

**Diseño e implementación de un software de reconocimiento de
placas vehiculares en tiempo real**

Luis Manuel Márquez Rodríguez

**Proyecto de Grado presentado como requisito para optar al título de
Ingeniero de Sistemas**

Director

Rubén Darío Ordóñez Mantilla

**Universidad Nacional Abierta y a Distancia, UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería (ECBTI)
Ingeniería de sistemas
CEAD Cúcuta
2018**

Tabla de Contenido

	pág.
INTRODUCCIÓN	2
1. OBJETIVOS	4
1.1 Objetivo General	4
1.2 Objetivos específicos	4
2. PLANTEAMIENTO DEL PROBLEMA.....	5
2.1 Definición del Problema	5
2.2 Formulación del problema	6
3. JUSTIFICACIÓN	7
4. MARCO TEÓRICO	9
4.1 Antecedentes de la Investigación	9
4.2 Marco Conceptual.....	12
4.2.1 Visión humana	12
4.2.2 Visión por computador	13
4.3 Marco Conceptual.....	15
4.4 Marco Legal.....	17
5. METODOLOGÍA.....	19
5.1 Tipo de Investigación.....	19
5.2 Fuentes de Información	19
5.3 Alternativa de trabajo de grado	19
5.4 Instrumentos.....	20
5.5 Procedimiento.....	28
5.6 Cronograma	30
6. DESARROLLO DEL PROYECTO.....	31
6.1 Descripción del Proyecto	31
6.2 Resultados y Análisis de los resultados	36
Conclusiones.....	38

Referencias 39

Lista de Tablas

	pág.
Tabla 1. <i>Tipo de placas en Colombia</i>	15
Tabla 2. <i>Información general de la Cámara Logitec 930e</i>	20
Tabla 3. <i>Especificaciones de cámara Web LOGITECH 930e</i>	21
Tabla 4. <i>Especificaciones técnicas del equipo informático (computador de escritorio)</i>	21
Tabla 5. <i>Clases y métodos básicos usados</i>	27
Tabla 6. <i>Cronograma</i>	30
Tabla 7. <i>Requisitos del Sistema para Java</i>	31
Tabla 8. <i>Requisitos mínimos del Sistema para Netbeans</i>	32
Tabla 9. <i>Requerimientos funcionales</i>	32

Lista de Figuras

	pág.
<i>Figura 1.</i> Visión humana	13
<i>Figura 2.</i> Procesos de la visión por computador	14
<i>Figura 3.</i> Cámara Logitech 930e	20
<i>Figura 4.</i> IDE Netbeans	22
<i>Figura 5.</i> Página de descarga de Netbeans	23
<i>Figura 6.</i> Componentes de la Máquina Virtual Java	24
<i>Figura 7.</i> Instalación de la librería JavaMail	25
<i>Figura 8.</i> Adición de la librería al proyecto en Netbeans	26
<i>Figura 9.</i> Formulario de Inicio de Sesión	33
<i>Figura 10.</i> Interfaz de resultados de búsqueda	34
<i>Figura 11.</i> Prueba de captura 1	35
<i>Figura 12.</i> Prueba de captura 2	35
<i>Figura 13.</i> Prueba de captura 3	36
<i>Figura 14.</i> Placa en escala de grises	37
<i>Figura 15.</i> Resultado de la detección de bordes	37

Resumen

El presente documento tiene como objetivo describir el desarrollo de un software para la captura y reconocimiento de placas vehiculares, el cual puede implementarse como apoyo a sistemas de vigilancia en parqueaderos y en la vía como complemento al sistema de tránsito. Metodológicamente se desarrolla una investigación de tipo descriptiva, las fases de desarrollo se realizan de acuerdo al ciclo de vida del software. Los resultados permiten observar que para obtener imágenes de mejor calidad, se requieren cámaras y monitores de altas resoluciones.

Palabras claves: Reconocimiento Óptico de Caracteres, Placa Vehicular, Resolución, Captura, Vigilancia, Seguridad

INTRODUCCIÓN

El presente Proyecto tiene como objetivo desarrollar un sistema de reconocimiento de placas vehiculares a partir de imágenes en tiempo real desarrollando una interfaz gráfica bajo Java y una librería de reconocimiento de caracteres, como apoyo a labores de vigilancia, búsqueda y ubicación del vehículo, pues uno de los delitos más comunes es el hurto de automotores, los cuales son destinados para la venta dentro o fuera del país; desmantelamiento y venta de autopartes; la comisión de delitos como secuestro, robo, fraude, atentados con explosivos, entre otros. (El Espectador, 2018)

El uso de este tipo de software permite la captura de la imagen de la placa en tiempo real, apoyado en el uso de cámaras e instalado en diferentes puntos de una ciudad o región, incluso en parqueaderos públicos o privados, vehículos policiales (motos, carros y helicópteros), drones, arcos en peajes y zonas fronterizas suelen ser de gran ayuda a las autoridades, pues la información puede ser enviada a la institución encargada de la vigilancia de la ciudad y hacer un cruce de información en una base de vehículos robados (Cruz, 2012), que en caso de ser positivo facilita su ubicación y posterior captura. (Imagen Noticias, 2014)

La metodología empleada para el desarrollo del software de reconocimiento se basa en el diseño ágil, teniendo en cuenta la caracterización de empresas pequeñas dedicadas al cuidado, parqueo y vigilancia de automotores, en donde sus recursos limitados no permiten grandes inversiones en tecnología, investigación, innovación y desarrollo, razón por la cual se acude a software de licencias públicas y de código abierto, por otro lado, a la portabilidad de los mismos, lo cual permite la instalación en equipos con diferentes sistemas operativos y de baja capacidad de memoria.

El trabajo se desarrolla en siete capítulos, el primero de ellos comprende los objetivos, posteriormente el planteamiento del problema, el tercero la justificación, el cuarto las bases teóricas, conceptuales y legales, el quinto hace referencia a la metodología de investigación; en el sexto capítulo se desarrolla el proyecto, va desde la etapa de análisis hasta la prueba y mantenimiento del software (mejoras). Por último, se encuentran las conclusiones y recomendaciones.

1. OBJETIVOS

1.1 Objetivo General

Desarrollar un sistema de reconocimiento de placas vehiculares a partir de imágenes en tiempo real desarrollando una interfaz gráfica bajo Java y una librería de reconocimiento de caracteres, como apoyo a las actividades de vigilancia en zonas de parqueo y parqueaderos privados.

1.2 Objetivos específicos

- Seleccionar las herramientas, técnicas y materiales necesarios para el desarrollo del sistema de reconocimiento en tiempo real.
- Diseñar la interfaz gráfica que utilizará el sistema de reconocimiento de placas vehiculares en tiempo real.
- Realizar pruebas y evaluación del sistema de reconocimiento de placas vehiculares en tiempo real.

2. PLANTEAMIENTO DEL PROBLEMA

2.1 Definición del Problema

El hurto de vehículos es una actividad delictiva con un alto nivel de organización el cual halla su oportunidad de ocurrencia por la rentabilidad que ofrece en la venta de autopartes a precios más bajos en el mercado y de manera informal; la facilidad de legalización de las partes; la experiencia que adquieren los delincuentes respecto a las tecnologías aplicadas en la vigilancia, rastreo y ubicación de vehículos; así mismo por la informalidad en la vigilancia y en el servicio de parqueadero; áreas de mayor índice delictual y con vías de escape; la baja rigurosidad en las penas, la falta de cooperación y comunicación entre los países lo que permite su salida y libre tránsito en otras naciones, y por último, los vehículos suelen ser utilizados para cometer otros actos criminales (secuestro, extorsión, etc.). (Norza, Duarte, Castillo, & Torres, 2013)

Desde esta perspectiva, el uso de software especializado en la captura, registro y control de ingreso y salida de vehículos representa un apoyo importante, ya que permite brindar a los propietarios de estos un grado de certeza sobre dónde está su vehículo, la hora de ingreso y salida, etc.

Aunque existe gran variedad de opciones de este tipo de software en el mercado, el valor de estos es elevado lo que se observa como un obstáculo para establecimientos o empresas pequeñas, siendo este uno de los motivos por los cuales se presenta la necesidad de desarrollar un sistema en donde el licenciamiento de las librerías utilizadas no incremente los costos en la producción de dicho software, ni restrinja a pequeñas empresas acceder a estos por su valor.

2.2 Formulación del problema

Dada la problemática anterior, el presente proyecto pretende dar respuesta al siguiente interrogante: ¿Qué características o requerimientos debe poseer el sistema de reconocimiento óptico de caracteres de placas vehiculares?

3. JUSTIFICACIÓN

Existen en el mercado variedad de productos a nivel de hardware y software para la detección, captura, procesado de imagen e identificación de placas vehiculares, la mayoría trabaja con dispositivos que utilizan diferentes tecnologías para la transmisión de datos como son los sistemas de radiofrecuencias, cableado de cobre, fibra óptica, protocolos RS y protocolo TCP/IP, este último implementado para el presente caso. (Kineo Ingeniería, s.f.)

La información es un recurso valioso que se debe proteger, así mismo debe ocurrir con la capturada en tiempo real ya que puede procesarse de forma inmediata al ser comparada con bases de datos de autoridades locales o internacionales, inclusive de empresas privadas o públicas, su envío debe realizarse por un medio de transmisión seguro.

Por otro lado, se busca que el destinatario de esta sea el autorizado para recibirla, por lo que es necesario tener en cuenta la identificación de la persona o equipo que la reciba, usando en este caso la dirección de Control de Acceso al Medio (MAC, por sus siglas en inglés, Media Access Control), la cual está asociada de forma única a una tarjeta o dispositivo de red a través de un código de 48 bits. (Ros, 2016)

Al utilizar filtrado MAC es posible crear una lista de los dispositivos que pueden conectarse o no a la red de la organización o empresa, lo que representa en sí una ventaja y forma de proteger la información. (Ros, 2018)

El software creado en el presente proyecto utiliza el protocolo TCP/IP para su transporte, fijando su destino con la dirección MAC del cliente, comparado frente a otros sistemas este no requiere de conversiones, como en los casos de los protocolos RS (RNDS, 2007); por otro lado, existen inhibidores de frecuencias que pueden llegar a afectar los dispositivos y señales que estos emiten.

Otra de las ventajas que ofrecen estas tecnologías, es que existen variedad de equipos desde cámaras alámbricas e inalámbricas a bajos precios ahorro significativo en infraestructura y tecnología; igualmente pueden utilizarse dispositivos móviles, siempre y cuando estos se registren en la lista de dispositivos permitidos para acceder a la red de la empresa u organización.

Así mismo, buscando el aprovechamiento de diferentes alternativas de software de código abierto y con licencia pública como librerías, máquinas virtuales, gestores de bases de datos como Java, OpenCV, MySQL, entre otros que son accesibles a través de las páginas de sus creadores y colocados a disposición del público en general de forma gratuita, que por sí representa ventaja de uso para las pequeñas empresas que no buscan explotación comercial de estas sino su utilidad para beneficio de la propia organización.

El sistema de reconocimiento de placas vehiculares sirve de apoyo a las empresas y organizaciones en actividades de vigilancia, guarda, captura y búsqueda de vehículos de especial interés, o bien como herramienta para automatizar el ingreso de vehículos a un edificio, etc.

En este caso, la utilización de una librería de visión artificial (OpenCV, Open Source Computer Vision), con licencia BSD permite su uso y modificación del código, aunque es utilizada a nivel comercial, es posible usarla como software libre para lo cual debe hacerse mención de esta en el desarrollo de programas de este tipo.

4. MARCO TEÓRICO

4.1 Antecedentes de la Investigación

El hurto de vehículos es un problema de alcance mundial, pues la falta de cooperación y comunicación entre las naciones se presenta como un escenario de oportunidad para las organizaciones delincuenciales en la movilización y comercialización del automotor o sus partes, además de la baja rigurosidad penal en esta clase de delito.

En este interés, los avances tecnológicos han servido de apoyo para el desarrollo y construcción de dispositivos, herramientas y software orientadas a la vigilancia, rastreo, seguimiento y ubicación de los vehículos. Sin embargo, es también necesario conocer que pueden presentarse problemas con respecto a la seguridad y su gestión al interior de las empresas, especialmente las pequeñas y medianas empresas, las cuales realizan inversiones relativamente bajas o casi nulas en tecnología y seguridad.

La transmisión de datos se realiza a través de cables de cobre, fibra óptica, radiofrecuencia, protocolos serie RS 232, RS 422 y RS 48, y protocolo TCP/IP, este último representa gran ventaja para empresas o negocios pequeños dedicados al parqueo y vigilancia de los vehículos, las cuales generalmente tienen contratado un servicio de internet, utilizan cámaras cableadas o inalámbricas que conlleva solo a la instalación del software, realizar la conexión al sistema ya existente, listar las direcciones IP que pueden acceder a este que representa finalmente una mínima inversión económica y de tiempo.

Existen también sistemas de rastreo que utilizan las constelaciones de satélites y tecnologías de posicionamiento geográfico que permiten desarrollar actividades de control, vigilancia y recuperación vehicular, seguimiento de paquetes, envíos y rutas, ubicación de automotores, vigilancia y seguimiento

en tiempo real, fotografía del área donde se ubica el mismo, entre otros, las cuales aplican empresas de transporte de carga y personas, así como por las autoridades de tránsito, complementado por software que realiza reconocimiento de placas vehiculares, fotografiado, seguimiento y ubicación de los vehículos.

En investigaciones internacionales, destaca el trabajo titulado "Sistema de localización monitoreo y control vehicular basado en los protocolos GPS/GSM/GPRS", desarrollado por Astudillo y Delgado (2012), cuyo objetivo es implementar el rastreo vehicular a través de satélite, información que es accesible a través de una plataforma web, que comprende una serie de formularios donde el usuario puede elegir fecha, hora, ruta, placa a fin de realizar las tareas de monitoreo y seguimiento del automotor. La información viaja por la red GSM/GPRS hacia los servidores donde se almacenan todos los datos de ubicación, la empresa y el mismo vehículo.

En estas aplicaciones también se hace uso de los Sistemas de Información Geográfica (GIS), el cual sirve de apoyo para el demarcado de áreas geográficas donde se logre ubicar el vehículo, así mismo de dispositivos de Localización Automática de Vehículos (AVL, por sus siglas en inglés, Advanced Vehicle Location), con los cuales se puede realizar seguimiento en vivo y de forma diferida (Tirabassi, 2011), igualmente es posible calcular la velocidad del vehículo dependiendo de la zona por la cual se desplace (baja, media y alta velocidad) teniendo en cuenta los límites de velocidad definidos por las autoridades y asignados para cada tipo de vía.

El Internet de las Cosas (IoT), hizo posible el seguimiento, activación y vigilancia desde cualquier dispositivo, da Silva (2016) desarrolló un sistema embebido para la vigilancia remota de vehículos para lo cual utilizó sensores inalámbricos como Arduino y Raspberri Pi, además de lenguajes como Java, Javascript y los frameworks Primefaces, JavaServer Faces, y la API de Google, para el seguimiento y visualización satelital, entre otros.

Así mismo, el desarrollo de aplicaciones hace posible el monitoreo desde la pantalla de un móvil que se conecte a internet (Betancourt & Gómez, 2015), adicional a esto la visión artificial ayuda en la identificación y lectura de placas vehiculares gracias al reconocimiento óptico de caracteres. (Álvarez, 2014)

En esta misma línea, Muñoz (2014) desarrolló un sistema detector y lector de matrículas vehiculares utilizando la visión artificial apoyada en tecnologías, el algoritmo de reconocimiento óptico de caracteres, la librería de código abierto OpenCV basada en C++, empleando para la extracción y levantamiento de las imágenes el detector SURF (Speeded Up Robust Features), basado en un sistema transformador de la información SIFT (Scale Invariant Feature Transform).

SURF ofrece ventajas en cuanto a velocidad de cálculo y robustez en la transformación de imágenes, con un impacto positivo en la calidad y precisión de estas, tardando menos tiempo en el reconocimiento y procesamiento de la imagen, reduciendo el ruido y límites verticales u horizontales en las mismas. (Muñoz, 2014, pág. 64)

Para llegar a una conclusión en cuanto a la tecnología a utilizar para la detección y lectura de las imágenes realiza un comparativo de estos dos sistemas, entre los cuales se presenta como desventaja los costos elevados de SIFT, de uso comercial.

Delgado (2010) hace referencia a cuatro métodos de localización de la placa: Análisis de gradiente de la imagen capturada; Análisis del valor de cuantización vectorial; Filtros a través de redes neuronales y Análisis de imagen de borde, encontrándose entre estas que las redes neuronales presentan un buen funcionamiento para el reconocimiento de patrones y símbolos segmentados.

En cuanto a la implementación del sistema se puede hacer uso de cámaras IP, si se requiere de calidad en la imagen es necesario adoptar resoluciones superiores, Espinosa & Salinas (2015) desarrollan un sistema utilizando una

cámara IP con una resolución de 1280 x 720 pixeles, con detector de movimiento con alarma email y FTP, y compatible con otros sistemas operativos.

Una vez captada la imagen es enviada al servidor, un computador con 4GB de memoria RAM, disco duro de almacenamiento de 250 GB, procesador Intel I3 y sistema operativo Windows. El sistema se desarrolló para el ingreso y salida de vehículos de un parqueadero, en el cual existe una barra de acceso la cual es accionada a través de Arduino y un servomotor. Posteriormente la cámara IP capta la imagen y es enviada de forma alámbrica al servidor local, para el cual se utilizó el gestor de base de datos MySQL y para el servidor Wamp Server.

Los autores concluyen que existieron factores que conllevaron a error en el reconocimiento y lectura de la imagen, como son el estado de la placa (golpes, suciedad, deformidad, desgaste, modificadas, alteradas, repintadas); el ambiente o espacio donde se realiza (poca luz, brillo); por otro lado, el número de capas en las redes neuronales pueden dar un mayor reconocimiento, siendo más para las capas ocultas y una por encima de estas, de esta forma se logró gran número de aciertos.

4.2 Marco Conceptual

4.2.1 Visión humana. El ojo humano captura imágenes del exterior, gracias a la luz que entra en el mismo, a través de la córnea y el iris, atravesando la lente del cristalino antes del alcanzar la retina. El iris es el encargado de controlar el nivel de luz que ingresa a la retina, esta recibe una imagen invertida y la transmite por el sistema óptico conformado por la córnea y el cristalino. Es decir, que el ojo humano es como una cámara oscura en donde el cristalino altera su forma para enfocar la imagen, esta capacidad de

adaptación se pierde con la edad y con ello la capacidad visual óptica. (Westland, 2001)

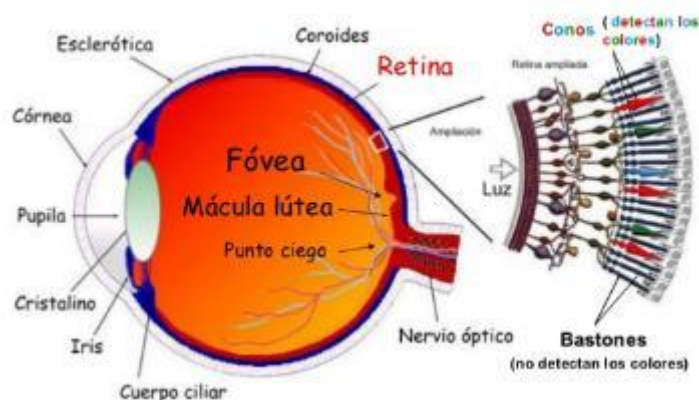


Figura 1. Visión humana

Fuente: (Óptica Luz, 2017)

La diversa luminosidad que posee la imagen y la posibilidad de que sea captada por el ojo se debe a los millones de células dentro de la retina llamadas bastones, capaces de generar la intensidad de brillo o luminosidad de la imagen. Según se aleja del área sensible, las células con forma de cono se vuelven más escasas y en los bordes exteriores de la retina sólo existen las células con forma de bastones.

4.2.2 Visión por computador. La visión por computador inicia con la realización de la primera fotografía por Joseph Nicéphore Niépce, en el año de 1825, en la cual se empleó una cámara oscura y una superficie fotosensible para fijar la imagen en esta. Posteriormente, en 1920 se utiliza el cable submarino para enviar y recibir imágenes entre Londres y Nueva York. Con la evolución en el sistema de las cámaras de video, en 1980, así como los microprocesadores y los computadores se logra desarrollar la visión por

computador que permite la extracción de características relevantes, incluso que este procedimiento se pueda realizar de forma automática.

La visión por computador realiza varios procesos (Somolinos, 2002):

Adquisición de la imagen. Comprende la captura de la imagen digital.

Pre-procesamiento. En esta etapa se hace el tratamiento de la imagen mediante filtros, transformaciones que hacen posible obtener una imagen más nítida y con mayor claridad.

Detección de bordes. Se selecciona y verifica el área que se va a trabajar.

Segmentación. Comprende la partición de la imagen en regiones con significado propio, de esta forma es posible interpretar y reconocer el área que se requiere identificar.



Figura 2. Procesos de la visión por computador

Fuente:

Extracción de características. En esta fase se realiza una representación matemática de las imágenes seleccionadas en partes determinadas.

Reconocimiento y Localización. Se clasifican los objetos con características comunes, que se asemejen al objeto, y dentro de la localización es necesario localizarlo dentro del espacio 3D y restringir mediante técnicas de triangulación el espacio, para que se seleccione nuestra única imagen.

Interpretación. En esta fase se procede a interpretar la escena resultante de la información y los cambios encontrados y generados en los procesos anteriores.

4.3 Marco Conceptual

Placas Vehiculares. Las placas permiten identificar un vehículo, la serie, rango y código los asigna el Ministerio de Transporte asignar sus series, su elaboración y entrega las realiza la Dirección de Tránsito o a quien designe el Ministerio.

En Colombia, las placas de vehículos se clasifican de acuerdo con el servicio que presta el automotor, y se deben portar adelante y atrás en un lugar visible en el caso de los carros. Para las motos, cuatrimotos o cualquier otro vehículo derivado de estos, la reglamentación indica que la placa debe estar ubicada en la parte posterior. De esta manera las placas están orientadas como se observa en la siguiente tabla.

Tabla 1. *Tipo de placas en Colombia*



Vehículos particulares: las placas son de color amarillo fuerte reflectivo, con borde negro, tres letras y tres números (AAA 123) de color negro separados por el logo del Ministerio de Transporte de Colombia y abajo solamente el nombre del municipio o distrito en donde se matriculó.



Vehículos de servicio público: le corresponden placas blancas reflectivas con letras y números negros (AAA 123) separados por el logo del Ministerio de Transporte de Colombia y abajo solamente el nombre del municipio o distrito en donde se matriculó.



Vehículos diplomáticos, consulares y de misiones especiales: usan placas azules y están compuestas por dos letras y cuatro números blancos (AA 1234) junto con la palabra Colombia. Se expiden a través del Ministerio de Relaciones Exteriores.



Vehículos de carga: algunos vehículos de carga utilizan placas de fondo rojo reflectivo con una letra (la mayoría de los casos "T") y cuatro números blancos (T 1234), abajo la palabra Colombia. Este tipo de matrícula es de tránsito libre.

Fuente: (Ortiz, 2016)

Por su parte las motocicletas o derivados tienen placas amarillas más pequeñas y con letras negras grabadas horizontalmente (AAA 12A).

Las placas autorizadas deben portarse siempre y no tienen fecha de caducidad, su reemplazo se hace única y exclusivamente con el mismo número y solo en casos de deterioro, reporte de pérdida o cambio de radicación del vehículo.

Las placas se ubican en el frente y parte trasera del vehículo.

4.4 Marco Legal

Constitución Política de Colombia

Referente a la protección de datos personales, el artículo 15 de la Carta Magna expresa que en aras de proteger la intimidad personal, familiar y buen nombre las personas tienen derecho a conocer, actualizar y rectificar las informaciones capturadas por los diferentes establecimientos y entidades estatales en bases de datos. (Superintendencia de Industria y Comercio, 2016)

En su artículo 61 establece el concepto de "Propiedad Intelectual", el cual es relevante al momento de desarrollar algún producto, o desarrollo, como lo es en el caso del Software.

Norma Técnica Colombiana NTC 5420-1

La cual hace referencia a la calidad en el proceso de desarrollo software como en el producto mismo. (ICONTEC Internacional, 2006)

Norma Técnica Colombiana NTC 5415-1

Hace referencia al proceso de evaluación del producto software, ya que el desarrollo de estos debe comprender procesos de calidad que conlleven a una calidad en el producto mismo.

ISO 12207 – Modelos de Ciclos de Vida del Software

Establece un estándar para los procesos de ciclo de vida del software de la empresa, dirigida especialmente a quienes adquieran este tipo de productos, al igual que para desarrolladores y proveedores.

Los procesos están agrupados en tres categorías: Principales, de apoyo y de organización, a la vez que agrupa las actividades del ciclo de vida en 5 procesos principales, 8 de apoyo y 4 organizativos.

ISO/IEC 14598

Estándar que define el proceso de evaluación del producto de software, el cual comprende seis partes: Visión General, Planeamiento y Gestión; Proceso para desensolvedores; Proceso para adquirentes, Proceso para avaladores y Documentación de Módulos.

ISO/IEC 25000 (SquaRE)

Norma basada en los estándares ISO 9126 y 14598, que refiere la calidad en el proceso de recolección de información (requerimientos) y de evaluación del software.

ISO 15939

Estándar que provee un modelo de información que apoya las etapas de planificación, performance y evaluación de la medición

IEEE (Institute of Electrical and Electronics Engineers)

Método de establecimiento y mejora del trabajo en equipo para procesos software, su creación data de 1884, su principal objetivo es "promover la creatividad, el desarrollo y la integración, compartir y aplicar los avances en las tecnologías de la información, electrónica y ciencias en general para beneficio de la humanidad y de los mismos profesionales". (Arciniega, 2017)

Dentro de este se encuentran los estándares VHDL, POSIX, IEEE 1394, IEEE 488, IEEE 802, IEEE 802.11, e IEEE 754.

Además, contiene una serie de documentos que sirven de guía para el desarrollo de software y proyectos de TI.

5. METODOLOGÍA

5.1 Tipo de Investigación

El tipo de investigación es experimental, por cuanto se pone a prueba un sistema de reconocimiento de placas vehiculares en tiempo real, desarrollado con lenguaje de programación y el uso de librerías de visión artificial.

El software puede implementarse en un parqueadero privado, de selección del estudiante, en donde se permitirá la instalación del software necesario para aplicar el sistema de reconocimiento de placas en vehículos en movimiento, al ingreso y salida de este.

5.2 Fuentes de Información

Las fuentes de información utilizadas para el proyecto de grado comprenden fuentes primarias y secundarias. Son fuentes primarias los libros de texto, manuales técnicos, artículos y tesis. Como fuentes secundarias se encuentra los informes generados en la prueba de codificación y la fase final del proyecto, donde se reseñarán el trabajo desarrollado y correcciones de este.

5.3 Alternativa de trabajo de grado

El tratado de este documento se ajusta a un proyecto aplicado. El cual, según reglamento estudiantil de la Unad, corresponde a la actividad de transferencia investigativa que le permite al estudiante diseñar proyectos de desarrollo empresarial, tecnológico y social comunitario para contribuir de manera novedosa a la solución de problemas focalizados (Unad, 2013, p 23).

5.4 Instrumentos

A continuación se presentan las características de los equipos a utilizar en el desarrollo del proyecto.



Figura 3. Cámara Logitech 930e

Fuente: LOGITECH (2014). Especificaciones técnicas de LOGITECH WEBCAM C930E. Recuperado de http://support.logitech.com/es_es/article/43549?product=a0qi00000069v0MAAQ

Tabla 2. Información general de la Cámara Logitech 930e

Categoría	Cámara Web
Software compatible	No requiere software. El software Logitech permite controlar la visión panorámica, la inclinación y el zoom desde el ordenador.
Sistemas operativos compatibles	Windows 8, Windows 7, Mac OS 10.7 o posterior
	Requisito HD
Requisitos del sistema	CPU mínima = Core 2 duo a 2,4 GHz CPU recomendada = Quad core a 3,06 GHz RAM mínima = 2 GB RAM recomendada = 4 GB

Fuente: LOGITECH (2014). Especificaciones técnicas de LOGITECH WEBCAM C930E. Recuperado de http://support.logitech.com/es_es/article/43549?product=a0qi00000069v0MAAQ

Tabla 3. *Especificaciones de cámara Web LOGITECH 930e*

Tipo de conexión	USB
USB VID_PID	0843
Protocolo USB	USB 2.0
Velocidad USB	Alta velocidad
Compatibilidad con UVC	Sí (UVC 1.5)
Micrófono	Sí
Tipo de micrófono	Dos, estéreo
Objetivo y tipo de sensor	Cristal
Tipo de foco	Auto
Resolución óptica	Real: 3 MP
Campo visual diagonal	90°
Longitud focal	N/A
Captura de imágenes (SD 4:3)	N/A
Captura de imágenes (W 16:9)	N/A
Captura de vídeo (SD 4:3)	N/A
Captura de vídeo (W 16:9)	N/A
Frecuencia de cuadro (máx.)	1080 p a 30 fps
RightLight	RightLight 2
Efectos de vídeo	N/A
Botones	N/A
Indicadores luminosos (LED)	Sí
Tapa de privacidad	Sí (hardware)
Opción de montaje sobre trípode	Sí
Clip universal ajustable (rango)	N/A
Longitud del cable	1,83 metros

Fuente: LOGITECH (2014). Especificaciones técnicas de LOGITECH WEBCAM C930E. Recuperado de http://support.logitech.com/es_es/article/43549?product=a0qi00000069v0MAAQ

Tabla 4. *Especificaciones técnicas del equipo informático (computador de escritorio)*

Procesador	Intel Core i5-2520M CPU 2.50 GHz
Memoria RAM	8,00 GB
Tipo de Sistema operativo	Windows 10 x 32 bits
Monitor	LED 20" LG 1080 px

Fuente: Autor

Software a utilizar

Netbeans. Es un entorno de desarrollo integrado (IDE) de código abierto y de uso gratuito, se presenta en su versión 8.0.2 con funcionalidades acordes a las necesidades del usuario. Fue especialmente desarrollado para el Lenguaje Java

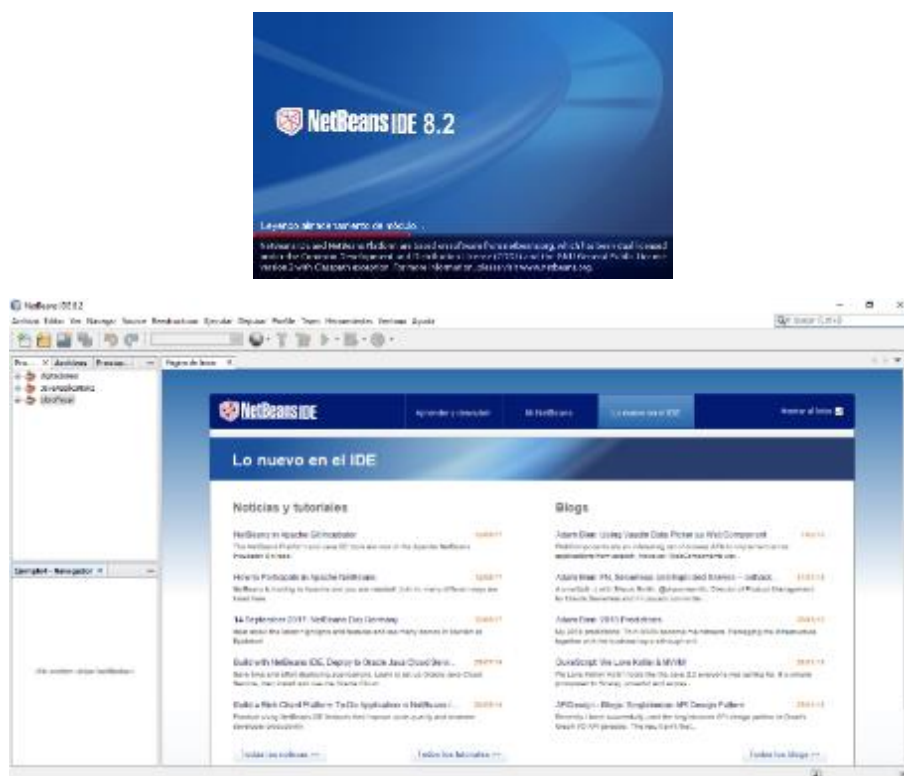


Figura 4. IDE Netbeans

Entre sus características principales se encuentran la gestión de proyectos, independientes de su tamaño; edición multilinguaje, uso de asistentes, optimización de proyectos, acceso a bases de datos y permite el uso de plugins.

Su instalación es un proceso bastante sencillo, se accede desde el navegador a la dirección URL: <https://netbeans.org/downloads/index.html>, donde se

pueden observar varias opciones de descarga, seleccionando a su vez el idioma, la versión, sus funcionalidades (lenguajes que se va a trabajar en este entorno), y la plataforma en que se utilizará como Windows, Linux, Mac OS o una versión independiente del sistema operativo en el que se instale o se use.

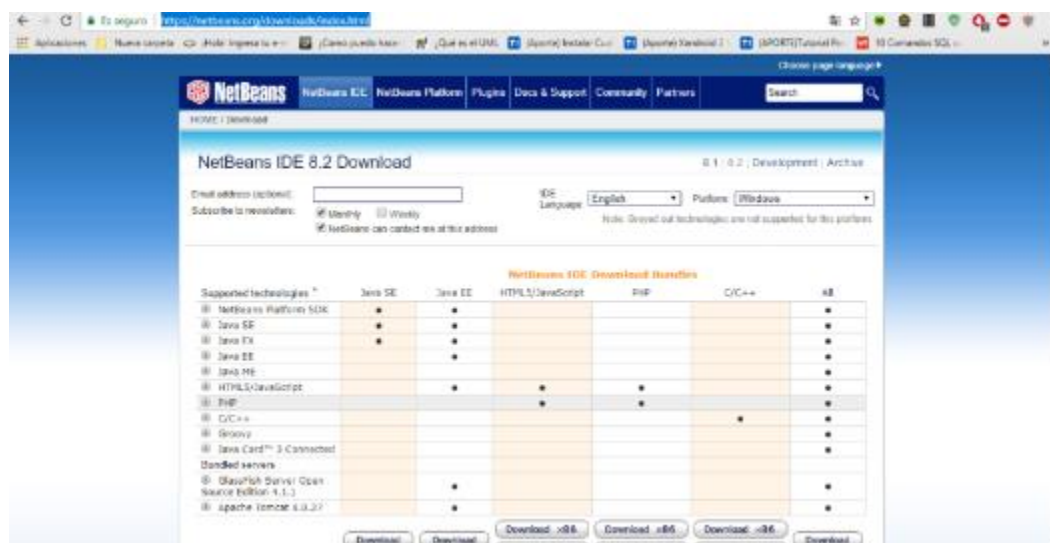


Figura 5. Página de descarga de Netbeans

Máquina Virtual de Java (JVM). Es un entorno que una vez instalado permite la ejecución de instrucciones generadas por un compilador Java. La máquina interpreta las instrucciones a través de un sistema que ejecuta los archivos de clase Java, esta presenta una gran ventaja, ya que es posible ejecutarlos en cualquier plataforma, brindando independencia del sistema operativo donde inicialmente se trabaja.

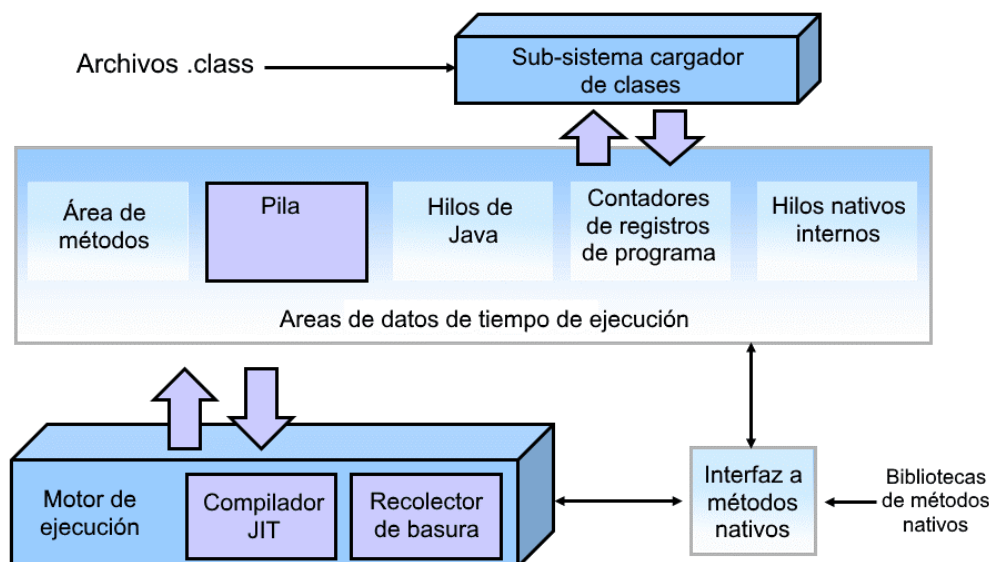


Figura 6. Componentes de la Máquina Virtual Java

Fuente: Alarcón, J. (2017). ¿Qué es la máquina virtual de Java o Java Virtual Machine? Recuperado de <https://www.campusmvp.es/recursos/post/que-es-la-maquina-virtual-de-java-o-java-virtual-machine.aspx>

Generalmente la máquina virtual de java se encuentra preinstalada en el sistema operativo, y es esta la que se encarga de leer las instrucciones contenidas en un archivo ejecutable con la extensión .jar, o aplicación de Java.

Kit de Desarrollo de Java (JDK, Java Development Kit). El kit de Java contiene variedad de utilidades que permite el desarrollo de aplicaciones Java, entre estas se encuentran las siguientes:

- *javac*. Compilador de Java, que interpreta el código fuente escrito en Java.
- *java*. Intérprete de Java, el cual ejecuta las instrucciones a partir de archivos class.
- *javadoc*. Permite crear documentos en formato HTML a partir del código fuente.

JavaMail. Librería desarrollada por Sun Microsystems Inc., que permite el envío de correos electrónicos desde la aplicación Java. Para su instalación es necesario descargarla desde la dirección URL: <http://www.oracle.com/technetwork/java/javamail/javamail145-1904579.html>.

Para su instalación del entorno de desarrollo Netbeans, se accede a la opción Tools Libraries, desde la carpeta del proyecto.

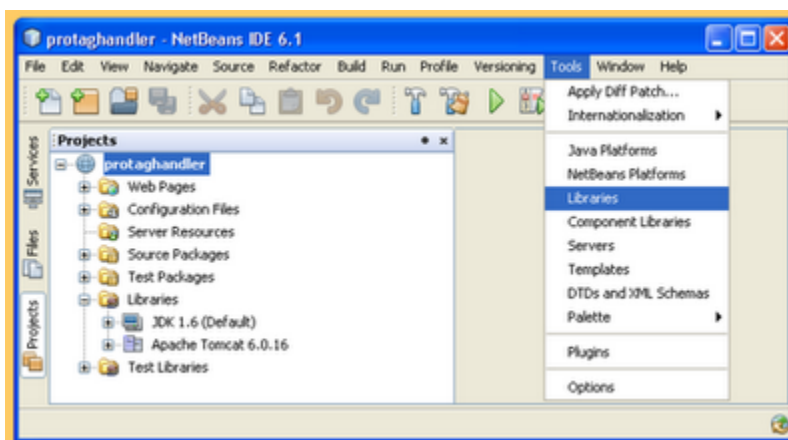


Figura 7. Instalación de la librería JavaMail

Una vez instalada se asocia la librería con el proyecto al dar clic derecho en la carpeta del proyecto y luego se selecciona la opción Propiedades (Properties), y se selecciona la categoría Libraries y clic sobre el botón, tal como se observa en la siguiente figura.

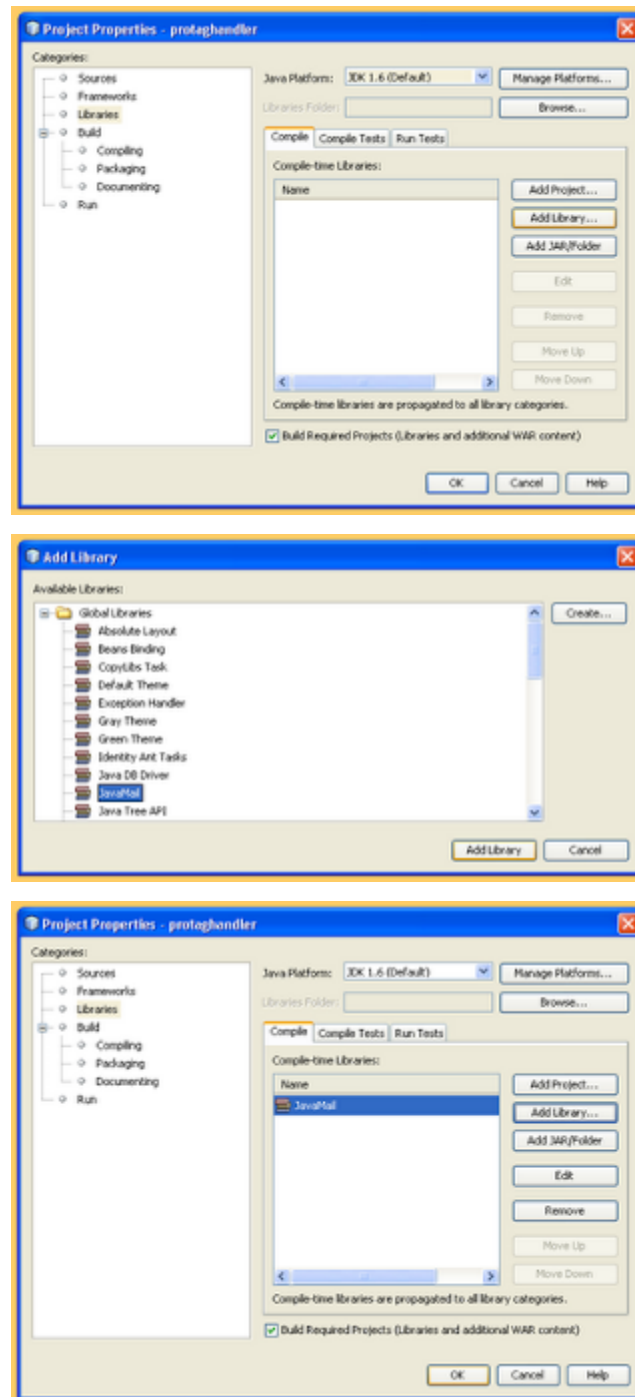


Figura 8. Adición de la librería al proyecto en Netbeans

Tabla 5. *Clases y métodos básicos usados*

Clases	Descripción
Clase Properties	Almacena las propiedades de conexión en el servidor de correo saliente SMTP.
Métodos	Descripción
Put	Asigna las propiedades requeridas como servidor SMTP, TLS, puerto, correo de quien envía, usuario de acceso al servidor SMTP.
Get	A través del cual se obtienen los parámetros mencionados en el método anterior.
GetDefaultInstance	Permite la creación de la sesión de las propiedades de la clase.
GetTransport	Al cual se le indica cuál es el protocolo de transporte (SMTP).
Clase MimeMessage	Se forma el mensaje a enviar.
Constructor	Se introduce la sesión sobre la que se va a enviar el mensaje.
Método SetFrom	Recibe la dirección del emisor, tipo InetAddress
Método AddRecipient	Recibe 2 parámetros tipo de receptor uno se escribe en la clase Message.RecipientType; y el segundo al método SetFrom con una variable de clase InetAddress que contiene la dirección del receptor.
Método SetSubject	Se escribe el asunto como parámetro, es una cadena o string.
Método SetText	Se escribe el texto del mensaje también en string.
Clase Transport	Se definen los parámetros del protocolo de transporte.
Método Connect	Método que realiza la conexión con el servidor (con usuario y contraseña)
Método SendMessage	Envía el mensaje creado.
Método Close	Cierra la conexión.

OpenCV. Es una biblioteca libre de visión artificial desarrollada por Intel. Contiene más de 500 algoritmos con multiplicidad de funciones en el procesamiento de imágenes, descripciones geométricas, segmentación, reconocimiento facial, calibración de cámaras, visión estéreo visión robótica,

seguimiento y computación. Así mismo, permite el uso de otras librerías de Intel dedicadas a procesadores Intel.

5.5 Procedimiento

El proyecto se realiza de acuerdo con las etapas del ciclo de vida del software (USR.CODE, s.f.):

Etapas 1: Análisis de requisitos. En esta se describen los requerimientos básicos para el desarrollo del sistema, herramientas y usuarios a los cuales va dirigido su uso y aplicación.

Etapas 2: Diseño y Arquitectura. Se procede al diseño gráfico de la interfaz, igualmente se describe su funcionalidad en términos generales. Se incorporan consideraciones de la implementación tecnológica como son: hardware, red, base de datos, etc. Se emplean los casos de uso para cubrir las funciones que realiza el sistema, y se transforman las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercano a la programación orientada a objetos.

Etapas 3: Programación. Se inicia con la codificación, para el presente caso se realiza utilizando el lenguaje JAVA, junto a la librería OpenCV.

Etapas 4: Pruebas. En esta se inicia la ejecución del sistema creado, al mismo tiempo se observa su funcionalidad y usabilidad, y si existen errores de codificación, se toman las notas necesarias sobre el funcionamiento en esta etapa, de esta forma es posible realizar y aplicar correcciones en la etapa de mantenimiento.

Etapa 5: Documentación. En esta etapa el desarrollo del software y de la gestión del proyecto pasan por modelaciones (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, etc.

Etapa 6: Mantenimiento. Se realiza con el fin de mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos.

5.6 Cronograma

Tabla 6. Cronograma

Actividades	Año 2018																			
	Feb.				Mar.				Abr.				May.				Jun.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Búsqueda de referencias, recopilación de información.			■	■																
Selección de herramientas (software y hardware, lenguaje, dispositivos, etc.)				■																
Etapa 1. Requerimientos					■															
Etapa 2. Diseño y Arquitectura						■	■	■	■											
Etapa 3. Programación (codificación)									■	■										
Etapa 4. Pruebas (aplicación del software)											■									
Etapa 5. Documentación (diagramas, informes, matriz de evaluación)												■	■							
Etapa 6. Mantenimiento (mejoras y correcciones)													■	■	■					
Presentación informe final.																	■			
Sustentación.																		■		

6. DESARROLLO DEL PROYECTO

6.1 Descripción del Proyecto

El software tiene como objetivo la detección, seguimiento y ubicación de vehículos a través del reconocimiento óptico de placas vehiculares en tiempo real. Su utilidad se encuentra en actividades de ingreso, salida y vigilancia en parqueaderos; en empresas de transporte de carga y de personas; vigilancia y control en las vías por las autoridades de tránsito o policiales, que puede ser complementario a sistemas ya instalados.

Se registra el cliente y los datos del vehículo, a través de una cámara de video o cámara IP se captura la imagen en movimiento o no, de la placa vehicular a través del reconocimiento óptico de caracteres (OCR) y se corrobora la información con la base de datos y luego cambiar el estado por "hurtado".

El sistema depende de gran variedad de tecnologías y librerías de Java y de reconocimiento óptico de caracteres (OCR), escalable, ya que pueden adaptarse posteriormente sensores alámbricos e inalámbricos, así como rastreo digital, o utilizar aplicaciones móviles.

Etapa 1. Requerimientos

Tabla 7. *Requisitos del Sistema para Java*

Versión Java	7.0, 8.0
Sistemas operativos	Windows 10 (8u51 y superiores) Windows 8.x (escritorio) Windows 7 SP1 Windows Vista SP2 Windows Server 2008 R2 SP1 (64 bits)

	Windows Server 2012 y 2012 R2 (64 bits)
Memoria RAM	128 MB
Espacio en disco	JRE: 124 MB Java Update: 2 MB
Procesador	Mínimo Pentium 2 a 266 MHz
Exploradores	Internet Explorer 9 y superior, Firefox

Fuente: (ORACLE, 2018)

Tabla 8. *Requisitos mínimos del Sistema para Netbeans*

Versión Netbeans	8.0
Sistemas operativos	Windows 7 Windows Vista Windows XP
Resolución de pantalla	1.024 x 768 pixeles
Tecnologías Java compatible	Java: JDK 8, Java SE, Java ME, Several Java, Java FX Java EE C C++

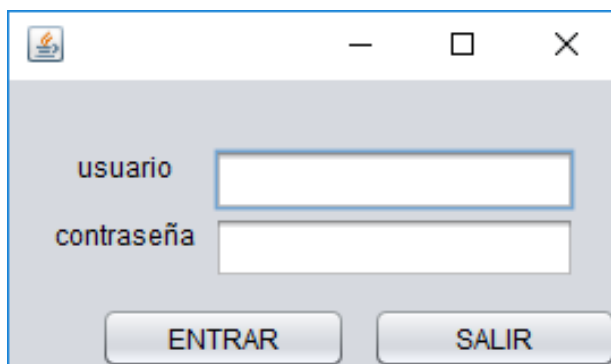
Fuente: (Netbeans Org, 2018)

Tabla 9. *Requerimientos funcionales*

Requerimiento	Descripción
RF-01	Nota 01
Inicio de sesión a través de la interfaz de usuario.	Se solicita el usuario y la contraseña.
RF-02	Nota 01
El programa permite la creación de usuarios (propietarios).	Los roles son usuario Administrador y Propietario o Cliente.
	Nota 02
	Se diligencian los siguientes datos: Cédula, Nombres y Apellidos, Placa, Marca, Color, Estado, Correo Electrónico.
RF-03	Nota 01

El programa permite eliminar un usuario (propietario)	Se solicita el número de identificación para la consulta.
	Nota 02 El administrador es el único que puede realizar la consulta.
	Nota 03 Los datos aparecen en la interfaz: Cédula, Nombres, Apellidos, Correo Electrónico, Placa, Marca, Color, Estado.
RF-04 El programa permite capturar la placa vehicular del propietario.	Nota 01 Se captura la imagen a través de la cámara web.
	Nota 02 Se guarda la imagen en la base de datos.
RF-05 El programa permite la lectura y reconocimiento de la placa.	Nota 01 El programa hace el reconocimiento y muestra los resultados en pantalla sobre el propietario del vehículo.
RF-06 El programa permite el envío de la imagen capturada al correo del propietario del vehículo.	Nota 01 Se envían los datos al correo del propietario del vehículo, la cual comprende la foto de la placa, ubicación.

Fuente: Autor



The image shows a standard Windows-style window with a title bar containing a small icon, a minus sign, a maximize button, and a close button. The main area of the window is light gray and contains a login form. The form has two text input fields. The first is labeled 'usuario' and the second is labeled 'contraseña'. Below these fields are two buttons: 'ENTRAR' on the left and 'SALIR' on the right. The entire window is outlined with a blue border.

Figura 9. Formulario de Inicio de Sesión

Fuente: Autor



Figura 10. Interfaz de resultados de búsqueda

Fuente: Autor

Etapas 2. Diseño y Arquitectura

Perfil del usuario: Administrador. Generalmente el vigilante o administrador encargado del parqueadero y su funcionamiento.

Perfil del usuario: Propietario del vehículo. Población de la ciudad de Cúcuta, propietaria de vehículo automotor, debidamente registrado y legalizado.

Etapas 3. Programación

Se anexa al final del documento.

Etapa 4. Pruebas



Figura 11. Prueba de captura 1

Fuente: Autor



Figura 12. Prueba de captura 2

Fuente: Autor



Figura 13. Prueba de captura 3

Fuente: Autores

Etapa 5. Documentación

Se anexa al final del documento.

Etapa 6. Mantenimiento

Esta etapa permitió realizar las mejoras al programa, entre ellas la configuración para el envío al correo del propietario del vehículo.

6.2 Resultados y Análisis de los resultados

Se tomó como imagen de análisis la placa presentada en la figura 11 (MHX-796), la cual se convirtió a escala de grises y se depuró con la utilidad de la librería de OpenCV para cambiar los tonos de colores a grises.



Figura 14. Placa en escala de grises

Fuente: Autor

En la segunda imagen se utilizó la función "Detector de bordes", con Java con lo cual se busca una mejor imagen en nitidez y de fácil reconocimiento.

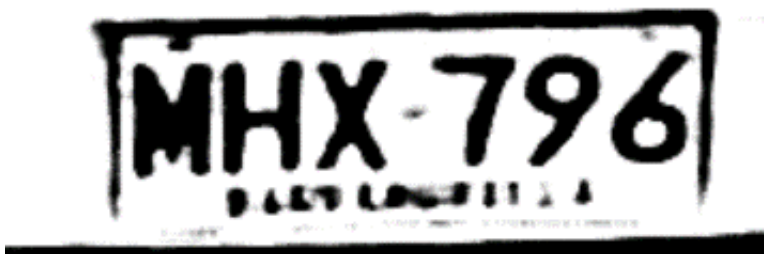


Figura 15. Resultado de la detección de bordes

Fuente: Autor

Se observa en las figuras anteriores que la imagen procesada pierde algunos caracteres de la parte inferior, el número de placa se mantiene, aunque la M presenta un relleno extra que puede generar error en el programa.

De acuerdo a lo anterior, se puede concluir que la funcionalidad de reconocimiento es posible con cámaras con resoluciones medias, sin embargo, para una imagen de mayor claridad se recomienda la adopción de cámaras de altas resoluciones, captadoras de imagen aun en movimiento. En resumen, la calidad utilizada es aceptable para pequeños negocios y empresas.

Conclusiones

- Las tecnologías y el internet permiten el acceso a un sinnúmero de herramientas en línea y de uso off-line, con licencia gratuita o de código abierto, que puede ser aprovechado por empresas o negocios pequeños, de los cuales es necesario aclarar la importancia de comprender el uso y el contenido de la licencia de este software.
- El diseño se realizó a partir de herramientas gratuitas y de código abierto, que puede instalarse en cualquier sistema operativo, como Java, que puede mejorar la portabilidad del programa.

Referencias

- Álvarez, M. (2014). Análisis, diseño e implementación de un sistema de control de ingreso de vehículos basado en visión artificial y reconocimiento de placas en el parqueadero de la Universidad Politécnica Salesiana - Sede Cuenca. Cuenca, Ecuador: Universidad Politécnica Salesiana, Ingeniería de Sistemas. Recuperado el 5 de octubre de 2017, de <https://dspace.ups.edu.ec/bitstream/123456789/7060/1/UPS-CT003790.pdf>
- Arciniega, F. (2017). *Normas y Estándares de calidad para el desarrollo de Software*. Recuperado el 21 de abril de 2018, de <https://fernandoarciniega.com/normas-y-estandares-de-calidad-para-el-desarrollo-de-software/>
- Astudillo, J., & Delgado, E. (2012). Sistema de localización, monitoreo y control vehicular basado en los protocolos GPS/GSM/GPRS. Cuenca, España: 15° Concurso de Trabajos Estudiantiles. Recuperado el 7 de octubre de 2017, de http://41jaiio.sadio.org.ar/sites/default/files/11_EST_2012.pdf
- Betancourt, D., & Gómez, G. (2015). Prototipo de sistema de vigilancia basado en la Internet de las Cosas con aplicativo para dispositivos móviles. Bogotá: Universidad Distrital Francisco José de Caldas, Ingeniería Industrial. Recuperado el 6 de octubre de 2017, de <http://repository.udistrital.edu.co/bitstream/11349/2918/1/PROTOTIPO%20DE%20SISTEMA%20DE%20VIGILANCIA%20BASADO%20EN%20LA%20INTERNET%20DE%20LAS%20COSAS%20CON%20APLICATIVO%20PARA%20DISPOSITIVOS.pdf>
- Cruz, F. (2012). Crean base de autos usados en el crimen. Recuperado el 15 de abril de 2018, de <http://www.excelsior.com.mx/2012/06/22/comunidad/842894>

- Da Silva, L. (2016). Sistema embebido para vigilancia remota de vehículos. Valencia, España: Universitat Politècnica de Valencia, Ingeniería de Computadores y Redes. Recuperado el 8 de octubre de 2017, de <https://riunet.upv.es/bitstream/handle/10251/74407/NUNES%20-%20Sistema%20embebido%20para%20vigilancia%20remota%20de%20vehículos.pdf?sequence=3>
- Delgado, J. (diciembre de 2010). Reconocimiento de Placas Vehiculares. México, D.F.: Instituto Politécnico Nacional, Ingeniería Mecánica y Eléctrica. Recuperado el 21 de abril de 2018, de <http://tesis.ipn.mx/jspui/bitstream/123456789/9880/1/221.pdf>
- El Espectador. (17 de enero de 2018). La persecución en helicóptero para recuperar un vehículo hurtado en Bogotá. Recuperado el 16 de abril de 2018, de <https://www.elespectador.com/noticias/bogota/la-persecucion-en-helicoptero-para-recuperar-un-vehiculo-hurtado-en-bogota-articulo-733821>
- Espinosa, D., & Salinas, C. (2015). Desarrollo de un sistema de reconocimiento de placas vehiculares. Cuenca, Ecuador: Universidad del Azuay, Ingeniería de Sistemas y Telemática.
- Hadhazy, A. (2015). *¿Cuáles son los límites de la visión humana?* Recuperado el 30 de septiembre de 2017, de http://www.bbc.com/mundo/noticias/2015/08/150810_vert_fut_vision_limites_lp
- ICONTEC Internacional. (2006). Norma Técnica Colombiana NTC 5420-1. Bogotá, D.C. Recuperado el 21 de abril de 2018, de <https://tienda.icontec.org/wp-content/uploads/pdfs/NTC5420-1.pdf>
- Imagen Noticias. (2014). Detectores de placas vehiculares combaten el robo de autos en el DF. México. Recuperado el 15 de abril de 2018, de <https://www.youtube.com/watch?v=zq-hr2jP0R0>

- Kineo Ingeniería. (s.f.). *Sistema de control de accesos personalizable: reconocimiento de matrícula, RFID, interfonía*. Recuperado el 21 de abril de 2018, de <https://www.interempresas.net/Medicion/FeriaVirtual/Producto-Aforo-y-clasificacion-vehicular-71059.html>
- Muñoz, R. (julio de 2014). Sistema de visión artificial para la detección y lectura de matrículas. Valladolid, España: Universidad de Valladolid, Ingeniería Electrónica Industrial y Automática. Recuperado el 21 de abril de 2018, de <https://uvadoc.uva.es/bitstream/10324/11848/1/TFG-P-165.pdf>
- Netbeans Org. (2018). Notas de la versión. Recuperado el 22 de mayo de 2018, de <https://netbeans.org/>
- Norza, E., Duarte, Y., Castillo, L., & Torres, G. (2013). Hurto de automotores y estrategias contra el delito: una mirada desde la academia, el victimario y la Policía. *Revista Criminalidad*, 55(2), 49-78. Recuperado el 19 de abril de 2018, de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-31082013000200004&lng=en&nrm=iso&tlng=es
- Óptica Luz. (27 de abril de 2017). ¿Cuáles son los límites de la visión humana? Recuperado el 21 de abril de 2018, de <http://visiontotal-opticaluz.blogspot.com.co/2017/04/cuales-son-los-limites-de-la-vision.html?sref=pi>
- ORACLE. (2018). ¿Cuáles son los requisitos del sistema para Java? Recuperado el 22 de mayo de 2018, de <https://www.java.com/es/download/help/sysreq.xml>
- Ortiz, M. (2016). ¿Sabes cuántos tipos de placas existen en Colombia? Recuperado el 21 de abril de 2018, de <http://andina.com.co/blog/sabes-cuantos-tipos-de-placas-existen-en-colombia/>

- Pacheco, S. (s.f.). *Geosistemas informáticos. Tema 1, Sistemas de visión artificial*. Recuperado el 2 de octubre de 2017, de <http://www.kumbaya.name/robotica/clases/vision%20robot/Tema%201.%20Sistemas%20de%20visi%C3%B3n%20artificial.pdf>
- Pajares, G., & de la Cruz, J. (2007). *Visión por computador. Imágenes Digitales y Aplicaciones* (2a. ed.). RA-MA.
- Pérez, R. (2009). Sistema de control de tráfico vehicular para la Empresa Unión Andina de Transportes S.A. "SISCONTRAV". Bogotá: Corporación Universitaria Minuto de Dios, Tecnología en Informática. Recuperado el 12 de octubre de 2017, de http://repository.uniminuto.edu:8080/xmlui/bitstream/handle/10656/437/TTI_PerezTorreglosaRaul_09.pdf?sequence=1
- Presidencia de la República de Colombia. (2002). Código Nacional de Tránsito Terrestre [Ley 769]. Bogotá.
- RNDS. (2007). *Conversores TCP/IP*. Informe Especial Conversión y Transmisión de datos, Argentina. Recuperado el 21 de abril de 2018, de http://www.rnds.com.ar/articulos/031/RNDS_128W.pdf
- Rodríguez, A. (2015). *OpenCV: Librería de Visión por Computador*. España: Universidad de la Laguna. Recuperado el 12 de octubre de 2017, de <https://osl.ull.es/software-libre/opencv-libreria-vision-computador/>
- Rodríguez, H., Vera, R., & Vintimilla, B. (2012). Detección y extracción de placas de vehículos en señales de video. *Revista Tecnológica ESPOL - RTE*, 25(1), 1-11. Recuperado el 18 de octubre de 2017, de <http://www.rte.espol.edu.ec/index.php/tecnologica/article/viewFile/95/63>
- Ros, I. (18 de julio de 2016). *Dirección MAC: qué es, por qué deberías cambiarla y cómo puedes hacerlo*. Recuperado el 21 de abril de 2018,

de <https://www.muycomputer.com/2016/07/18/direccion-mac-deberias-cambiarla/>

Ros, I. (31 de marzo de 2018). *Filtrado MAC, una buena opción para proteger tu pyme*. Recuperado el 21 de abril de 2018, de <https://www.muypymes.com/2018/03/31/filtrado-mac-proteger-pyme>

Somolinos, J. (2002). *Avances en robótica y visión por computador*. Cuenca: Ediciones de la Universidad de Castilla-La Mancha.

Superintendencia de Industria y Comercio. (2016). Registro Nacional de Bases de Datos - RNBD. Bogotá, D.C. Recuperado el 21 de abril de 2018, de <http://www.ccc.org.co/file/2016/09/Presentacion-RNBD-SIC.pdf>

Tirabassi, A. (2011). Monitoreo satelital de vehículos mediante una aplicación web. Argentina: Universidad Nacional de La Plata. Recuperado el 5 de octubre de 2017, de http://sedici.unlp.edu.ar/bitstream/handle/10915/46920/Documento_completo.pdf?sequence=1

USR.CODE. (s.f.). Ciclo de vida del Software. Implementación y Debugging, Capítulo 1. Recuperado el 25 de mayo de 2018, de <https://ingsw.pbworks.com/f/Ciclo+de+Vida+del+Software.pdf>

Westland, S. (2001). *Cómo funciona el ojo humano*. Recuperado el 20 de octubre de 2017, de http://www.gusgsm.com/funciona_ojo_humano