin más que notificarle por el momento agradezco su atención a la presente.

Atentamente,



Ing. Kenneth S. Palacio Baus, M.Sc.
Docente Escuela de Ingeniería de Sistemas y Telemática

Ingeniero

Xavier Ortega Vásquez

Decano de la Facultad de Ciencias de la Administración

Presente

De mis consideraciones:

Reciba un cordial saludo, al tiempo que mediante la presente, me permito informarle que he revisado el diseño de tesis presentado por los estudiantes de la Escuela de Ingeniería en Sistemas y Telemática: Darwin Darío Espinoza Saquicela y Christian Antonio Salinas Escobar, con el tema *Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares*, elaborado como requisito previo para la obtención del título de Ingeniero de Sistemas y Telemática.

Al respecto, es importante mencionar que el diseño de tesis presenta una estructura teórica, metodológica y técnica coherente, para la obtención de un sistema computarizado para reconocimiento de placas vehiculares y un sistema integral de gestión de la información manejada en un parqueadero como complemento a la aplicación a desarrollar.

Por lo expuesto, emito informe favorable y recomiendo su aprobación.

Atentamente,



Ing. Kenneth S. Palacio Baus, M.Sc.

Docente de la Escuela de Ingeniería de Sistemas y Telemática

Cuenca, diciembre 18 de 2014

Ingeniero

Xavier Ortega Vásquez

Decano de la Facultad de Ciencias de la Administración

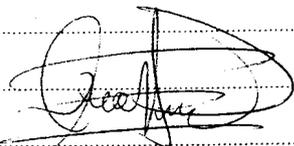
Ciudad

Señor Decano:

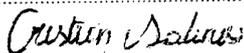
Nosotros, Darwin Darío Espinoza Saquicela y Christian Antonio Salinas Escobar, estudiantes de la Escuela de Ingeniería de Sistemas y Telemática con los códigos 49883 y 48989, respectivamente, presentamos a usted la denuncia del trabajo de titulación denominado "Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares". Adjuntamos también el informe favorable del Prof. Ing. Kenneth S. Palacio Baus, M.Sc., a quien proponemos como Director.

Solicitamos comedidamente se sirva autorizar el trámite correspondiente con el fin de que podamos cumplir con los requisitos reglamentarios para la graduación.

Atentamente,



Darío Espinoza Saquicela



Christian Salinas Escobar



UNIVERSIDAD DEL
AZUAY

DOCTORA JENNY RIOS COELLO SECRETARIA DE LA FACULTAD DE
CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, el Señor Darwin Darío Espinoza Saquicela registrado con código 49883 perteneciente a la Escuela de Ingeniería de Sistemas tiene aprobado más del 80% de pensum de estudios.

Cuenca, 12 de Diciembre de 2014

UNIVERSIDAD DEL
AZUAY
FACULTAD DE
ADMINISTRACION
SECRETARIA

Derecho 70353
vcf.-



UNIVERSIDAD DEL
AZUAY

DOCTORA JENNY RIOS COELLO SECRETARIA DE LA FACULTAD DE
CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, el Señor Christian Antonio Salinas Escobar registrado con código 48989
pertenece a la Escuela de Ingeniería de Sistemas tiene aprobado más del 80% de
pensum de estudios.

Cuenca, 12 de Diciembre de 2014

UNIVERSIDAD DEL
AZUAY
FACULTAD DE
ADMINISTRACION
SECRETARIA

Derecho 70461
vcf.-



Universidad del Azuay

Ingeniería de Sistemas y Telemática

Metodología de la Investigación

Guía para la presentación de la propuesta de proyecto de titulación

1. Datos generales

1.1 Nombres de los estudiantes:

Salinas Escobar Christian Antonio.

1.1.1 Código: 48989

1.1.2 Contacto:

Teléfono convencional: 4024182 Celular: 0984724200

Correo electrónico: ua048989@uazuay.edu.ec.

Espinoza Saquicela Darwin Darío.

1.1.3 Código: 49883

1.1.4 Contacto:

Teléfono convencional: 2243924 Celular: 0992896013

Correo electrónico: ua049883@uazuay.edu.ec

1.2 Director sugerido: Palacio Baus, Kenneth: Ingeniero en Electrónica.

1.2.1 Contacto: kpalacio@uazuay.edu.ec

1.3 Co-director sugerido: Mogrovejo, Ítalo: Ingeniero en Electrónica.

1.3.1 Contacto: imogrovejo@uazuay.edu.ec

1.4 Asesor metodológico: (opcional). Responsable del seguimiento de procesos de investigación.

1.5 Tribunal designado: (de acuerdo a la normativa interna de cada Facultad).

1.6 Aprobación: fecha de Junta Académica y fecha de Consejo Facultad.

1.7 Línea de Investigación de la carrera: señala la línea (declarada por la carrera), en la que se enmarca la investigación.

1.7.1 Código UNESCO

3311 01

1.7.2 Tipo de trabajo:

Investigación e Implementación de un Prototipo

a) Proyecto integrador (*capstone project*)

b) Investigación formativa

1.8 Área de estudio:

Tratamiento Digital de Señales, Telecomunicaciones, Microcontroladores.

1.9 Título propuesto: "Desarrollo de un Sistema de Reconocimiento de Placas Vehiculares".

1.10 Subtítulo: (opcional) "LPR UDA"

1.11 Estado del proyecto:

Un sistema capaz de reconocer placas vehiculares presenta como factor principal la identificación unívoca de un vehículo y la obtención de información relacionada al mismo [1]. Los beneficios que aporta un sistema de reconocimiento de placas vehiculares pueden ser relacionados principalmente a mejoras en la seguridad en diversas tareas, tales como:

- Parqueaderos sin requisitos de seguridad [2]. Estos parqueaderos establecen tarifas para calcular el costo por el tiempo de uso, emiten tickets sencillos que solo detallan una fecha y hora de entrada. Al no tener requisitos de seguridad, la posibilidad de fraude es alta. Como se describe a través del siguiente escenario:

Una persona con un vehículo *A* obtiene un ticket *T_a* en el que se detalla su hora y fecha de entrada. Tiempo después la misma persona regresa al mismo parqueadero con un vehículo *B* y obtiene un ticket *T_b*; se ha dado cuenta que podrá retirar del parqueadero su vehículo *A* con menor costo

usando el ticket T_b , ya que T_a fue expedido antes que T_b . La persona opta por realizar fraude, y usará el ticket T_b para retirar su vehículo A.

El proceso puede ser repetido. Ahora la persona regresa con su vehículo A y obtiene un nuevo ticket T_a que usa para retirar su vehículo B. Esto generaría enormes pérdidas al parqueadero, ya que no se estaría controlando de manera real cuáles son los vehículos y usuarios que usan el servicio y en qué cantidad.

- Parqueaderos con requerimientos de seguridad [2]. La ventaja aquí es notable. Se puede gestionar quién tiene acceso y quién no.
- Cobros [2]. En lugares semejantes a los peajes. Además si el sistema de reconocimiento de placas vehiculares es enlazado a una base de datos que contenga datos relacionados de los usuarios, se podrían realizar cobros en línea, notificaciones por correo electrónico, por ejemplo.
- Áreas con acceso restringido [2]. En lugares protegidos, como centros históricos, un sistema de reconocimiento de placas vehiculares podría gestionar el acceso de vehículos a dichos lugares dependiendo de un horario programado.

2. Contenido

2.1 Motivación de la investigación:

La aplicación de técnicas de procesamiento digital de imágenes en muchas actividades de la vida contemporánea, se ha visto incrementada notablemente en la actualidad. La reducción de los costos de producción de dispositivos de adquisición como cámaras digitales, y el mejoramiento paulatino de sus características han atraído la atención de los consumidores y al mismo tiempo, de investigadores en el área de procesamiento y análisis digital de imágenes [3], quienes aprovechan las robustas plataformas de hardware existentes hoy en día, y que permiten el desarrollo de novedosas aplicaciones, tanto en terminales fijos como móviles.

Los sistemas de tratamiento digital de imágenes poseen típicamente una cámara como "sensor universal" [3], el cual define ventajas como:

- Fácil reconfiguración.
- Costos de hardware reducidos: actualmente es fácil conseguir cámaras digitales de resoluciones aceptables entre 2 megapíxeles en adelante.

- **Facilidad de programación:** para desarrolladores de aplicaciones de visión por computador, y la disponibilidad de librerías de acceso abierto como OpenCV¹.

Las aplicaciones de tratamiento digital de imágenes tratan de emular y realizar tareas simples del sistema de visión humano, llegando incluso a superarlo en tareas muy específicas [3].

En el contexto de este proyecto, se plantea el uso de herramientas de visión por computador para el reconocimiento de placas vehiculares. Esta aplicación ha sido ampliamente utilizada en sistemas de control y análisis de tráfico vehicular a nivel mundial, principalmente por las ventajas que incorpora a los sistemas que hacen uso de ella [2]:

- **Reducción de costos.** Debido a los bajos costos de los componentes de hardware y herramientas de software involucradas, así como también debido a la automatización y consecuente sustitución del elemento humano en ciertas tareas.
- **Aumento de la seguridad.** Los vehículos requeridos o buscados por la policía pueden ser encontrados de manera unívoca.
- **Ganancias exactas.** Los servicios tarifas vinculados al uso de vehículos pueden beneficiarse del uso de sistemas de identificación basados en imágenes digitales.

Con el presente proyecto, se pretende impulsar el uso de herramientas de procesamiento de imágenes en el entorno de un sistema automatizado de control, monitoreo y gestión de parqueaderos, el cual constituirá un sólido punto de partida que sustentará el desarrollo de aplicaciones más complejas en el futuro.

2.2 Problemática:

En general, los sistemas destinados a gestionar el acceso de vehículos a parqueaderos tarifados, carecen de un mecanismo de identificación automático sus clientes, y normalmente requieren la presencia de un operario humano de manera constante. Adicionalmente, la mayoría de sistemas de gestión existentes no poseen herramientas de manejo de la información relacionada a los vehículos y sus usuarios. Un ejemplo concreto se presenta en el campus de la Universidad del Azuay, cuyo acceso al parqueadero para profesores es gestionado por una

¹ www.opencv.org

persona encargada de verificar que el vehículo posea una etiqueta que le autoriza el ingreso.

En este contexto, es importante mencionar también que el uso de la información recopilada por las videocámaras de vigilancia en las ciudades es muy limitado, ya que principalmente están destinadas a tareas de monitoreo y registro de eventos. Muchos de estos sistemas únicamente almacenan las grabaciones sin analizar la información asociada al contenido del video. Aprovechando las infraestructuras de video vigilancia implementadas en la actualidad, es posible integrar sistemas de Procesamiento Digital de Imágenes que permitan realizar el análisis de contenido de los videos para la extracción de información.

2.3 Resumen:

Hoy en día, el rápido aumento de vehículos en las grandes ciudades ha propiciado el incremento de la infraestructura vial y consecuentemente, la necesidad de espacios de aparcamiento en busca de solucionar los múltiples problemas causados por el excesivo tráfico. Los sistemas para el reconocimiento de Placas de Vehículos han sido desarrollados para automatizar las tareas de monitoreo, control y regulación de la movilidad, utilizando la tecnología de las cámaras de video vigilancia y varias técnicas de Procesamiento de Imágenes [4], particularmente, en sistemas de control de infracciones.

En el caso de sistemas de administración de parqueaderos, el Reconocimiento de Placas Vehiculares (LPR) facilita la gestión de ingreso y salida de vehículos, agilizando las tareas de registro y facturación. Considerando que puede llegar a manejarse un gran número de vehículos, se espera que el elemento humano sea propenso a generar errores [1] y que un sistema automático sea más eficiente y rentable. Además de mejorar la calidad del servicio de un parqueadero, un sistema LPR contribuye a la seguridad de los vehículos ya que al reconocer los caracteres de una placa, se puede vincular la información del propietario del vehículo y verificar si está autorizado o no, para ingresar al establecimiento donde se haya implementado el sistema.

La complejidad de los sistemas que adoptan las técnicas LPR es alta, ya que la tarea de detectar una placa no siempre es simple, dadas las diferentes situaciones del medio ambiente en donde se adquieran las imágenes. La robustez del sistema deberá contemplar situaciones tales como: color del vehículo y color de la placa similares; imágenes con bajo nivel de contraste y; condiciones en donde la iluminación es muy pobre [4]. Por este motivo, se evaluarán diferentes enfoques

y técnicas, de tal modo que se puedan comparar y determinar la o las mejores técnicas, o su combinación para obtener resultados satisfactorios.

2.4 Indagación exploratoria y base conceptual

Actualmente, en el mercado, podemos observar una gran cantidad de aplicaciones y servicios que utilizan tecnología de cámaras y también técnicas de tratamiento de imágenes para resolver problemas diarios como el reconocimiento de un patrón en particular. En todas las ciudades del mundo, se han presentado problemas de tráfico vehicular, accidentes y conductores que infringen la ley [5]. Este tipo de problemas han impulsado la creación de sistemas de identificación y seguimiento de vehículos y su uso se ha extendido incluso a la identificación de vehículos robados [6]. Es común también el uso de este tipo de sistemas en ambientes cerrados, donde es necesario automatizar accesos y/o emitir tickets de identificación para los usuarios del servicio. [6]

Muchas investigaciones dirigidas a la identificación de vehículos se han enfocado en la extracción y reconocimiento de caracteres de placas vehiculares. [1] Lotufo, Morgan y Jhonson [7] proponen un reconocimiento de placas vehiculares usando reconocimiento óptico de caracteres. Jhonson y Bird [8] proponen un aprendizaje guiado para seguimiento de bordes y emparejamiento con plantillas para identificación automática de vehículos. Fahmy [9] propuso una red neuronal BAM (Bidirectional Associative Memory) para el reconocimiento de placas mientras que Ter Brugge y Helmholf [10] proponen un sistema de reconocimiento de placas que usa lógica difusa y redes neuronales.

La compleja tarea del reconocimiento de placas vehiculares, se matiza con diversidad de formatos de placas encontrados a nivel mundial, incluso aquellos utilizados dentro de una misma jurisdicción para diferenciar vehículos en sus diferentes categorías (motos, vehículos oficiales, diplomáticos, emergencias, etc). Además, las condiciones cambiantes del clima y la velocidad de desplazamiento de los automotores contribuyen a la obtención de diferentes niveles de iluminación presentes durante la etapa de adquisición de las imágenes. Hoy en día, diferentes técnicas de LPR continúan siendo evaluadas y muchas otras se han propuesto [11].

El proceso realizado por un sistema LPR está compuesto principalmente por tres fases [4] [11]:

1. Detección de Placa Vehicular (LPD, Licence Plate Detection)

2. Segmentación de caracteres

3. Reconocimiento de caracteres

Comúnmente, la primera fase es considerada la más importante. Existen varios estudios que resuelven el problema de esta fase, por ejemplo, haciendo uso de métodos de *extracción de bordes*, *diferencias de contraste* de las placas vehiculares, *uso de plantillas*, entre otros.

Los algoritmos empleados en el reconocimiento de Placas Vehiculares deben tener la capacidad de operar lo suficientemente rápido, con el fin de cubrir las necesidades de los Sistemas Inteligentes de Transporte (ITS). Al hablar de procesamiento en tiempo real se hace referencia a las operaciones inmediatas de manera que no se puede perder ningún objeto de interés que se mueve a través de una escena. Sin embargo, con el crecimiento exponencial de la capacidad de procesamiento computacional, las últimas tecnologías en reconocimiento de placas vehiculares operan en menos de 50ms, procesando más de 20 fotogramas por segundo para el caso de videos [11].

A continuación se presenta los principales métodos empleados para la extracción de la región de una placa vehicular en una imagen digital:

a) Procesamiento de Imagen Binaria

Esta técnica se basa en combinaciones de estadísticas de borde y morfología matemática. (Utilizada para extraer componentes útiles de imágenes, para la representación y descripción de forma de regiones), como se documenta en [12]. El método, se fundamenta en el principio de que el cambio de brillo en la región de la placa, es más remarcable y más frecuente que en cualquier otro lugar de la imagen. Una desventaja de este método, es que puede llegar a ser demasiado sensible a los bordes no deseados, mostrándolos como una elevada magnitud, por ejemplo, en la región del radiador en la vista frontal del vehículo. A pesar de esta desventaja, si este método se combina con una técnica para eliminar los bordes no deseados de las imágenes procesadas, la tasa de extracción de la región de la placa es relativamente alta y además, la técnica es muy rápida en comparación con otros métodos, como se detalla en [11].

En [13] el modelo conceptual del algoritmo propuesto, se basa en la operación morfológica llamada "*top-hat*", que es capaz de localizar objetos pequeños con diferencias de brillo. Este algoritmo, con una tasa de detección de 80%, es altamente dependiente de la distancia entre la cámara y el vehículo, ya que las

operaciones morfológicas se refieren a la dimensiones de los objetos binarios. Una aplicación similar se describe en [14] con un nivel de precisión del 93%.

Por otra parte existen también algoritmos híbridos de extracción de Placas Vehiculares como los que se implementa en [15], **basados en estadísticas de borde y la morfología** para los sistemas de emisión de tickets para vigilancia de carreteras. Este enfoque se puede dividir en la siguientes cuatro secciones:

- La detección de bordes verticales.
- Análisis estadístico de bordes.
- Ubicación jerárquica basada en Placas Vehiculares.
- Extracción de Placas basado en la morfología.

En la precisión promedio de localización de una placa vehicular, empleando estos algoritmos híbridos, se obtiene una impresionante tasa del 99,6%, ya que sus autores mediante diferentes pruebas obtuvieron 9786 aciertos de las 9825 imágenes procesadas.

Los sistemas LPR implementan visión artificial o también llamada visión por computadora, al igual que técnicas de tratamiento digital de señales (DSP), en específico se enfoca en el tratamiento digital de imágenes. Un punto relevante en la identificación de una placa vehicular es el reconocimiento del contorno de la misma, y es aquí donde entra en juego el tratamiento digital de las imágenes. Para lograr la identificación del contorno es típicamente usada la transformada de Hough, la cual permite la detección de figuras dentro de una imagen.

Para una implementación en ambientes reales un sistema LPR requiere de hardware y software [1] Es aquí donde las 3 fases mencionadas anteriormente pueden ser más detalladas, como se indica en la figura 1.

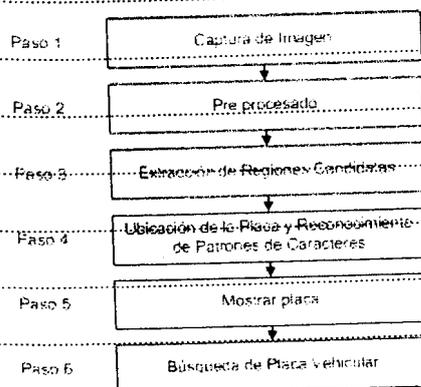


Figura 1. Software de un Sistema de Reconocimiento de Placas Vehiculares [1]

En el paso 1, la imagen frontal del vehículo es capturada en tiempo real por una cámara, para ser pre procesada en el paso 2. En el tercer paso se extraen regiones candidatas, provenientes de la imagen pre procesada, y acorde a las regulaciones para el formato de placas vehiculares. En el paso 4 se obtiene los caracteres de la placa vehicular a partir de las regiones candidatas extraídas en el paso anterior, para esto se pueden usar varios métodos como por ejemplo redes neuronales multi capa. En el quinto paso, la ubicación e identificación de la placa son mostradas al usuario en una pantalla. Finalmente, en el paso 6 se comprueba si el vehículo está registrado en el sistema; en caso de ser correcto se le da acceso automáticamente, caso contrario se requiere apertura manual.

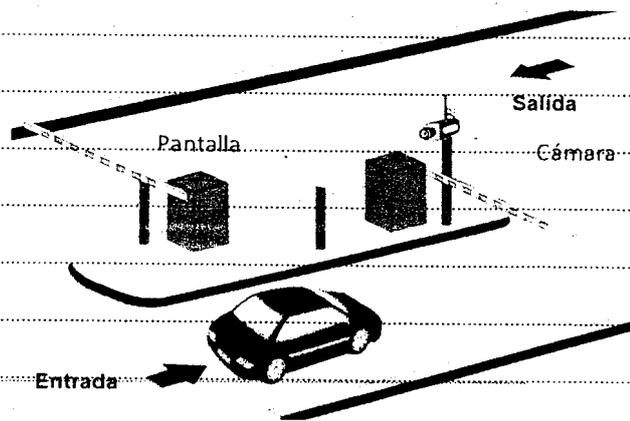


Figura 2 Prototipo para una implementación de un ingreso a parqueadero automatizado

En cuanto al hardware requerido para una implementación de un LPR, se puede apreciar en la Figura 2 la aparición de ciertos elementos como la cámara o las barreras de entrada. Como en muchos otros sistemas de tratamiento de imagen, la cámara juega un papel fundamental.

En la implementación que sugiere este proyecto, el procedimiento a seguirse es el siguiente:

1. El vehículo ingresa al parqueadero y se ubica frente a la barrera de ingreso, donde es visible por la cámara.
2. La cámara, conectada a un computador, captura las imágenes en tiempo real para que sean procesadas por el sistema LPR.
3. El sistema LPR reconocerá la placa vehicular y con la ayuda de información adicional decidirá si el vehículo puede ingresar o no.
4. Se muestra en la "Pantalla" un mensaje de bienvenida en caso de que el vehículo consiga acceso al parqueadero o un mensaje de error en caso de que no sea autorizado a ingresar.

La barrera de ingreso se abrirá o cerrará dependiendo de las decisiones tomadas por el sistema. Además se cuenta botones para operaciones de abrir y cerrar de forma manual.

2.5 Objetivo general:

Implementar un sistema de gestión y automatización del acceso a un parqueadero, mediante el reconocimiento de placas vehiculares a través de una cámara como mecanismo de identificación de clientes.

2.6 Objetivos específicos:

- Desarrollar un sistema capaz de reconocer placas vehiculares.
- Desarrollar un sistema de manejo de parqueaderos integrando técnicas de reconocimiento de placas vehiculares para la identificación de usuarios.
- Diseñar e implementar un prototipo que muestre la automatización del acceso a un parqueadero utilizando una cámara IP y una barrera móvil controlada por un motor activado mediante una tarjeta Arduino UNO.
- Generar una interfaz de administración de usuarios autorizados a acceder al parqueadero.
- Generar una interfaz web para el acceso remoto a la administración de usuarios del parqueadero.

2.7 Metodología:

La secuencia de actividades que se considera para cumplir con los objetivos de este proyecto de titulación es la siguiente:

1. Indagación y adquisición de bases conceptuales. Es en esta fase en donde se debe adquirir los conocimientos necesarios para emprender el camino a la realización del proyecto propuesto, en donde se verán involucrados temas como los algoritmos y las técnicas utilizadas para el reconocimiento de las placas vehiculares.

Estudio de los métodos de tratamiento de imágenes para la detección de placas, como primer paso en el proceso de la identificación de vehículos.

Evaluar diferentes técnicas para el reconocimiento de caracteres, como técnicas basadas en redes neuronales y reconocimiento de patrones.

2. Definir los elementos que integrarán la Plataforma para la Administración de un Parqueadero. Previo a un análisis entre las diferentes alternativas tecnológicas, se debe seleccionar la opción más adecuada para implementar el sistema. Entre estos elementos se tiene:

Red de Datos

Protocolos de Red

Base de Datos

Lenguaje de Programación

Librerías para el tratamiento digital de imágenes

Aplicaciones Web

Cámara IP

3. Desarrollo del Sistema que sea capaz de Reconocer Placas Vehiculares.

4. Pruebas de laboratorio.

A medida que se desarrolla el sistema, se realizarán experimentos y pruebas de verificación, para lo cual se recopilará un número considerable de imágenes de vehículos diferentes y bajo condiciones de iluminación y ubicación de la cámara, totalmente controladas. Estas imágenes permitirán determinar la mejor técnica a aplicarse en el sistema final, así como parámetros técnicos del proceso de adquisición de imágenes.

5. Implementación del prototipo de automatización. El prototipo exhibirá la automatización del acceso a un parqueadero y será capaz de reconocer placas vehiculares reales a través de una cámara IP. Así también el prototipo tendrá la capacidad de mover una barrera para dar acceso a los vehículos autorizados mediante un motor controlado por una tarjeta Arduino UNO conectada a un computador, que al mismo tiempo recibirá las imágenes de la cámara IP y las procesará mediante el aplicativo de reconocimiento de placas vehiculares.

2.8 Alcances y resultados esperados:

Desarrollar un prototipo de un parqueadero automatizado para el acceso de vehículos que serán identificados mediante el reconocimiento de su placa. El sistema será implementado en una maqueta. Se espera un nivel de identificación correcta de vehículos de al menos 90% para lo cual se realizarán diferentes pruebas con placas reales.

2.9 Supuestos y riesgos:

- ✓ Que el porcentaje de eficacia de al menos una de las técnicas evaluadas para el sistema no supere el 90%.
- ✓ Carencia de referencias bibliográficas referentes a las técnicas de extracción y reconocimiento de caracteres.
- ✓ Conexiones de red deficientes que no permitan una velocidad aceptable de transmisión de datos entre la cámara IP y el servidor de tratamiento de imagen.
- ✓ Condiciones ambientales extremas, en donde no se pueda obtener información de las imágenes capturadas de las placas vehiculares.
- ✓ El tiempo real de Desarrollo del Sistema, puede extenderse más allá de lo planificado, según el cronograma de Actividades.
- ✓ El presupuesto muy limitado, puede influir en el desempeño del sistema, ya que se requiere de componentes hardware con suficiente velocidad de procesamiento (pc), y resolución aceptable (cámara).

2.10 Presupuesto:

Rubro-Denominación	Costo USD (detalle)	Justificación ¿para qué?
Motor	\$ 30	Para simular la barrera de acceso al parqueadero. El Motor es parte del prototipo.

Arduino UNO	\$ 40	Controla la operación del servomotor en el prototipo.
Implementos varios (resistencias, leds, transistores, etc.)	\$ 20	Necesarios para armar el circuito que controla el prototipo.
Cámara IP	\$ 90	Instalación en el acceso principal del parqueadero, es donde el sistema permite o no el ingreso.
Computadora	\$500	Dedicada al procesamiento de las imágenes capturadas por la cámara IP. Cuenta con sistema operativo Windows 7 y se hace uso de las herramientas: Visual Basic 2010, Qt Project y las librerías de visión por computador, OpenCV.

2.11 Financiamiento: El financiamiento para la realización de este proyecto será propio; usaremos fondos propios para sustentar los gastos involucrados.

2.12 Esquema tentativo: Incluye el listado de temas y subtemas que nos guiarán en el desarrollo del trabajo. Este esquema podrá ser revisado durante el desarrollo del trabajo. Debe estar en relación directa con los objetivos específicos.

1. INTRODUCCIÓN

- 1.1 Antecedentes
- 1.2 Planteamiento del Problema
- 1.3 Justificación
- 1.4 Definición de Objetivos

2. ANÁLISIS VEHICULAR A TRAVÉS DE IMÁGENES

- 2.1 Introducción
- 2.2 Visión por computadora en el análisis del tráfico vehicular.
- 2.3 Técnicas para la detección de placas
- 2.4 Algoritmos y técnicas de reconocimiento de caracteres.

3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

- 3.1 Análisis y Planificación de Requisitos
- 3.2 Diseño del Sistema
- 3.3 Especificaciones Hardware del Sistema

3.4 Especificaciones Software del Sistema

4. PRUEBAS Y EVALUACIÓN DEL SISTEMA

4.1 Modelo de evaluación del sistema

4.2 Evaluación cuantitativa

4.3 Evaluación cualitativa

5. CONCLUSIONES Y TRABAJO FUTURO

6. REFERENCIAS BIBLIOGRAFICAS

7. ANEXOS

7.1 Manual del Usuario del Sistema

7.2 Diagramas Electrónicos del prototipo.

7.3 Código Fuente del Sistema



2.13 Cronograma:

CRONOGRAMA LPR UDA

3	INTRODUCCION	18-01-2015
3.1	Antecedentes	18-01-2015
3.2	Planteamiento del Problema	18-01-2015
3.3	Justificación	18-01-2015
3.4	Objetivos e Impactos	18-01-2015
2	ANÁLISIS DE LA BIBLIOTECA DE MATERIAS	05-02-2015
2.1	Introducción	05-02-2015
2.2	Motivos para la computación en el análisis de tráfico vehicular	05-02-2015
2.3	Formas para la detección de tráfico	05-02-2015
2.4	Requisitos de almacenamiento de imágenes	05-02-2015
3	ANÁLISIS DE IMPLANTACIÓN DE LA RED DE CÁMARA	18-01-2015
3.1	Requisitos de instalación de cámaras	18-01-2015
3.2	Requisitos de software	18-01-2015
3.3	Requisitos de hardware	18-01-2015
3.4	Requisitos de conectividad	18-01-2015
3.5	Requisitos de almacenamiento	18-01-2015
3.6	Requisitos de seguridad	18-01-2015
3.7	Requisitos de mantenimiento	18-01-2015
3.8	Requisitos de capacitación	18-01-2015
3.9	Requisitos de documentación	18-01-2015
3.10	Requisitos de presupuesto	18-01-2015
3.11	Requisitos de evaluación	18-01-2015
3.12	Requisitos de seguimiento	18-01-2015
3.13	Requisitos de cierre	18-01-2015
3.14	Requisitos de mantenimiento	18-01-2015
3.15	Requisitos de actualización	18-01-2015
3.16	Requisitos de mejora	18-01-2015
3.17	Requisitos de optimización	18-01-2015
3.18	Requisitos de escalabilidad	18-01-2015
3.19	Requisitos de flexibilidad	18-01-2015
3.20	Requisitos de interoperabilidad	18-01-2015
3.21	Requisitos de portabilidad	18-01-2015
3.22	Requisitos de reutilización	18-01-2015
3.23	Requisitos de accesibilidad	18-01-2015
3.24	Requisitos de privacidad	18-01-2015
3.25	Requisitos de seguridad	18-01-2015
3.26	Requisitos de confiabilidad	18-01-2015
3.27	Requisitos de disponibilidad	18-01-2015
3.28	Requisitos de sostenibilidad	18-01-2015
3.29	Requisitos de responsabilidad	18-01-2015
3.30	Requisitos de transparencia	18-01-2015
3.31	Requisitos de integridad	18-01-2015
3.32	Requisitos de autenticidad	18-01-2015
3.33	Requisitos de no repudio	18-01-2015
3.34	Requisitos de confidencialidad	18-01-2015
3.35	Requisitos de integridad	18-01-2015
3.36	Requisitos de disponibilidad	18-01-2015
3.37	Requisitos de sostenibilidad	18-01-2015
3.38	Requisitos de responsabilidad	18-01-2015
3.39	Requisitos de transparencia	18-01-2015
3.40	Requisitos de integridad	18-01-2015
3.41	Requisitos de autenticidad	18-01-2015
3.42	Requisitos de no repudio	18-01-2015
3.43	Requisitos de confidencialidad	18-01-2015
3.44	Requisitos de integridad	18-01-2015
3.45	Requisitos de disponibilidad	18-01-2015
3.46	Requisitos de sostenibilidad	18-01-2015
3.47	Requisitos de responsabilidad	18-01-2015
3.48	Requisitos de transparencia	18-01-2015
3.49	Requisitos de integridad	18-01-2015
3.50	Requisitos de autenticidad	18-01-2015
3.51	Requisitos de no repudio	18-01-2015
3.52	Requisitos de confidencialidad	18-01-2015
3.53	Requisitos de integridad	18-01-2015
3.54	Requisitos de disponibilidad	18-01-2015
3.55	Requisitos de sostenibilidad	18-01-2015
3.56	Requisitos de responsabilidad	18-01-2015
3.57	Requisitos de transparencia	18-01-2015
3.58	Requisitos de integridad	18-01-2015
3.59	Requisitos de autenticidad	18-01-2015
3.60	Requisitos de no repudio	18-01-2015
3.61	Requisitos de confidencialidad	18-01-2015
3.62	Requisitos de integridad	18-01-2015
3.63	Requisitos de disponibilidad	18-01-2015
3.64	Requisitos de sostenibilidad	18-01-2015
3.65	Requisitos de responsabilidad	18-01-2015
3.66	Requisitos de transparencia	18-01-2015
3.67	Requisitos de integridad	18-01-2015
3.68	Requisitos de autenticidad	18-01-2015
3.69	Requisitos de no repudio	18-01-2015
3.70	Requisitos de confidencialidad	18-01-2015
3.71	Requisitos de integridad	18-01-2015
3.72	Requisitos de disponibilidad	18-01-2015
3.73	Requisitos de sostenibilidad	18-01-2015
3.74	Requisitos de responsabilidad	18-01-2015
3.75	Requisitos de transparencia	18-01-2015
3.76	Requisitos de integridad	18-01-2015
3.77	Requisitos de autenticidad	18-01-2015
3.78	Requisitos de no repudio	18-01-2015
3.79	Requisitos de confidencialidad	18-01-2015
3.80	Requisitos de integridad	18-01-2015
3.81	Requisitos de disponibilidad	18-01-2015
3.82	Requisitos de sostenibilidad	18-01-2015
3.83	Requisitos de responsabilidad	18-01-2015
3.84	Requisitos de transparencia	18-01-2015
3.85	Requisitos de integridad	18-01-2015
3.86	Requisitos de autenticidad	18-01-2015
3.87	Requisitos de no repudio	18-01-2015
3.88	Requisitos de confidencialidad	18-01-2015
3.89	Requisitos de integridad	18-01-2015
3.90	Requisitos de disponibilidad	18-01-2015
3.91	Requisitos de sostenibilidad	18-01-2015
3.92	Requisitos de responsabilidad	18-01-2015
3.93	Requisitos de transparencia	18-01-2015
3.94	Requisitos de integridad	18-01-2015
3.95	Requisitos de autenticidad	18-01-2015
3.96	Requisitos de no repudio	18-01-2015
3.97	Requisitos de confidencialidad	18-01-2015
3.98	Requisitos de integridad	18-01-2015
3.99	Requisitos de disponibilidad	18-01-2015
3.100	Requisitos de sostenibilidad	18-01-2015

2.14 Referencias:

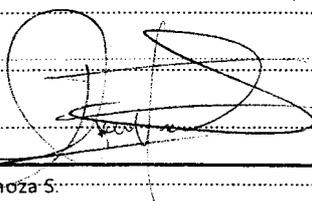
- [1] T. Sirithinaphong y K. Chamnongthai, The Recognition of Car Licence Plate For Automatic Parking System, Thailand: King Mongkut's University of Technology Thonburi, 1999.
- [2] S. Draghici, A neural network based artificial vision system for licence plate recognition, Wayne State University.
- [3] T. Klinger, Image Processing with LabVIEW and IMAQ Vision, New Jersey: Prentice Hall Professional, 2003.
- [4] K. JaeDo, H. YoungJoon y H. Hernsoo, License Plate Detection Using Topology of Characters and Outer Contour, Korea: Soongsil University.
- [5] H. Ching- Tang, C. Liang-Chun, H. Kuo-Ming y H. Hsieh-Chang, A Real-time Mobile Vehicle License Plate Detection and Recognition for vehicle monitoring and management, Taiwan: Tamkang University.
- [6] S. N. H. Sheikh Abdullah, License Plate Recognition using Multilayer Neural Networks, IEEE, 2006.
- [7] R. Lotufo, A. Morgan y A. Jhonson, Automatic Number-Plate Recognition, Proceedings of the IEE Colloquium on Image Analysis for Transport Applications, 1990.
- [8] A. Jhonson y B. Bird, Number-plate matching for Automatic Vehicle Identification, IEE Colloquium on Electronic Image and Image Processing in Security and Forensic, 1990.
- [9] M. Fahmy, Automatic Number-plate recognition: Neural Network Approach, Proceedings of VNIS'94 Vehicle Navigation and Information System Conference, 1994.
- [10] J. Nijhuis, M. Ter Brugge, K. Helmholt, J. Pluim, L. Spaanenburg, R. Venema y M. Westenberg, Car License Plate Recognition with Neural Networks and Fuzzy Logic, IEEE International Conference on Neural Networks, 1995.
- [11] Christos-Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, , Ioannis D. Psoroulas, Vassili Loumos y Eleftherios Kayafas, License Plate Recognition From Still Images and Video Sequences: A Survey, IEEE, 2008.
- [12] R. Radha y C.P. Sumathi, A novel approach to extract text from license plate of vehicles, 2012.
- [13] M. F. G. M y A. J. L., New methods for automatic reading of VLP's (Vehicle License Plates), SPPRA, 2002.
- [14] C. Wu, L. C. On, C. H. Weng y T. S. Kuan, A Macao license plate recognition system, Guangzhou, China: Learn Cybern, 2005.
- [15] H. B y C. L, A hybrid license plate extraction, ICPR, 2004.

2.15 Anexos:

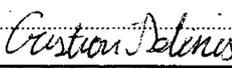
Laboratorio LPR: contiene una base de datos, con una gran cantidad de imágenes y videos de Placas Vehiculares que pueden ser utilizados para evaluar el sistema propuesto

<http://www.medialab.ntua.gr/research/LPRdatabase.html>

2.16 Firma de responsabilidad (estudiantes)

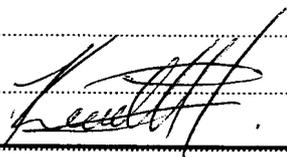


Darío Espinoza S.



Christian Salinas E.

2.17 Firma de responsabilidad (director sugerido)



Inq. Kenneth Palacio B.
Director de Tesis.