

Universidad Católica de Cuenca

Comunidad Educativa al servicio del pueblo



CARRERA DE INGENIERIA ELECTRÓNICA

TRABAJO DE INVESTIGACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA

TÍTULO:

**MONITOREO DE TRÁFICO VEHICULAR MEDIANTE EL TRATAMIENTO
DE IMÁGENES USANDO OPENCV**

AUTOR:

HÉCTOR ADRIÁN CARANGUI SIGUENCIA.

TUTOR:

ING. JEAN PAUL MATA QUEVEDO.

AZOGUES – ECUADOR

2018

ÍNDICE GENERAL

ÍNDICE GENERAL.....	I
ÍNDICE DE TABLAS.....	IV
ÍNDICE DE FIGURAS.....	V
AGRADECIMIENTOS.....	XII
DEDICATORIA	XIII
RESUMEN	XIV
ABSTRACT.	XV
CAPÍTULO I.....	1
PROPUESTA.....	1
1.1. Antecedentes	1
1.2. Justificación	1
1.3. Definición del problema.....	3
1.4. Objetivos	3
1.4.1. Objetivo general	3
1.4.2. Objetivos específicos.....	3
1.5. Metodología	3
CAPÍTULO II.....	5
FUNDAMENTACIÓN TEÓRICA.....	5
2.1. Antecedentes de la investigación.....	5
2.2. Visión artificial	5
2.3. Tareas que realiza la visión Artificial	6
2.3.1. Principales aplicaciones	7
2.3.2. Métodos de captación	7
2.4. Etapas de la visión por computador	7
2.5. Secuencia en la obtención de datos	7
2.6. Procesamiento de imágenes.....	9
2.7. Procesamiento de imágenes analógicas.....	10

2.8.	Procesamiento de imágenes digitales.....	11
2.9.	Sistemas de tiempo real (RTS)	12
2.10.	Características de un sistema de tiempo real (RTS)	13
2.11.	Sistemas de reconocimiento vehicular	15
2.12.	OPEN CV.....	18
2.13.	Características de OPEN CV	19
2.14.	Ventajas y desventajas de OPEN CV	19
2.15.	Funciones destacadas de OPEN CV	20
2.16.	Aplicaciones de OPEN CV.....	20
2.17.	Tarjeta electrónica LattePanda	20
2.18.	Especificaciones de LattePanda	21
2.19.	Cámara web	22
2.20.	Microsoft Visual Studio	22
2.21.	Características de Microsoft Visual Studio.....	23
2.22.	Unidad de control y reconocimiento.....	24
CAPÍTULO III		25
DISEÑO, DESARROLLO DEL SISTEMA (MOTRAVE)		25
3.1.	Requerimientos del sistema (MOTRAVE).....	25
3.2.	Definición del software	25
3.3.	Selección de la tarjeta electrónica base.....	26
3.4.	Instalación de Visual Studio en LattePanda.....	28
3.4.1.	Verificamos que el equipo esté listo para Visual Studio	28
3.4.2.	Descargar Visual Studio	28
3.4.3.	Instalación de Visual Studio	29
3.4.4.	Selección de las cargas de trabajo.....	29
3.4.5.	Selección de componentes individuales.....	30
3.4.6.	Procedimiento para instalar paquetes de idioma.....	31
3.5.	Pasos para la creación de una aplicación.....	31
3.6.	Algoritmo de reconocimiento vehicular.	34

3.7. Diseño del sistema MOTRAVE	35
3.8. Base de datos y lenguaje de desarrollo	37
3.9. Conexión del equipo	39
3.10. Diseño de la carcasa del equipo	40
CAPÍTULO IV.....	45
VALIDACIÓN E IMPLEMENTACION DEL SISTEMA DE MONITOREO DE TRÁFICO VEHICULAR (MOTRAVE)	45
4.1. Planeación para la ubicación y verificación del equipo MOTRAVE	45
4.2. Visita in situ y levantamiento de información	46
4.3. Recopilación de la información.	46
4.4. Análisis de volúmenes vehiculares en la intersección de las calles Luis Cordero y Bartolomé Serrano	48
CAPÍTULO V.....	60
CONCLUSIONES Y RECOMENDACIONES	60
5.1. Conclusiones	60
5.2. Recomendaciones	61
5.3. Trabajos futuros	62
ANEXOS	68

ÍNDICE DE TABLAS

Tabla 1: Técnicas de procesamiento.....	11
Tabla 2: Funciones OPEN CV.....	18
Tabla 3: Características de las tarjetas Raspberry Pi 2 y LattePanda.....	26
Tabla 4. Días de pruebas	47

ÍNDICE DE FIGURAS

Figura 1. Proceso de Visión Artificial.....	6
Figura 2. Secuencia en la obtención de datos.....	8
Figura 3. Ejemplo de sistema en tiempo real	13
Figura 4. Sistema de monitoreo de tráfico.....	17
Figura 5. Tarjeta electrónica LattePanda.	21
Figura 6. Funcionamiento de la unidad de control y reconocimiento.....	24
Figura 7. Página oficial de Visual Studio.	28
Figura 8. Términos y condiciones de Visual Studio.....	29
Figura 9. Activación de cargas de trabajo de Visual Basic.....	30
Figura 10. Selección de componentes individuales.....	30
Figura 11. Instalación de paquete de idioma.....	31
Figura 12. Introducción al desarrollo de Visual Basic.....	32
Figura 13. Página de inicio de Microsoft Visual Studio.....	33
Figura 14. Secuencia para la creación de un nuevo proyecto en Visual Studio.....	33
Figura 15. Selección de la aplicación de Visual Studio.	34
Figura 16. Ventana del proyecto de Visual Studio.....	34
Figura 17. Algoritmo de reconocimiento vehicular.....	35
Figura 18. Diseño del sistema de reconocimiento.....	36
Figura 19. Algoritmo de conexión de la base de datos.....	37
Figura 20. Base de datos con phpMyAdmin.....	38
Figura 21. Conexión del equipo.....	39

Figura 22. Diseño posterior de la carcasa en el software sketch up 2018. .	41
Figura 23. Diseño frontal de la carcasa en el software sketch up 2018	41
Figura 24. Diseño lateral de la carcasa en el software sketch up 2018.....	42
Figura 25. Diseño central y soporte de la pantalla de la carcasa en el software sketch up 2018.	42
Figura 26. Diseño superior e inferior de la carcasa en el software sketch up 2018.	43
Figura 27. Carcasa terminada y ensamblada.....	43
Figura 28. Carcasa con todos los componentes terminados y ensamblados.	44
Figura 29. Localización del área de estudio.	45
Figura 30. Localización específica de la implementación del equipo.	47
Figura 31. Análisis vehicular martes 15 de mayo.	48
Figura 32. Análisis vehicular miércoles 16 de mayo.....	49
Figura 33. Análisis flujo jueves 17 de mayo.	50
Figura 34. Análisis vehicular viernes 18 de mayo.	51
Figura 35. Análisis vehicular martes 22 de mayo.	52
Figura 36. Análisis vehicular jueves 24 de mayo.....	53
Figura 37. Análisis vehicular viernes 1 de junio.....	54
Figura 38. Análisis vehicular viernes 5 de junio.....	55
Figura 39. Análisis del flujo vehicular total por semana durante el mes de mayo.	56
Figura 40. Análisis del flujo vehicular total por semana durante el mes de junio.	57

Figura 41. Análisis total del flujo vehicular del estudio del mes de mayo. ...	58
Figura 42. análisis total del flujo vehicular del estudio de los días 1 y 5 de junio.	59

PERMISO DEL AUTOR DE TESIS PARA SUBIR AL REPOSITORIO INSTITUCIONAL

Yo Héctor Adrián Carangui Siguencia portador (a) de la cédula de ciudadanía Nro. 0302628847 En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación **“MONITOREO DE TRÁFICO VEHICULAR MEDIANTE EL TRATAMIENTO DE IMÁGENES USANDO OPEN CV”** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de Los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, Así mismo; autorizo a la Universidad para que realice la publicación de éste trabajo de titulación en Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, 31 de julio del 2018

F:

CESIÓN DE DERECHOS DE AUTOR

Yo, Héctor Adrián Carangui Siguencia con documento de identificación N° 0302628847 manifiesto mi voluntad y cedo a la Universidad Católica de Cuenca la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación: **“MONITOREO DEL TRAFICO VEHICULAR MEDIANTE EL TRATAMIENTO DE IMÁGENES USANDO OPENCV”**, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Católica de Cuenca, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Católica de Cuenca.

Azogues, 31 de julio del 2018

Héctor Adrián Carangui Siguencia.

CI:0302628847

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de grado, presentado por el Sr. Héctor Adrián Carangui Siguencia para optar por el título de INGENIERO ELECTRÓNICO, doy fe que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a presentación pública y evaluación por parte del jurado examinador que se designe.

En la ciudad de Azogues, a los 31 días del mes de julio del 2018.

Firma

.....
Ing.: Jean Paul Mata Quevedo

DECLARATORIA DE RESPONSABILIDAD

Yo, **Héctor Adrián Carangui Siguencia** con número de cédula de identidad 0302628847 autor del trabajo de titulación “**MONITOREO DEL TRÁFICO VEHICULAR MEDIANTE EL TRATAMIENTO DE IMÁGENES USANDO OPEN CV**” certifico que el total contenido del Proyecto Técnico, es de mi exclusiva responsabilidad y autoría

Azogues, julio del 2018

Héctor Adrián Carangui Siguencia

CI:0302628847

AGRADECIMIENTOS

Agradezco a Dios y a la Virgen que me dio la vida y la fuerza durante todas las adversidades vividas, por darme tan estupenda familia, poner a tantas personas buenas en mi camino y por enseñarme a valorar todo lo que tengo. Gracias a mi madre y mis abuelitos que han estado conmigo durante todo el trayecto, brindándome ánimo y ayuda. Agradezco a toda mi familia, y también a amigos y compañeros que me han ayudado y a todos aquellos que me han brindado una palabra de aliento.

Héctor Adrián Carangui Sigüencia

DEDICATORIA

Dedico este trabajo a mi familia que me ha apoyado incondicionalmente en todo, a mi madre y abuelitos que me han guiado desde los estudios hasta en la vida misma que han estado apoyándome siempre.

Héctor Adrián Carangui Siguencia

RESUMEN

El presente trabajo engloba el uso de visión artificial y procesamiento digital de imágenes para el monitoreo de tráfico vehicular en tiempo real, aborda el uso de dispositivos electrónicos, entre los cuales destaca la tarjeta electrónica LattePanda como cerebro del sistema, además utiliza el software Microsoft Visual Basic y las librerías de OPEN CV para cumplir con el objetivo planteado.

Se desarrolla el proyecto denominado MOTRAVE por sus siglas que significa: Monitoreo de Tráfico Vehicular, para determinar el flujo vehicular en distintas zonas de la ciudad, y que permite ser ajustable a diferentes variables del entorno a medir, esto se puede lograr mediante programación, es decir el equipo permite aislar objetos del área de censado ofreciendo distintas formas de realizar estudios. Asimismo, dispone de una base de datos la cual ayuda hacer que los datos recolectados vayan siendo almacenados por fecha y hora. El equipo desarrollado ayuda a determinar estudios de tráfico vehicular los cuales determinan la fluidez vehicular por las diferentes calles de la ciudad de Azogues.

Palabras clave: Visión artificial, Microsoft Visual Studio software, Open CV, monitoreo.

ABSTRACT.

This work is about the use of artificial vision and the digital processing of images to monitor traffic in real time, also it targets the use of electronic devices such as the electronic card Latte Panda it is the electronic brain, additionally Microsoft Visual Basic software and the OPEN CV bookshops were used to reach the objective.

The MOTRAVE project was developed, it means “Monitoreo de Trafico Vehicular” this study determines the vehicular flow in different zones of the city and permits to be adaptable to the return variable to measure, it could be reached through out the programming, it means this device can isolate objects from the census offering some ways to do studies.

It has a database, which helps that data could be stored by date and time. This study determines the traffic flow on the streets in Azogues.

KEY WORDS: Artificial vision, Microsoft visual studio software, OPEN CV. Monitoring.

CAPÍTULO I

PROPUESTA

1.1. Antecedentes

Los métodos de monitoreo vehicular en los últimos períodos se han fundamentado en sensores de luz infrarroja, magnéticos, células fotoeléctricas las cuales han sido usadas hasta la actualidad en su mayoría. Dichos métodos de monitoreo vehicular en la actualidad tratan de tener como base una técnica automatizada de conteo y categorización vehicular basados en su mayoría en sucesiones de video y algoritmos de programación, en donde se puede observar un procedimiento automático de conteo y categorización vehicular. Una de las diversas técnicas que se puede considerar son las de procesamiento de imágenes, la cual es aplicada a sucesiones de video obtenidas a través de una cámara, las que posteriormente serán utilizadas en la detección y conteo vehicular, resolviendo de esta manera el problema de clasificación vehicular (Bravo , Carrera , & Moya, 2014).

Con los avances tecnológicos y de control computarizado, se ha desarrollado la capacidad de tener un manejo automático sin la necesidad de intervención humana para el monitoreo del mismo de tal manera que se aprovecha las características de los avances tecnológicos.

En este sentido, los métodos de monitoreo vehicular con los avances tecnológicos se han orientado básicamente a la eliminación de la intervención humana, dentro de la aplicación de esos métodos de monitoreo, consiguiendo de esta manera un sistema seguro.

1.2. Justificación

Los métodos de monitoreo vehicular en los últimos períodos se han basado en sistemas tradicionales, hace poco tiempo los métodos de visión artificial

operaban información sumamente limitada, en cierto sentido. Hoy en día se puede procesar más información a mayor velocidad, por esta razón surge la necesidad de crear algoritmos para el tratamiento de imágenes. A través del tiempo se ha observado que el monitoreo de tráfico vehicular ha sido un problema, ya que por diferentes motivos ocasionan atascamientos en diferentes zonas de la ciudad lo que genera malestar en los conductores (Miranda & Encalada, 2015).

La congestión vehicular surge cuando la demanda de tráfico es mayor a la que ofrece la vía, lo que obliga a que el tiempo de viaje sea excesivo, dependiendo del transporte, lugar territorial, y el período del día, esto da a conceptuar que existen cuantiosos vehículos circulando y cada uno de ellos circulan de forma pausada e irregular (COMPILADOR, 2013).

El monitoreo vehicular es importante para determinar un estudio de tráfico, el cual ayuda a establecer una fluidez, no obstante, el congestionamiento en zonas urbanas es mayor, esto origina atrasos y pérdidas de tiempo. La existencia de varios vehículos ocasiona que las calles y avenidas se saturen, generando ruido, contaminando el medio ambiente, causando malestar en los transeúntes y dificultando el acceso a diferentes puntos de la ciudad (Jerez & Morales, 2015).

Uno de los factores que afecta al flujo vehicular en la ciudad es la distribución del transporte público, ya que ninguna entidad controla la fluidez de este medio a través de ella, el cual origina atascamientos, la coincidencia de varias unidades en un mismo punto, la infraestructura o quizá porque la ciudad no se desarrolló de una manera adecuada, para que los vehículos puedan transitar de una mejor forma (Municipio del Distrito Metropolitano de Quito, 2015). Así mismo, hay que considerar que el flujo vehicular en una vía depende del nivel de cultura de las personas, el cumplimiento de estas normativas beneficia la aptitud de vida de los individuos, el tráfico y la

economía, brindando un ambiente agradable o todos los que forman parte de ella.

1.3. Definición del problema

El monitoreo de tráfico vehicular es el rastreo y determinación de objetos en movimiento, lo cual admite el uso de la visión artificial en aplicaciones útiles y competentes, en estudios de monitoreo vehicular, lo cual es posible ajustar de manera automática y a un bajo costo. Esto ayudaría a mejorar los procedimientos de señalización en función de las peculiaridades del tráfico vehicular (Bravo , Carrera , & Moya, 2014).

1.4. Objetivos

1.4.1. Objetivo general

Elaborar un modelo de plan el cual monitoree el tráfico vehicular, mediante el tratamiento de imágenes usando Microsoft Visual Studio y librerías de OPEN CV.

1.4.2. Objetivos específicos

- Investigar sobre aportes teóricos y metodológicos acerca sistemas de monitorio a base de tratamiento de imágenes.
- Desarrollar el prototipo de monitoreo vehicular usando el tratamiento de imágenes con OPEN CV.
- Comprobar el funcionamiento del sistema de monitoreo vehicular.

1.5. Metodología

Se utiliza el método analítico – sintético, ya que se analizará a diversos autores de tal manera que se resuman todas las ideas que puedan ser útiles para tener solidez en la información y el uso de método histórico debido a que

gran parte de la información se encuentran en textos y trabajos de titulación en donde exponen sus experiencias en el transcurso de la investigación.

En la segunda fase del trabajo de titulación, el método utilizado en el desarrollo es sistémico, analítico-sintético. El cual es un método que permite conocer más el objeto de estudio. El método sintético da la posibilidad de hacer un resumen de todas las ideas fundamentales respecto al tema de estudio. Y para el nivel práctico será la utilización del software Microsoft Visual Basic con librerías de OPEN CV a base de tratamiento de imágenes.

En la parte final del trabajo de investigación se utiliza el método inductivo – deductivo. Luego del desarrollo completo de las primeras partes se encontrarán las respuestas a las preguntas que fueron planteadas por el investigador, debido a que la deducción es afín a la inducción, con la cual se obtendrán las conclusiones y recomendaciones pertinentes sobre el trabajo.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA

2.1. Antecedentes de la investigación

Este capítulo dará a conocer los temas que intervienen en el desarrollo del proyecto, como son: visión artificial, la importancia de OPEN CV, tarjeta electrónica, software y cámara a usarse de manera que estén citados puntualmente y expliquen en que intervienen en este proyecto.

2.2. Visión artificial

Tiene por objetivo simular el comportamiento de un sistema visual humano, en este aspecto se concierne métodos para adquirir, procesar, analizar y comprender imágenes del mundo exterior, para un posterior análisis por medio de un elemento tecnológico, y así desempeñar diversas funciones de acuerdo a los datos obtenidos (Cantero & Acides, 2016).

La visión por computador se desarrolla con el objetivo de automatizar tareas cíclicas y repetitivas, para realizar el seguimiento de un objeto en específico, detección de gestos, conteo de productos, entre otros (Mathworks, 2015).

Según Merchán (2010) la visión artificial es la sustitución del sentido visual humano para realizar una tarea de inspección, mediante la adquisición y análisis automático de imágenes para obtener los datos deseados, ya sea para monitorear o evaluar una parte o actividad en específico.

La visión artificial sigue un proceso, el cual se lo muestra en la **Figura 1** a continuación.

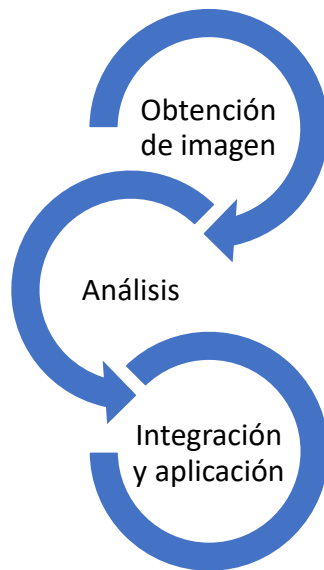


Figura 1. Proceso de Visión Artificial

Fuente: (Merchán, 2010)

Elaborado por: El Autor.

El rastreo y determinación de cuerpos en movimiento consiente el uso de la Visión Artificial, admitiendo ajustes y mejoras a un bajo costo, los sistemas de señalización en función de las peculiaridades que se obtengan de los estudios de tráfico vehicular.

La Visión Artificial es definida como un campo de la inteligencia artificial, admite la obtención, proceso y análisis de información a través de imágenes, en la actualidad es de gran utilidad en la industria ya que reduce el ciclo en procesos industrializados.

A continuación, se detalla algunas de las funciones que realiza la Visión Artificial.

2.3. Tareas que realiza la visión Artificial

- Automatiza tareas repetitivas.
- Realiza el control de calidad de productos.
- Inspección de objetos sin contacto físico.

- Inspección a grandes velocidades.

2.3.1. Principales aplicaciones

- Identifica e inspecciona objetos.
- Determinación de la posición de objetos.
- Relaciones espaciales entre varios objetos.
- Mediciones angulares.

2.3.2. Métodos de captación

- Digital.
- Pixel.
- Nivel de grises.
- Imagen binaria.
- Escena.
- Window (ventana de medida)

2.4. Etapas de la visión por computador

- **Localización de borde de la imagen** Un borde es una parte de la imagen a través de la cual la intensidad lumínica cambia de manera abrupta.(Cancelas, González, Álvarez, & Enguita, 2016)
- **Segmentación de la imagen por regiones.** Una región es una parte de la imagen en la que los cambios de intensidad lumínica son graduables.(Cancelas et al., 2016)

2.5. Secuencia en la obtención de datos

La **Figura 2.** Secuencia en la obtención de datos. Muestra el método con el cual se desarrolla la obtención de datos.



Figura 2. Secuencia en la obtención de datos.

Fuente: (visionartificial.pdf, 2010)

Elaborado por: El Autor.

Los modelos avanzados que utilizan visión por ordenador se desarrollan generalmente en física (radiometría, óptica y diseño de sensores) y en gráficos por ordenador. Ambos campos modelan la forma en que los objetos se mueven y animan, la forma en que la luz irradia en sus superficies, se dispersa por la atmósfera, se refracta a través de lentes de cámara y finalmente se proyecta en un plano de imagen. Como humanos, percibimos la estructura tridimensional del mundo que nos rodea con aparente facilidad. Mirando un retrato de grupo enmarcado, usted puede fácilmente contar (y nombrar) a todas las personas en la imagen e incluso adivinar sus emociones por su apariencia facial (Burgo, 2014).

En la visión por computador, se trata de hacer lo inverso, es decir, describir el mundo que vemos en una o más imágenes y reconstruir sus propiedades, como la forma, la iluminación y las distribuciones de color. Algo asombroso que se puede mencionar es que los seres humanos y los animales lo realizan tan fácilmente, mientras que los algoritmos de visión computarizada son tan propensos a errores (Ramos, Vargas, & Aceves, 2015).

La identificación de vehículos en una sola instantánea proporcionaría mediciones directas de la densidad del vehículo en el momento de la imagen.

En ausencia de algoritmos de comparación de vehículos, podría sustituirse una velocidad media estimada de la autopista por velocidades individuales de cada vehículo. Acoplar las velocidades de la autopista con densidades imaginadas generaría estimaciones del flujo de tráfico en el segmento (Echaveguren, Díaz, & Arellano, 2013).

2.6. Procesamiento de imágenes

El procesamiento de imágenes utiliza operaciones matemáticas sobre imágenes digitales, como una fotografía o video, con la finalidad de mejorarla, utilizando filtros para eliminar los diferentes tipos de distorsiones que puedan presentar ya sea por la intensidad lumínica que cambia de una manera abrupta, o la segmentación de imagen las cuales son graduables (Castillo & Hernandez, 2013).

La tecnología digital moderna ha hecho posible la manipulación de señales multidimensionales con sistemas que van desde simples circuitos digitales hasta computadoras paralelas avanzadas (Šípal, 2012).

Por lo tanto, es importante comprender las características y limitaciones del sistema visual humano para comprender el "receptor" de las señales 2D. Al principio es importante darse cuenta de que:

- el sistema visual humano no es bien comprendido,
- no existe ninguna medida objetiva para juzgar la calidad de una imagen que corresponda a la evaluación humana de la calidad de la imagen, y,
- no existe el observador humano "típico". Sin embargo, la investigación en psicología perceptiva ha proporcionado algunas ideas importantes sobre el sistema visual.

El procesamiento de imágenes es el estudio de cualquier algoritmo que toma una imagen como entrada y devuelve una imagen como salida. Esto incluye,

visualización, edición, manipulación, mejora, detección de características y compresión de imágenes. El procesamiento de imágenes es una técnica la cual permite mejorar las imágenes crudas recibidas de cámaras tomadas en la vida cotidiana normal para diversas aplicaciones (Jimenez, 2018).

Durante las últimas cuatro a cinco décadas se han desarrollado varias técnicas en el Procesamiento de Imágenes. La mayoría de las técnicas se desarrollan para mejorar las imágenes obtenidas. Los sistemas de procesamiento de imágenes son cada vez más populares debido a la fácil disponibilidad de potentes ordenadores personales, dispositivos de memoria de gran tamaño, software de gráficos, etc. (Salvador & Moctezuma, 2015).

El procesamiento de imágenes se utiliza en varias aplicaciones como:

- Teledetección.
- Imágenes médicas
- Evaluación no destructiva
- Estudios forenses
- Textiles - Textiles
- Ciencia de los Materiales.
- Militar
- Industria cinematográfica
- Procesamiento de documentos
- Artes gráficas
- Industria gráfica

Actualmente, existen dos métodos disponibles en el Procesamiento de Imágenes.

2.7. Procesamiento de imágenes analógicas

El procesamiento de imágenes analógicas se refiere a la alteración de la imagen por medios eléctricos. El ejemplo más común es la imagen de televisión. La señal de televisión es un nivel de voltaje que varía en amplitud para representar el brillo a través de la imagen. Al variar eléctricamente la señal, se altera la apariencia de la imagen visualizada. Los controles de brillo y contraste de un televisor sirven para ajustar la amplitud y referencia de la señal de vídeo, lo que resulta en el brillo, oscurecimiento y alteración del rango de brillo de la imagen visualizada (Alvarado, 2011).

2.8. Procesamiento de imágenes digitales

Se define como la representación numérica de objetos en una serie de operaciones para obtener un resultado deseado. Comienza con una imagen y produce una versión modificada de la misma. Por lo tanto, es un proceso que lleva una imagen a otra. En este caso, se utilizan ordenadores para procesar la imagen. La imagen se convertirá a formato digital utilizando un escáner - digitalizador y luego se procesará (Chato, 2017).

La principal ventaja de los métodos de procesamiento digital de imágenes es su versatilidad, repetitividad y la conservación de la precisión de los datos originales.

Las diversas técnicas de procesamiento de imágenes son:

Tabla 1: Técnicas de procesamiento.

Representación de la imagen	Representación de la imagen Una imagen definida en el "mundo real" se considera una función de dos variables reales, por ejemplo, $f(x, y)$ con f como la amplitud (por ejemplo, el brillo) de la imagen en la posición real de coordenadas (x, y) .
-----------------------------	--

Escala	El tema de la técnica de ampliación es tener una visión más cercana mediante la ampliación o el aumento de la parte interesada en las imágenes. Mediante la reducción, podemos llevar el tamaño inmanejable de los datos a un límite manejable.
Aumento	Esto se hace generalmente para mejorar la escala de visualización para interpretación visual o, a veces, para ajustar la escala de una imagen a otra.
Reducción	Para reducir una imagen digital a los datos originales, se seleccionan y visualizan cada fila y cada columna de las imágenes originales.
Rotación	La rotación se utiliza en mosaico de imágenes, registro de imágenes, etc.
El mosaico	Es un proceso de combinar dos o más imágenes para formar una sola imagen grande sin desequilibrio radiométrico.

Elaborado por: El Autor.

Fuente: (Chato, 2017)

2.9. Sistemas de tiempo real (RTS)

Es cualquier sistema de procesamiento de información que responde a estímulos de entrada, generados externamente dentro de un instante de tiempo definido y especificado. Un RTS se define como un sistema en el tiempo en el que se producen los resultados es significativo. Los resultados deben producirse dentro de unos plazos determinados. La exactitud de un RTS no sólo depende de los resultados lógicos que se produzcan, sino también de los tiempos en los que se produzcan.

“El sistema puede introducir un estado incorrecto si el resultado correcto se produce demasiado pronto o demasiado tarde con respecto a los límites de tiempo o plazos especificados” (Zhañay, 2015).

La **Figura 3** a continuación muestra una representación en forma de diagrama de bloques de un ejemplo de un RTS. Las entradas pueden provenir de fuentes como sensores, interruptores, enlaces de comunicación o un teclado y las salidas pueden ir a actuadores, alarmas, motores, lámparas, enlaces de comunicación o pantallas. Las salidas del sistema tienen límites de tiempo asociados dentro de los cuales deben ser producidas (Zhañay, 2015).

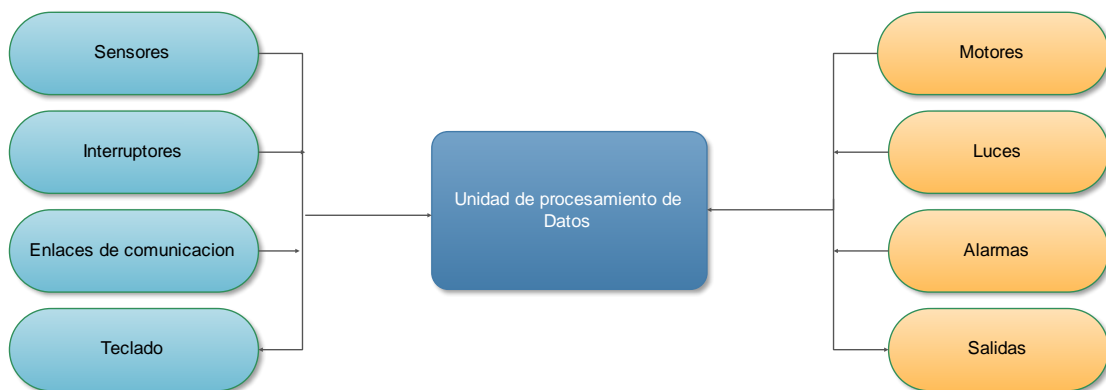


Figura 3. Ejemplo de sistema en tiempo real
Fuente: (Zhañay, 2015)
Elaborado por: El Autor

2.10. Características de un sistema de tiempo real (RTS)

Las siguientes características se encuentran a menudo en muchos RTS, sin embargo, un sistema no necesita tener todas estas características para ser un RTS.

Un RTS se usa dentro de un sistema grande para proporcionar funciones de monitoreo, control y computación. Estos sistemas se denominan "sistemas informáticos integrados". A menudo contienen dispositivos que actúan como los sentidos (por ejemplo, sensores de calor o sensores de luz) y dispositivos

que actúan como el efecto de los cambios físicos (por ejemplo, actuadores mecánicos, electromecánicos y electrónicos). El sistema se construye generalmente en torno a un microcontrolador utilizado para ejecutar diversas funciones de monitorización y control del sistema. El sistema lee la entrada de las cámaras digitales, aplica varios algoritmos de reconocimiento aplica filtrado, calibración y procesamiento en los datos de entrada y produce datos de salida a hacia la base en los que va a ser almacenado. (Miranda & Encalada, 2015).

Los sistemas RTS a menudo requieren el procesamiento concurrente de múltiples entradas. Esto implica el procesamiento correlacionado de múltiples entradas en el mismo intervalo de tiempo, un ejemplo es, un sistema de control de proceso industrial podría ser necesario para correlacionar valores de temperatura, presión y concentración de una reacción química para realizar ajustes simultáneos de calentadores y válvulas para mantener una reacción en el estado deseado (González, 2016).

Se deben crear y gestionar tareas concurrentes para cumplir las funciones de un RTS. La programación de tareas es uno de los aspectos importantes de la gestión de la concurrencia. Dado que las tareas competirán por los mismos recursos, como una CPU (en un sistema de uni-procesador), es necesario utilizar una programación prioritaria en la que las tareas con plazos más estrictos tendrán una prioridad mayor para ser programadas en la CPU antes que otras tareas de menor urgencia (UNED, 2011).

Las escalas de tiempo de muchos sistemas en tiempo real son rápidas según los estándares humanos. Los dispositivos complejos monitoreados o controlados a menudo funcionan en escalas de tiempo rápidas, a manera de ejemplo, un sistema de control de crucero de automóviles para mantener una conducción suave con sólo pequeñas variaciones de la velocidad deseada, la velocidad real debe ser monitoreada muchas veces por segundo.

La precisión de respuesta requerida para un RTS es mayor que la requerida por otros sistemas. Una respuesta temprana o tardía puede constituir un comportamiento erróneo. Un cierre prematuro de una planta química podría causar daños extensos al equipo o daños ambientales.

Los RTS tienen requisitos de fiabilidad y seguridad superiores a los que exigen otros sistemas. El fracaso de un sistema involucrado en la transferencia automática de fondos entre bancos puede llevar a que se pierdan millones de dólares, el fracaso en un sistema embebido podría resultar en el fracaso de un sistema vital de soporte vital (Salvador & Moctezuma, 2015).

2.11. Sistemas de reconocimiento vehicular

El sistema de vigilancia de tráfico vehicular es un tema de investigación activo en la visión por ordenador ya que trata de detectar, y reconocer los vehículos a través de una secuencia de imágenes, sustituyendo el viejo método tradicional de vigilancia de las cámaras operadas por controladores humanos (Albusac, 2008).

Un sistema de visión computarizada puede monitorear tanto el comportamiento inmediato no autorizado como el comportamiento sospechoso a largo plazo, y por lo tanto alerta al operador humano para una investigación más profunda del evento. El sistema de video vigilancia puede ser manual, semiautomático o totalmente automático dependiendo de la participación humana (Ferro, 2010).

Estos sistemas están demostrando ser ineficaces para lugares muy ocupados, ya que el número de afluencia de vehículos excede el número de cámaras. Estos sistemas están muy extendidos en todo el mundo. El sistema semiautomático de vigilancia del tráfico requiere la ayuda tanto del operador humano como de la visión por ordenador. El objeto es censado por el

algoritmo de visión por computador y el trabajo de calibración del equipo es realizado por el operador. (Sucar, 2010).

En estos sistemas se utiliza un nivel bajo de procesamiento de vídeo, y gran parte de la tarea se realiza con la ayuda de la intervención del operador humano. En el sistema totalmente automático no hay intervención humana y todo el trabajo lo hace la visión por ordenador (Perez E. , 2015). Un sistema de vigilancia típico consiste en una red de cámaras de tráfico, que procesa el vídeo de tráfico capturado y transmite los parámetros extraídos en tiempo real.

Generalmente estos tipos de sistemas constan principalmente de tres patrones:

- Adquisición de Imagen
- Detección de vehículos
- Unidad de reconocimiento de caracteres

Este sistema se utiliza para detectar y contar coches en marcos de imágenes en escenas dinámicas y luego extraerlas. Se ha encontrado numerosas aplicaciones tan amplias como ha sido posible, tales como: control de accesos en áreas sensibles a la seguridad, seguridad para comunidades y edificios importantes, detección de áreas objetivo militares, vigilancia de tráfico en ciudades y carreteras, detección de comportamientos anómalos, gestión de control de tráfico para el reconocimiento de vehículos que cometen infracciones de tráfico, tales como la ocupación de carriles reservados para el transporte público, la superación de límites de velocidad, el cruce de semáforos en rojo, la entrada a áreas restringidas sin permiso y entre muchas otras aplicaciones (Dolores, 2010).

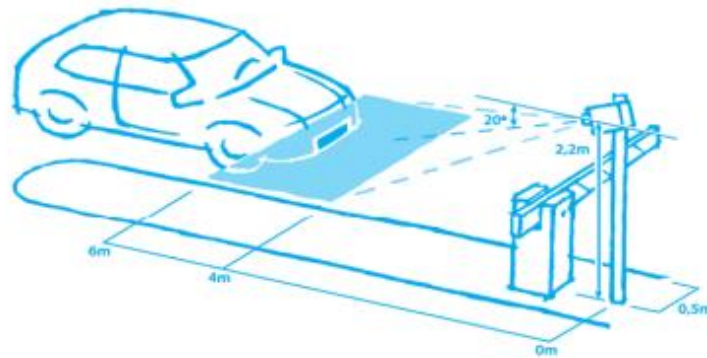


Figura 4. Sistema de monitoreo de tráfico
Fuente: (Dolores, 2010)

La vigilancia del tráfico es utilizada por empresas privadas, gobiernos y organismos públicos para la gestión eficiente de las redes de transporte, la seguridad vial, la seguridad pública en carreteras y calles transitadas. La identificación de objetos intrusos es un paso importante en el análisis de la escena y la segmentación exitosa del objeto en primer plano en movimiento desde el fondo asegura la clasificación de los objetos, la identificación del vehículo, el seguimiento y el análisis de la actividad, haciendo que estos últimos pasos sean más eficientes (Chato, 2017).

La tarea de la segmentación de objetos en movimiento es extraer información significativa sobre el vehículo en movimiento a partir de secuencias de vídeo, lo que proporciona la comodidad de la representación basada en objetos y la manipulación del contenido de vídeo en tiempo real.

Los enfoques convencionales para la segmentación de objetos en movimiento incluyen métodos de diferencia de fotogramas, métodos de sustracción de fondo y métodos de flujo óptico. La estimación y segmentación en primer plano es la primera etapa de varios sistemas de vigilancia del tráfico. Las regiones de primer plano están marcadas (ej., imagen de máscara) para el procesamiento en los pasos siguientes. Existen dos enfoques principales para estimar los conocimientos previos, que utilizan fuertes supuestos para cumplir con la definición antes mencionada. En primer lugar, se puede utilizar un

modelo de fondo de algún tipo para acumular información sobre el fondo de escena de una secuencia de vídeo (Perez E. , 2015).

2.12. OPEN CV

Es la abreviatura de Open Source Computer Vision library, el cual se usa para el procesamiento de imágenes, está compuesto por varios grupos de funciones, aplicaciones desarrolladas en lenguaje C y algunas clases en C++. El objetivo de OPEN CV es reunir librerías y código que permitan procesar imágenes y mediante este lenguaje de programación tener visión artificial. Las aplicaciones que brinda este recurso son ilimitadas, cada día las programaciones crecen de manera inimaginable, el desarrollo, su conexión y compartirlas es una de sus grandes ventajas. La librería OPEN CV puede ser utilizada en diferentes plataformas como Windows, Linux, Mac OS, iOS y Android, está diseñado con un enfoque en las aplicaciones de tiempo real, por la cual esta librería se divide en cinco grandes grupos en donde se encuentran las funciones dependiendo del uso y funcionalidad de las mismas, las cuales se mencionan a continuación (OpenCV, 2018).

Tabla 2: Funciones OPEN CV

CXCORE	Donde se encuentran las estructuras y algoritmos básicos que usaran las demás funciones como son el manejo de memoria, errores y demás aspectos del algebra básica y matricial.
CV	Computer Vision, donde se encuentran implementadas las funciones principales de procesamiento de imágenes y algoritmos de visión por computadora.
HIGHUI	Donde se encuentra la interfaz gráfica de Open CV y las funciones de guardado y carga de videos e imágenes

ML	Machine Learning, donde se encuentran los algoritmos de aprendizaje
CV AUX	Donde se encuentran las funciones experimentales como la detección de rostros, visión estereoscópica, entre otros.

Elaborado por: El Autor.

Fuente: (OpenCV, 2018)

2.13. Características de OPEN CV

- No usa librerías numéricas externas, aunque puede hacer uso de alguna si está disponible.
- Es compatible con Intel Process Library (IPL).
- Dispone de interfaces para algunos métodos de programación y entornos.
- Es de uso libre tanto en el área comercial como no comercial.

2.14. Ventajas y desventajas de OPEN CV

Ventajas

En la mayoría de los casos consigue un ratio de comprensión, ofreciendo una calidad mucho mejor que en otros formatos, además su nivel de comprensión es ajustable típicamente entre 1: 10 y 1: 100. (La, Investigación, Arévalo, González, & Ambrosio, 2002)

Desventajas

- Posee compresiones / descompresiones complejas y costosas.
- No incluye transparencias ni aplicaciones.
- La información perdida no se recupera. (La et al., 2002)

2.15. Funciones destacadas de OPEN CV

- Cv WarpPerspective. Función que transforma una imagen origen mediante una matriz de mapeo.
- Cv Size. Estructura que contiene las dimensiones de una imagen se emplea para asignar valores al campo correspondiente de una imagen.
- Cv LoadImage. Lee una imagen desde una memoria no volátil de forma que pueda ser procesada por el conjunto de funciones de la librería.
- Cv SaveImage. Guarda en la memoria no volátil la imagen indicando que se encuentra en memoria.
- CvUndistort2. Calibra una imagen y la almacena en ImCalib basándose en la matriz de los parámetros intrínsecos de la cámara y en el vector de coeficientes de distorsión. (Leal, Leal, & Br, 2010)

2.16. Aplicaciones de OPEN CV

- Es usado en sistemas de visión de vehículos no tripulados.
- Se usa en sistemas de video vigilancia.
- Es la clave para el reconocimiento rastreo de objetos, análisis estructural y operaciones básicas, interfaces gráficas y adquisición de video. (La et al., 2002)

2.17. Tarjeta electrónica LattePanda

Considerada como un mini ordenador completo con Windows 10 en una sola tarjeta, que cabe en la mano nos permite interactuar con Arduino dentro del mismo hardware a través de un coprocesador ATmega32u4. Es compatible casi con todos los aparatos como impresoras, cámaras y más, todas las aplicaciones del computador trabajan de forma normal en dicho dispositivo, incluye potentes herramientas como Visual Studio, Java, y más. Principalmente está formada por un chipset Intel Atom-Cherry Trail, capaz de poner a funcionar sus cuatro núcleos de 1.8Ghz que está muy bien flanqueado

por 4Gb de RAM y hasta 64 Gb en formato micro SSD.(Titulación, La, Del, Ingeniero, & Electrónica, 2016)

En cuanto a conectividad la placa cuenta con Wi-Fi, Bluetooth 4.0, salida HDMI, Ethernet e incluso un puerto USB 3.0 y dos puertos de 2.0, dispone de un puerto de 3.5 mm para conectar altavoces. La compañía que está detrás de esta tarjeta es la china DFRobot que pensó que Raspberry Pi o Intel eran poca cosa para crear una placa compatible con Arduino (LattePanda, Studio, 2017).

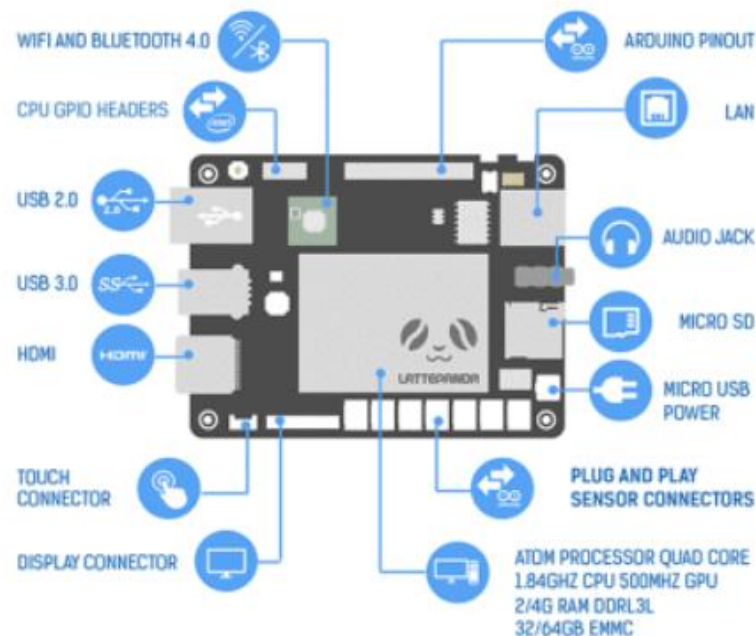


Figura 5. Tarjeta electrónica LattePanda.

Fuente: (LattePanda, Studio, 2017)

2.18. Especificaciones de LattePanda

- Procesador Intel Cherry Trail.
- Frecuencia base 1.44GHz
- Frecuencia ráfaga 1.92GHz
- Sistema operativo Windows 10
- RAM 4GB DDR3L

- Almacenamiento en micro SSD 64GB
- GPU Intel HD graphics, 12EUs @ 200-500MHz memoria de un solo canal.
- USB 3.0 uno
- USB 2.0 dos
- WI-FI 802.11n 2.4G
- Bluetooth 4.0
- Co-procesador ATmega32u4.
- Salida de video HDMI y MIPI-DSI
- Conector superpuesto de panel táctil integrado.
- Admite ethernet de 100Mbps.
- Procesador Intel GPIO seis
- ATmega Procesador GPIO veinte.
- Conector de interfaz de gravedad seis
- Voltaje 5v a 2 A.
- Dimensiones de la placa 88 x 70 mm.
- Peso neto 55g.(LattePanda, Studio, 2017)

2.19. Cámara web

La cámara web cuenta con el sistema de conexión plug and play, sin requerir un controlador de dispositivo, con una resolución de 640 x 480 pixeles/30fps, el cual soporta Windows 8/7/10/vista/xp SP2. Para que la cámara funcione de manera correcta el sistema requiere por lo mínimo un procesador Intel 1.6GHz o AMD, un espacio de memoria RAM de 512MB y de 200MB. La resolución digital. Hace referencia la numero de bits que se usa para codificar los niveles de intensidad luminosa.(Cancelas et al., 2016).

2.20. Microsoft Visual Studio

Visual Studio es un lenguaje de programación dirigido a eventos, el cual utiliza un ambiente de desarrollo netamente basada en dialecto Basic con importantes agregados, aunque provee facilidades para el desarrollo de

aplicaciones y base de datos, contiene un desarrollo integrado o IDE que integra editor de textos para la edición del código fuente, un depurador, compilador y enlazador, además de un editor de interfaces graficas o GUI. (Microsoft visual basic, 2017).

2.21. Características de Microsoft Visual Studio

Los compiladores generan código que requieren librerías de enlace dinámico para que funcione, conocidas comúnmente como DLL de las siglas (Dynamic-Link library) o reside con el archivo de nombre MSVBVMxy.DLL (siglas de Microsoft Visual Basic Virtual Machine x.y, donde (x.y es la versión) ya que estas librerías contienen rutinas de código ejecutable que son cargadas en tiempo de ejecución. Además, permite dentro del IDE (Entorno de Desarrollo Integrado) de visual Studio ejecutar el programa que se está desarrollando es decir compila el programa rápidamente lo ejecuta simulando la función de un intérprete puro. (Microsoft visual basic, 2017).

“ Existe un único entorno de desarrollo para Visual Studio desarrollado por Microsoft de nombre Microsoft Visual Basic x.0, correspondientes a las versiones desde la 2.0 hasta la 20.0 con sus respectivas diferencias entre versiones de lenguaje. ” (Microsoft visual basic, 2017).

En la parte superior del software aparecen tres elementos de la siguiente manera, la barra de título donde figura el nombre del proyecto en curso, su estado o ejecución la barra de menú y una barra de herramientas, esta última se puede personalizar posibilitando los comandos del IDE. En la central ocupando la parte de mayor área se encuentra el sitio de programación, este incluye y muestra ventanas del proyecto las vistas del código fuente, objetos, módulos y los controladores, que contienen las ventanas de la aplicación y el panel de control, este último normalmente se encuentra ubicado al lado derecho. (Microsoft visual basic, 2017)

2.22. Unidad de control y reconocimiento

Es el encargado de recibir la información en este caso de las cámaras, para procesarla y luego tomar una decisión ya sea para almacenarla o ignorarla, dicha unidad es la de mayor importancia del proyecto ya que todo funciona en base a las decisiones tomadas por este, en la **Figura 6**. Funcionamiento de la unidad de control y reconocimiento. Se da a conocer el funcionamiento de la unidad de control.

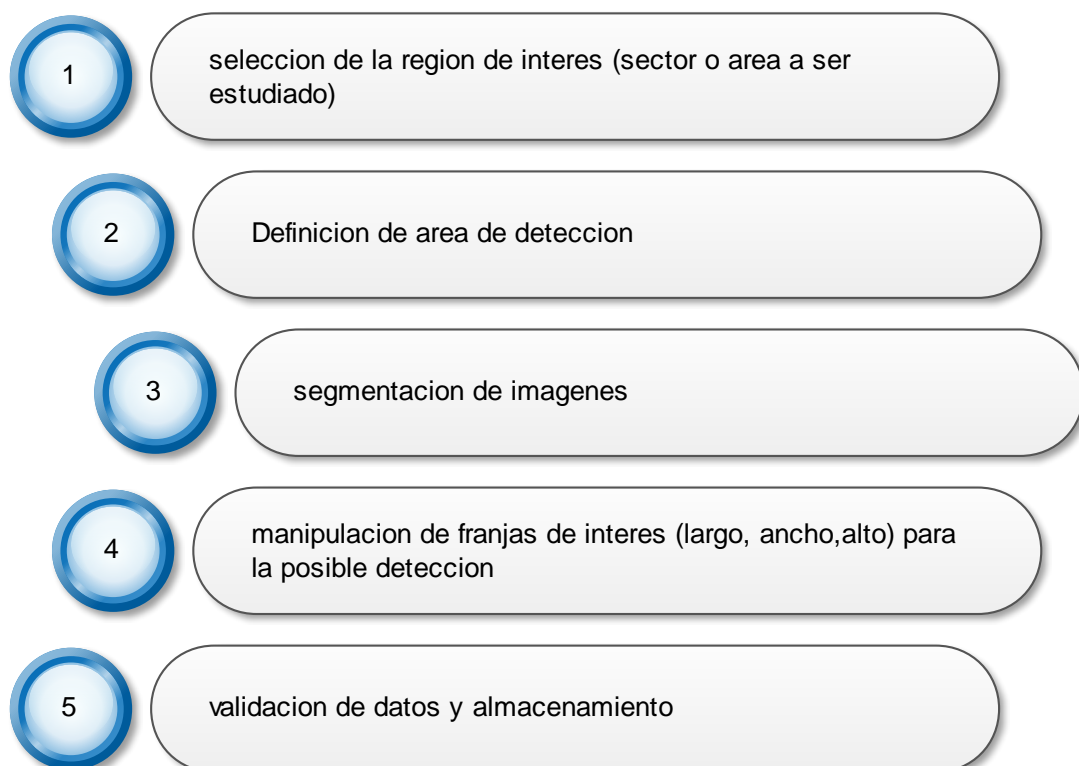


Figura 6. Funcionamiento de la unidad de control y reconocimiento.

Fuente:(Base & Franjas, 2010)

Elaborado por: El Autor.

CAPÍTULO III

DISEÑO, DESARROLLO DEL SISTEMA (MOTRAVE)

El presente capítulo tiene como finalidad describir el diseño desarrollo e implementación del equipo (tarjeta electrónica LattePanda, pantalla y táctil, baterías, regulador de corriente y voltaje, cámara, carcasa), así como el sistema de reconocimiento y la interacción con la base de datos. Se ha creído conveniente dividirla dos secciones.

En una primera sección, se discute sobre el diseño y desarrollo del equipo, donde se da a conocer la forma en la cual se lleva a cabo la instalación del software a usarse, el desarrollo del algoritmo de reconocimiento y la interacción con la base de datos; además se explica el diseño de la carcasa y el procedimiento de ensamblaje.

En la sección complementaria se da a conocer la metodología usada para la implementación del equipo, las condiciones y características en las que opera el equipo para obtener los resultados de las pruebas pertinentes.

3.1. Requerimientos del sistema (MOTRAVE)

El propósito del proyecto es desarrollar un sistema capaz de censar, reconocer, validar y almacenar información de tráfico vehicular en tiempo real, de forma independiente al entorno al cual se le asigne realizar determinada tarea, Para ello se describe el proceso para cumplir con dicho objetivo.

3.2. Definición del software

Para el desarrollo del proyecto se utiliza Microsoft Visual Studio, debido a que este software admite librerías y desarrollo de OPEN CV, además facilita el procesamiento de imágenes en tiempo real. En si es un entorno de desarrollo para sistemas operativos de Windows el cual soporta múltiples lenguajes de

programación tales como c, c++, Visual Basic.NET, Java, Python, Ruby, y Php al igual que entornos de desarrollo Web. Las aplicaciones creadas están desarrolladas mediante objetos y son manejadas a través de eventos, las cuales no siguen una ruta predefinida, lo que hace que el programa ejecute diferentes secciones de código en respuesta a eventos. (Basic, 2006)

3.3. Selección de la tarjeta electrónica base

Para definir la tarjeta electrónica a usarse, se ha visto conveniente realizar una comparación entre las tarjetas electrónicas Raspberry Pi 2 y LattePanda, que son las tarjetas más comunes dentro de nuestro medio y que cumplen las características básicas para el desarrollo del proyecto. La **Tabla 3:** Características de las tarjetas Raspberry Pi 2 y LattePanda muestra la el análisis técnico – comparativo.

Tabla 3: Características de las tarjetas Raspberry Pi 2 y LattePanda

Características	Raspberry Pi 2	Latte Panda
Procesador.	ARM Cortex-A3 Quad core 900Mhz	Intel charry trail Quad core 1.8Ghz.
Sistema operative	Linux/Android	Windows 10
Cache	128KB	2MB
Ram	1GB DDR2	4GB DDR3L
Capacidades Graficas	Broadcom video core	Intel gen8
Soporte de video	1080p30H264/mpeg- 4	1080p60HEVC DECODE H264,VP8
Salida de video	HDMI	HDMI conector Display
Almacenamiento	No posee	64 GB
Micro SDD	SI ADMITE	SI ADMITE
CO- PROSESADOR	No posee	Atmega32u4

Puerto serial	1	1 puerto para chip Intel 1 puerto para arduino.
Interfaces	2 conectores SPI	2 conectores para chip Intel. Conector de pantalla y tactil 3.5mm Puerto de audio
Proposito general	17	2 GPIOs para chip Intel 20 GPIOs para Arduino.
Fuente de alimentación.	5v	5v
Activación de LAN	Si admite	Si admite
Dimensiones	3.37"x2.22"	3.46"x2.76"
USB	4x2.0	1x3.0 2x2.0
Ethernet	RJ45	RJ45
MODULO WI-FI	No posee	Si
Bluetooth	No posee	Si

Fuente:(LattePanda, Studio, 2017)
Elaborado por: El Autor.

De acuerdo a los resultados mostrados y haciendo un análisis respecto a los requerimientos, se puede concluir que la tarjeta electrónica a utilizarse en este proyecto es LattePanda, debido a que el sistema operativo que admite es Windows 10, y este a su vez permite la instalación y correcto uso de Microsoft Visual Studio, está formada por un chipset Intel Atom-Cherry Trail, capaz de poner a funcionar sus cuatro núcleos de 1.8Ghz permitiendo aprovechar en su totalidad la tarjeta, está muy bien flanqueado por 4Gb de RAM la cual ayuda al procesamiento de datos además soporta una micro SSD de hasta 64 Gb la cual nos sirvió para almacenar la información. Para la utilización de las cámaras usamos un puerto USB 3.0 y un puerto 2.0.

3.4. Instalación de Visual Studio en LattePanda

Para la instalación de Visual Studio en la tarjeta base, nos basamos en el siguiente procedimiento:

3.4.1. Verificamos que el equipo esté listo para Visual Studio

De acuerdo a (“Instalar Visual Studio,” 2018) es necesario que se verifique las siguientes características de soporte del equipo.

- Soporte el programa a instalar, para esto hay que revisar los requisitos del sistema y si el equipo es compatible con el Visual Studio a instalar.
- Revisar las actualizaciones de Windows ya que estas garantizan, que el equipo tiene las actualizaciones de seguridad más reciente y componentes necesarios.
- Reiniciar el equipo garantiza la instalación correctamente.
- Asegurarse que estén habilitadas únicamente las aplicaciones necesarias a usarse.

3.4.2. Descargar Visual Studio

Para descargar accedemos a la siguiente dirección.

<https://www.visualstudio.com>

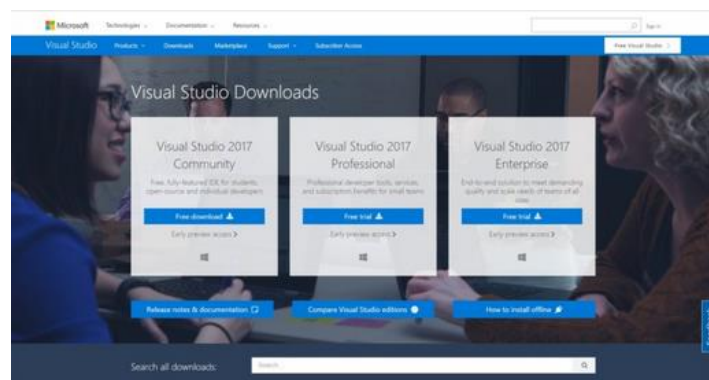


Figura 7. Página oficial de Visual Studio.
Fuente: (“Instalar Visual Studio,” 2018)

Luego seleccionamos la versión de visual Community2017 y procedemos a descargar.

3.4.3. Instalación de Visual Studio

Una vez descargado procedemos a la instalación ejecutando el archivo de programa como administrador, este instalador incluye todo lo necesario para personalizar. Desde la carpeta de descargas hacemos doble clic en el archivo de programa previo que coincida o sea similar con el siguiente archivo.

➤ **Vs_community.exe** para Visual Studio Community.

Luego recibimos un aviso de control de cuenta de usuario, el cual aceptamos y continuamos con la instalación. Posteriormente nos pidió que aceptemos términos de licencia y declaración de privacidad como muestra la **Figura 8.Términos** y condiciones de Visual Studio.

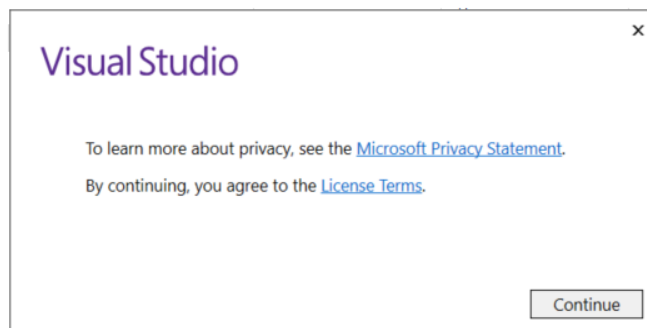


Figura 8.Términos y condiciones de Visual Studio.
Fuente:(“Instalar Visual Studio,” 2018)

3.4.4. Selección de las cargas de trabajo

Una vez instalado procedemos a personalizar la instalación, mediante la selección de cargas de trabajo las cuales son necesarias para el desarrollo del proyecto como se muestra en la **Figura 9**. Activación de cargas de trabajo de Visual Basic.

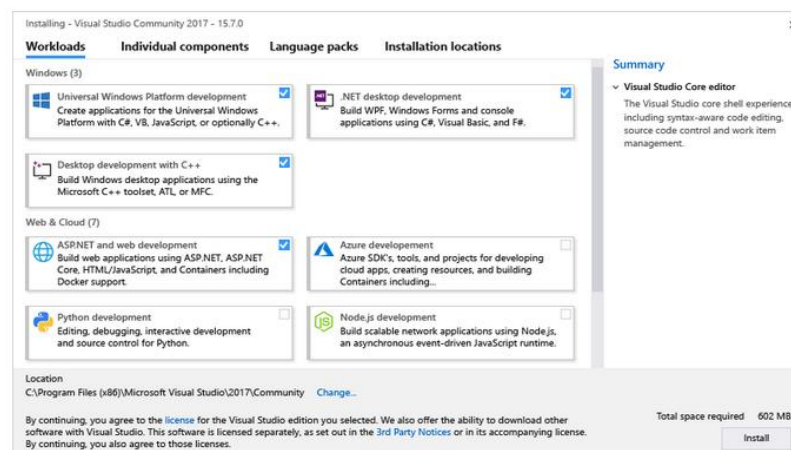


Figura 9. Activación de cargas de trabajo de Visual Basic.
Fuente: (“Instalar Visual Studio,” 2018)

Para este proyecto seleccionamos la plataforma de desarrollo universal de Windows, el desarrollo de programación con C++, el desarrollo web de ASP.NET, y el desarrollo de escritorio de .NET, este incluye el editor principal determinado, que contiene compatibilidad de edición de código básica para más de 20 lenguajes, capacidad de abrir y editar código desde cualquier carpeta sin la necesidad de crear una carpeta o proyecto.

3.4.5. Selección de componentes individuales

Si no deseamos utilizar cargas para activar los niveles de trabajo podemos hacerlo mediante la instalación de componentes individuales como muestra la **Figura 10**. Selección de componentes individuales.

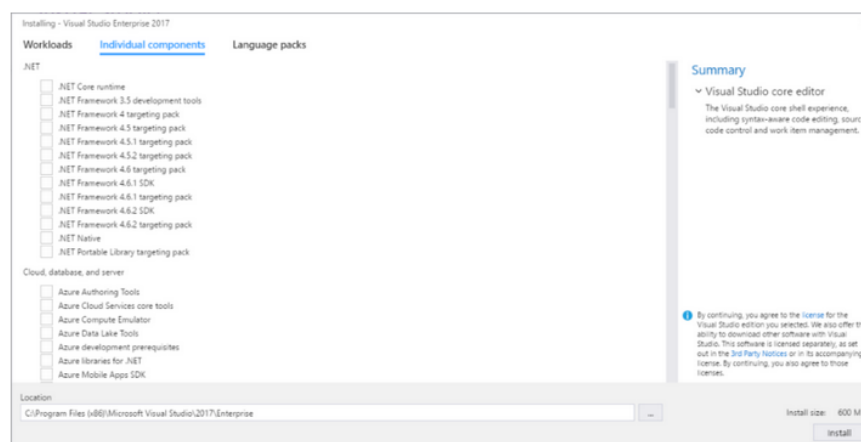


Figura 10. Selección de componentes individuales.
Fuente: (“Instalar Visual Studio,” 2018)

3.4.6. Procedimiento para instalar paquetes de idioma

Una vez personalizada el área de trabajo instalamos los paquetes de idioma en Visual Studio 2017, hacemos clic en la opción paquetes de idioma del instalador y seguimos con las instrucciones, o podemos realizarlo mediante código así el instalador recordara esta configuración cuando se ejecute el software, la **Figura 11**. Instalación de paquete de idioma.

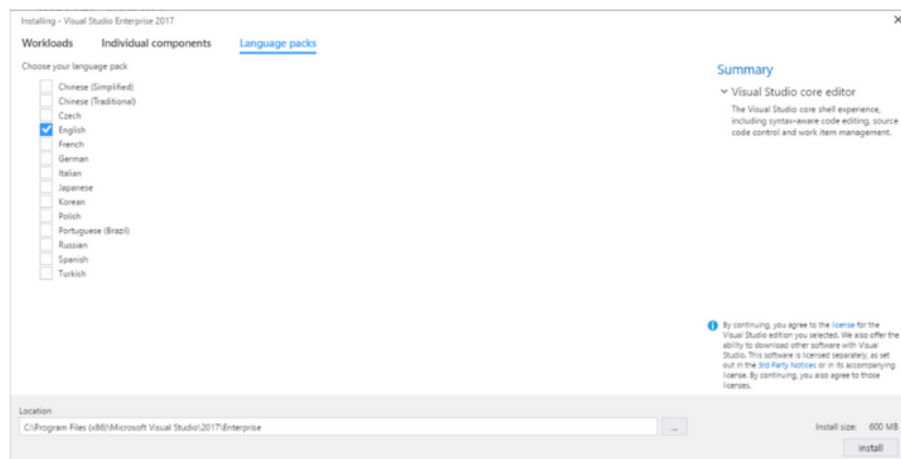


Figura 11. Instalación de paquete de idioma.
Fuente: ("Instalar Visual Studio," 2018)

3.5. Pasos para la creación de una aplicación

- Una vez creada la interfaz de usuario se observa los controladores y objetos del formulario, con el fin de facilitar su lectura y depuración debe asignar nombres a los objetos.
- Para establecer las propiedades de los objetos de interfaz, establecemos valores usando la ventana de propiedades en tiempo de diseño o con código para modificarlas en período de ejecución.
- Para los eventos añadimos código que se ejecute en respuesta a estos, un evento ocurre cuando diferentes acciones ocurren sobre un objeto.

- Cuando creamos el proyecto nos aseguramos al darle un nombre al proyecto, es recomendable guardar frecuentemente el proyecto conforme añadamos código.
- Para la depuración la aplicación usa el comando iniciar para examinar el código, esto nos sirvió para verificar errores y modificarlos.
- Al completar el proyecto creamos un archivo ejecutable usando el comando Generar archivo.exe del menú de archivos.
- Debido a que es ejecutable depende de otros archivos en tiempo de ejecución tales como Visual Basic (Vbrun50032.dll), algunos archivos OCX y DLL requeridos por controladores necesarios para la activación del mismo.

A continuación, en la **Figura 12**. Introducción al desarrollo de Visual Basic. Se muestra el software Microsoft Visual Studio instalado en LattePanda.

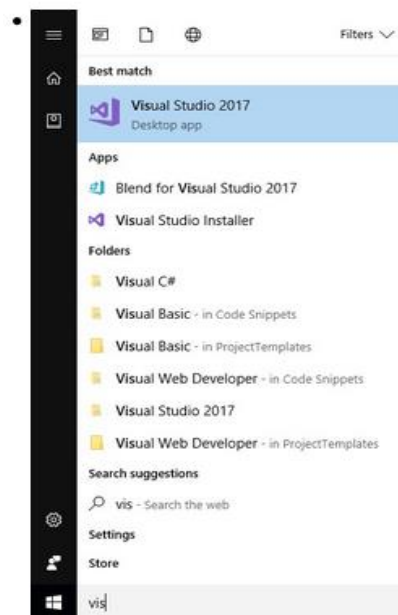


Figura 12. Introducción al desarrollo de Visual Basic.
Fuente: El Autor. ("Instalar Visual Studio," 2018)

Luego de abrir Visual Studio aparece la siguiente página como se muestra en la **Figura 13**. Página de inicio de Microsoft Visual Studio. Ofreciendo diferentes

opciones tales como: abrir un proyecto existente, crear un proyecto nuevo, abrir un proyecto reciente, o abrir un proyecto desde un sitio web.

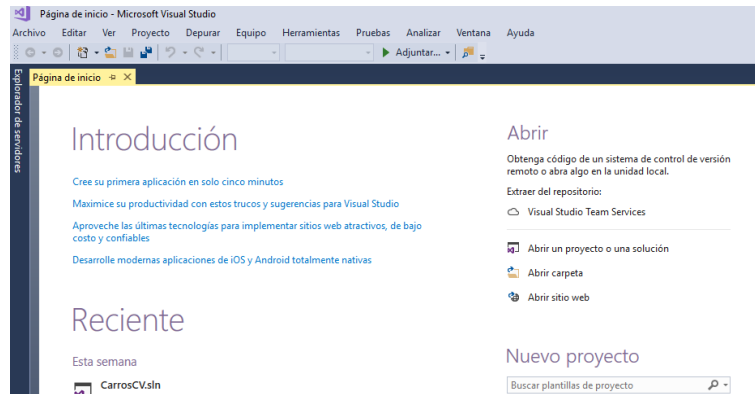


Figura 13. Página de inicio de Microsoft Visual Studio.
Fuente: ("Instalar Visual Studio," 2018)

Para el desarrollo de este proyecto seleccionamos, archivo, luego nuevo, y por último proyecto, la **Figura 14**. Secuencia para la creación de un nuevo proyecto en Visual Presenta la secuencia a seguir para la creación de un nuevo proyecto.

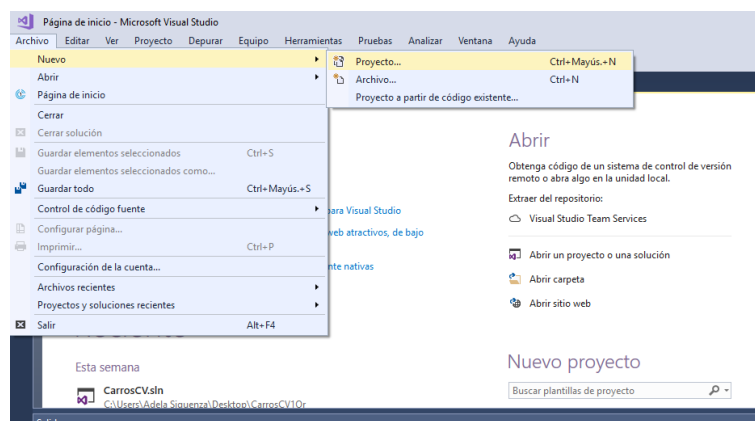


Figura 14. Secuencia para la creación de un nuevo proyecto en Visual Studio
Fuente: ("Instalar Visual Studio," 2018)

Luego aparece la ventana de Visual pidiéndonos que seleccionemos la aplicación sobre la cual instalar como muestra la **Figura 15**. Selección de la aplicación de Visual Studio. Nosotros seleccionamos la Aplicación de Windows formas.

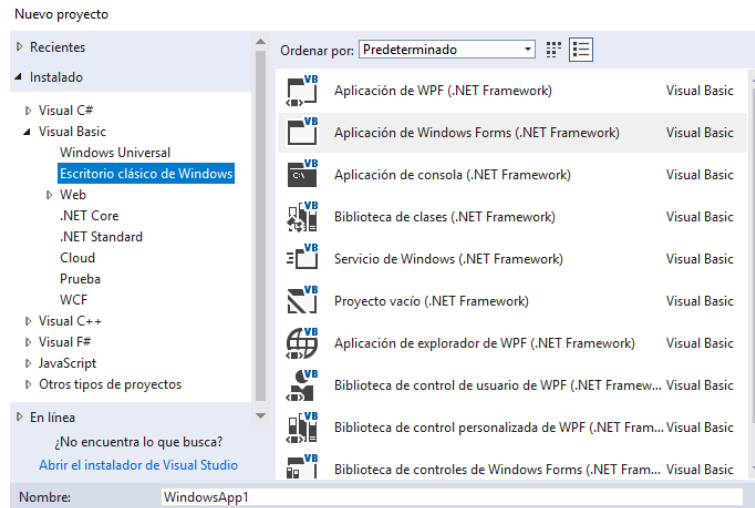


Figura 15. Selección de la aplicación de Visual Studio.
Fuente: ("Instalar Visual Studio," 2018)

Una vez seleccionado se crea la ventana en la cual desarrollamos el proyecto como muestra la **Figura 16**. Ventana del proyecto de Visual Studio.

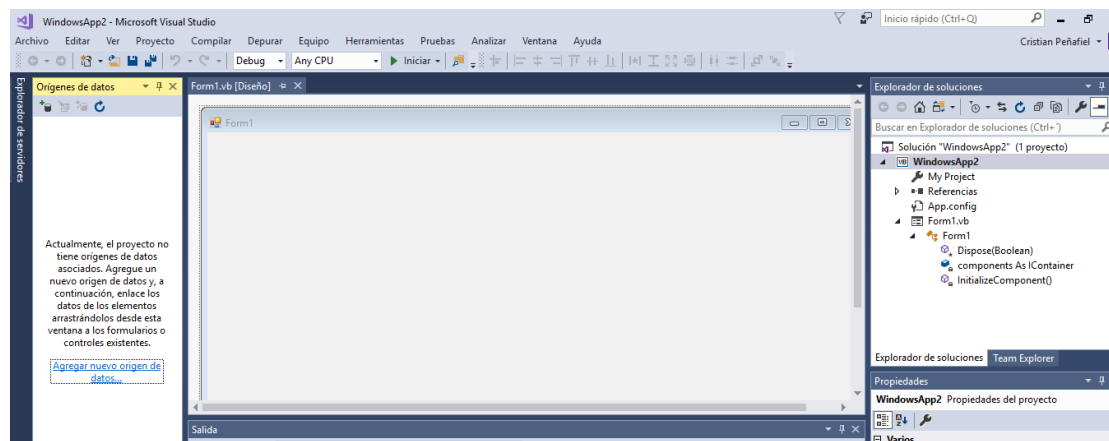


Figura 16. Ventana del proyecto de Visual Studio.
Fuente: ("Instalar Visual Studio," 2018)

3.6. Algoritmo de reconocimiento vehicular.

El siguiente algoritmo explica la funcionalidad del sistema de reconocimiento vehicular, desde el muestreo, hasta la validación de un posible vehículo.

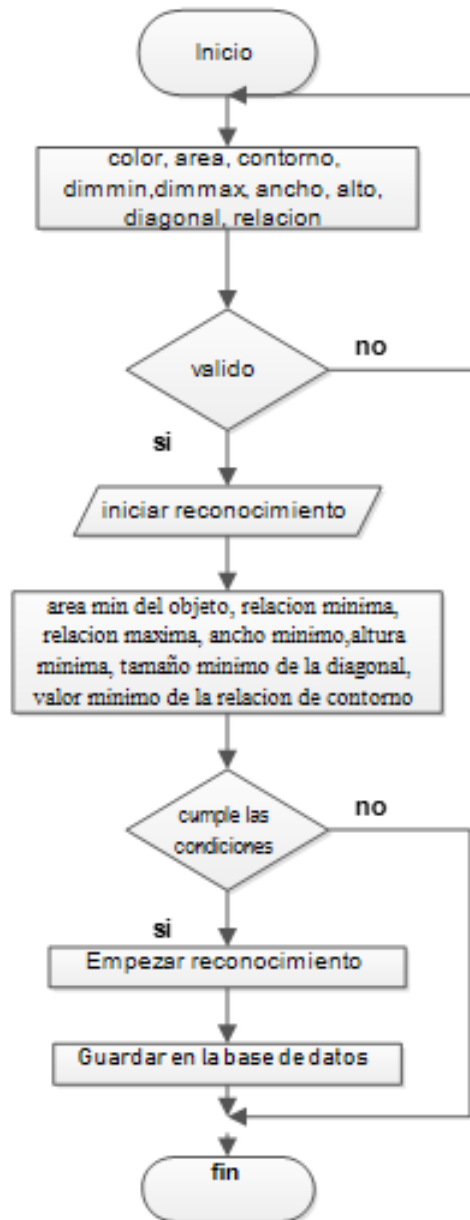


Figura 17. Algoritmo de reconocimiento vehicular.
Fuente: El Autor.

3.7. Diseño del sistema MOTRAVE

El diseño del sistema parte de la obtención de una imagen digital la cual se obtiene mediante una cámara, luego se realiza un filtrado y segmentación para que el objeto pase a un posterior análisis donde se le realiza una serie de comparaciones, si la imagen cumple las condiciones de programación será reconocido como un posible automóvil, el procesamiento y reconocimiento

valida el análisis realizado entonces la registra y la envía a que se almacene en la base de datos como un posible automóvil que atraviesa una determinada vía. A continuación, la **Figura 18**. Diseño del sistema de reconocimiento. Muestra el diseño del sistema de reconocimiento vehicular.

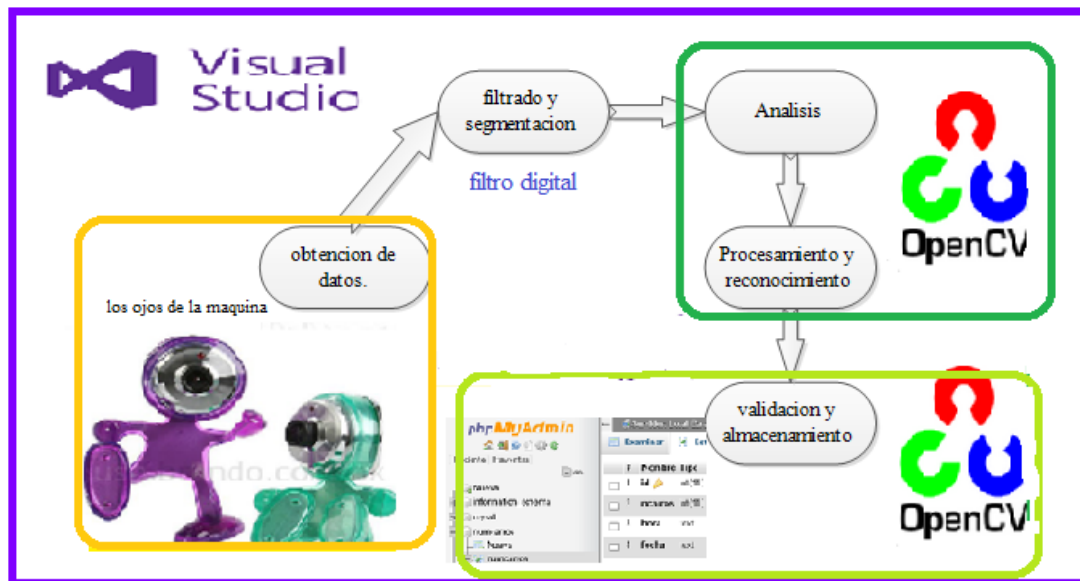


Figura 18. Diseño del sistema de reconocimiento.
Elaborado por: El Autor.

Luego de crear el sistema de reconocimiento se realiza la conexión con la base de datos, como muestra el siguiente algoritmo en la **Figura 19**. Algoritmo de conexión de la base de datos, para lo cual se debe establecer una conexión con el programa de reconocimiento vehicular. Una vez conectado comienza a registrar y enviar la información a SQL en donde se adjunta la fecha, hora y manda a almacenar en la Base de Datos, el contador se reinicia cada hora así el almacenamiento de la información se lo realiza de una forma ordenada.

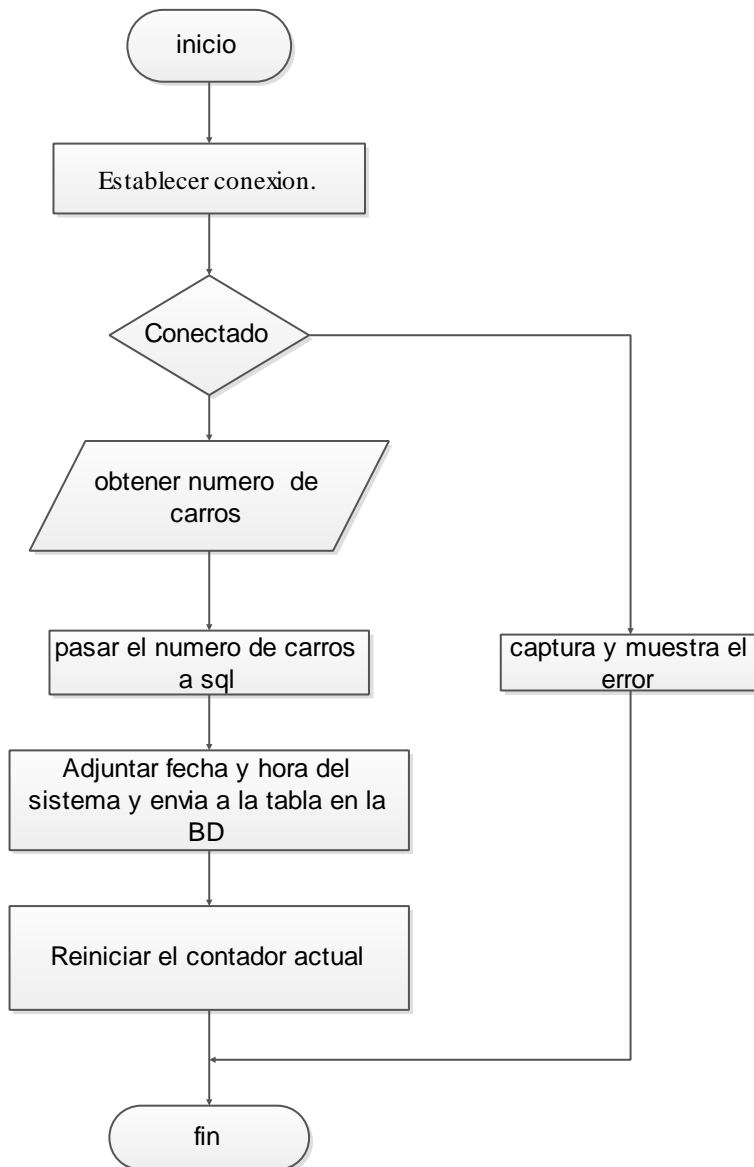


Figura 19. Algoritmo de conexión de la base de datos
Elaborado por: El Autor.

3.8. Base de datos y lenguaje de desarrollo

El lenguaje usado para realizar la conexión a la base de datos es (MySQL) con HTML. En este caso, se usó la versión 3.0.6 a 64 bits que contiene Apache en su versión 2.4.23, (MySQL), en la versión 5.7.14 y PHP con la opción de usar la versión 5.6.25. En el cual se crea tablas con el **id**, **hora**, **fecha** y **número de vehículos**.

La administración del servidor se realiza mediante apache el cual permite crear la lista inicial instaurando un usuario para evitar conexiones externas, está formado de cuatro columnas, la primera columna es de tipo entero autoincremento, la segunda es de tipo entero, la tercera y cuarta es de tipo texto con la codificación utf8_spanish_ci para el reconocimiento de todos los caracteres en español incluyendo ñ y tildes, la selección de este formato se realiza para evitar limitaciones presentes, al dejar en su idioma original que es el inglés, en la base de datos solo se registra símbolos si no se usa el formato adecuadamente puede causar daños tales como limitando el acceso o negándolo por completo.

La **Figura 20**. Base de datos con phpMyAdmin. Muestra la base de datos ya creada en phpMyAdmin, la cual muestra, el id, numero de carros, hora y fecha.

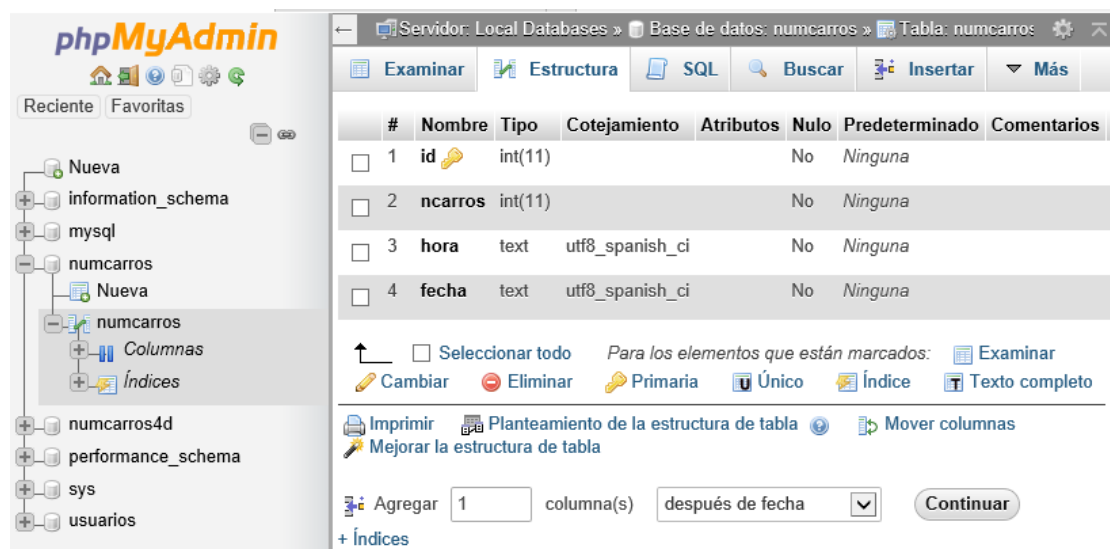


Figura 20. Base de datos con phpMyAdmin.
Fuente: ("phpMyAdmin," 2003)

3.9. Conexión del equipo

Para hacerlo portátil el equipo queda conectado de la siguiente manera. La alimentación es en corriente continua para lo cual se utiliza dos baterías de 3.7V. a 1.5A. cada una dando un voltaje de 7.4V a 3A en el sumatorio total, este voltaje ingresa a un controlador estabilizador de corriente y voltaje, el cual es regulado con un potenciómetro el mismo que está incorporado en el controlador dando a la salida un voltaje y corriente estable.

La carga de las baterías con las que opera es útil durante el lapso de una hora, ya que el consumo de corriente del dispositivo es de 2.5 A. para la validación opero conectado a una fuente de 120 V.

A los puertos USB 2.0 y 3.0 van conectadas las cámaras digitales las cuales obtienen los datos a ser analizados, la pantalla y táctil van conectadas a sus respectivos puertos, a continuación, en la siguiente

Figura 21. Conexión del equipo. se explica la conexión del equipo.

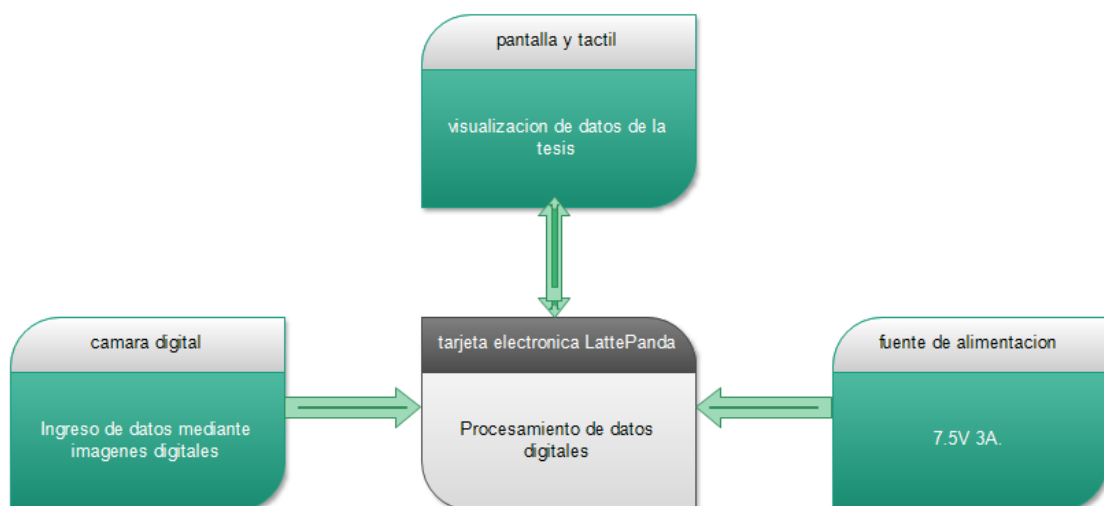


Figura 21. Conexión del equipo.
Fuente: El Autor.

3.10. Diseño de la carcasa del equipo

El equipo de monitoreo de tráfico vehicular, está compuesta por una carcasa la misma que tiene por finalidad dar, protección al proyecto de manera que no sufra daños al entrar en contacto con la intemperie. Se diseñó con el grado de protección IP 0.0, ya que no tiene ninguna protección especial, el único objetivo es fijar la tarjeta electrónica y sus componentes dentro de esta, lo cual ayuda a mejorar la presentación y el manejo de los diferentes componentes.

La carcasa es de acrílico debido a que es material resistente, posee un buen aislamiento y es ligero permitiendo al usuario el acceso al área de control de la placa, y evita cargar un peso exagerado. A continuación, se detalla las características de cada una de las partes de la carcasa.

La estructura de la **Figura 22**. Diseño posterior de la carcasa en el software sketch up 2018. es la parte posterior de la carcasa, la cual posee las siguientes medidas, largo = 267mm, ancho = 190mm, además cuenta con una rejilla misma que sirve de ventilación tiene las siguientes medidas, largo = 37mm, y ancho = 30mm, y un soporte para la cámara de 24mm de diámetro.

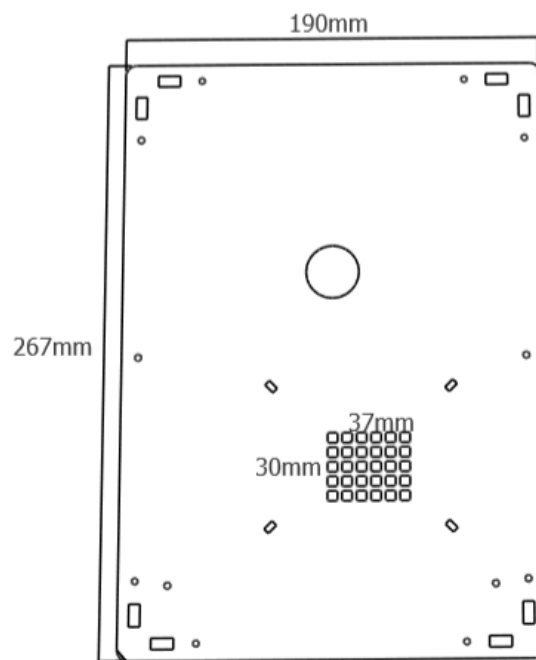


Figura 22. Diseño posterior de la carcasa en el software sketch up 2018.
Fuente: El Autor.

La **Figura 23.** Diseño frontal de la carcasa en el software sketch up 2018 muestra la parte frontal tiene las siguientes medidas, largo = 254mm, ancho = 180mm, esta es notablemente más pequeña.

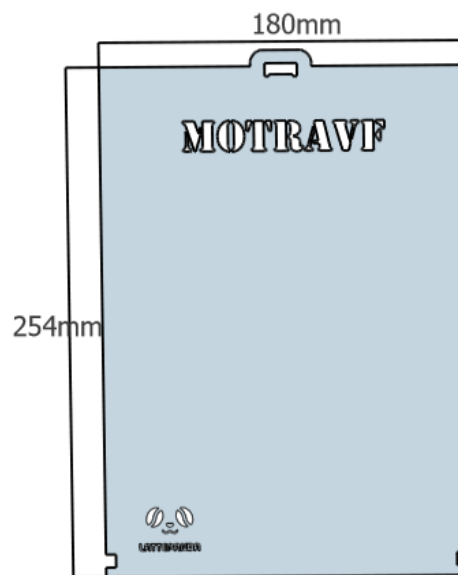


Figura 23. Diseño frontal de la carcasa en el software sketch up 2018

Fuente: El Autor.

La **Figura 24** muestra las estructuras laterales, las cuales constan con diferentes muescas, que sirven de soporte para otros componentes y ventilación de la tarjeta electrónica.

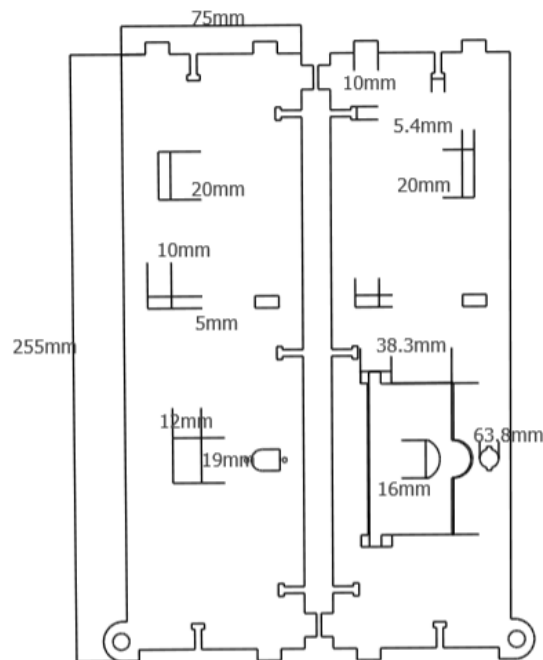


Figura 24. Diseño lateral de la carcasa en el software sketch up 2018.
Fuente: El Autor.

La **Figura 25.** Diseño central y soporte de la pantalla de la carcasa en el software sketch up 2018. muestra los componentes que divide a la caja de protección con sus medidas y los soportes de la pantalla.

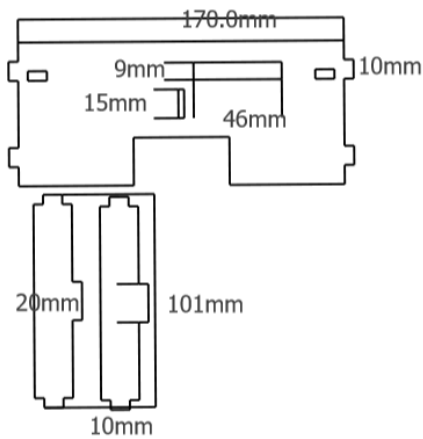


Figura 25. Diseño central y soporte de la pantalla de la carcasa en el software sketch up 2018.

Fuente: El Autor.

La **Figura 26.** Diseño superior e inferior de la carcasa en el software sketch up 2018. Muestra las estructuras superior e inferior de la carcasa.

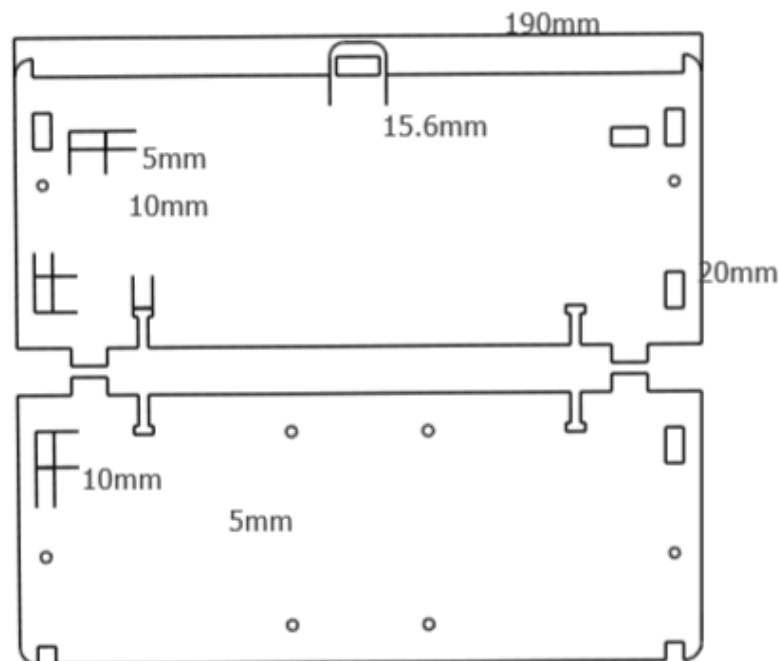


Figura 26. Diseño superior e inferior de la carcasa en el software sketch up 2018.

Fuente: El Autor.

En la **Figura 27.** Carcasa terminada y ensamblada. Se puede observar la carcasa terminada y ensamblada.

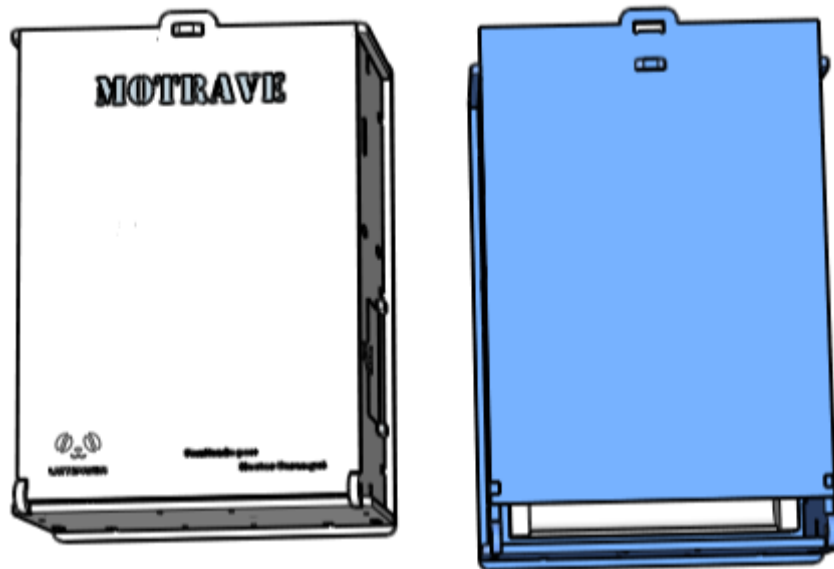


Figura 27. Carcasa terminada y ensamblada.

Fuente: El Autor.

La **Figura 28.** Carcasa con todos los componentes terminados y ensamblados. Muestra la carcasa con todos los componentes ensamblados.

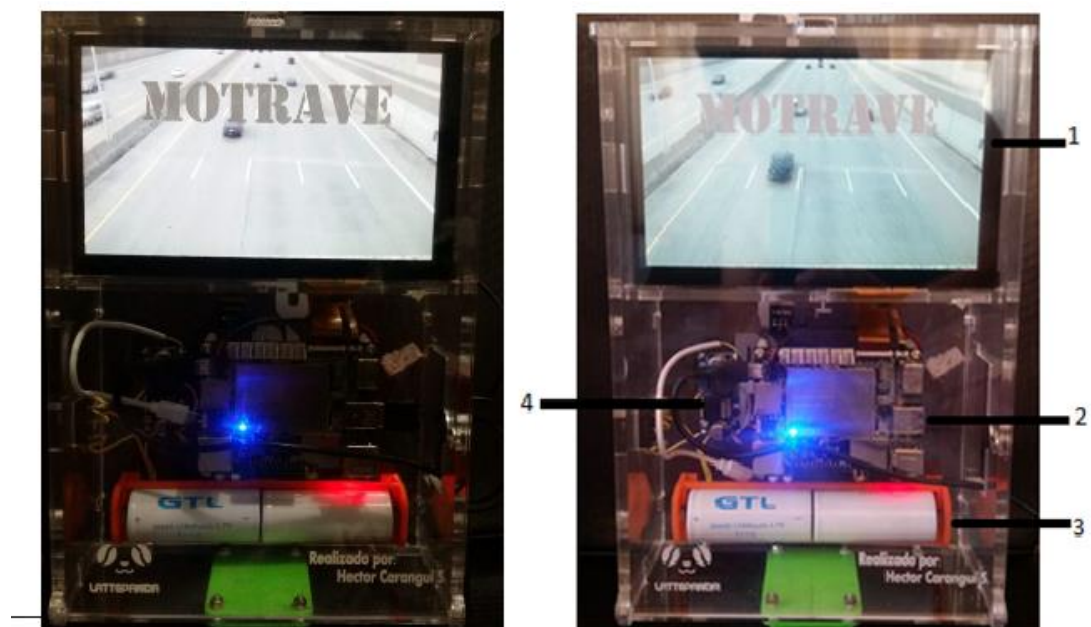


Figura 28. Carcasa con todos los componentes terminados y ensamblados.

Fuente: El Autor.

1. Pantalla y táctil de LattePanda.

2. Tarjeta electrónica LattePanda.
3. Baterías de alimentación.
4. Regulador de voltaje y corriente.

CAPÍTULO IV

VALIDACIÓN E IMPLEMENTACION DEL SISTEMA DE MONITOREO DE TRÁFICO VEHICULAR (MOTRAVE)

Este capítulo dará a conocer el procedimiento que se siguió para efectuar la validación e implementación del proyecto desarrollado, para lo cual se acudió a la Dirección de Movilidad de la ciudad de Azogues que es el ente que regula los procesos de tráfico vehicular y así puedan verificar y certificar su funcionalidad. A continuación, se da a conocer el estudio realizado para la obtención de la validación.

De acuerdo a los requerimientos por parte de la Dirección de Movilidad, se ajustó el equipo para probar en la intersección: Luis Cordero y Bartolomé Serrano y se obtuvieron los resultados mostrados en los apartados siguientes.

4.1. Planeación para la ubicación y verificación del equipo MOTRAVE

La ubicación se la efectuó en la intersección de las calles Bartolomé Serrano y Luis Cordero en la ciudad de Azogues de acuerdo a los requerimientos de la Dirección de Movilidad. A continuación la **Figura 29**. Localización del área de estudio.muestra la localización del área de estudio.



Figura 29. Localización del área de estudio.
Fuente: (google maps, 2015)

4.2. Visita in situ y levantamiento de información

Inicialmente se realizó la visita a la intersección que se nos fue asignada para proceder con dicho estudio, luego verificamos el lugar y posicionamiento tanto del equipo como de las cámaras, a continuación, determinamos los ángulos de captación y nivel de censado, así como la alimentación para el equipo (MOTRAVE) y finalmente definimos la codificación y representación de los movimientos vehiculares que se presentan en la intersección con base en el algoritmo de programación. Se cuantifican los datos obtenidos por día y horas.

El equipo se implementó en el controlador del semáforo de dicha intersección para lo cual se utilizaron dos cámaras web. Para el desarrollo del estudio se contempló tres etapas.

- Recopilación de la información con el equipo de nombre (MOTRAVE).
- Tabulación de la información.
- Análisis de la información.

4.3. Recopilación de la información.

La información recolectada proviene de la cuantificación directa realizada en las calles Luis Cordero y Bartolomé Serrano, mediante el equipo MOTRAVE el cual realiza el conteo usando visión artificial. Para lo que se utilizó dos cámaras digitales ubicadas sobre uno de los semáforos.

El conteo se realizó solo de vehículos de 3.5 toneladas hacia abajo es decir se consideró camiones pequeños, automóviles, camionetas. Se excluyó motociclistas, ciclistas buses de transporte público y personas. La **Figura 30.** Localización específica de la implementación del equipo. Da a conocer exactamente la ubicación del equipo MOTRAVE en la ciudad de Azogues.

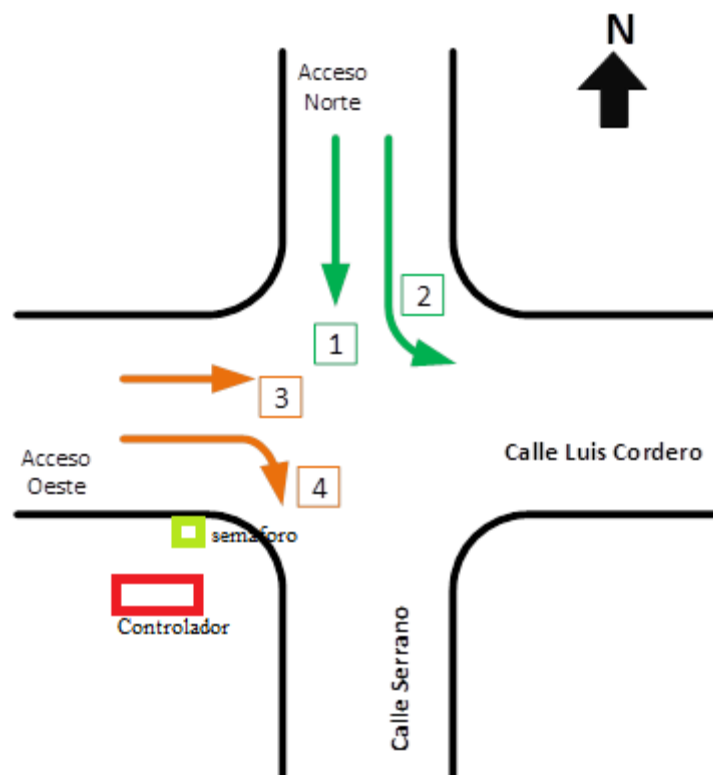


Figura 30. Localización específica de la implementación del equipo.
Fuente: El autor.

El equipo se ubicó dentro del controlador del semáforo de la intersección de estas calles opero conectado a una fuente de voltaje de 120V AC durante los siguientes días como se explica a continuación en la **Tabla 4**.

Tabla 4. Días de pruebas

Mayo											Junio				
15	16	17	18	19	20	21	22	23	24		1	2	3	4	5

Fuente: El Autor

El estudio se lo realiza determinando cuatro formas de circulación vehicular como muestra la **Figura 30. Localización** específica dela implementación del equipo.

1. Vehículos que circulen sobre la calle serrano y continúen sobre el paso a desnivel.
2. Vehículos que circulen sobre la calle serrano y luego circulen a través de la calle Luis Cordero.
3. Vehículos que circulen sobre la calle Luis Cordero desde el acceso oeste y continúen sobre esta.
4. Vehículos que circulen desde el acceso oeste de la calle Luis Cordero y luego se desvíen al paso a desnivel.

4.4. Análisis de volúmenes vehiculares en la intersección de las calles Luis Cordero y Bartolomé Serrano

Con el fin de determinar el número de vehículos que circulan atreves de las vías ya mencionadas como muestra la **Figura 30. Localización** específica dela implementación del equipo.Realizando la cuantificación del flujo vehicular, para lo cual se utilizó, el equipo (MOTRAVE) y una base de datos para almacenamiento.

Con los datos obtenidos, el Director de Movilidad de la ciudad de Azogues, efectuó una comparación con los datos que ellos recogieron de forma manual durante los mismos días que se realizó el estudio, y en base a esta comparación procedió a la entrega del certificado validando el trabajo de titulación el cual se adjunta en los anexos.

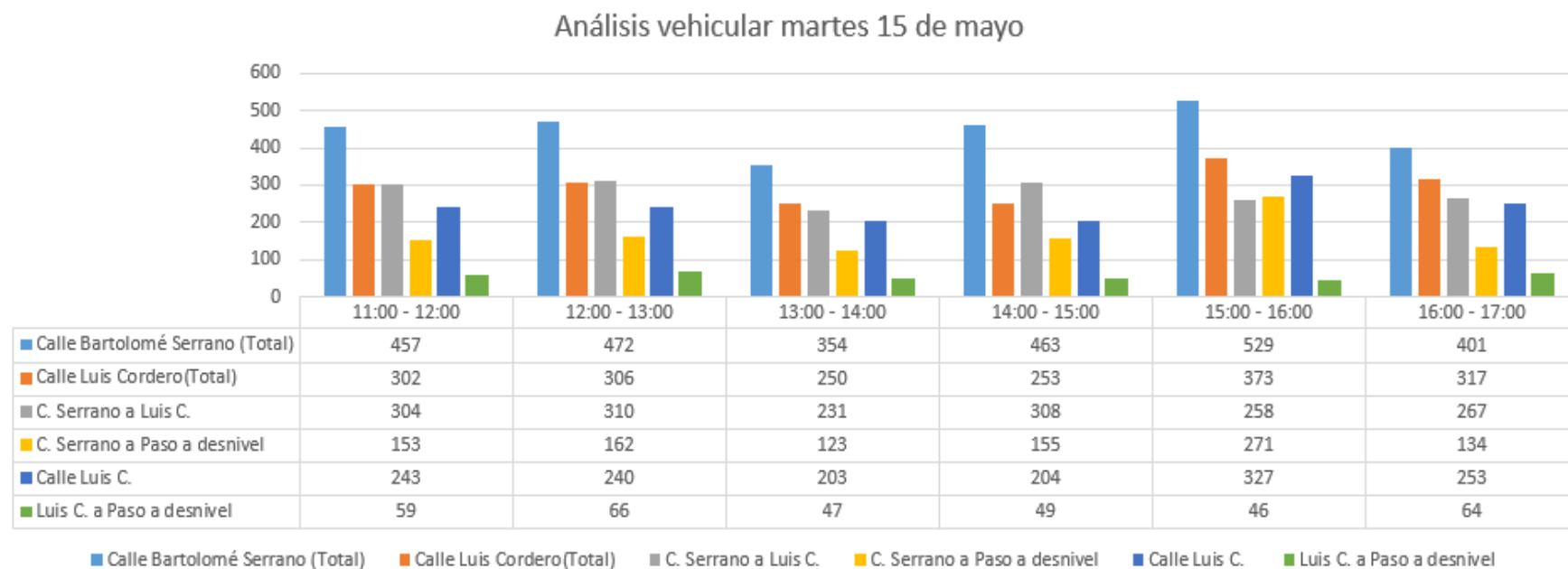


Figura 31. Análisis vehicular martes 15 de mayo.
Fuente: El autor.

Análisis vehicular miércoles 16 de mayo

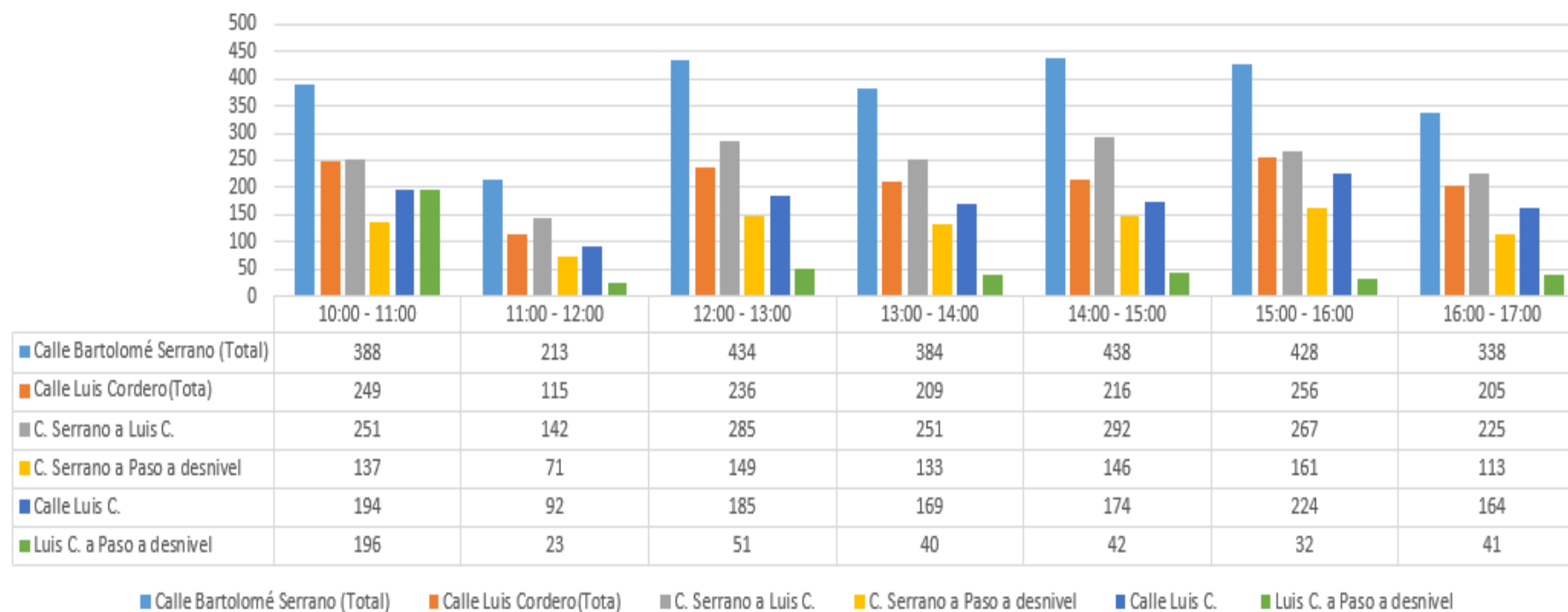


Figura 32. Análisis vehicular miércoles 16 de mayo.
Fuente: El autor.

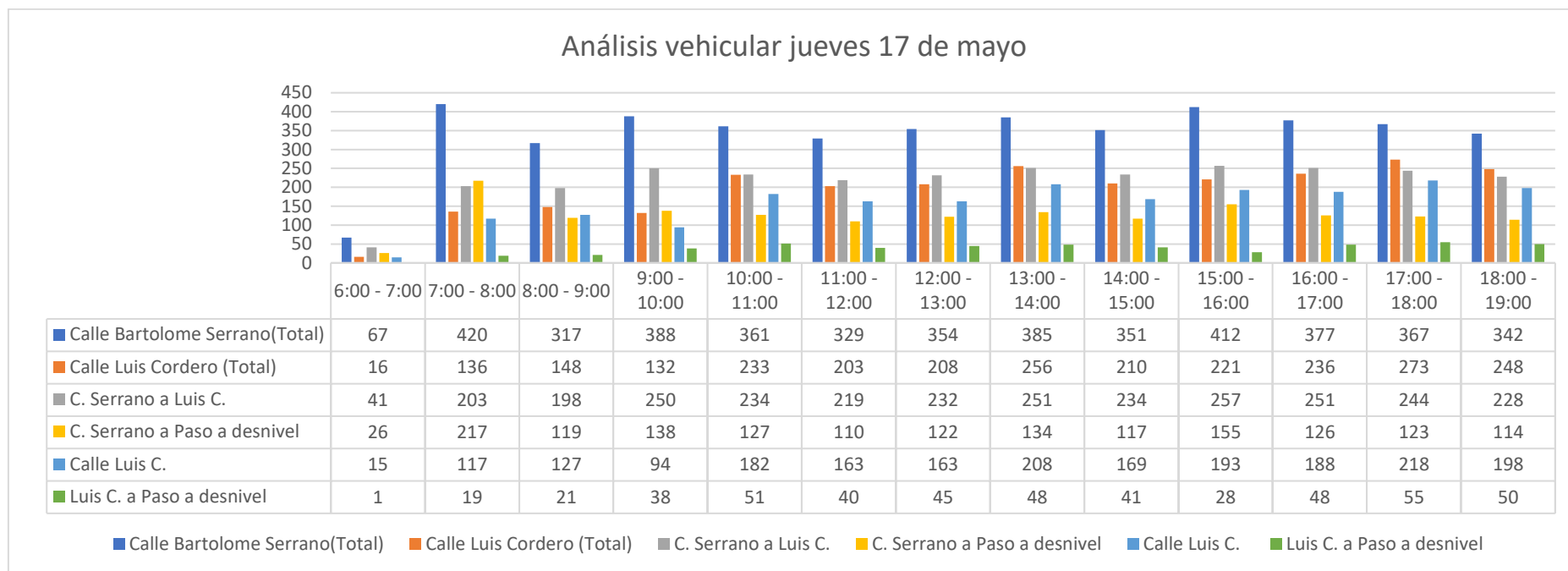


Figura 33. Análisis flujo jueves 17 de mayo.
Fuente: El autor.

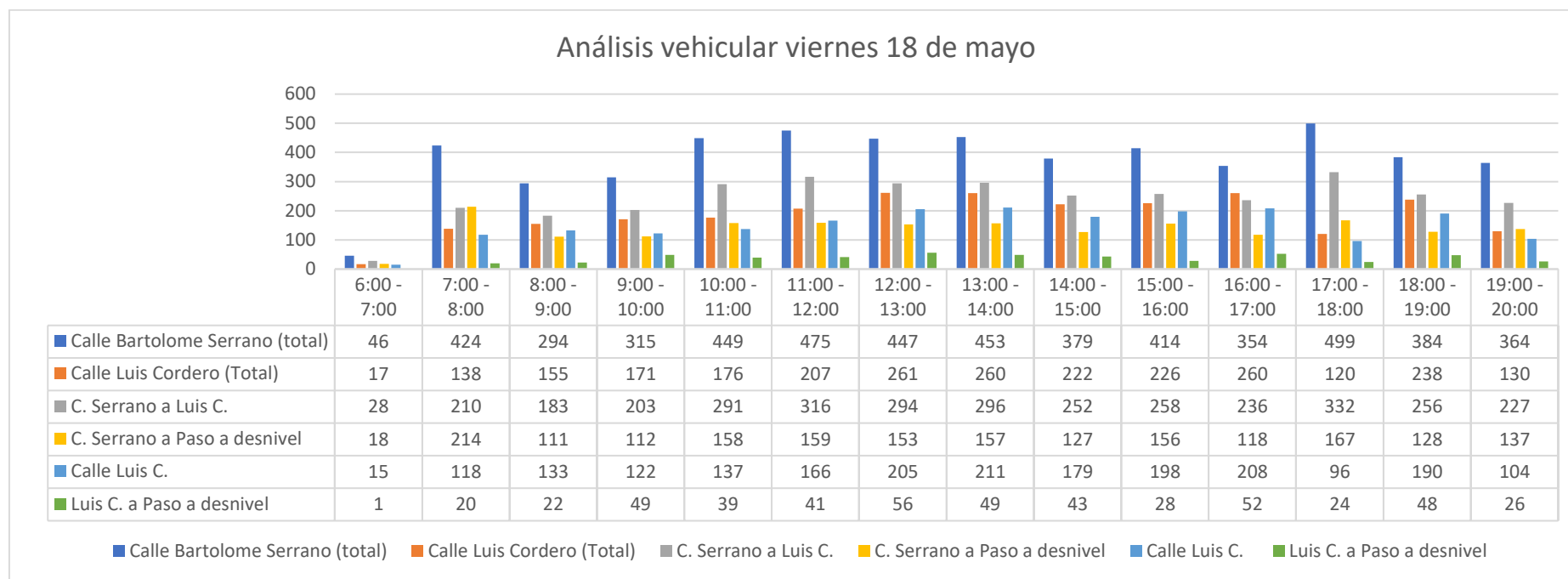


Figura 34. Análisis vehicular viernes 18 de mayo.
Fuente: El autor.

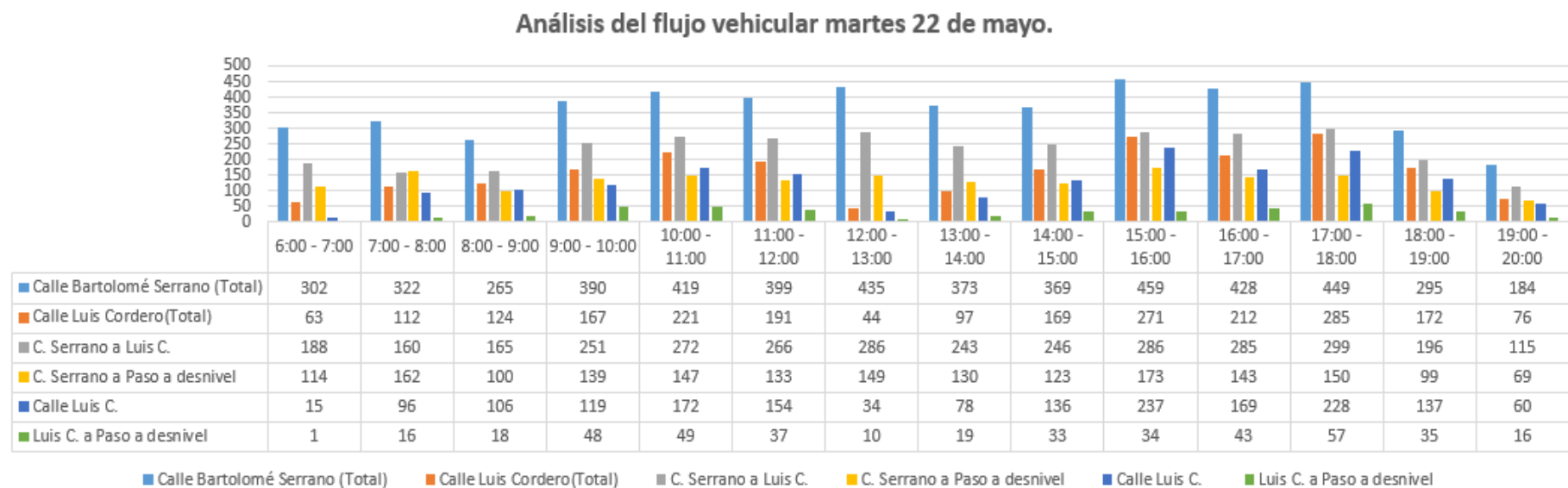


Figura 35. Análisis vehicular martes 22 de mayo.
Fuente: El autor.

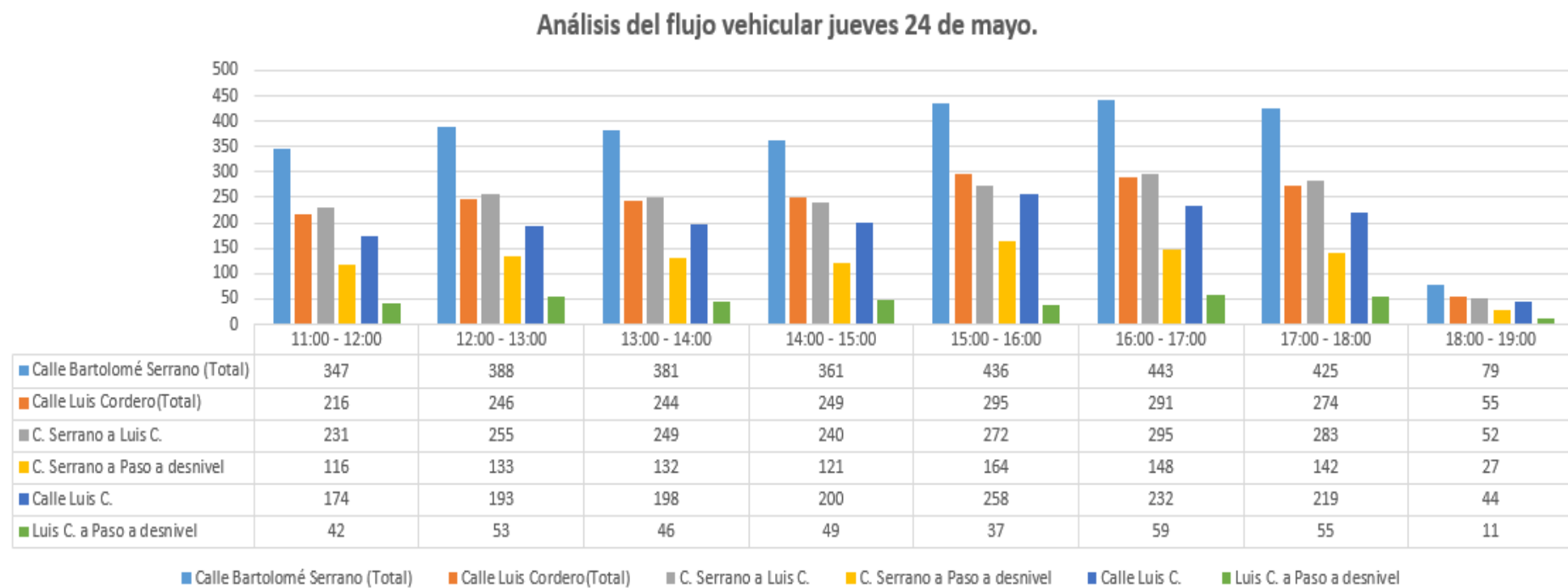


Figura 36. Análisis vehicular jueves 24 de mayo.
Fuente: El autor.

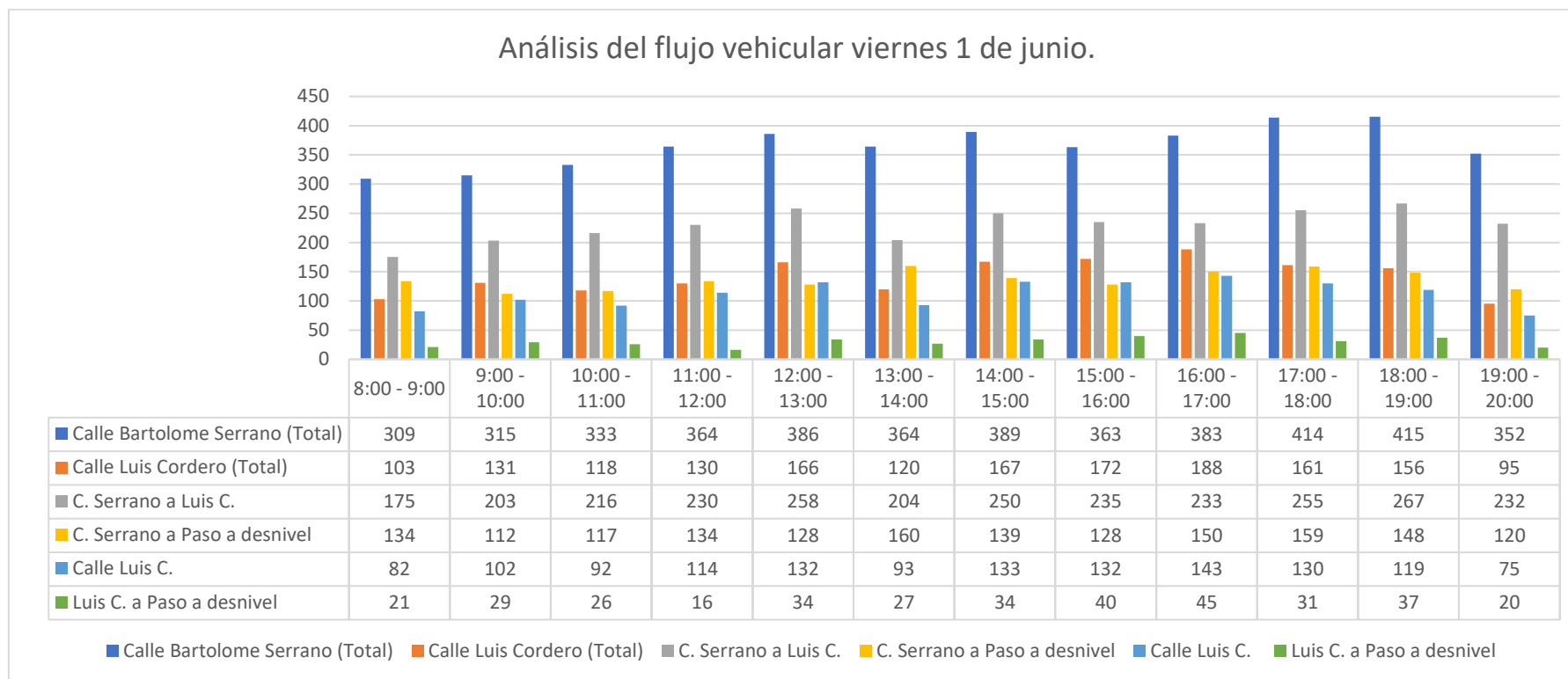


Figura 37. Análisis vehicular viernes 1 de junio.
Fuente: El autor.

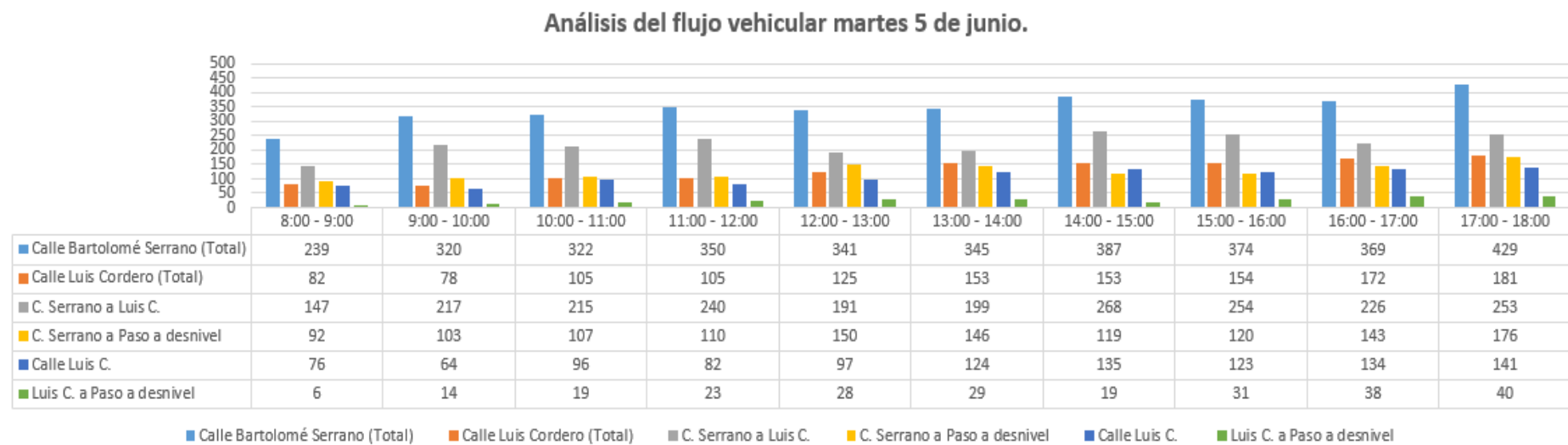


Figura 38. Análisis vehicular viernes 5 de junio.
Fuente: El autor.

Análisis del flujo vehicular total por semana durante el mes de mayo

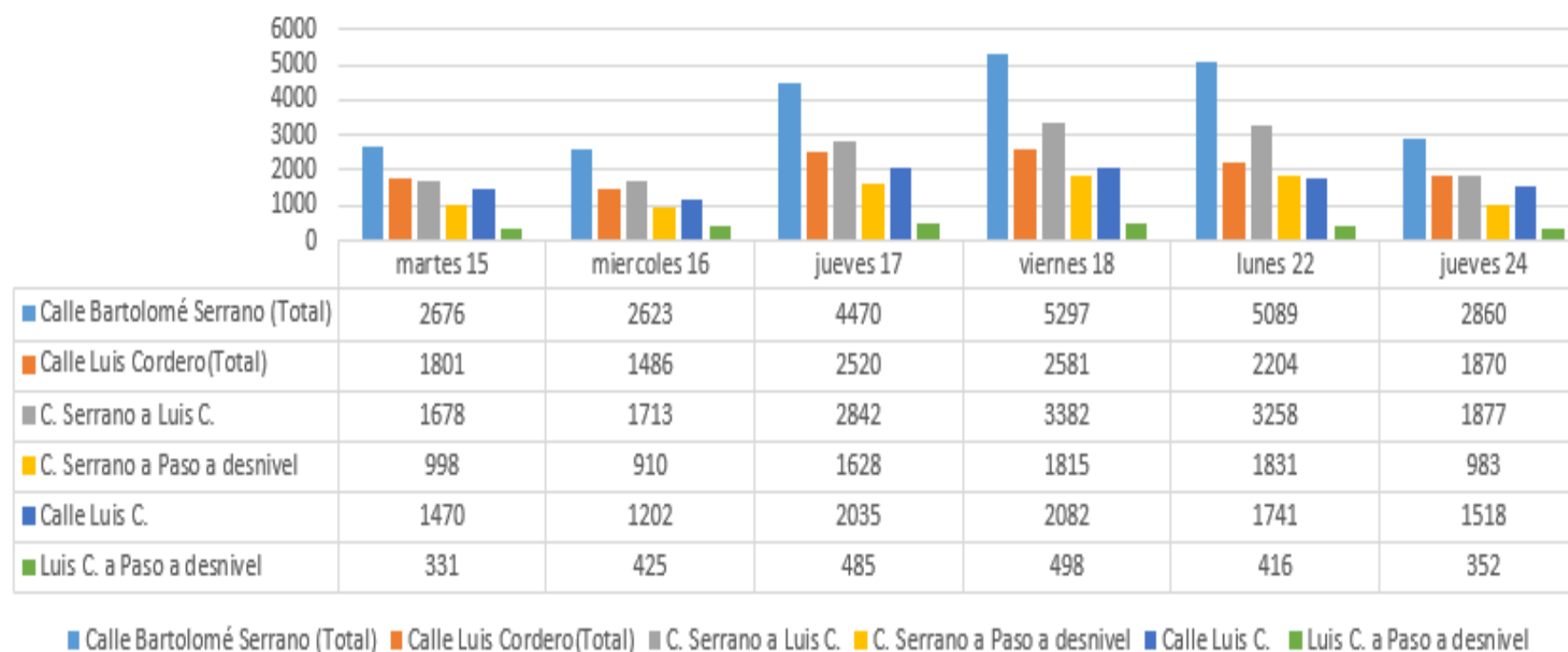


Figura 39. Análisis del flujo vehicular total por semana durante el mes de mayo.

Fuente: El autor.

Análisis del flujo vehicular total por semana durante el mes de junio

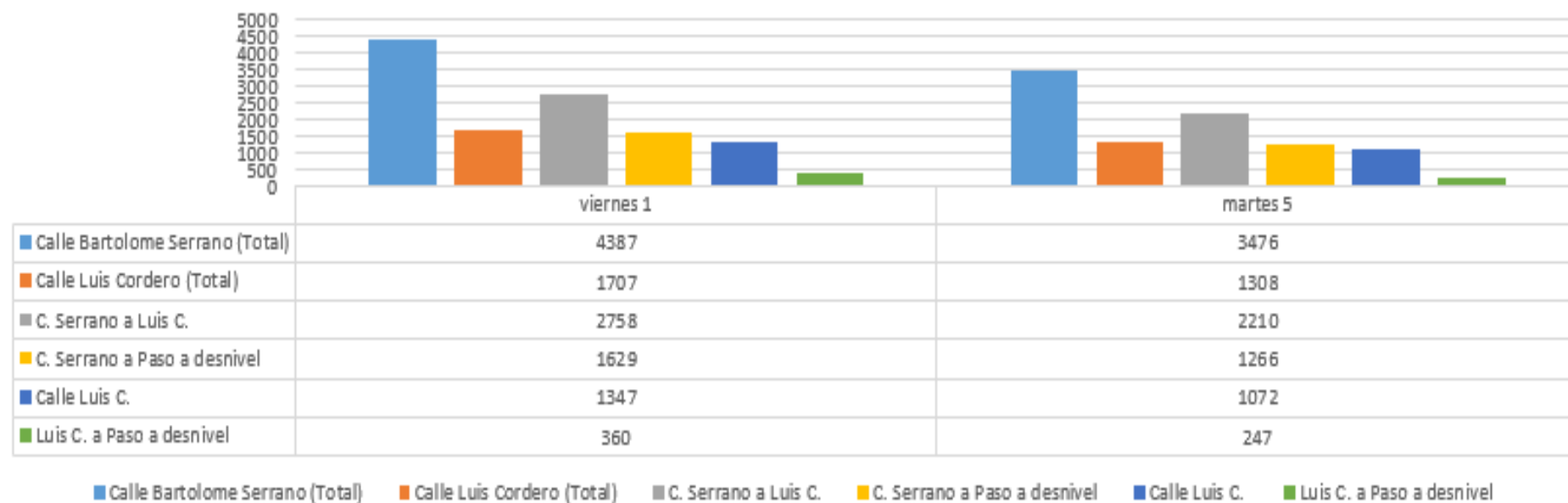


Figura 40. Análisis del flujo vehicular total por semana durante el mes de junio.

Fuente: El autor.

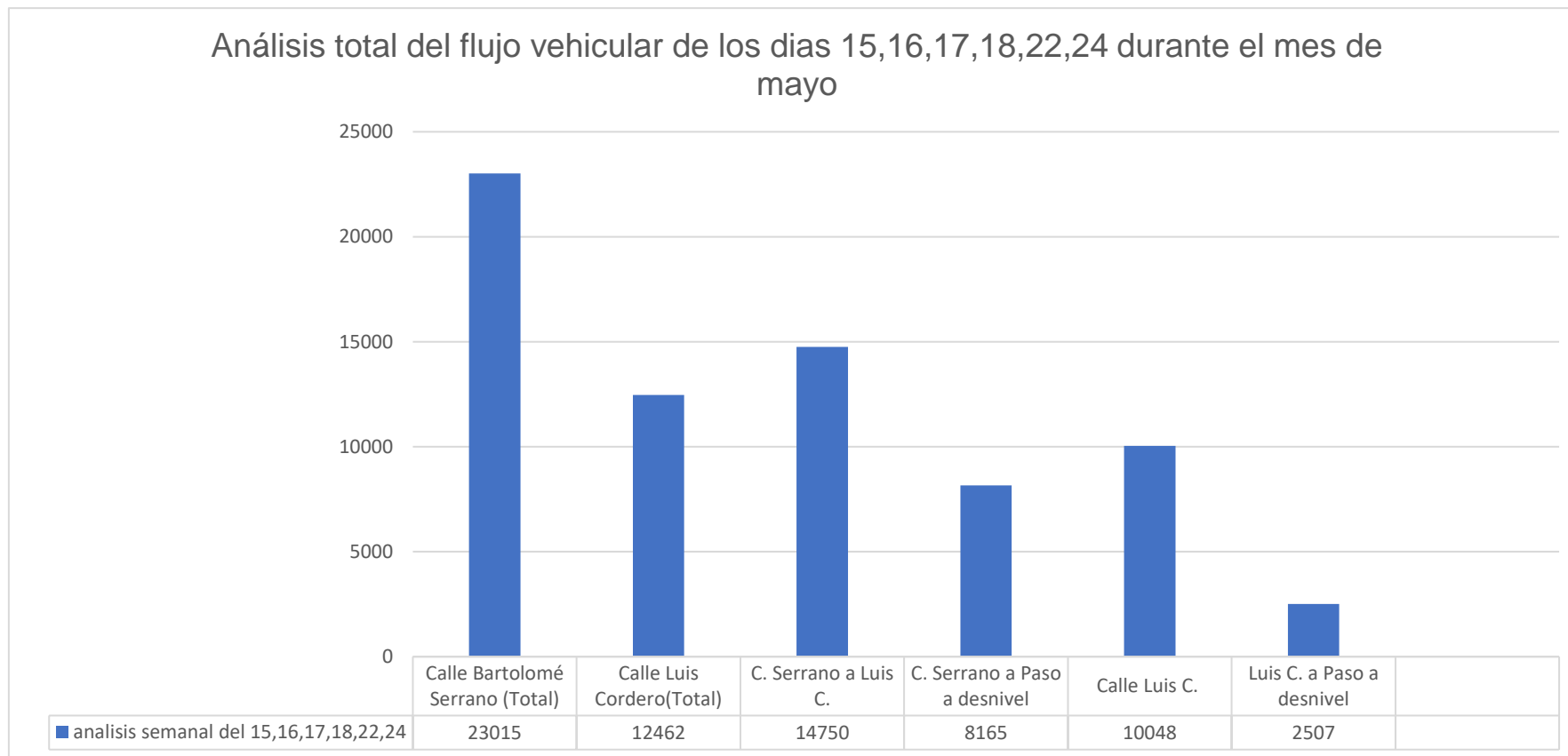


Figura 41. Análisis total del flujo vehicular del estudio del mes de mayo.
Fuente: El Autor.

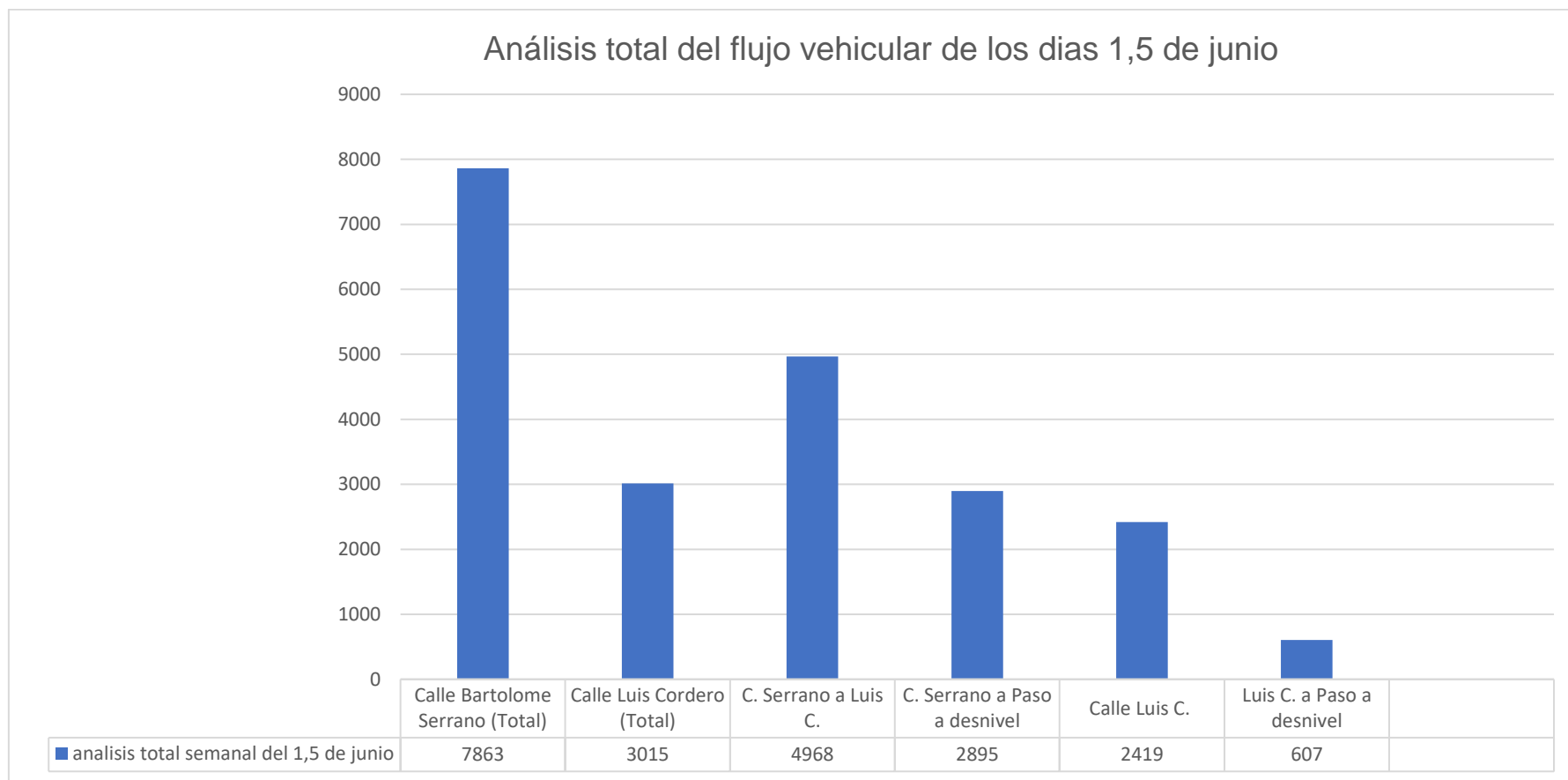


Figura 42. análisis total del flujo vehicular del estudio de los días 1 y 5 de junio.

Fuente: El Autor.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

La posibilidad de integrar la tecnología en nuestras vidas es una realidad, aunque existen limitaciones, no solo por los recursos económicos como se cree sino por la falta de información sobre estos; el limitado desarrollo en lo que hace referencia a software, en esta área como es la inteligencia artificial o ya sea por falta de capacitación o por los escasos antecedentes sobre estos en nuestro medio. Sin embargo, en la última década en nuestro país se ha introducido en la educación nuevas herramientas tecnológicas, además de acceso a ordenadores e internet que han contribuido a crear investigación y desarrollo en dicho campo.

Con la digitalización y el avance tecnológico que han hecho que la mayoría de personas estén familiarizadas con los distintos dispositivos, la inclusión de las TICs en la educación cambiando el paradigma educativo; hacen posible la introducción de nuevas técnicas de programación facilitando el desarrollo y comprensión de software, mejorando de cierta manera el medio en el cual habitamos, pero también mostrando que sería necesaria una capacitación para su correcto uso, así no se desperdiciaría tiempo ni dinero en los diferentes estudios de tráfico vehicular que se realicen en la ciudad.

El equipo desarrollado fue bien recibido por parte de la Dirección de Movilidad quienes consideran que es una herramienta útil y que sería usada para posteriores estudios de tráfico vehicular en la ciudad, de implementarse a largo plazo sería necesario hacerla remota ya que de esta manera se pudiese acceder al equipo desde cualquier parte y descargar datos o verificar la funcionalidad del mismo.

En base a lo desarrollado, se le da el control del proceso censado, reconocimiento, y almacenamiento a la tarjeta electrónica LattePanda, considerando que la tarjeta posee todas las funciones necesarias para desarrollar los algoritmos con los cuales ha sido programada, haciendo que sea tan puntual o tenga la mayor exactitud posible, sin embargo, se considera necesario las posibilidades de ampliar tanto el software como el hardware para tener una mayor área de cobertura. También brinda la posibilidad de ampliar aún más el software las opciones que se pueden integrar al código de programación son varias, algunas de ellas fuese el reconocimiento de placas de vehículos, desarrollo de redes neuronales para semaforización, control de rutas, entre otras.

5.2. Recomendaciones

Al estar desarrollado sobre Microsoft Visual Basic, con código de programación y librerías de OPEN CV ofrece un alto estándar de programación. Esto brinda la convicción de que al partir de un modelo sencillo se pueden ir incrementando variables en el programa fortaleciéndolo, y haciéndolo de fácil comprensión tanto para un desarrollador con experiencia, así como para uno que recién se esté iniciando en este.

Según las pruebas de funcionamiento que se realizó para la validación, es necesario usar cámaras de alta definición, que sean autoajustables para que el equipo opere de una mejor manera esto ayudaría a evitar posibles errores en la obtención de datos de información.

Además, como una recomendación especial debe analizar las capacidades de cada objeto ya que a estos se le atribuye estados. En base a las mejoras que se desee dar a futuro a la programación, se debe definir los objetivos ya que dependiendo de ello se podrá continuar con el mismo equilibrio de desarrollo o se tendría que usar uno con mayor capacidad.

Será optativo el uso de cámaras de alta definición y autoajustables a la radiación ya que esto ayudaría a tener un mejor censado y recopilación de datos, evitando la pérdida de información por cambios abruptos de resplandor de la calle que se esté realizando el estudio.

5.3. Trabajos futuros

Se pueden mejorar o añadir varias funciones tales como hacer que el dispositivo envíe los datos inalámbricamente hacia un servidor, hacerlo remoto para facilitar el control del equipo, incrementar un switch para habilitar más cámaras.

Permite incrementar funciones como las de determinar el transporte pesado, urbano, y particular para determinar las capas de asfalto en una vía, dicho objetivo se realiza mediante software sin la necesidad de un nuevo hardware.

Es posible integrar nuevas funciones al equipo, tales como la de reconocimiento de placas y determinación de velocidad sin afectar el funcionamiento actual de la aplicación, por el mismo sistema de escenas.

REFERENCIAS BIBLIOGRÁFICAS

- Aguilar, G. (1995). *Procesamiento digital de imagenes utilizando filtros morfológicos*. Quito: Escuela politécnica nacional.
- Albusac, J. (2008). *Vigilancia Inteligente: Modelado de Entornos reales e interpretación de conductas para la seguridad*. Castilla: Universidad de Castilla.
- Alvarado, J. (2011). *Procesamiento digital de señales*. Costa Rica: Tecnológico de Costa Rica.
- Bradski, G. O. (24 de diciembre de 2012).
<http://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV.pdf>. Obtenido de <https://www.google.com.ec/>
- Bravo , Y., Carrera , D., & Moya, D. (2014). *Analisis de tráfico vehicular usando visión artificial en la ciudad de San Juan de Pasto*. San Juan de Pasto: Universidad de Nariño.
- Bravo, Y., Carrera, H., & Moya, P. (2005). Diseño e implementacion de un sistema de monitoreo de tráfico vehicular utilizando procesamiento digital de imagen. *Jornada en Ingenieria Eléctrica y Electrónica*, 140-150.
- Bull, A. (2003). *Congestión de tránsito*. CEPAL.
- Burgo, M. (2014). *Autoncontrol psicoemocional*. Madrid: Universidad Complutense de Madrid.
- Cantero, A., & Acides, E. (2016). Visión por computadora: identificación, clasificación y seguimiento de objetos. *Universidad Nacional del Este*, 17-28.
- Castillo, R., & Hernandez, J. (2013). Procesamiento Digital de Imágenes Empleando Filtros Espaciales. *Universidad Autónoma de Baja California* , 1-6.

Chato, P. (2017). *Diseño e implementacion de un sistema de conteo de personas para seguridad del transporte colectivo utilizando vision artificial*. Latacunga: Universidad de las fuerzas armadas.

COMPILADOR, A. B. (08 de 03 de 2013).

http://www.cleanairinstitute.org/cops/bd/file/gdt/10-CEPAL-GTZ-Congestion_de_%20transito.pdf. Obtenido de Congestión de Tránsito, el problema y cómo enfrentarlo - 10-CEPAL-GTZ-Congestion_de_transito.pdf: <https://www.google.com.ec/>

Dolores, M. (2010). *Un modelo para el desarrollo de sistemas de detección de situaciones de riesgo de detección de situaciones de riesgo capaces de integrar informacion de fuentes heterogéneas*. Granada: Universidad de Granada.

Echaveguren, T., Díaz, A., & Arellano, D. (2013). Comparacion entre mediciones de velocidad obtenidas con los equipos GPS y pistola laser. *Obras y proyectos*, 47-55.

Espinoza, R. (2012). *Sistemas inteligentes de transportes-ITS*. Lima: Ministerio de transportes y comunicaciones.

Ferro, J. (2010). *Manual operativo del director y jefe de seguridad*.

González, J. (2016). *Calidad, ISO 9000, Formación y control de procesos*.

google. (04 de 08 de 2015).

https://www.google.com.ec/maps/place/Luis+Cordero+Crespo+%26+Calle+Bartolom%C3%A9+Serrano,+Azogues/@-2.7381935,-78.8486429,3a,75y,317.99h,80.18t/data=!3m9!1e1!3m7!1sCZ2R865pGy_VfygQ0H9mEQ!2e0!7i13312!8i6656!9m2!1b1!2i7!4m5!3m4!1s0x91cd12a4d987545d:0xbda3b. Obtenido de https://www.google.com.ec/maps/place/Luis+Cordero+Crespo+%26+Calle+Bartolom%C3%A9+Serrano,+Azogues/@-2.7381935,-78.8486429,3a,75y,317.99h,80.18t/data=!3m9!1e1!3m7!1sCZ2R865pGy_VfygQ0H9mEQ!2e0!7i13312!8i6656!9m2!1b1!2i7!4m5!3m4!1s0x91cd12a4d987545d:0xbda3b.

- Hernandez, Fernandez, & Baptista. (1998). *Metodologia de la investigacion*. Mexico: Mc Graw-hill.
- Informe_OpenCV_Tratamiento_Imagenes.pdf, A. M. (20 de febrero de 2015). http://www.informatica-juridica.com/wp-content/uploads/2014/01/Informe_OpenCV_Tratamiento_Imagenes.pdf. Obtenido de <https://www.google.com.ec/>
- Jimenez, L., & Puerto, R. (2004). *Sistemas informaticos en tiempo real: Teoria y aplicaciones*.
- Jimenez, M. (2018). *Procesamiento de imagenes digitales*. Sevilla: Universidad de Sevilla.
- Magaña, S. (1998). *Redes lan de microcomputadoras*. México: Consultores ICIMEX.
- Mathworks. (18 de Marzo de 2015). *Mathworks*. Obtenido de Visión artificial: <https://es.mathworks.com/discovery/vision-artificial.html>
- Merchán, E. (2010). *Sistema de visión artificial para el control de calidad en piezas cromadas*. México: Instituto politécnico nacional.
- Microsoft visual basic. (15 de 05 de 2017). www.microsoft.com. Obtenido de Visual Basic 6.0 to be supported on windows 10.
- Miranda, A., & Encalada, L. (2015). Sistema de control vehicular utilizando reconocimiento óptico de caracteres. *Escuela Superior Politécnica del Litoral ESPOL*, 3-8.
- Municipio del Distrito Metropolitano de Quito. (2015). *Diagnostico de la movilidad en el distrito metropolitano de Quito para el plan metropolitano de desarrollo territorial*. Quito: Secretaría de Movilidad.
- OpenCV. (10 de Marzo de 2018). *OpenCV*. Obtenido de OpenCV help: <https://opencv.org/>
- Perez, A. (2004). Una visión sobre la psicología como ciencia. *Cooperación nuevos rumbos*, 187-196.

- Perez, E. (2015). Los sistemas SCADA en la automatización industrial. *Tecnología en Marcha*, 3-14.
- Platero, C. (2009). *Formación de las imágenes*.
- Pozueta, J. (2000). *Movilidad y planeamiento sostenible: Hacia una consideración inteligente del transporte y la movilidad en el planeamiento y en el diseño urbano*. Madrid: Instituto Juan de Herrera.
- Quintero, J. (2015). Sistemas inteligentes de transporte y nuevas tecnologías en el control y administración del transporte. *Universidad Pedagógica y Tecnológica de Colombia*, 53-62.
- Ramon, J., & Serantes, J. (2006). *Técnicas Experimentales en hidráulica*. Coruña: Universidade da Coruña.
- Ramos, J., Vargas, J., & Aceves, M. (2015). *Desarrollo de software y hardware en mecatrónica*. México: Asociación mexicana de Mecatrónica.
- Redeszone. (01 de 01 de 2016). LattePanda, un mini-ordenador con Windows 10. *LattePanda, un mini-ordenador con Windows 10 por 60 euros*.
- Salvador, E., & Moctezuma, V. (2015). Tecnologías de almacenamiento de información en el ambiente digital. *Ciencias de la Información*, 50-62.
- Sampieri, H., & Fernandez, C. (2003). *El proceso de Investigación y los enfoques cuantitativo y cualitativo*.
- Sanchez, S. (2006). *Integración de técnicas de manejo de la imagen digital para la web y aplicaciones multimedia*. Sangolquí: Escuela Politécnica del Ejército.
- Šípal, J. (2012). *Sistemas lógicos digitales*. Fakulta elektrotechnická .
- Sucar, E. (2010). *Visión Computacional*. México: Instituto nacional y electrónica.

- UNED. (2011). *Comunicaciones industriales y en Tiempo Real*.
Departamento de Ingeniería Eléctrica, Electrónica y de control.
- Valdivia, Y. (2008). *Segmentación espacio-temporal de objetos en un video digital basada en un modelo con campos aleatorios de Markov*.
México: Universidad Nacional Autónoma de México.
- visionartificial.pdf, M. W.-V. (27 de 05 de 2010).
<http://www.etitudela.com/celula/downloads/visionartificial.pdf>.
Obtenido de VISIÓN ARTIFICIAL.doc - visionartificial.pdf:
<https://www.google.com.ec/>
- Zhañay, O. (2015). *Estudio del sistema de adquisición y monitoreo de temperatura de las unidades de generación de la central paute molino y elaboración del manual de operaciones*. Cuenca: Universidad Politécnica Salesiana.

ANEXOS

Anexo 1. Oficio dirigido al Director de movilidad de la ciudad de Azogues.

GAD MUNICIPAL DE AZOGUES
Teléfono: (01) 8114-2018-6753-Ext.
Fax: 20186753-03
E-mail: 1644
Martha Carvajal Salamea

Azogues 03 de mayo del 2018

Ing. Paulo Sacoto, Director de Movilidad
Ilustre Municipalidad de Azogues.

De mis consideraciones:

Yo Héctor Adrián Carangui Siguencia con C.I. 0302628847, estudiante de la Universidad Católica de Cuenca sede Azogues, de la facultad de Ingeniería electrónica, solicito a usted muy comedidamente, me ayude con la validación de mi trabajo de titulación para la obtención del título de Ingeniero Electrónico, para lo cual necesito me ayude con la revisión del sistema que se ha creado.

Por la atención brindada a la presente quedo a usted muy agradecido.

Atentamente,

Héctor Adrián Carangui Siguencia.
C.I. 0302628847
adriancarangui@yahoo.com
0987401517

AZOGUES
Municipio de Inocencio Urbina
GAD MUNICIPAL DE AZOGUES
03 MAY 2018
VENTANILLA ÚNICA DE SERVICIOS
Hora: 15:00 Firma: 

Anexo 2. Implementación del equipo (MOTRAVE) con los ingenieros encargados del controlador.



Anexo 3. Código del programa.

Option Explicit On

Option Strict On

Imports Emgu.CV

Imports Emgu.CV.CvEnum

Imports Emgu.CV.Structure

Imports Emgu.CV.Util

Imports System

Imports System.IO

Imports System.Collections

Public Class Form1

Dim Ubic_PCB As String

Dim COLOR_BLANCO As New MCvScalar(255.0, 255.0, 255.0)

Dim COLOR_NEGRO As New MCvScalar(0.0, 0.0, 0.0)

Dim COLOR_ROJO As New MCvScalar(0.0, 0.0, 255.0)

Dim COLOR_VERDE As New MCvScalar(0.0, 200.0, 0.0)

Dim CAM_VIDEO As Capture

Dim cerrando_form As Boolean = False

Dim factor_forma As Mat =
CvInvoke.GetStructuringElement(ElementShape.Rectangle, New Size(5, 5), New
Point(-1, -1))

'Condiciones para que un objeto reconocido sea posiblemente un carro

Dim CONDICION_AREA As Integer = 800

Dim CONDICION_REL_MIN As Double = 0.2

Dim CONDICION_REL_MAX As Double = 4.0

Dim CONDICION_ANCHO As Integer = 60

Dim CONDICION_ALTO As Integer = 60

Dim CONDICION_DIAGONAL As Double = 60.0

Dim CONDICION_REL_CON As Double = 0.5

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
Me.RegistrosTableAdapter.Fill(Me.DatosVEHDataSet.Registros)

CAM_VIDEO = Nothing

"C:\Users\Corell7\Documentos\VidCAR.mp4"

"C:\Users\lattepanda\Desktop\CarrosCV\VidCARI.avi"

'CAM_VIDEO = New Capture("C:\Users\Adela
Siguenza\Videos\WIN_20180510_10_08_58_Pro.mp4")

TimerOn.Start()

End Sub

```
Dim contaCarros As Integer = 0
Dim contaCarrosA As Integer = 0
Private Sub EmpezarReconocimiento()
TimerOn.Stop()
Dim frame1 As Mat = CAM_VIDEO.QueryFrame() 'Anterior
Dim frame2 As Mat = CAM_VIDEO.QueryFrame() 'Actual
Dim frame1A As Mat = frame1.Clone() 'Copia
Dim frame2A As Mat = frame2.Clone() 'Copia
Dim frameDif As Mat = New Mat(frame1.Size, DepthType.Cv8U, 1) 'Diferencia
Dim limiteY1 = frame1.Rows / 4 '4
Dim limiteX1 = 200 'Cols/6
Dim limiteY2 = limiteY1 + 200 '350
Dim limiteX2 = limiteX1 + 200 '900
'segunda area
Dim limiteY1A = frame1.Rows / 2 '4
Dim limiteX1A = 600 'Cols/6
Dim limiteY2A = limiteY1 + 200 '350
Dim limiteX2A = limiteX1 + 450 '900

Dim primerFrame As Boolean = True
Dim objetos As New List(Of objetoX)

Dim centroTexto As Point = New Point(650, 370)

Dim flagSleep As Boolean = False

While Not cerrando_form
If Not pausaFlag Then
#Region "Procesar imagen"
frame1A = frame1.Clone() 'Copia
frame2A = frame2.Clone() 'Copia
frameDif = ProcesarImagen(frame1A, frame2A, frameDif)
#End Region

#Region "Obtener Contornos"
Dim contornos As New VectorOfVectorOfPoint()
CvInvoke.FindContours(frameDif, contornos, Nothing, RetrType.External,
ChainApproxMethod.ChainApproxSimple) 'Encontrar contornos
```

```

CvInvoke.DrawContours(frameDif, contornos, -1, COLOR_BLANCO, 1) 'Dibujar
contornos
#End Region

#Region "Suavizar contornos"
Dim convex_hulls As New VectorOfVectorOfPoint(contornos.Size) 'Donde se
almacenan los recubrimientos
For i As Integer = 0 To contornos.Size - 1
CvInvoke.ConvexHull(contornos(i), convex_hulls(i)) 'Convertir el contorno en
recubrimiento
Next
#End Region

#Region "Obtener contornos validos"
'Obtener conornos validos (que cumplen las condiciones)
Dim objetosActuales As New List(Of objetoX)
For i As Integer = 0 To contornos.Size - 1
Dim objetoActual As New objetoX(convex_hulls(i))
If objetoActual.rect_Area > CONDICION_AREA And 'Condición de area
objetoActual.relacion > CONDICION_REL_MIN And 'Condición de relación de
aspecto mínima
objetoActual.relacion < CONDICION_REL_MAX And 'Condición de relación de
aspecto máxima
objetoActual.rect_exterior.Width > CONDICION_ANCHO And 'Condición de tamaño
objetoActual.rect_exterior.Height > CONDICION_ALTO And 'Condición de tamaño
objetoActual.diagonal > CONDICION_DIAGONAL And 'Condición de diagonal del
rectángulo
(CvInvoke.ContourArea(objetoActual.contorno) / 'Condición de relación:
contorno/area
objetoActual.rect_Area) > CONDICION_REL_CON And
(objetoActual.centro.X > (limiteX1)) And
(objetoActual.centro.X < (limiteX2)) Then
'Si cumple las condiciones el posible objeto, pasará a formar
'parte de los objetos "carros"
objetosActuales.Add(objetoActual)
End If
'condicion de segundos limites
If objetoActual.rect_Area > CONDICION_AREA And 'Condición de area
objetoActual.relacion > CONDICION_REL_MIN And 'Condición de relación de
aspecto mínima
objetoActual.relacion < CONDICION_REL_MAX And
objetoActual.rect_exterior.Width > CONDICION_ANCHO And
objetoActual.rect_exterior.Height > CONDICION_ALTO And
objetoActual.diagonal > CONDICION_DIAGONAL And 'Condición de diagonal del
rectángulo

```

```

(CvInvoke.ContourArea(objetoActual.contorno) / 'Condición de relación:
contorno/area
objetoActual.rect_Area) > CONDICION_REL_CON And
(objetoActual.centro.X > (limiteX1A)) And
(objetoActual.centro.X < (limiteX2A)) Then
'Si cumple las condiciones añadir el posible objeto, que ahora pasará a formar
'parte de los objetos "carros"
objetosActuales.Add(objetoActual)
End If

```

```

Next
#End Region

```

```

Dim frameCentros As Mat = frame2.Clone()
Dim carroContado As Boolean = False
Dim carroContadoA As Boolean = False

```

```

If primerFrame And objetosActuales.Count > 0 Then
primerFrame = False
For Each objetoActual As objetoX In objetosActuales
objetos.Add(objetoActual)
Next
Else
For Each objetoActual As objetoX In objetosActuales

```

```

CvInvoke.Rectangle(frameCentros, objetoActual.rect_exterior, COLOR_ROJO, 2)

```

```

Dim menorDistancia As Integer = 999999
Dim menorDistancia_i As Integer = 0

```

```

For i As Integer = 0 To objetos.Count - 1
Dim P1 As Point = objetoActual.centro
Dim P2 As Point = objetos(i).centro
Dim Pdx As Integer = Math.Abs(P1.X - P2.X)
Dim Pdy As Integer = Math.Abs(P1.Y - P2.Y)
Dim distancia As Integer = CInt(Math.Sqrt((Pdx ^ 2) + (Pdy ^ 2)))
If distancia < menorDistancia Then
menorDistancia = distancia
menorDistancia_i = i
End If
Next

```

```

If menorDistancia < objetoActual.diagonal / 2 Then
objetos(menorDistancia_i).centro = objetoActual.centro

```

```

If      objetos(menorDistancia_i).centro.Y      >      limiteY1      And
objetos(menorDistancia_i).centro.Y < limiteY2 Then
objetos(menorDistancia_i).dentro += 1
Else
If objetos(menorDistancia_i).dentro > 5 Then
contaCarros += 1
carroContado = True
End If
End If

If      objetos(menorDistancia_i).centro.Y      >      limiteY1A      And
objetos(menorDistancia_i).centro.Y < limiteY2A Then
objetos(menorDistancia_i).dentro += 1
Else

If objetos(menorDistancia_i).dentro > 5 Then
contaCarrosA += 1
carroContadoA = True
End If
objetos.Remove(objetos(menorDistancia_i))
End If

Else

If (objetoActual.centro.Y > (limiteY1)) And (objetoActual.centro.Y < (limiteY2)) Then
objetos.Add(objetoActual)
End If

If (objetoActual.centro.Y > (limiteY1A)) And (objetoActual.centro.Y < (limiteY2A))
Then
objetos.Add(objetoActual)
End If

End If
Next
End If

Dim LIMITE As Rectangle = New Rectangle()
Dim LIMITEA As Rectangle = New Rectangle()
LIMITE.X = CInt(limiteX1)
LIMITE.Y = CInt(limiteY1)
LIMITE.Width = CInt(limiteX2 - limiteX1)
LIMITE.Height = CInt(limiteY2 - limiteY1)

LIMITEA.X = CInt(limiteX1A)
LIMITEA.Y = CInt(limiteY1A)

```

```

LIMITEA.Width = CInt(limiteX2A - limiteX1A)
LIMITEA.Height = CInt(limiteY2A - limiteY1A)

If carroContado Then
    CvInvoke.Rectangle(frameCentros, LIMITE, COLOR_VERDE, 2)
Else
    CvInvoke.Rectangle(frameCentros, LIMITE, COLOR_ROJO, 2)
End If

If carroContadoA Then
    CvInvoke.Rectangle(frameCentros, LIMITEA, COLOR_VERDE, 2)
Else
    CvInvoke.Rectangle(frameCentros, LIMITEA, COLOR_ROJO, 2)
End If

lbl_contador.Text = contaCarros.ToString
Label1.Text = contaCarrosA.ToString
'CvInvoke.PutText(frameCentros, contaCarros.ToString, centroTexto,
FontFace.HersheySimplex, 2.0, COLOR_NEGRO, 4)

ImageBox1.Image = frameCentros

frame1 = frame2.Clone()
frame2 = CAM_VIDEO.QueryFrame()

If flagSleep Then
    flagSleep = False
    'Thread.Sleep(500)
End If

If carroContado Then
    flagSleep = True
End If

If frame2 Is Nothing Then
    cerrando_form = True
    MsgBox("Fin de video")
End If
End If
Application.DoEvents()
End While
End Sub

Private Function ProcesarImagen(frame1A As Mat, frame2A As Mat, frameDif As
Mat) As Mat
    CvInvoke.CvtColor(frame1A, frame1A, ColorConversion.Bgr2Gray) 'Grises

```

```

CvInvoke.CvtColor(frame2A, frame2A, ColorConversion.Bgr2Gray) 'Grises
CvInvoke.GaussianBlur(frame1A, frame1A, New Size(5, 5), 0) 'Difuminar
CvInvoke.GaussianBlur(frame2A, frame2A, New Size(5, 5), 0) 'Difuminar
CvInvoke.AbsDiff(frame1A, frame2A, frameDif) 'Diferencia
CvInvoke.Threshold(frameDif, frameDif, 30, 255.0, ThresholdType.Binary) 'Threshold
For i As Integer = 0 To 1
    CvInvoke.Dilate(frameDif, frameDif, factor_forma, New Point(-1, -1), 1,
        BorderType.Default, New MCvScalar(0, 0, 0)) 'Dilatar
    CvInvoke.Dilate(frameDif, frameDif, factor_forma, New Point(-1, -1), 1,
        BorderType.Default, New MCvScalar(0, 0, 0)) 'Dilatar
    CvInvoke.Erode(frameDif, frameDif, factor_forma, New Point(-1, -1), 1,
        BorderType.Default, New MCvScalar(0, 0, 0)) 'Erosionar
Next
Return frameDif
End Function

```

```

Private Sub TimerOn_Tick(sender As Object, e As EventArgs) Handles TimerOn.Tick
    empezarReconocimiento()
End Sub

```

```

Private Sub btnClose_Click(sender As Object, e As EventArgs) Handles
    btnClose.Click
    cerrando_form = True
    TimerOff.Start()
End Sub

```

```

Private Sub TimerOff_Tick(sender As Object, e As EventArgs) Handles TimerOff.Tick
    Me.Close()
End Sub

```

```

Dim pausaFlag As Boolean = False
Private Sub btnPausa_MouseDown(sender As Object, e As MouseEventArgs)
    pausaFlag = True
End Sub

```

```

Private Sub btnPausa_MouseUp(sender As Object, e As MouseEventArgs)
    pausaFlag = False
End Sub

```

```

Private Sub btnPausa_Click(sender As Object, e As EventArgs)
    'empezarReconocimiento()
End Sub

```

```

Private Sub btnReset_Click(sender As Object, e As EventArgs)
    contaCarros = 0
    lbl_contador.Text = "0"

```

End Sub

Private Sub TableLayoutPanel2_Paint(sender As Object, e As PaintEventArgs)

End Sub

Private Sub lbl_contador_Click(sender As Object, e As EventArgs) Handles
lbl_contador.Click

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
contaCarros = 0
lbl_contador.Text = "0"
End Sub

Private Sub RegistrosBindingNavigatorSaveItem_Click(sender As Object, e As
EventArgs)
Me.Validate()
Me.RegistrosBindingSource.EndEdit()
Me.TableAdapterManager.UpdateAll(Me.DatosVEHDataSet)

End Sub

'//////////Base de Datos*****

End Class

Anexo 4. Certificación del abstract por parte del centro de idiomas de la universidad.



CENTRO DE IDIOMAS

ABSTRACT

This work is about the use of artificial vision and the digital processing of images to monitor traffic in real time, also it targets the use of electronic devices such as the electronic card Latte Panda it is the electronic brain, additionally Microsoft Visual Basic software and the OPEN CV bookshops were used to reach the objective.

The MOTRAVE project was developed, it means "Monitoreo de Trafico Vehicular" this study determines the vehicular flow in different zones of the city and permits to be adaptable to the return variable to measure, it could be reached through out the programming, it means this device can isolate objects from the census offering some ways to do studies.

It has a data base which helps that data could be stored by date and time. This study determines the traffic flow on the streets in Azogues.

KEY WORDS: Artificial vision, Microsoft visual studio software, OPEN CV. Monitoring.

Azogues, 27 de julio de 2018

EL CENTRO DE IDIOMAS DE LA UNIVERSIDAD CATÓLICA DE CUENCA SEDE AZOGUES, CERTIFICA QUE EL DOCUMENTO QUE ANTECEDE FUE TRADUCIDO POR PERSONAL DEL CENTRO PARA LO CUAL DOY FE Y SUSCRIBO.

**LCD.A. GABRIELA ESTRELLA G. MST.
COORDINATOR**



Anexo 5. Certificación del funcionamiento del equipo.

AZOGUES
Academia de jóvenes abogados

**ING. PAULO SACOTO GARCIA, DIRECTOR DE MOVILIDAD DEL
GOBIERNO AUTÓNOMO DESCENTRALIZADO MUNICIPAL DE
AZOGUES.**

CERTIFICA:

Que, una vez probado el equipo denominado de Monitoreo de Tráfico Vehicular **"MOTRAVE"**, los días martes 15, miércoles 16, jueves 17, viernes 18, martes 22, y jueves 24 de mayo; viernes 1 y martes 5 de junio, y que, comparando con los conteos manuales programados certifico que el equipo de propiedad del Señor Hector Carangui funciona dentro de los rangos para los cuales ha sido diseñado, es decir el 5% de error, por lo tanto el equipo funciona correctamente.

Autorizo al Sr. Hector Carangui portador de la cedula de ciudadanía 0302628847 dar al presente certificado el uso que estime conveniente.

Atentamente,



Ing. Paulo Sacoto
DIRECTOR DE MOVILIDAD