ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

"IMPLEMENTACIÓN DE UN SISTEMA PROTOTIPO DE LUCES FRONTALES CON SEGMENTACIÓN PARA AUTOMOTORES EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL DIFUSAS"

TRABAJO DE TITULACIÓN

TIPO: DISPOSITIVO TECNOLÓGICO

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES

AUTORES: JHINSON EDGAR COYAGO TOMALO

JONATHAN JAVIER COYAGO TOMALO

TUTOR: Ing. Mgs. PABLO EDUARDO LOZADA YÁNEZ

Riobamba-Ecuador

2019

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

El tribunal de Trabajo de Titulación certifica que: El trabajo de investigación: IMPLEMENTACIÓN DE UN SISTEMA PROTOTIPO DE LUCES FRONTALES CON SEGMENTACIÓN PARA AUTOMOTORES EMPLEANDO TÉCNICAS DE VISIÓN ARTIFICIAL DIFUSAS, de responsabilidad de los señores Jhinson Edgar Coyago Tomalo; Jonathan Javier Coyago Tomalo, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	**FIRMA**	**FECHA**
Ing. Washington Gilberto Luna E. **DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**	_____	_____
Ing. Freddy Enrique Chávez Vásquez **DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES**	_____	_____
Ing. Pablo Eduardo Lozada Yánez **DIRECTOR DEL TRABAJO DE TITULACIÓN**	_____	_____
Ing. Alberto Leopoldo Arellano Aucancela **MIEMBRO DE TRIBUNAL**	_____	_____

Nosotros, Jhinson Edgar Coyago Tomalo; Jonathan Javier Coyago Tomalo, somos responsables de todos los resultados, ideas y modos expuestos en este Trabajo de Titulación y el patrimonio intelectual del trabajo de titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

Jhinson Edgar Coyago Tomalo	Jonathan Javier Coyago Tomalo
210048721-0	210048722-8

DEDICATORIA

El presente trabajo está dedicado a Dios, a mis padres, a mis abuelitas, familiares y a todas las personas que han influenciado de manera positiva a mi vida, haciendo que cada día sea mejor persona que lo era antes.

Jonathan

Me complace dedicar este trabajo primeramente a Dios, mi esposa e hijos y a mis padres porque han depositado la confianza y apoyo durante toda mi trayectoria de formación profesional.

Jhinson

AGRADECIMIENTO

Un agradecimiento sincero a la Escuela Superior Politécnica de Chimborazo, por permitirme la oportunidad de obtener una profesión y ser de ayuda a la sociedad.

A mi familia por todo el apoyo brindado durante todo el tiempo de estudio.

Jonathan

Agradezco a mis padres, esposa e hijos por estar siempre presentes en cada paso y decisión que he tomado para culminar este largo caminar y que ahora está llegando a su etapa final.

Jhinson

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

ABREVIATURAS

°C	Grados Celsius / Unidad de temperatura
µs	Microsegundos / Unidad de tiempo equivalente a una millonésima parte de un segundo
ARM	Advanced RISC Machine / Máquina RISC Avanzada
BLE	Bluetooth Low Energy / Bluetooth de baja energía
BMP	Bits Maps Protocole / Protocolo de mapas de bits
BMW	Bayerische Motoren Werke / Fábricas de motores bávara
cm	centímetro / Unidad de longitud equivalente a una centésima parte de un metro
CSI	Camera Serial Interface / Interfaz Serial de Cámara
DSI	Display Serial Interface / Interfaz Serial de Monitor
EEPROM	Electrically Erasable Programmable Read-Only Memory / Memoria de solo lectura programable y borrable eléctricamente.
Fps	Frame per second / Fotogramas por segundo
GB	GibaByte / Unidad de almacenamiento de información equivalente a mil millones de bytes
GHz	Giga Hertz / Gigahercio
GPIO	General Purpose Input-Output / Entrada/Salida de Propósito General
GPU	Graphics Processing Unit / Unidad de procesamiento gráfico
GUI	Graphical User Interface / Interfaz Gráfica de Usuario
HD	High-Difinition / Alta definición
HDMI	High-Definition Multimedia Interface / interfaz multimedia de alta definición
HTML	HyperText Markup Language / Lenguaje de Marcas de Hipertexto
kB	Kilobyte / Unidad de almacenamiento de información equivalente a mil bytes
LAN	Local Area Network / Red d Área Local
LCD	Liquid Cristal Display / Representación visual por cristal líquido
LED	Light-Emitting Diode / Diodo Emisor de Luz
LPDRR	Low-Power Double Data Rate / Tasa de datos doble de bajo consumo
MHz	MegaHertz / Megahercio
mm	Milímetro / Unidad de longitud equivalente a una milésima parte de un metro
ms	Milisegundo / Unidad de tiempo equivalente a una milésima parte de un segundo
pH	Medida de acidez o alcalinidad de una disolución
PWM	Pulse-Width Modulation / Modulación por Ancho de Pulsos
RISC	Reduced Instruction Set Computer / Ordenador con Conjunto Reducido de Instrucciones

RGB	Red-Green-Blue / Rojo-Verde-Azul
ROI	Region of Interest / Región de Interés
SD	Secure Digital / Dispositivo en formato de tarjeta de memoria para dispositivos portátiles
SDRAM	Synchronous Dynamic Random-Access Memory / Memoria de acceso aleatorio síncrona y dinámica
SIFT	Scale-invariant feature transform / Transformación de la característica invariante de escala
USB	Universal Serial Bus / Bus Universal en Serie
V	Volts / Unidad de tensión eléctrica
VAC	Voltage AC / Voltaje de Corriente Alterna
VDC	Voltage DC / Voltaje de Corriente Directa
VGA	Video Graphics Array / Matriz de gráficos de vídeo
W	Watts / Unidad de medida de la potencia eléctrica
WIFI	Wireless Fidelity / Fidelidad sin cables o inalámbrica

RESUMEN

Debido a la gran cantidad de accidentes de tránsito que ocurren en la noche por imprudencia e impericia de conductores que no realizan los cambios de luces de altas a bajas, se implementó un sistema de visión artificial difusa para la segmentación respectiva de luces largas, con la finalidad de lograr el no deslumbramiento hacia los demás conductores que transiten en la vía. La construcción del prototipo se realizó con la conformación de una Raspberry Pi 3, Arduino UNO, Servomotores, Barra LED, relé automotriz y elementos de protección. Para el desarrollo del sistema de detección se hizo énfasis en el algoritmo de detección "Haar Cascade", la interfaz gráfica fue desarrollada en el entorno Qt Creator. Cuando el sistema detecta la presencia de un vehículo que circule en sentido contrario o en el mismo sentido, este envía el ángulo determinado por las reglas de la lógica difusa hacia el Arduino haciendo que los servomotores se muevan, logrando la respectiva segmentación de luz. El clasificador en cascada permite crear una serie de datos admisibles al reconocimiento de vehículos determinando la posición del mismo. La interfaz gráfica será visualizada por el usuario permitiendo manipular 6 botones, cada uno cumple funciones específicas como: iniciar cámara, parar, detectar, graficar fuzzy, conectar y desconectar. Según las pruebas realizadas, se establece que la distancia máxima a la que puede detectar un vehículo es aproximadamente 200 metros; de igual manera el sistema presenta mejores resultados a bajas velocidades debido a un retardo en la adquisición de las imágenes en tiempo real.

PALABRAS CLAVE: <VISIÓN ARTIFICIAL DIFUSA> <DETECCIÓN DE VEHÍCULOS> <DESLUMBRAMIENTO> <SEGMENTACIÓN DE LUZ> <ALGORITMO HAAR CASCADE> <PROCESAMIENTO DE IMÁGENES> <INTERFAZ GRÁFICA> <BARRA LED>

ABSTRACT

Due to the large number of traffic accidents that occur at night due to recklessness and inexperience of drivers who do not perform the lighting raises from high to low, a diffuse artificial vision system was implemented for the respective segmentation of long lights, with the purpose of achieving non-glare towards other drivers who travel on the road The prototype construction was made with the conformation of a Raspberry Pi 3, Arduino ONE, Servomotors, LED bar, automotive relay and protection elements. For the development of the detection system, emphasis was placed on the detection algorithm "Haar Cascade", the graphic interface was developed in the Qt Creator environment. When the system detects the presence of a vehicle traveling in the opposite direction or in the same direction, it sends the angle determined by the rules of fuzzy logic to the Arduino causing the servomotors to move, achieving the respective light segmentation. The cascade classifier allows you to create a series of data admissible to the vehicle recognition determined the position of it. The graphical interface will be visualized by the user allowing to manipulate 6 buttons, each one fulfills specific functions such as: start camera, stop, detect, graph fuzzy, connect and disconnect. According to the tests carried out, it is established that the maximum distance at which a vehicle can detect is approximately 200 meters; in the same way, the system presents better results at low speeds due to a delay in the acquisition of the images in real time.

KEYWORDS: <DIFFUSE ARTIFICIAL VISION> <DETECTION OF VEHICLES> <DAZZLING> <LIGHT SEGMENTATION> <HAAR CASCADE ALGORITHM> <PROCESSING OF IMAGES> <GRAPHIC INTERFACE> <LED BAR>

INTRODUCCIÓN

En la actualidad una de las técnicas de mayor impacto tecnológico y revolucionario a escala mundial es la visión artificial, misma que es empleada en múltiples disciplinas científicas de tal modo que sus aplicaciones pueden ser tanto industrial, como para sistemas no industriales en los cuales necesitan un reconocimiento visual de un entorno físico para tomar acciones que cumpla con el objetivo deseado.

Los vehículos convencionales en el Ecuador no están dotados un sistema de luces inteligentes que evite el deslumbramiento de los conductores que circulan por la misma vía tanto en sentido contrario como en el mismo sentido (por reflexión de luz en los retrovisores), generando directamente molestias visuales a los usuarios de automotores. Muchas de las veces el deslumbramiento es causado por la imprudencia e impericia de los conductores, esto ha generado muchos accidentes de tránsito, mismos que han sido leves, graves e incluso mortales.

La implementación del sistema prototipo pretende ser la base para el futuro desarrollo de dispositivos genéricos adaptables a cualquier tipo de vehículo convencionales, generando una solución accesible a los múltiples usuarios sin la necesidad de cambiar de vehículo para gozar de este servicio, actualmente existen autos que ya tienen integrados este tipo de tecnología por distintas marcas reconocidas tales como Toyota, Mercedes-Benz, BMW y otras, pero que su costo es muy elevado y solo se encuentra disponible para determinados modelos.

El prototipo contará con una cámara Raspberry Pi encargada de la adquisición de imágenes en tiempo real, un algoritmo de visión artificial que procese las mismas dentro de una unidad central de procesos Raspberry Pi3, que permita conocer los valores de las variables independientes para proceder a ejecutar la segmentación de luz, es decir, una o más porciones se desviarán de su proyección normal para no deslumbrar y las porciones restantes se mantendrán para aumentar la visibilidad de la carretera por la que se está circulando, de modo que el conductor no tendrá que preocuparse por este tipo de acciones y su concentración al conducir será la más óptima.

El sistema prototipo será capaz de detectar los vehículos mediante un clasificador de cascada (*haar cascade*), el cual previamente ha sido entrenado con imágenes captadas por la propia cámara, esto con el fin de adquirir las variables independientes que mediante condiciones de lógica difusa procede a enviar las señales respectivas a cada actuador y así lograr el desvío del haz luminoso tratando lograr el no deslumbramiento hacía los demás conductores.

PLANTEAMIENTO DEL PROBLEMA

¿Cómo la implementación de un sistema prototipo de luces inteligentes frontales con segmentación evitará el deslumbramiento hacia los conductores de automotores?

¿Cómo se podría detectar las luces o presencia de otros vehículos, mediante técnicas de visión artificial difusas?

¿Cómo se realizará la segmentación de luz controlada por actuadores y un circuito electrónico?

JUSTIFICACIÓN

La implementación del sistema prototipo planteado en el presente trabajo de titulación pretende ser la base para el futuro desarrollo de dispositivos genéricos adaptables a cualquier tipo de vehículo convencionales, generando una solución accesible a los múltiples usuarios sin la necesidad de cambiar de vehículo para gozar de este servicio, actualmente existen autos que ya tienen integrados este tipo de tecnología por distintas marcas reconocidas tales como Toyota, Mercedes-Benz, BMW y otras, pero que su costo es muy elevado y solo se encuentra disponible para determinados modelos.

El prototipo contará con una cámara HD oficial de Raspberry Pi encargada de la adquisición de imágenes en tiempo real, un algoritmo de visión artificial que procese las mismas dentro de una unidad central de procesos Raspberry Pi que permita conocer los valores de las variables independientes para proceder a ejecutar la segmentación de luz, es decir, una o más porciones se mantendrá en baja para no deslumbrar y las porciones restantes en alta para aumentar la visibilidad de la carretera por la que se está circulando, de modo que el conductor no tendrá que preocuparse por este tipo de acciones y su concentración al conducir será la más óptima.

OBJETIVOS

Objetivo general

- Implementar un sistema prototipo de luces inteligentes frontales con segmentación para automotores, empleando técnicas de visión artificial difusas.

Objetivos específicos

- Investigar algoritmos de visión artificial enfocados al reconocimiento de patrones.

- Experimentar métodos de segmentación o patrones de luz de acuerdo a la proyección de sombras frente a una fuente lumínica.

- Desarrollar el algoritmo de visión artificial que permita el procesamiento de los patrones y con la información obtenida generar las acciones requeridas en los actuadores.

- Diseñar un circuito electrónico que permita la comunicación entre todos los elementos que conformarían todo el sistema.

- Implementar el sistema prototipo con su respectivo algoritmo de visión y circuito electrónico para manipular la segmentación de luz.

- Comprobar el comportamiento del sistema prototipo implementado en su totalidad mediante pruebas experimentales.

CAPÍTULO I

1. MARCO TEÓRICO

1.1 Iluminación en vehículos

El sistema de alumbrado en vehículos permite ejercer la conducción con seguridad al aportar la iluminación necesaria para ver y ser vistos con claridad. La seguridad en los vehículos se debe primordialmente al alumbrado que posee, ya que se puede circular en lugares de baja visibilidad, ayudando a visualizar con claridad el entorno e informando a los demás conductores sobre la presencia del mismo en la carretera. (Jiménez, J. 2018)

Según la normativa nacional (INEN 2009) establece que el color de las luces emitidas por el vehículo está estandarizado, los faros posteriores serán de color rojo, los direccionales de color ámbar y los frontales amarillos o luz blanca, a excepción de los vehículos de emergencia. (Jiménez, J. 2018)

En la Figura 1-1 se observa la ubicación y detalle de las luces que posee un vehículo en general y de acuerdo a la ubicación, existen 3 grupos:

- Faros y luces auxiliares de iluminación delantera
- Faros frontales, laterales y traseros de iluminación
- Luz interior de cortesía y otros dispositivos lumínicos

Figura 1-1: Sistema de Iluminación en el vehículo
Fuente: https://goo.gl/Lts1oh

1.2 Tipos de lámparas o emisores de luz

Existen cuatro tipos de lámparas que en la actualidad son usados en los automóviles y no necesariamente luces modernas. Las lámparas incandescentes tienen más de 130 años de antigüedad (aunque han ido evolucionando en el pasar del tiempo), los más modernos son las lámparas LEDs tienen más de 80 años desde su creación. (Ibañez, P. 2011)

1.2.1 Incandescentes

Las lámparas incandescentes pasan por un filamento metálico (wolframio) que actúa como resistencia a la corriente eléctrica provocando el cambio de temperatura desprendiendo luz y calor. El filamento está encerrado en una ampolla de vidrio al vacío, en algunas ocasiones es rellenada con un gas noble llamado kriptón. Estas lámparas son las más ineficientes y de duración escasa. (Ibañez, P. 2011)

1.2.2 Halógenas

Estas lámparas mantienen el mismo principio que las incandescentes, la diferencia está en la ampolla que contiene un gas halógeno haciendo que el filamento dure más, proyecte mayor luz de color blanco. Debido a que alcanza mayores temperaturas, se reemplaza la ampolla convencional (vidrio de arena de silicio) por una ampolla de vidrio de cuarzo, en este tipo de ampollas se debe evitar tocarlas con los dedos desnudos, debido a que el pH puede dañar el vidrio. Fueron el primer cambio en los faros de los vehículos permitiendo tener más luz. (Ibañez, P. 2011)

1.2.3 Xenón

Las lámparas Xenón también son conocidas como lámparas de descarga de gas, dentro de la ampolla de cuarzo ya no se coloca un filamento metálico sino dos electrodos de tungsteno demasiado cercanos, pero no en contacto. El relleno de la ampolla es de mercurio, sales metálicas y gas xenón. El funcionamiento de esta lámpara se debe a que llega la corriente eléctrica a uno de los electrodos, esta salta hasta el otro produciendo un arco eléctrico que desprende una gran cantidad de luz muy blanca ligeramente azulada. (Ibañez, P. 2011)

El consumo en relación a las lámparas halógenas es menor a pleno funcionamiento, con este tipo de lámparas se ha otorgado un gran cambio con respecto a la iluminación en los vehículos, pues

se obtiene más luz, homogénea y blanca, la desventaja que representa se ve afectado en lo económico, ya que su precio supera elevadamente a las antes mencionadas. (Ibañez, P. 2011)

Para tener una mejor perspectiva de la proyección del haz de luz en la Figura 2-1, se puede observar una comparación en cuanto a profundidad de iluminación entre una lámpara halógena y de xenón.

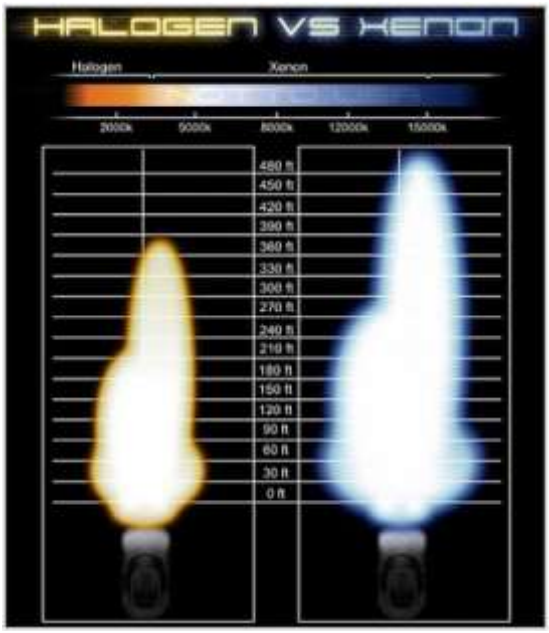


Figura 2-1: Comparación entre lámpara halógena y de xenón
Fuente: https://goo.gl/sfhQiK

1.2.4 LED

Los Diodos Emisores de Luz (*Ligth-Emitting Diode*) básicamente se constituyen de un material semiconductor encapsulado en un pequeño lente de plástico, para la emisión de luz es necesario pasar una corriente eléctrica a baja tensión. Son los que más sobresalen en los últimos años, en la Figura 3-1 se muestra un ejemplo de faro con diodo emisor de luz, ya sea por su estética o por su inferior consumo, cada vez se ven más coches con faros o pilotos LEDs, se está imponiendo en las luces de posición diurna debido a su alto brillo, también en las luces posteriores, luces de freno y las intermitentes. (Ibañez, P. 2011)

Figura 3-1: Faros LED
Fuente: https://goo.gl/szztPK

1.3 Diseño del faro, reflectores y proyectores

En la antigüedad los faros eran translúcidos, el cristal de dispersión no permitía ver el interior del cristal, estaba tallado interiormente desde un punto de vista óptico formando prismas horizontales que son los encargados de distribuir el haz de luz.

En la actualidad se ha cambiado a cristales de dispersión transparentes (normalmente de policarbonato). El diseño geométrico del reflector, ya sea parabólico o elíptico, son los encargados de controlar la distribución del haz de luz que genera dicha lámpara. (Ibañez, P. 2011)

La luz generada por una lámpara se irradia en todas direcciones, el objetivo principal de un faro es de redireccionar dichos rayos de luz hacia zonas específicas.

En función del enfoque de las lámparas, se puede encontrar tres tipos de faros:

- **Faros de superficie simple:** se constituye de un reflector en forma de parábola y un cristal que es el encargado de distribuir los rayos de luz (son costosos de fabricar y roban algo de luz).
- **Faros de superficie compleja:** el reflector es diseñado para redirigir los rayos de luz, ya que el cristal es más liso y por ende existen menos perdida de luz.
- **Faros elipsoidales:** en este tipo se sustituye el faro por una lente y el reflector tiene forma elipsoide con el fin de concentrar mayor luz en el foco de la lente. Presentaban un gran defecto al ser usados con lámparas halógenas, éstas se oscurecían por ello se aprovecha mejor con lámparas de descarga de gas porque producen menor temperatura. (Fidalgo, R. 2013)

En función al cambio de luz baja a luz alta, existen los siguientes tipos:

- **Faros de una parábola y lámpara H4:** en este tipo existe una sola bombilla en el faro con la peculiaridad que presentan dos filamentos distanciados milimétricamente, con el fin de

provocar una distancia focal en el faro y que su luz se proyecte de dos formas. Haciendo el encendido de uno u otro filamento se puede observar el cambio de luces bajas a altas. El defecto que presentan los dos filamentos es que no se pueden encender al mismo tiempo, y esto provoca que, al pasar a luces largas, se pierda visibilidad de zonas cercanas del vehículo.

- **Faros de doble parábola:** este se constituye de dos faros en un solo cristal, dicho reflector está dividido en dos secciones, una realiza la luz baja y la otra las luces altas. Las luces bajas emitidas por este faro es de menor intensidad que las que presentan los faros de una parábola, pero es más homogénea y se puede mantener encendidas las luces al mismo tiempo.
- **Faros de casquillo móvil:** un pequeño motor es el encargado del cambio de luces bajas a altas, haciendo mover la lámpara dentro del faro ocasionando un cambio de la distribución de la luz.
- **Faros de cortinilla:** son los más usados en la actualidad, el diseño de este faro está destinado a crear una huella de luz enorme en la carretera (ocasiona deslumbramiento), para evitar deslumbrar a los conductores se ha diseñado una cortinilla en las luces bajas, la función que cumplen es tapar parte de la luz que irradia la lámpara. Cuando se activan las luces altas, dichas cortinillas se abren y deja emitir toda la luz. (Fidalgo, R. 2013)

1.4 Asistente de luz de carretera

En la actualidad muchas de las marcas de automóviles ofrecen este tipo de servicio que lo denominan asistente de luz de carretera. Este dispositivo hace que cambien las luces de bajas a altas de manera automática, es decir, sin la intervención del conductor. Pero este sistema ya ha sido implementado en vehículos americanos en los años 50, en aquel entonces se lo conocía como "el ojo autrónico", el cual consistía de un sensor de luz (célula fotoeléctrica) ubicado a la altura de los ojos del conductor, haciendo que este active un relé para cambiar las luces de alta a baja cuando le llegaba la luz del carro contrario. (Fidalgo, R. 2013)

En los autos modernos, se reemplaza el sensor por una cámara (colocada en el retrovisor central), lo cual detecta las luces de otro automóvil e inmediatamente envía la información hacia la computadora (unidad de control de luces), ejecutando dos posibilidades de accionamiento:

- El sistema ordena pasar de luces largas a cortas o viceversa.
- El sistema puede mantener las luces de largo alcance, pero evitando el deslumbramiento hacia los demás conductores, esto lo logra tapando parte del haz de luz, creando una sombra en dicha zona, o en el caso de un sistema Matrix LED hace que apague y encienda los leds correspondientes para evitar deslumbrar al conductor del vehículo que circula en sentido contrario y del mismo sentido ubicado al frente, como se observar en la Figura 4-1. (Fidalgo, R. 2013)

Figura 4-1: Sistema automático de luces frontales
Fuente: https://goo.gl/vZAWMJ

1.5 Visión Artificial

1.5.1 Definición

Considerada como una rama de la Inteligencia Artificial, la visión por computadora se encarga de aplicar métodos para la adquisición, procesamiento y análisis de imágenes que son tomadas del mundo real mediante una cámara (ver Figura 5-1), con la finalidad de enviar la información hacia la máquina para que puedan distinguir y entender las cosas, mediante una imagen o secuencia de imágenes y que ésta realice la actividad para lo cuál sea diseñada. (Montoya, D. & Pachacama, A. 2016. p1)

Figura 5-1: Visión Artificial
Fuente: https://bit.ly/2DT3ZJ

Los avances tecnológicos han hecho que la visión artificial se desarrolle a pasos enormes, debido a los nuevos hardware que procesan mayor cantidad de información y en tiempo casi real, otra de las ventajas es que pueden ser incorporados en cualquier sistema electrónico, debido a que su tamaño físico es cada vez más óptimo. (Montoya, D & Pachacama, A. 2016. p2)

En cuanto a lo que se refiere a software, también hay que destacar que existen múltiples herramientas de software libre que ayudan a desarrollar los proyectos con la utilización de librerías previamente ya desarrolladas, donde se hace uso de las funciones que permiten cumplir con el propósito al cual se está proyectando. (Montoya, D. & Pachacama, A. 2016. p2)

1.5.2 Beneficios

La visión artificial tuvo gran impacto en el campo del control de calidad de los productos finalizados, donde se realiza análisis y procesamiento de imágenes de los productos terminados, con el fin de verificar que dicho producto cumpla con todas las normas de calidad establecidas por la empresa. (Montoya, D. & Pachacama, A. 2016. p2)

Con el desarrollo tecnológico cada vez es mayor la capacidad de realizar el análisis y procesamiento de imágenes en tiempo real, gracias a esto es posible realizar las correcciones respectivas o aplicar una acción de modo inmediato en un determinado proceso. (Montoya, D. & Pachacama, A. 2016. p2)

Empleando dispositivos ópticos como cámaras se puede diferenciar ciertos objetos en las imágenes y sus características, claramente esta función también es realizada por el ojo humano. Y si se requiere captar características de los objetos que son imperceptibles al ojo humano, en este caso la visión artificial toma gran ventaja ya que con la ayuda de distintos tipos de cámaras se puede detectar estas características. Las personas no son capaces de detectar las radiaciones por calor, infrarrojo, ultravioletas o en ambientes de poca iluminación como se muestra en la Figura 6-1, es factible reconocer estas características particulares de los objetos, haciendo uso de cámaras que sean capaces de detectar este tipo de radiaciones es viable hacerlo. (Montoya, D. & Pachacama, A. 2016. p3)

Figura 6-1: Cámara de Visión Nocturna
Fuente: https://goo.gl/3LK466

En la actualidad los sistemas de visión artificial son accesibles para todos los usuarios, permite mejorar o desarrollar nuevos proyectos con la finalidad de seguir aportando un avance a la visión por computador. La accesibilidad a estos componentes (procesadores, cámaras, ópticas, etc.) hacen de la visión artificial un hecho que ha ido reduciendo sus costos debido al gran impacto tecnológico que se vive hoy en día. (Montoya, D. & Pachacama, A 2016 p3)

1.5.3 Aplicaciones

Existen numerosas y diversas aplicaciones basadas en visión artificial, que pueden ser aplicados en cualquier campo que se desee.

- Identificación e inspección de objetos.
- Determinación de la posición de los objetos en el espacio.
- Establecimiento de relaciones espaciales entre varios objetos (guiado de robots).
- Determinación de las coordenadas importantes de un objeto.
- Realización de mediciones angulares.
- Mediciones tridimensionales. (ETI 2010)

1.6 Definiciones Preliminares

A continuación, se detallan diversos conceptos que hará más fácil el entendimiento de visión artificial.

1.6.1 Luz Visible

La luz está establecida por ondas luminosas con diferentes longitudes, el espectro visible para el ojo humano lo conforman las longitudes que van de los 200nm a los 700nm de acuerdo al espectro que se muestra en la Figura 7-1. Éstas ondas luminosas son captadas por los ojos, posteriormente envía estímulos al cerebro el cual interpreta las diferentes amplitudes y longitudes de onda de luz, produciendo las sensaciones que se conocen como brillo y color de los objetos presentes en las imágenes captadas. (Garzón, P. & Sola, J. 2016 p2)

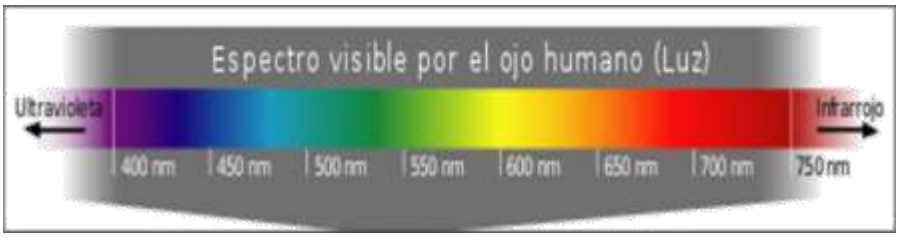


Figura 7-1: Espectro Visible de la luz
Fuente: https://goo.gl/o2cr4X

1.6.2 Pixel

Un pixel es la unidad mínima que forma parte de una imagen digital, haciendo énfasis a la superficie más pequeña uniforme que hace parte de la imagen. Cada uno de los pixeles presentes en la imagen están definidas por su nivel de gris, intensidad o de coordenadas (x, y), donde el origen de coordenadas es la esquina superior izquierda de la imagen. Cada pixel almacena el color que contiene en formato binario mismo que se puede apreciar en la Figura 8-1, el número de bits que contiene cada pixel se denomina *profundidad de color.* (Montoya, D & Pachacama, A.2016 p7)

- Un bit por pixel: $2^1 = 2$ colores (blanco y negro)
- Cuatro bits por pixel: $2^4 = 16$ colores (VGA)
- Ocho bits por pixel: $2^8 = 256$ colores (Súper VGA)
- 24 bits por pixel: $2^{24} = 16{,}7$ millones de colores (RGB)

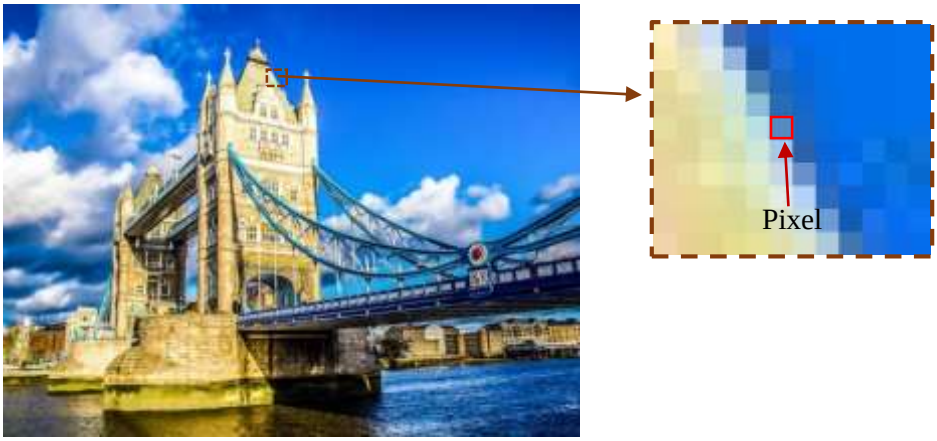


Figura 8-1: Vista de un pixel en una imagen digital
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

1.6.3 Imagen digital

En la actualidad se está presentando procesos de digitalización a toda la información del mundo real. La imagen digital se obtiene mediante dispositivos electrónicos como son las cámaras fotográficas, filmadoras, scanner, celulares, entre otros.
Las imágenes digitales se pueden clasificar en dos grandes grupos, los cuales se detallan a continuación. (Montoya, D. & Pachacama, A. 2016 p8)

1.6.3.1 Mapa de bits

En la Figura 9-1 se observa un ejemplo de una imagen en mapa de bits.

Figura 9-1: Imagen en mapa de bits
Fuente: https://www.elvisualista.com/2016/05/05/que-es-un-mapa-de-bits-1/

Estas imágenes se encuentran denotadas por una matriz de puntos o bits, una imagen es el resultado de la yuxtaposición de pixeles, cada pixel con su respectiva profundidad de color. (Julio, 2012, p 1)

1.6.3.2 Imágenes Vectoriales

La imagen vectorial es aquella que está formada digitalmente por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), la Figura 10-1 es un claro ejemplo, de los cuales cada uno se encuentra definido por diversos atributos matemáticos, ya sea de forma, posición y color. (Julio, 2012, p 5)

Figura 10-1: Imagen Vectorial
Fuente: https://goo.gl/zyMy9B

Al hacer modificaciones a una imagen digital vectorial, ésta no se distorsiona o pixela, ya que realmente no se está modificando la imagen, sino que se está redefiniendo los elementos que la constituyen, con lo cual se puede ampliar, disminuir, estirar o retorcer la imagen digital.

Las imágenes vectoriales tienen una serie de ventajas en comparación a las imágenes de mapa de bits (BMP):

- Ocupan poco espacio de memoria.
- No se pierde la calidad de la imagen al ser modificada.
- Con menos cálculos del microprocesador se puede dar animación a los objetos.

La desventaja que tiene la imagen vectorial ante los mapas de bits, es que no se puede almacenar grandes cantidades de información, ya que requieren ciertos cálculos por parte del microprocesador.

1.6.4 Resolución de una imagen digital

Figura 11-1: Imagen digital con distinta resolución
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

La resolución de una imagen se puede definir como la calidad visual o cantidad de pixeles que tiene la imagen, como se observa en la Figura 11-1.

La cantidad de información que deberá ser procesada, depende de la resolución de la imagen, lo que implica un mayor coste computacional, también es entendible que una imagen con mayor resolución sea la más óptima al momento de procesarla. (Montoya, D & Pachacama, A. 2016, p 8)

1.6.5 Región de Interés

Figura 12-1: Región de interés (ROI)
Fuente: https://goo.gl/twfrx8

También conocida por sus siglas en inglés como ROI (*Region of Interest*), lo cual define a una parte seleccionada de la imagen que se requiere al momento de realizar algún tipo de procesamiento, en la Figura 12-1 se puede observar cómo se delimita la ROI de una imagen captada, esta región de interés depende de la aplicación que se está realizando y la capacidad del hardware para procesar mayor o menor cantidad de datos. (Montoya, D & Pachacama, A. 2016, p 9)

1.6.6 Brillo

Se entiende por brillo a la capacidad de un objeto, dependiendo de su color, para reflejar la luz blanca que incide sobre él. (Montoya, D & Pachacama, A. 2016, p 9)

1.6.7 Contraste

Se define como la diferencia de intensidad que existe entre un pixel de la imagen y sus pixeles cercanos a este. (Montoya, D & Pachacama, A. 2016, p 9)

1.6.8 Tono

También conocido como matiz, el tono se lo puede interpretar como la sensación visual captada por la vista en donde indica si determinada área de la imagen se hace similar a los colores amarillo, azul, rojo o verde. (Montoya, D & Pachacama, A. 2016, p 9)

1.6.9 Saturación

La saturación es la propiedad que describe la pureza del color, mientras haya mayor saturación más puro, vivo o colorido se verá la imagen, en la Figura 13-1 se observa los niveles de saturación para un color. (Montoya, D & Pachacama, A. 2016, p. 9)

Figura 13-1: Niveles de saturación
Fuente: https://goo.gl/Ztso6w

1.7 Modelo Fisiológico de Visión

1.7.1 Sistema visual

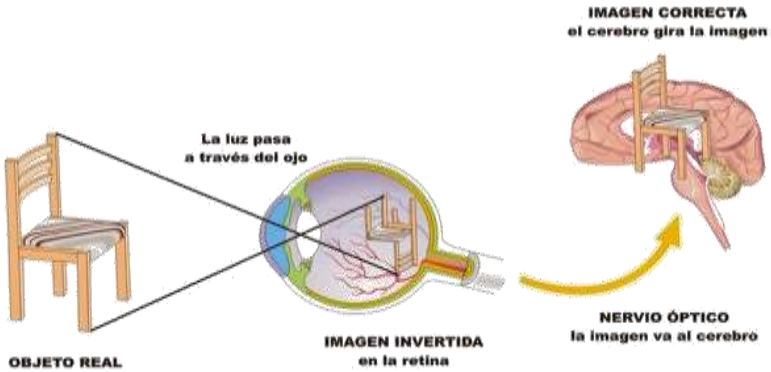


Figura 14-1: Funcionamiento del Sistema visual humano
Fuente: https://goo.gl/S3JnHP

El órgano fundamente para el sistema visual es el ojo humano, éste ayuda a capturar la luz y la convierte en impulsos permitiendo ser procesada por el cerebro, generando sensaciones para completar el sistema de visión, como se muestra en la Figura 14-1.

1.7.2 Esquema de un sistema de visión artificial

Un sistema de visión artificial trata de asimilar el comportamiento que tiene el cerebro al recibir las imágenes captadas por la vista, este proceso se manifiesta en cuatro etapas, los cuáles de detallan a continuación en la Figura 15-1. (Velez, J. *et* 2003 p. 18)

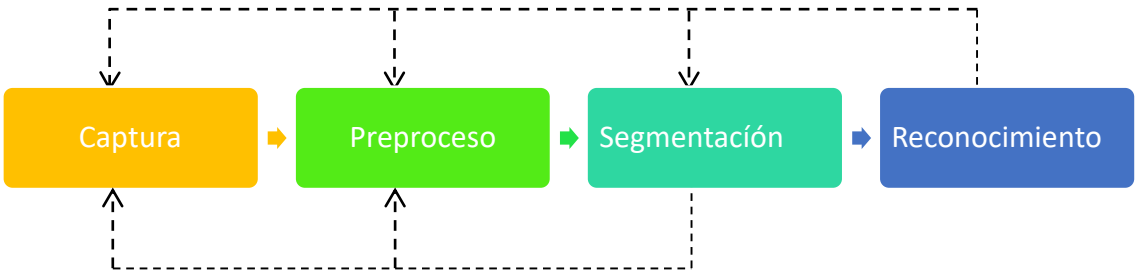


Figura 15-1: Etapas de un sistema de visión por computador
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2018

- **Captura:** en esta etapa se adquieren las imágenes mediante un sensor (cámara, scanner, filmadoras, celulares, entre otros) y se digitalizan.
- **Pre-proceso:** se realiza un filtrado a la imagen con la finalidad de eliminar elementos innecesarios y dar énfasis a las características de interés.
- **Segmentación:** se realiza la separación de los elementos de interés de la imagen para comprender y realizar el respectivo procesamiento.
- **Reconocimiento:** se distingue los elementos que han sido separados previamente, mediante parámetro y características. (Garzón, P. & Sola, J. 2016, p. 8)

No necesariamente se debe seguir una secuencia, porque se recomienda regresar a etapas anteriores con la finalidad de corregir ciertos errores o hipótesis falsas, que ayuden al desarrollo de las mismas.

En la fase de pre-proceso se basa en el modelo de visión general Marr-Palmer, presentado en la Figura 16-1. Palmer basándose en el modelo de David Marr hace la presentación de un modelo de visión que ayudan a realizar el análisis del problema a nivel computacional. (Garzón, P. & Sola, J. 2016, p.8)

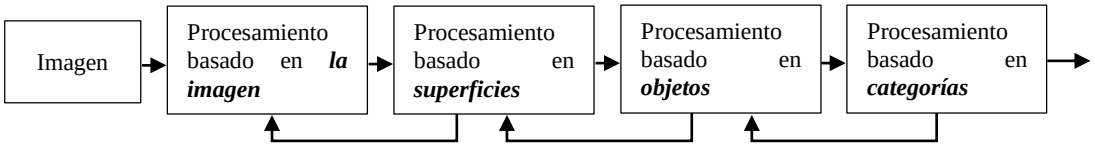


Figura 16-1: Modelo general de visión Marr-Palmer
Realizado por: Coyago Jhinson; Coyago Jonathan; 2018

- **Procesamiento a nivel de la imagen:** en este proceso se obtiene el contenido de cada pixel presente en la imagen.
- **Procesamiento basado en la superficie:** aquí se introduce información adicional que hará que el problema se regularice, tales como configuraciones de luz, colore, entre otros.
- **Procesamiento basado en objetos:** al igual que en el procedimiento aquí se introducen información o conocimiento adicional para regularizar los problemas.
- **Procesamiento basado en categorías:** se busca relaciones entre objetos en una imagen. (Garzón, P. & Sola, J. 2016, p. 9)

1.8 Segmentación

Se define como el proceso de separar en regiones la imagen con sus objetos de interés para determinada aplicación.

1.8.1 Segmentación supervisada

Es necesario la intervención humana durante el proceso para determinar qué características de la imagen son de interés.

1.8.2 Segmentación no supervisada

Es innecesario adquirir información adicional dada por un ser humano.

1.8.3 Segmentación a nivel de objeto

La segmentación S de una imagen I es un conjunto de regiones de la imagen que satisface la ecuación (1).

$$|L(\mathcal{R}_i)| = 1 \, para \, \mathcal{R}_i \in S, i = 1 \dots n_{opt} \qquad (1)$$

Esta condición establece que el operador de reconocimiento (L) para una región es un solo objeto. Se asegura que no hay varios objetos para una región.

n_{opt} es el valor óptimo de regiones etiquetadas con el operador de reconocimiento para un objeto a segmentar.

$n_{opt} = |S|\ es\ mínimo$

Esta condición indica que no se está sobre-segmentando. (Garzón, P. & Sola, J. 2016, p. 10)

1.8.4 Segmentación a nivel de imagen

Basado en el criterio de homogeneidad, se definen regiones como grupos de pixeles, los cuales comparten cierta característica, suponiendo que los objetos se componen de superficies homogéneas.

Se define la segmentación basada en imagen en términos de homogeneidad H_l y adyacencia $\mathcal{A}$.

La segmentación a nivel de imagen satisface que para cada región $\mathcal{R}_i \in S_i$, $H_l(\mathcal{R}_i) = verdadero$, y $H_l(\mathcal{R}_i \cup \mathcal{R}_j) = falso$ para $\mathcal{A}(\mathcal{R}_i, \mathcal{R}_j) = verdadero$, es decir, para cada región en un grupo de regiones el criterio de homogeneidad da verdadero si se analiza cada región por separado, pero si se unen dos regiones adyacentes este criterio da falso. Con esto se logra prevenir la sobre-segmentación. (Garzón, P. & Sola, J. 2016, p. 10)

Cada método de segmentación provee su manera particular de definir la homogeneidad e impone sus restricciones para una región.

1.8.5 Métodos de segmentación a nivel de imagen.

La segmentación con base al espacio de características toma en cuenta la definición de pixel $e_i = (p_i, c_i)$, aunque se ignora la información p (posición del pixel), se muestran a continuación los siguientes métodos.

1.8.5.1 Umbralización

En este método se parte la imagen en dos regiones: el objeto y el fondo. Todos los pixeles que tengan valores característicos (c) mayores a un umbral (T), se asignan como fondo y el resto como objeto.

El umbral o threshold puede ser definido por el usuario o aproximado automáticamente (algoritmo de Otsu). (Garzón, P. & Sola, J. 2016, pp. 10–11)

1.8.5.2 Histogramas

Este método se aplica a imágenes en grises, y con esto se obtiene un solo histograma en la cual se hace la detección de máximos para determinar las diferentes clases dentro la imagen.

1.9 Algoritmos de detección de imágenes

1.9.1 Algoritmo de Canny

Figura 17-1: Algoritmo de detección Canny
Fuente: https://goo.gl/b3gyiy

Para detectar bordes como se muestra en la Figura 17-1, se usa un método que hace relación a la matemática como la primera derivada, esta se utiliza porque adquiere el valor de cero en las regiones donde no se altera la intensidad, teniendo un valor constante en toda la transición de intensidad. (Castillo, Y. & Yancha, M. 2017, p. 23)
En este algoritmo se encuentran tres importantes pasos:

- **La obtención del gradiente:** Se calcula la magnitud y orientación del vector gradiente por cada pixel.
- **Supresión no máxima:** Como resultado se puede lograr una reducción del ancho de los bordes, logrando así, bordes de un pixel de ancho.
- **Histéresis de umbral:** Permite disminuir la posibilidad de la aparición de contornos falsos. (Rebaza, J. 2014, p. 3)

1.9.2 Algoritmo SIFT

En este algoritmo la función principal que desempeña es extraer características específicas de las imágenes en escala de grises, teniendo por resultado el reconocimiento de dicha imagen que esté

dentro de una base de datos. Mediante una serie de pasos se puede desarrollar este tipo de algoritmo, los cuáles se mencionan a continuación. (Flores, P. & Braun, J. 2011 pp. 1–5)

- **Detección de extremos en el espacio-escala:** En este paso se localizan puntos invariantes a la traslación, escalado y rotación de la imagen teniendo en cuenta que tenga una afectación mínima a los ruidos.
- **Localización exacta punto clave:** La búsqueda en este paso produce varios extremos en los cuales se encuentran puntos con escaso contraste, dichos puntos no permanecen estables a cambios de ruido e iluminación.
- **Asignación de orientación:** En este paso los puntos clave llegan a ser descriptivos relativos a estas orientaciones y así obtener características invariantes a las rotaciones.
- **Descriptor de puntos clave:** Se determina para cada punto clave un descriptor invariante a los diversos cambios de iluminación y afinidades.

1.9.3 Algoritmo Tracking de objetos

Este método es uno de los que mayormente se usa debido a su facilidad de entendimiento y su entorno de trabajo. Este método reemplaza al MeanShift, el cual trataba de predecir su nueva posición mediante el grado de pertenencia de un conjunto de datos adquiridos por una región de interés (ROI). El CamShift se basa en el método antes mencionado y lo complementa la nueva posición del objeto haciendo un análisis y comparación entre sus histogramas de las posiciones anteriores. Se puede observar el funcionamiento de este método en la Figura 18-1.

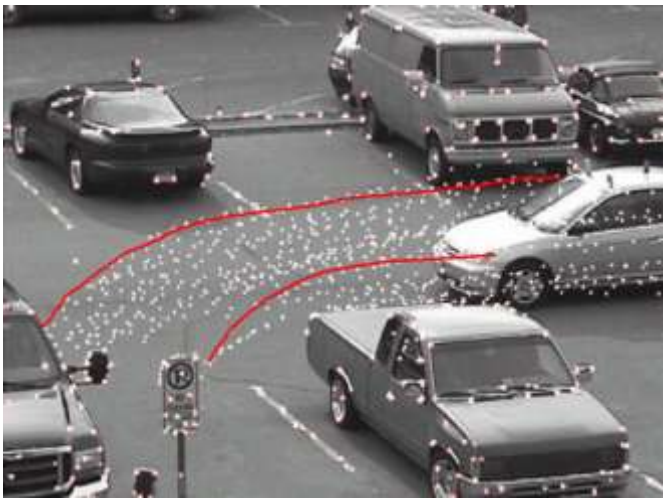


Figura 18-1: Tracking de objeto
Fuente: http://www.dfists.ua.es/controlvisual/ponencias/SVCPCMumh.pdf

1.9.4 Clasificar Cascada (Haar Cascade)

Para la detección de objetos se puede implementar el algoritmo de clasificador en cascada que se basan en características de Haar, este método es efectivo y fue propuesto por Paul Viola y Michael Jones en su artículo *"Detección rápida de objetos usando una cascada aumentada de características simples"* en 2001. La función de cascada se entrena a partir de múltiples imágenes positivas y negativas, una vez culminado el entrenamiento se lo utiliza para detectar los objetos en otras imágenes. (Berger, W. 2018)

Este algoritmo básicamente se basa en 4 etapas.

- **Selección de funciones Haar:** se recoge todas las características de Haar, se considera características de regiones rectangulares que estén adyacentes a la ubicación especifica de la ventana de detección, aquí se calcula la diferencia de las cantidades de intensidades de los pixeles en cada región, como muestra el ejemplo en la Figura 19-1. (Berger, W. 2018)

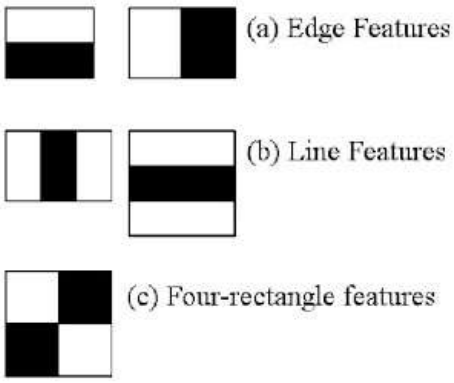


Figura 19-1: Características Haar
Fuente: https://goo.gl/Jg1E1q

- **Creando imágenes integrales:** las imágenes integrales se las utiliza para que esto se desarrolle mucho más rápido.
- **Adaboost Training:** aquí se selecciona las mejores características y hace el entrenamiento a los demás clasificadores que lo usan. Finalmente, este algoritmo construye un clasificador fuerte mediante la combinación lineal de clasificadores simples (débiles) ponderados.
- **Clasificador Cascada:** Esta etapa consiste en la colección de etapas, donde cada etapa es un conjunto de aprendices débiles (simples clasificadores denominados *tocones de decisión*). (Berger, W. 2018)

Para cada una de las etapas del clasificador, se crean etiquetas de la región definida por la ubicación actual en la ventana deslizante los que pueden ser positivos o negativos. La etiqueta positiva señala que se encontró el objeto y la etiqueta negativa indica que no se encontraron objetos dentro de la imagen de interés. Estas etapas se han diseñado con el fin de rechazar las muestras negativas lo más rápido posible, debido a que en la mayoría de las ventanas no se encuentra el objeto de interés. Caso contrario ocurre cuando se detecta los verdaderos positivos porque son escasos y es necesario comprobar su veracidad. (Berger, W. 2018)

- Un verdadero positivo es cuando se clasifica una muestra positiva correctamente.
- Un falso positivo se da cuando una muestra negativa se clasifica por error como una muestra positiva.
- Un falso negativo sucede cuando una muestra positiva se lo clasifica erróneamente como una muestra negativa.

Para realizar el entrenamiento de los clasificadores en cascada se necesita de un conjunto de muestras positivas y un conjunto de muestras negativas, las imágenes positivas se deben proporcionar con regiones de interés específicas. Para conseguir una precisión aceptable del detector se debe de configurar el número de etapas, el tipo de función y otros parámetros. A continuación, se muestra en la Figura 20-1, el esquema básico de un clasificador en cascada. (Berger, W. 2018)

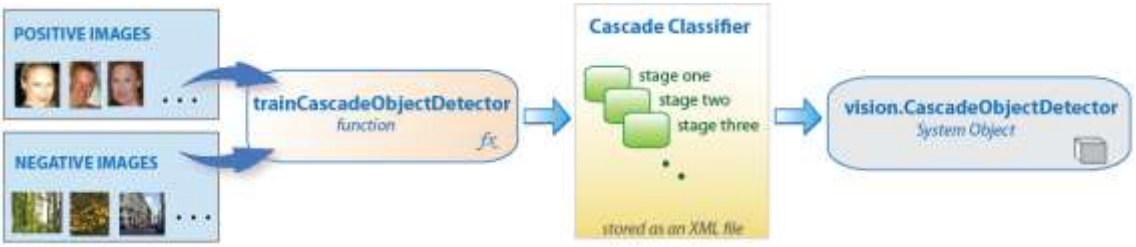


Figura 20-1: Clasificador Cascada
Fuente: http://www.mathworks.com/help/vision/ug/traincascadeobjectdetector_overview.png

1.10 Lógica Difusa

1.10.1 Introducción

El ser humano tiene dificultades para tomar una decisión cuando la información no es precisa, para ello se ha creado la lógica difusa con el objetivo de emular la lógica humana y así poder tomar una decisión correcta a pesar de la información, en otras palabras, se la puede definir como un conjunto de principios matemáticos basándose en jerarquía de membresía o pertenencia con el

fin de modelar dicha información. Se diferencia de la lógica convencional porque no toma dos únicos valores como son el cero y el uno, sino que crea rangos dentro del intervalo de 0 y 1. (Ponce, P. 2010, p. 33)

Una de las ventajas que presenta la lógica difusa, es que no se necesita conocer con exactitud el modela matemático del sistema real, sino que se comporta como una caja negra en la cual se le añaden las entradas y al final se obtendrá la salida deseada de acuerdo al sistema elaborado. (Ponce, P. 2010, p. 34)

Una variable lingüística es una variable que adopta valores con palabras, que permiten determinar el estado de un objeto o fenómeno. Todos los valores lingüísticos forman un conjunto de términos o etiquetas. (Ponce, P. 2010, p. 34)

Para el desarrollo de este control con lógica difusa, se debe de tener las entradas del sistema las cuales se van a mapear a las variables lingüísticas, que se le denomina *difusificación.* Adicionalmente con estas variables se crearán reglas, las cuales serán las encargadas de regir las acciones de control de la salida del sistema. (Ponce, P. 2010, p. 34)

En la Figura 21-1 se muestra las partes básicas que conforma un controlador difuso.

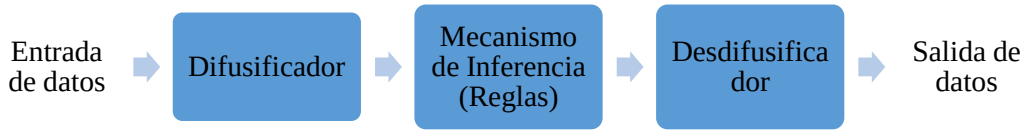


Figura 21-1: Diagrama de bloques de un controlador difuso
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

- **Reglas:** estas reglas son las que dictan la acción de control que se va a tomar, su estructura se basa en relaciones, las cuales se determinan mediante el cálculo de relaciones "SI-ENTONCES".
- **Difusificador:** es la conexión entre las entradas reales y difusas.
- **Desdifusificador:** simplemente toma un valor difuso de las reglas y la convierte a una salida real. (Ponce, P. 2010, p. 34)

1.10.2 Aplicaciones

La lógica difusa es aplicada en áreas multidisciplinarias, que surgen de la evolución tecnológica de los electrodomésticos hasta los programas computacionales. Gracias a este método de lógica

difusa se ha logrado resolver problemas que han sido intratables usando la lógica clásica. (Castillo, O. 2008)

Se puede dividir la aplicación de la lógica difusa en tres grupos

- Productos enfocados al consumidor: lavadoras difusas, hornos microondas, sistemas térmicos, traductores lingüísticos, cámaras de video, televisores, estabilizadores de imágenes digitales y sistema de foco automático en cámaras fotográficas.
- Sistemas: elevadores, trenes, automóviles, controles de tráfico, sistemas de control de aires acondicionados (evitando las oscilaciones de temperatura), y sistema de reconocimiento de escritura.
- Software: Diagnóstico médico, seguridad, comprensión de datos, tecnología informática, y bases de datos difusas. (Castillo, O. 2008)

1.10.3 Funciones de Membresía

Para la identificación de los grados de pertenencia de los elementos del conjunto difuso, es necesario extraer los datos de los fenómenos que se va a representar y de ahí definir su función de membresía. Existen funciones de membresía convencionales y no convencionales, entre las convencionales se tiene los siguientes. (Ponce, P. 2010, pp. 50–51)

1.10.3.1 Función de Saturación

Es la más sencilla de todas, en esta se tiene un valor de 0 hasta cierto punto y luego crece con pendiente constante hasta que alcanza el valor de 1, donde se mantiene hasta el final, como se muestra en la Figura 22-1. (Ponce, P. 2010, p. 50)

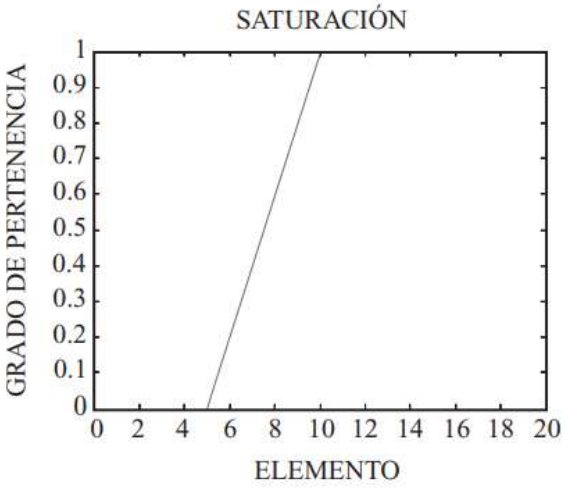


Figura 22-1: Función de Saturación
Fuente: https://goo.gl/kNmyrm

1.10.3.2 Función hombro

Se la considera como la contraparte de la función de saturación, debido a que inicia con un valor unitario hasta un cierto punto y luego desciende constantemente hasta alcanzar el valor 0, mostrado en la Figura 23-1. Esta función es útil cuando el grado de pertenencia es total en valores muy pequeños y va decayendo conforme el valor de la variable aumente. Un claro ejemplo es aplicado en una pecera, ya que la cantidad de oxigeno es aceptable cuando el número de peces no exceda el límite, una vez que los peces aumenten, el oxígeno irá decayendo hasta el punto de que no sea suficiente para los peces. (Ponce, P. 2010, p. 50)

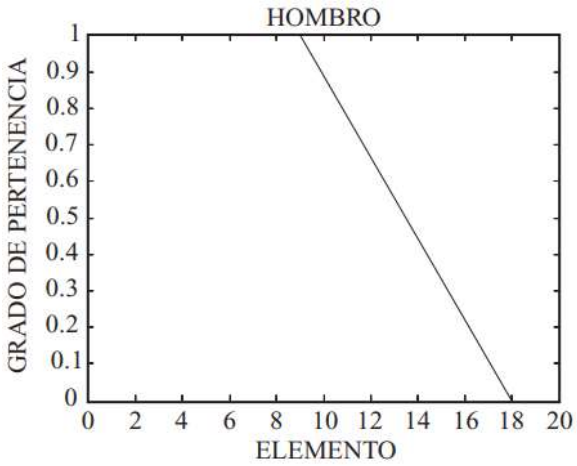


Figura 23-1: Función hombro
Fuente: https://goo.gl/kNmyrm

1.10.3.3 Función Triangular

Como su nombre lo indica, su forma es triangular el cual consta de una parte con pendiente positiva hasta alcanzar la unidad y una vez alcanzado este punto desciende de manera uniforme hasta alcanzar el 0, como se muestra a continuación en la Figura 24-1. (Ponce, P. 2010, p. 51)

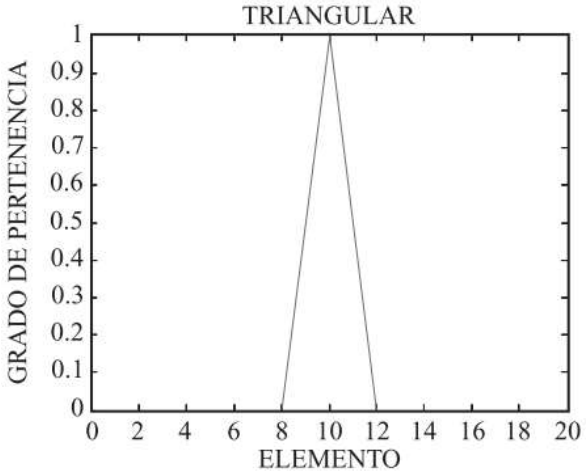


Figura 24-1: Función Triangular
Fuente: https://goo.gl/kNmyrm

Esta función es de utilidad cuando se tiene un valor óptimo central, que se va perdiendo a medida que se aleja de dicho valor.

1.10.3.4 Función Trapecio o Pi

Es una forma generalizada de la función triangular, esto se debe a que no sólo tiene un punto en la cual la pertenencia sea la unidad, sino toda una franja que varía su ancho dependiendo del fenómeno observado, en la Figura 25-1 se muestra dicha función de membresía. La cual es muy utilizada gracias a su desempeño cuando existen rangos de valores óptimos a pesar de que las condiciones no son las adecuadas. (Ponce, P 2010, p. 52)

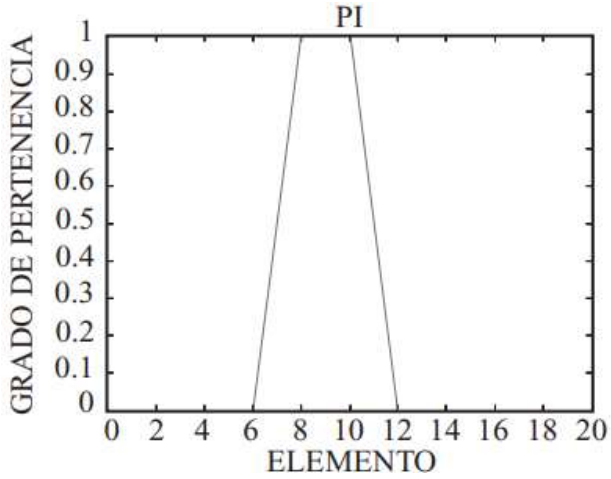


Figura 25-1: Función Trapecio
Fuente: https://goo.gl/kNmyrm

1.10.3.5 Función "S" o Sigmodal

Su forma es similar a la de saturación, con la diferencia es que el segmento de subida no es una línea recta, sino una curva de segundo orden, esta tiende a cambiar de concavidad en un punto dado hasta que llega al valor de 1 y se mantiene allí, como se muestra en la Figura 26-1. (Ponce, P. 2010, p. 52)

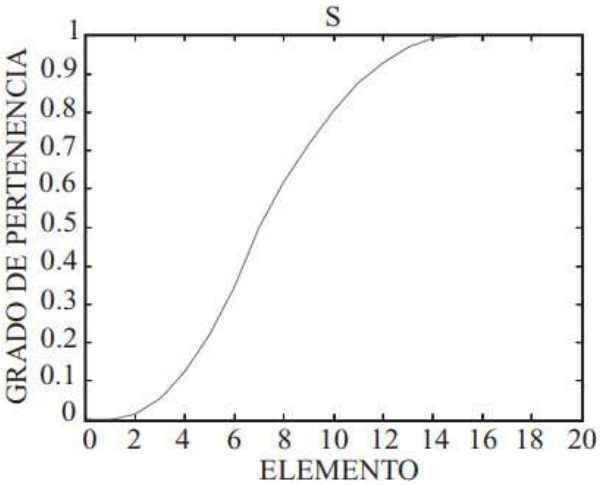


Figura 26-1: Función Sigmodal
Fuente: https://goo.gl/kNmyrm

1.10.4 Métodos de Desdifusificación

El conjunto de salida difusa (resultado del bloque de inferencia) entra al bloque desdifusificador, con el fin de obtener una salida de valor concreto, para hallar ese valor específico se aplican métodos matemáticos, entre estos métodos los más utilizados son:

- **Método del máximo:** el valor de salida se lo escoge el máximo de la función de pertenencia del conjunto difuso.
- **Método del centroide:** toma como valor de salida, al valor centro de gravedad de la función de pertenencia del conjunto difuso.
- **Método de la altura:** en este método se calcula el centro de gravedad para cada regla, con el fin de obtener la media ponderada y asignarle como el valor de salida. (Hernández, J. 2010, p. 52)

1.11 Raspberry Pi

La Raspberry pertenece a la serie de pequeñas computadoras o micro ordenadores que han sido desarrolladas en el Reino Unido y presentado en una pequeña placa, donde abarcan todos los componentes que hace de su funcionamiento similar al de una computadora convencional, presenta la desventaja en cuánto a la limitación de velocidad en el procesador, limitando de esta manera el número de tareas o procesos que pueda desempeñar al mismo tiempo.

1.11.1 Tabla comparativa Raspberry Pi

Al paso de los años, estas pequeñas computadoras han ido mejorando tecnológicamente con el fin de cumplir con las necesidades que el usuario necesita para el desarrollo de proyectos o aplicaciones en un tamaño más pequeño que lo usual.

Las características entre los diferentes modelos de Raspberry se detallan a continuación en la Tabla 1-1.

Tabla 1-1: Tabla comparativa entre los modelos Raspberry

Raspberry Pi	**Modelo A**	**Modelo A+**	**Modelo B**	**Modelo B+**	**RPi 2 modelo B**	**RPi 3 Modelo B**
SoC	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2836	Broadcom BCM 2837
CPU	700MHz ARM1176JFZ-S	700MHz ARM1176JFZ-S	700MHz ARM1176JFZ-S	700MHz ARM1176JFZ-S	900MHz Quad-Core ARM Cortex-A7	1.2GHz Quad Cortex A53
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV	250MHz VideoCore IV	400MHz VideoCore IV
RAM	256MB	512MB	512MB	512MB	1GB	1GB
USB	1	1	2	4	4	4
Video	RCA, HDMI	Jack, HDMI	RCA, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI
Audio	Jack, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI
Boot	Memoria SD	Micro SD	Memoria SD	Micro SD	Micro SD	Micro SD
Wireless	No	No	No	No	No	802.11n / Bluetooth 4.1
Red Ethernet	No	No	Ethernet 10/100	Ethernet 10/100	Ethernet 10/100	Ethernet 10/100
Alimentación	5V / 2Amp	5V / 2Amp	5V / 2Amp	5V / 2Amp	5V / 2Amp	5V / 2.5Amp
GPIO	26 pines	40 pines	26 pines	40 pines	40 pines	40 pines
Tamaño (mm)	85.6x53.98	65x56	85.6x53.98	85x56x17	85x56x17	85x56x17

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

1.11.2 Raspberry Pi 3 Modelo B

Para el año 2016 con la llegada de la Raspberry Pi 3, mostrado en a Figura 27-1, se ha solucionado la limitación que presentaba estos micro-ordenadores con lo que refiere a la conectividad inalámbrica (incluyendo WIFI y Bluetooth sin necesidad de adaptadores), además presenta 1 GB de RAM, con un procesador BCM2837 ARMv8, cuenta con 4 núcleos a 1.2 GHz, esta versión es la primera en usar instrucciones de 64 bits. (Alonso, R et 2018)

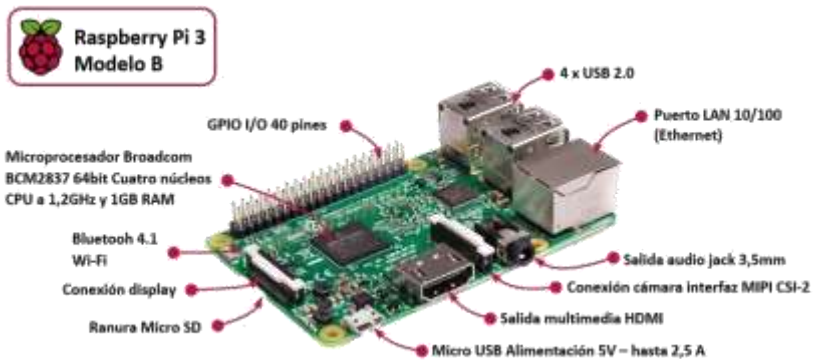


Figura 27-1: Raspberry Pi 3 Modelo B
Fuente: https://goo.gl/Q8N74C

1.12 Sistemas Operativos – Raspberry

Existen algunos Sistemas Operativos que pueden ser instalados en una Raspberry, pero el sistema oficial de Raspberry y que es soportado por cualquier modelo es el Raspbian. Entre los sistemas, se encuentra los siguientes:

- Raspbian
- Ubuntu Mate
- Snappy Ubuntu Core
- Windows 10 IoT Core
- OSMC
- LibreELEC
- PiNet
- RISC OS
- IchigoJam RPi

1.13 Arduino

Arduino es una plataforma de código abierto, es decir, está basado en hardware y software abierto, con la finalidad de que los usuarios y desarrolladores creen diferentes tipos de microordenadores de una sola placa. Este hardware permite diseñar proyectos autónomos que se conectan e interactúan entre sí, tanto el hardware como el software. (Yubal, F. 2018)

Se define un microcontrolador como un circuito integrado programable con la capacidad de ejecutar ordenes que han sido grabadas en su memoria.

Existen diferentes tipos de Arduino, entre los que se destacan los siguientes:

- Arduino UNO, como se muestra en la Figura 28-1.
- Arduino Leonardo
- Arduino MEGA
- Arduino Yún
- Arduino DUE
- Arduino Mini
- Arduino Micro
- Arduino Zero (García, J. 2018)

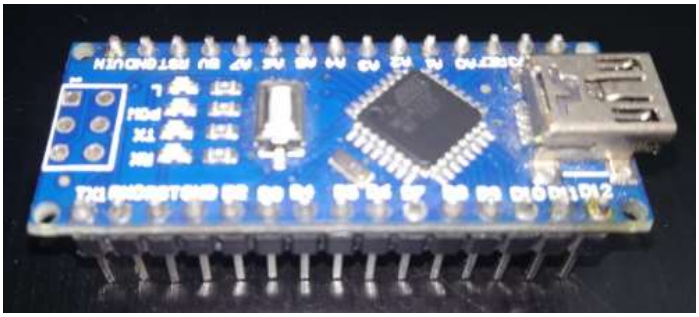


Figura 28-1: Placa Arduino Nano
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

CAPÍTULO II

2. MARCO METODOLÓGICO

Para el desarrollo e implementación del sistema prototipo de luces frontales inteligentes, se describe el software y hardware utilizado para el funcionamiento del mismo, la metodología utilizada se detalla en la Figura 1-2.

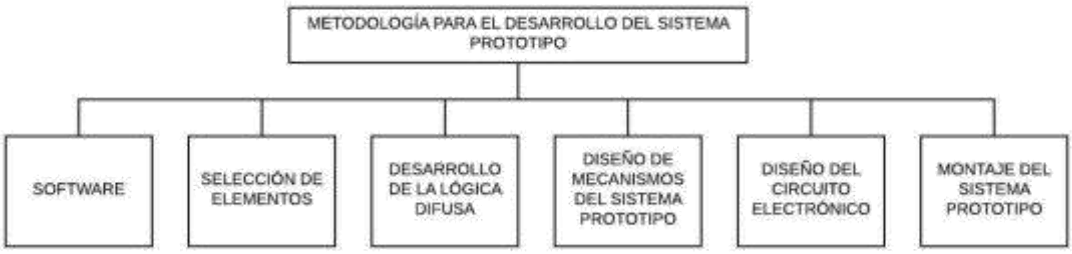


Figura 1-2: Esquema de desarrollo del sistema prototipo
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1 Desarrollo del software

2.1.1 Selección de algoritmo

Existen varios tipos de algoritmos que permiten la detección y seguimiento de objetos, pero no todos son factibles para el desarrollo del sistema, como se muestra en la Tabla 1-2. Por ejemplo, el algoritmo Canny detecta solo los bordes del objeto que pretende analizar, por lo cual no es factible para el sistema porque se necesita detectar específicamente la luz que emite los faros de los vehículos los mismos que no poseen un borde fijo.

El algoritmo SIFT extrae características específicas de imágenes que están en escala de grises las mismas se encuentran registradas en una base de datos, este algoritmo no es factible para aplicarlo en el desarrollo del sistema porque sufre alteraciones al exponer los objetos a cambios de ruido e iluminación.

Algoritmo Camshift predice la nueva posición del objeto en base a sus posiciones anteriores, no es factible desarrollar el sistema con este algoritmo porque la posición de los vehículos no siempre es la misma, es decir, están en constante cambio debido a los factores externos que se presenta en las carreteras.

Se utiliza el algoritmo Haar Cascade porque facilita la creación de un clasificador en cascada, en base al número de muestras que se recopilen del entorno o de los objetos de interés, las muestras se procesan mediante el meta-algoritmo Adaboost que crea varios clasificadores de menor precisión para al final combinar todas las iteraciones y crear solo un clasificador de mayor

precisión, genera un archivo .XML el cual puede ser procesado de forma ligera por la Raspberry Pi 3. La capacidad de generalizar ante la presencia de nuevos ejemplos a clasificar es el principal objetivo de un clasificador.

Tabla 1-2: Características de los algoritmos

Algoritmo / **Características de interés**	**Canny**	**SIFT**	**CamShift**	**Haar Cascade**
Reconocimiento de objetos	Convierte la imagen a la escala a grises y umbraliza los bordes de mayor relevancia. (Poco Satisfactorio)	Elige puntos clave invariantes en el tiempo de una base de datos. (Bueno)	Depende de histogramas ponderados anteriores en una ROI (Malo)	Mediante base de datos almacenados en un archivo .XML (Bueno)
Seguimiento de objetos	Los bordes se pueden seguir, pero con dificultad de mantenerse ante cambios de escenarios o iluminación. (Poco Satisfactorio)	La respuesta de seguimiento es fluida, pero necesita imágenes de referencia durante todo el proceso. (Bueno)	Por predicción de la nueva posición, dedicado solo para encontrar cambios de un escenario. (Muy Malo)	Gracias a la efectividad de combinación de Adaboost con filtros Haar, el seguimiento de objetos es excelente ya que no los pierde una vez encontrados, le afecta muy poco el cambio de escenario o iluminación. (Bueno)

Capacidad de generalizar el aprendizaje supervisado	No permite (Muy malo)	No permite (Muy Malo)	No permite (Muy Malo)	El archivo XML contiene muchas características del objeto de interés que se desea encontrar. (Muy Bueno)
Etiqueta el objeto	No permite (Muy Malo)	Por medio de comparaciones del número de puntos clave encontrados. (Poco Satisfactorio)	No permite (Muy Malo)	El clasificador permite realizar un fácil etiquetado, depende del entrenamiento realizado. (Bueno)

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Con estos antecedentes se compara los algoritmos candidatos para este proyecto, se hace un análisis cualitativo que cumpla las necesidades requeridas, de acuerdo a la escala de Likert se pondera las cualidades descritas de cada uno definiendo un rango de 1 a 5 con las siguientes etiquetas:

1 = Muy malo
2 = Malo
3 = Poco Satisfactorio
4 = Bueno
5 = Muy bueno

Tabla 2-2: Escala de Likert para la selección del algoritmo

Algoritmo / Pregunta	Canny	SIFT	CamShift	Haar Cascade
¿Qué capacidad tiene para reconocer objetos?	3	4	2	4
¿Qué capacidad tiene para dar seguimiento a un objeto?	3	4	1	4
¿Qué capacidad tiene para generalizar el aprendizaje supervisado?	1	1	1	5
¿Qué capacidad tiene para etiquetar un objeto?	1	3	1	4
Promedio	**2**	**3**	**1.25**	**4.25**

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De acuerdo a la ponderación tabulada en la Tabla 2-2 respecto al análisis presentado en la Tabla 1-2 se escoge el algoritmo de Haar Cascade porque presenta un promedio de 4.25 equivalente a la etiqueta de "Buena", siendo el más conveniente para el desarrollo del prototipo.

2.1.2 Python - OpenCV

Figura 2-2: Logo Oficial - Python OpenCV
Fuente: https://goo.gl/ZQqvBS

Es un lenguaje de programación (Figura 2-2). A diferencia con otros lenguajes como C / C++, Python presenta lentitud ante estos, pero con la facilidad de extenderlo con C / C++. Presenta dos ventajas en el código, en primer lugar, es rápido como el original de C / C++, en segundo lugar, es más fácil codificar en Python. La librería OpenCV facilita el desarrollo del sistema ya que ayuda al funcionamiento de reconocimiento de objetos mediante visión artificial.

2.1.3 Cascade Trainer GUI

Facilita la creación de clasificadores en cascada previamente entrenados, probados o mejorados dispone de una interfaz gráfica (Figura 3-2), que permite establecer diferentes parámetros como número de imágenes positivas o negativas. numero de iteraciones, recurso de memoria RAM para el entrenamiento, tamaño de las muestras entre otros.

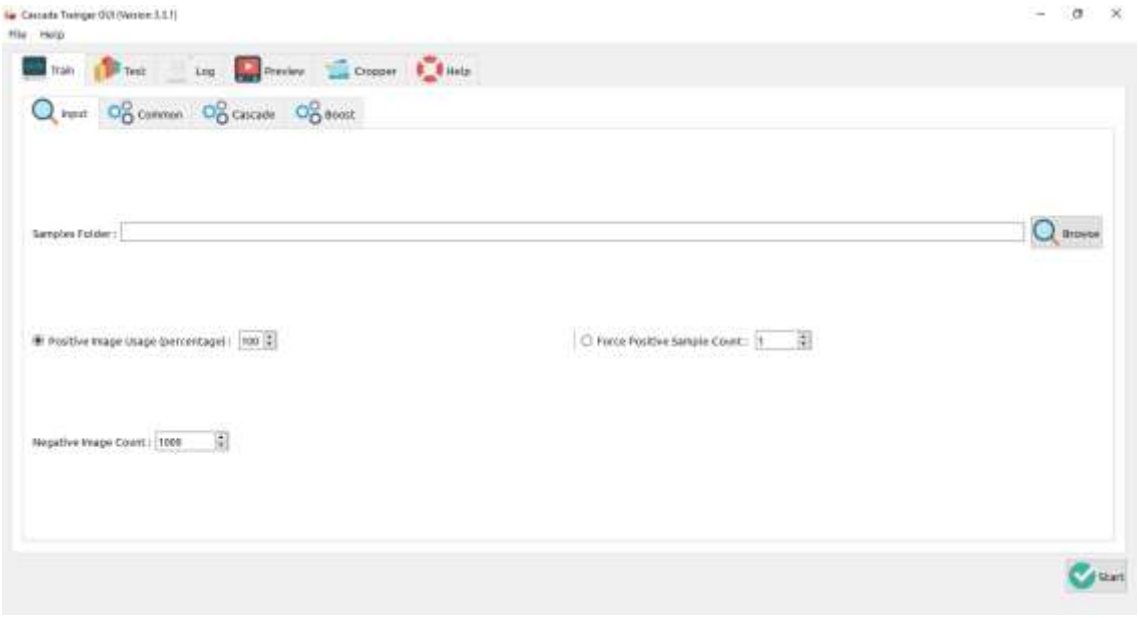


Figura 3-2: Pantalla de inicio Cascade Trainer GUI
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Antes de iniciar con el entrenamiento del clasificador en cascada, es indispensable disponer de numerosas muestras de imágenes positivas e imágenes negativas, dichas muestras deben estar contenidas dentro de una carpeta y clasificadas en subcarpetas que contengan las imágenes positivas (carpeta "p") y las imágenes negativas (carpeta "n").

Las muestras de imágenes positivas son aquellas que contienen el objeto de interés, caso contrario, las muestras de imágenes negativas son las que no contienen el objeto a entrenar parcial o totalmente.

Por consiguiente, se debe seleccionar la carpeta creada (Figura 4-2), que contiene las subcarpetas "n" y "p". Se deja el parámetro de análisis del 99% de muestras positivas, en el ítem de muestras negativas se establece el número de imágenes que contiene dicha carpeta.

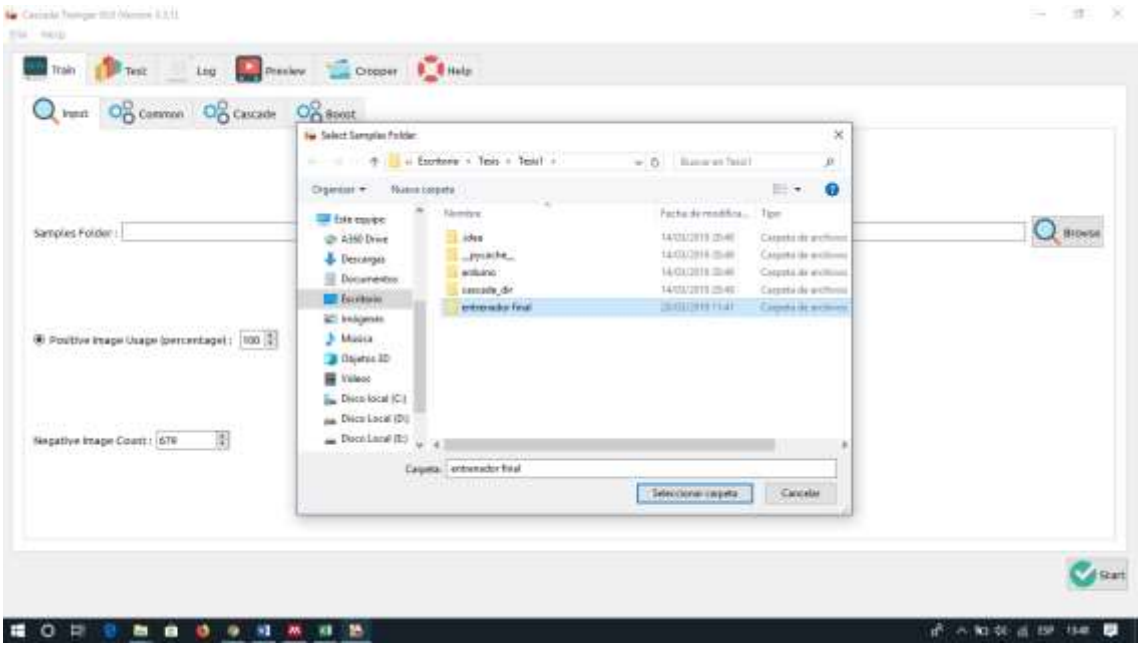


Figura 4-2: Selección de la carpeta a entrenar
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Las pestañas Common, Cascade y Boost, proporcionan parámetros necesarios para personalizar el entrenamiento, por defecto se mantiene los parámetros óptimos y recomendados por el software, como muestra en la Figura 5-2 y 6-2. Para cambiar los parámetros siguientes es necesario tener un amplio conocimiento en las técnicas de clasificación en cascada, para obtener mejores resultados en lo que refiere al clasificador final.

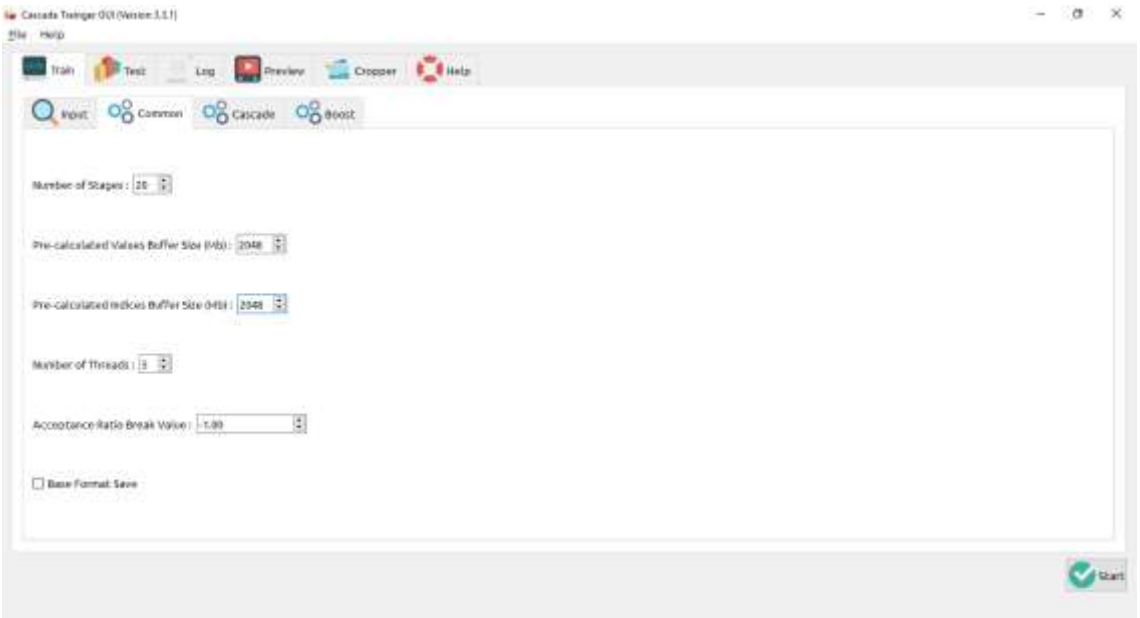


Figura 5-2: Parámetros configurados - Pestaña Common
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

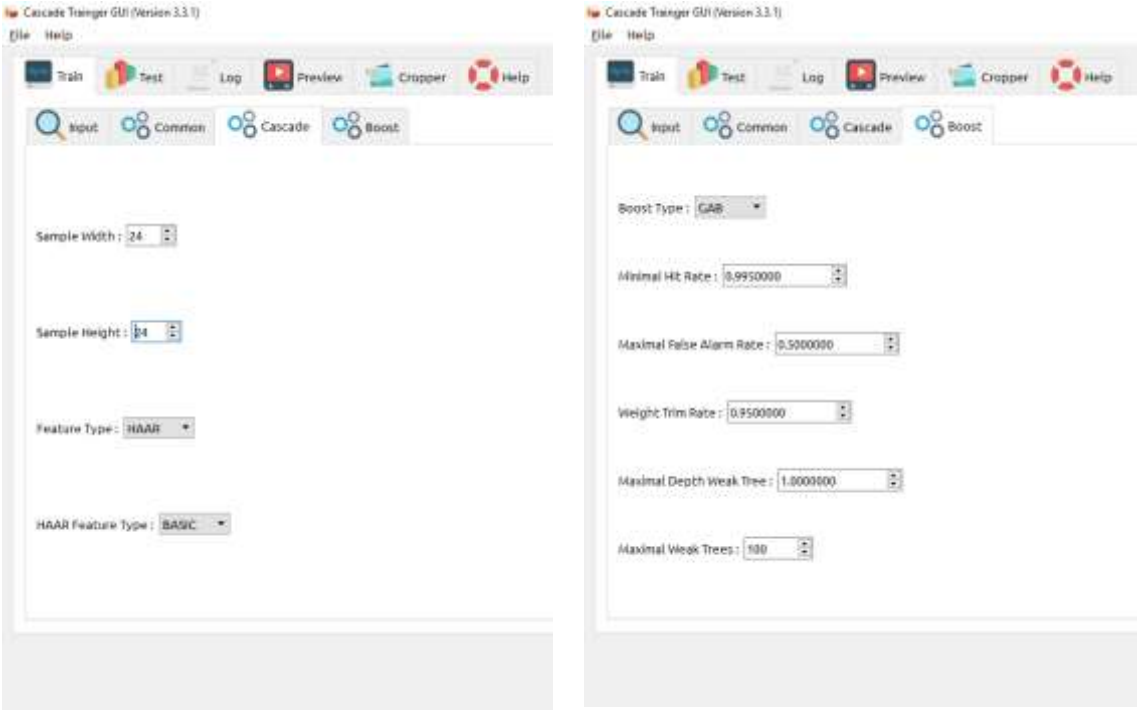


Figura 6-2: Parámetros por defecto - Pestaña Cascade y Boost
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.4 SolidWorks

Se realiza diseños CAD 3D, se desarrolla planos en 2D, se modela piezas y ensamblajes en 3D. Facilita la creación, diseño, simulación, fabricación, publicación y gestión de datos del diseño, su interfaz principal se muestra en la Figura 7-2, con ayuda de esta plataforma se diseña la caja que abarque todos los componentes necesarios para el funcionamiento del sistema prototipo.

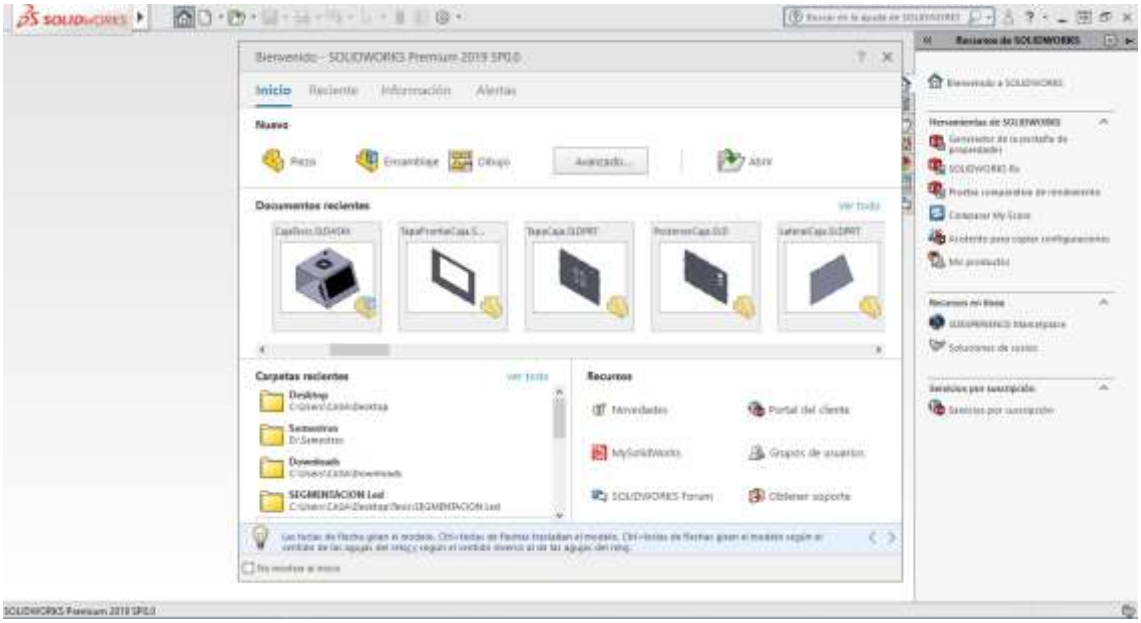


Figura 7-2: Interfaz principal SolidWorks 2019
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.5 Eagle (Easily Applicable Graphical Layout Editor)

Se utiliza para el diseño de diagramas y PCBs que son necesarios para el sistema, contiene un editor de diagramas electrónicos que los elementos pueden ser colocados con un solo click y rutear con otros componentes mediante cables o etiquetas, la Figura 8-2 muestra el entorno de trabajo de este software.

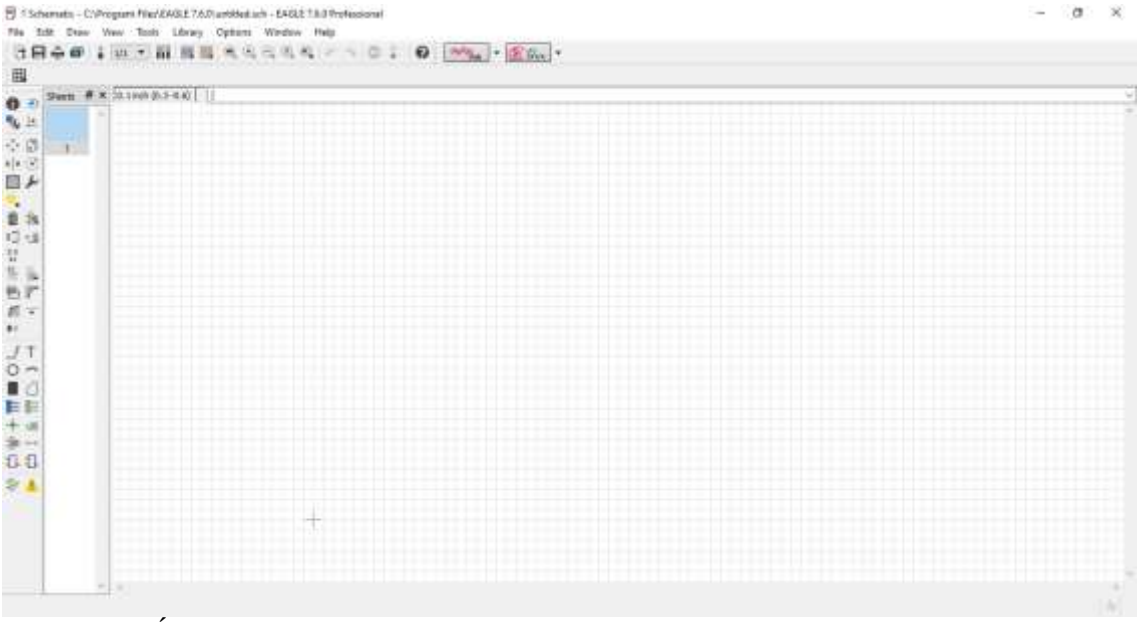


Figura 8-2: Área de trabajo EAGLE
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.6 Arduino IDE

Se cataloga como un entorno de desarrollo en el cual se realiza la programación para cada placa de Arduino, para crear un nuevo programa de Arduino se ejecuta la aplicación, posteriormente se muestra el entorno de trabajo (Figura 9-2), donde se escribirá las líneas de código necesarios para la comunicación con la Raspberry, y el accionamiento necesario de los servomotores cada vez que reciba el dato del ángulo que se encuentra en el puerto serial.

Al conectar el Arduino por cable USB hacia la placa Raspberry, este se energizará dando paso a la alimentación de los servos, a la espera de los ángulos de cierre para cada servomotor independientemente.

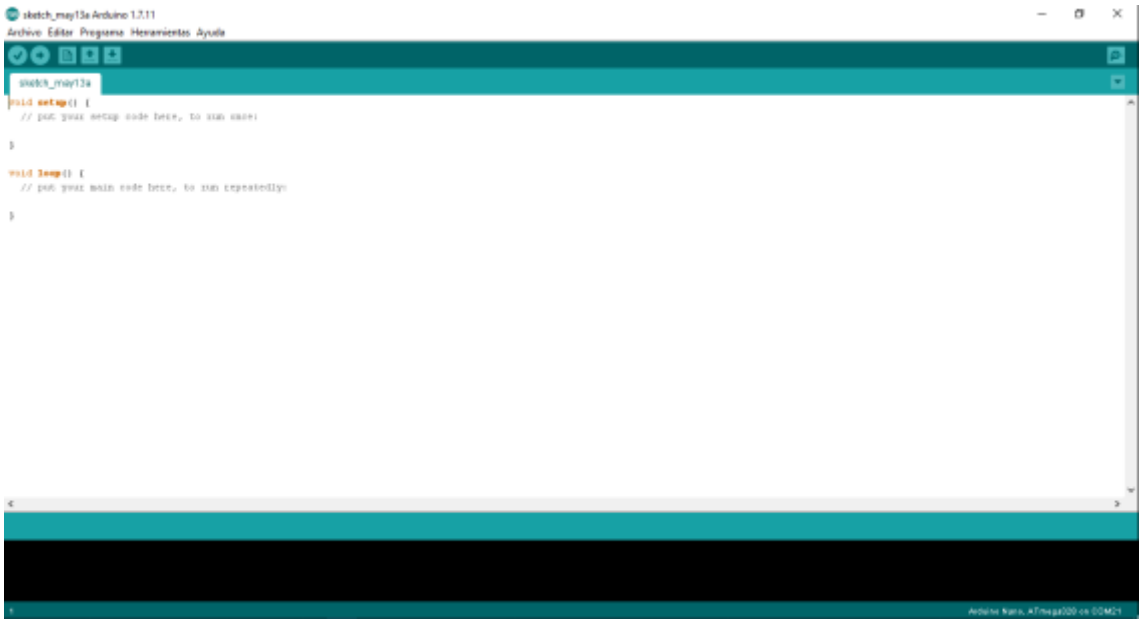


Figura 9-2: Entorno principal Arduino IDE
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

La programación de Arduino (Figura 10-2), se basa en la lectura del puerto serial para obtener el ángulo y la identificación del servo que actuará en ese momento, esta placa Arduino controla cada servomotor que están conectados a los pines PWM que trae diseñado.

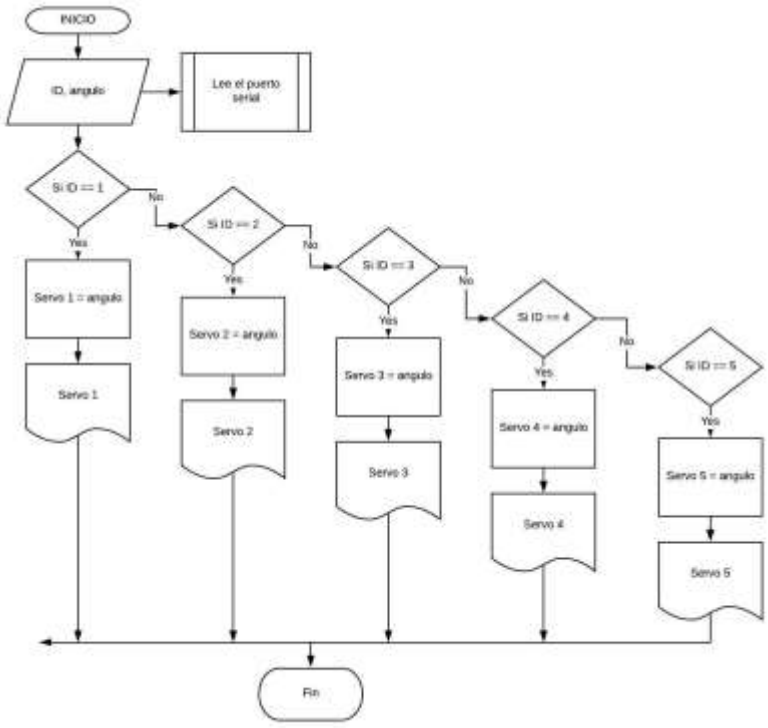


Figura 10-2: Diagrama de Flujo - Arduino Nano
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.7 Sistema Operativo – Raspberry

2.1.7.1 Instalando Raspbian Stretch

Para instalar el sistema operativo en la Raspberry, primero se descarga la imagen ISO de la página oficial mostrado en la Figura 11-2**:** Raspbian Sistema Operativo Oficial, una vez descargado se lo instala en una micro SD, para este proyecto se usó una con capacidad de 32GB, dando mayor espacio de almacenamiento cuando se generen archivos de video captados en las carreteras y que sirvan de muestras para el entrenamiento del clasificador en cascada

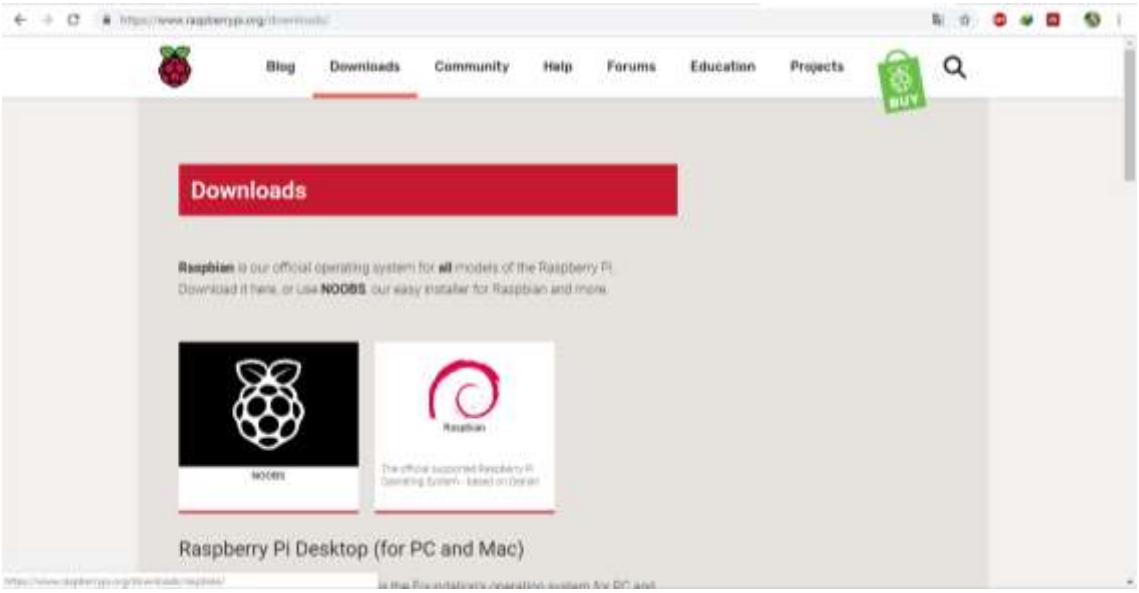


Figura 11-2: Raspbian Sistema Operativo Oficial
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Una vez cargado el sistema a la micro SD, se coloca dicha tarjeta en la ranura que posee la placa Raspberry, luego se alimenta con la fuente de 5V a 3A, para dar paso a que inicie el sistema, como muestra en la Figura 12-2, una vez cargado el sistema se observa la pantalla principal de Raspbian donde se encuentra programas básicos instalados que ayudan al desarrollo de aplicaciones, la programación es de código abierto y accesible al usuario.

Figura 12-2: Iniciando Sistema Operativo Raspbian
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.8 Qt Designer

Es una plataforma IDE programado en C++, JavaScript y QML, desarrollada por TrollTech, diseñado para ser compatible con los sistemas operativos más usados, como lo son: GNU/LINUX, MAC OS X y Windows. En la Figura ***13***-2, se muestra el comando para instalar la librería en Raspbian.

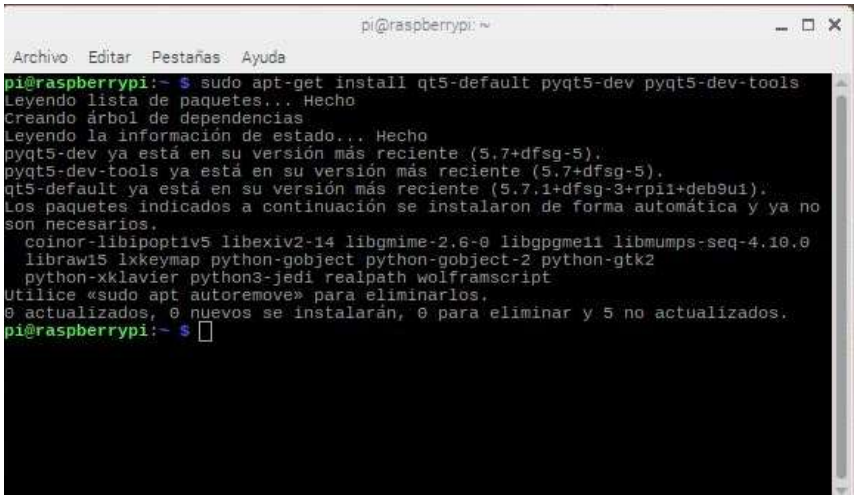


Figura 13-2: Instalando la librería Qt Designer
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

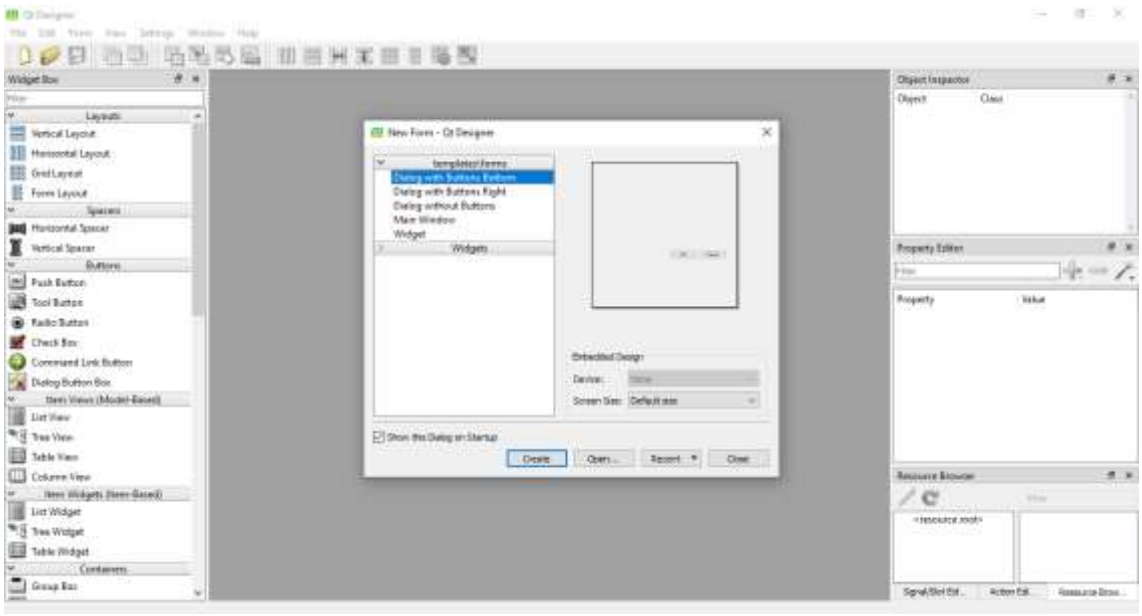


Figura 14-2: Ventana principal Qt Designer
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Figura 14-2 se muestra en entorno de trabajo del software Qt Designer. Para el desarrollo de la Interfaz Gráfica de Usuario (GUI), se usa las siguientes clases que posee.

2.1.8.1 Label

Sirve para crear etiquetas (textos) dentro de la GUI y gracias a la propiedad pixmap (Figura 15-2) permite incluir imágenes en tal campo.

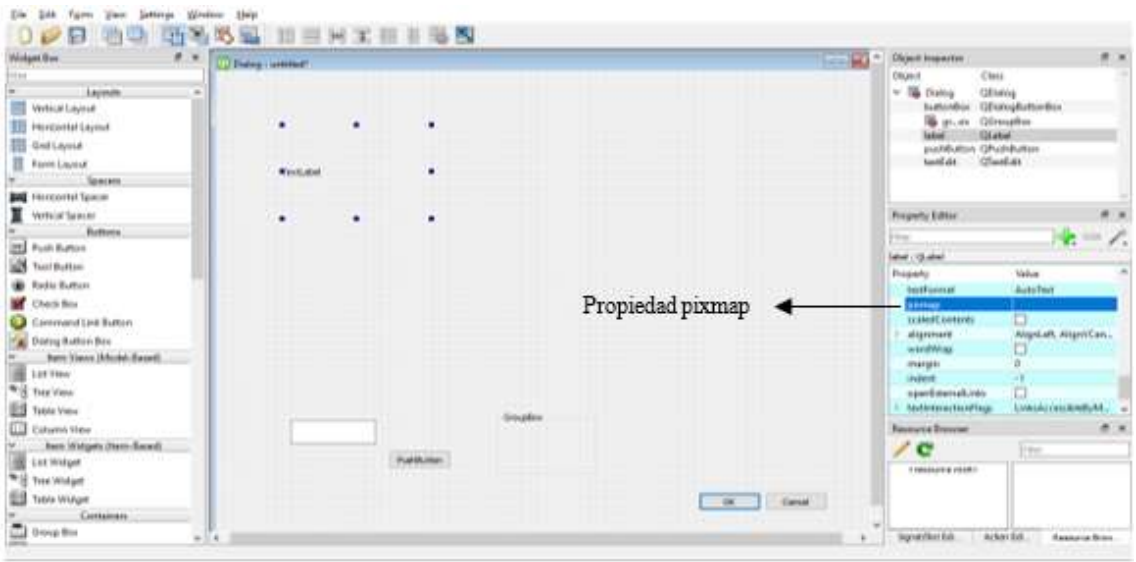


Figura 15-2: Clase Label - Qt Designer
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.8.2 Push Button

Como su nombre lo indica, proporciona un botón de comando, la forma es rectangular y normalmente muestra una etiqueta de texto que representa su acción, como muestra la Figura 16-2. Se usa el botón cuando la aplicación o ventana de dialogo deba realizar una acción al momento que el usuario haga clic en ella.

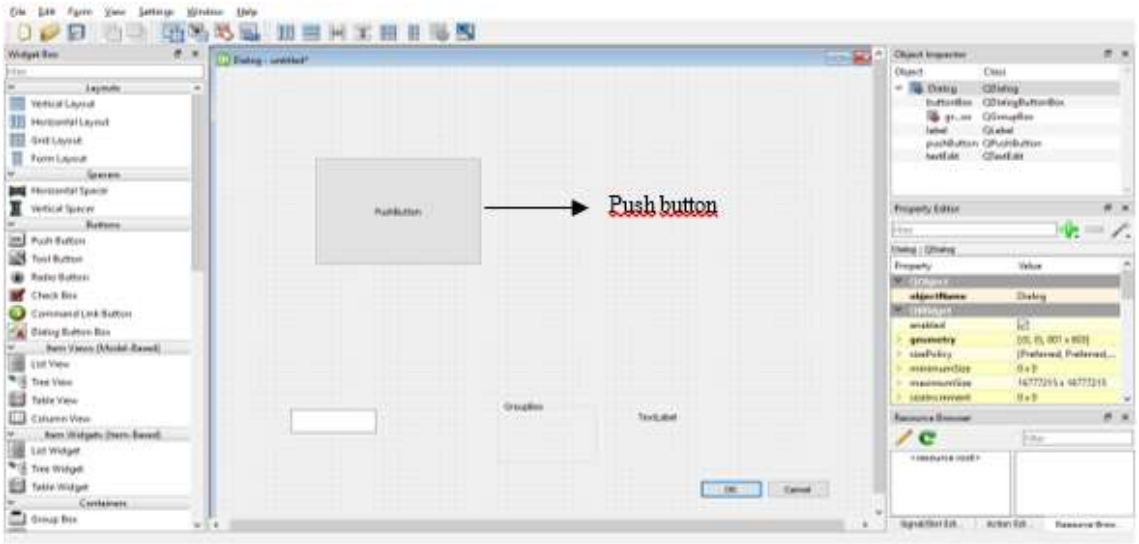


Figura 16-2: Clase Push Button - Qt Designer
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.8.3 Text Edit

Esta clase representa a un visor avanzado que admite el formato de texto enriquecido mediante las etiquetas de estilo HTML (Figura ***17***-2), funciona también en párrafos y caracteres. El soporte de texto enriquecido está diseñado con el fin de proporcionar una manera rápida, portátil y eficiente de agregar facilidades razonables de ayuda en línea a las aplicaciones.

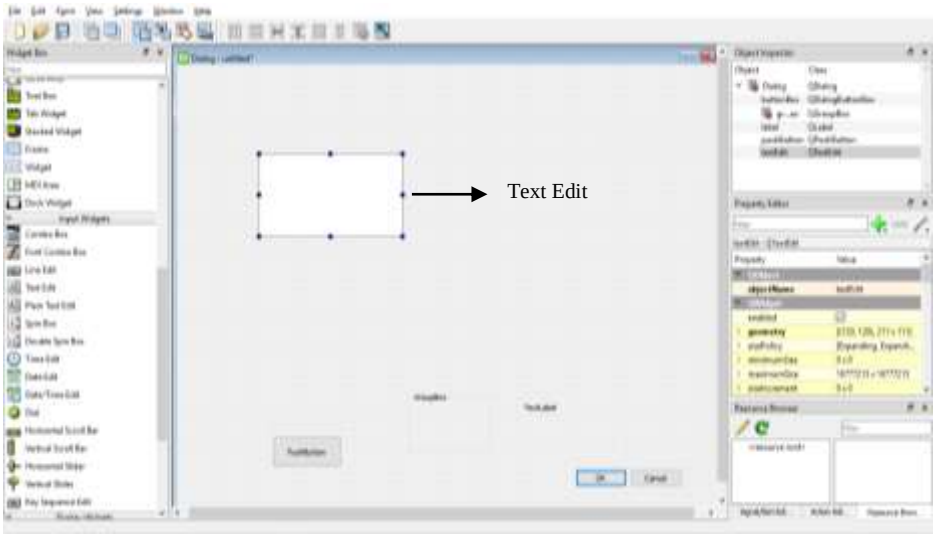


Figura 17-2: Clase Text Edit - Qt Designer
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.1.8.4 Group Box

Esta clase se define como widgets contenedores que organizan los botones en grupos, de manera lógica y de pantalla, tal como se muestra en la Figura 18-2.

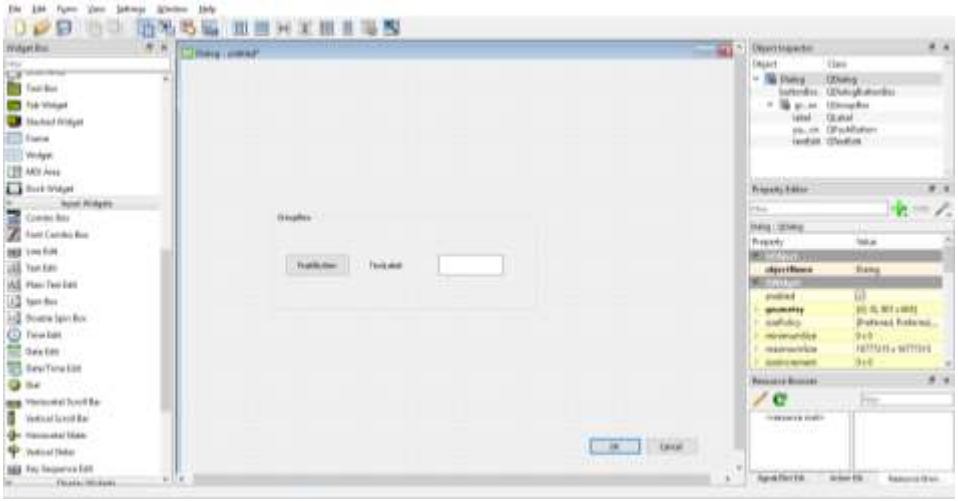


Figura 18-2: Clase Group Box - Qt Designer
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Figura 19-2 se muestra la interfaz gráfica final diseñada en la plataforma de Qt Designer, con botones que cumplen una función específica cada una, que se detallan a continuación.

Figura 19-2: Interfaz Gráfica de Usuario
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

1) **Iniciar:** Inicia la captura del video.
2) **Stop:** Detiene la captura de video.
3) **Detectar:** Inicializa el Clasificador en Cascada.
4) **Detectar Fuzzy:** Muestra las gráficas de la lógica difusa, tanto entradas como salidas.
5) **Puerto COM:** Permite establecer el puerto en el que se encuentra el Arduino.
6) **Conectar:** Establece la comunicación con el Arduino.
7) **Desconectar:** Finaliza la comunicación con el Arduino.
8) Espacio designado para mostrar el video
9) **Group Box (Servos):** Muestra el ángulo de cada servo accionado.

2.1.9 SciKit-Fuzzy

Es una colección de múltiples algoritmos de lógica difusa diseñados para el uso en SciPy Stack, que está escrito en lenguaje de programación Python. En la Figura 20-2 se muestra el comando necesario para instalar la librería de lógica difusa en la Raspbian.

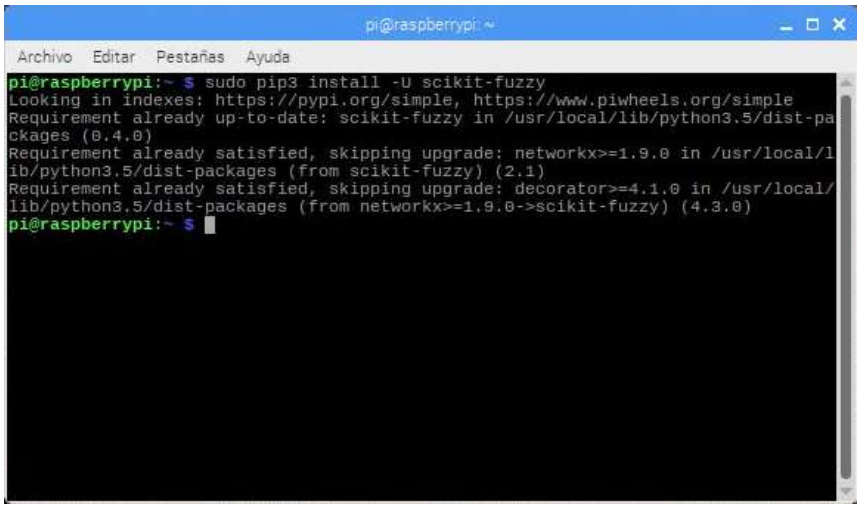


Figura 20-2: Instalando la librería SciKit-Fuzzy
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2 Selección de elementos

2.2.1 Lámpara LED

Para desarrollar el sistema de luces se emplea un faro para modificar el haz de luz, pero las lámparas convencionales presentan dificultad para la innovación, porque consisten de una bombilla incandescente, halógena o de xenón.

Para la aplicación del sistema en carreteras con poca iluminación es favorable incorporar barras LED, como se muestra en la Figura 21-2, para obtener mayor cantidad de visibilidad. Esta barra LED tiene una potencia de consumo de 240W a 12VDC, intensidad lumínica de 1500lm, protección IP68 y una vida útil de 50000 horas.

Figura 21-2: Lámpara de barra LED
Fuente: https://goo.gl/SrQTnN

2.2.2 Servomotor SG90

Figura 22-2: Servomotor SG90
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Debido al sistema de componentes electrónicos y electromagnéticos permite controlar la posición del eje de acuerdo a una cantidad de grados acoplando varios engranajes en una transmisión similar que permite potenciar el torque del motor manteniendo fijo el eje una vez culminado el movimiento.

El servomotor SG90 que se muestra en la Figura ***22***-2, posee características que permiten el desarrollo eficaz del sistema, siendo controlado por el valor del ángulo que se le asigne en todo momento. Las características técnicas se presentan en la Tabla 3-2.

Tabla 3-2: Especificaciones Técnicas del Servo Motor SG90

Micro Servo:	Tower-pro
Velocidad:	0.10 sec/60° @ 4.8V
Torque:	1.8 Kg-cm @ 4.8V
Voltaje de Funcionamiento:	3.0-7.2V
Temperatura de funcionamiento:	-30 ℃ ~ 60 ℃
Ángulo de rotación:	180°
Ancho de pulso:	500-2400 µs
Longitud de cable conector:	24.5 cm

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Posee un conector tipo "S" universal que permite la conexión con entradas tipo JR, Hitec, Futaba, GWS y Cirrus.

En la Figura 23-2 se muestra la distribución de cables para el servomotor mencionado.

- **Rojo:** Alimentación positiva (+)
- **Café:** Alimentación negativa o a tierra (-)
- **Naranja:** Señal PWM

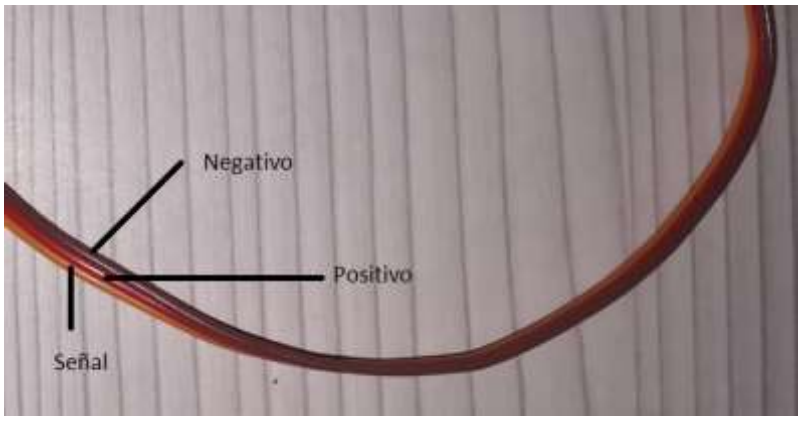


Figura 23-2: Distribución de cables de servomotor
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

La energía que necesita para la alimentación es baja por lo cual es posible conectar directamente a la fuente del circuito de control o Arduino y continuar realizando pruebas.

2.2.3 Arduino Nano

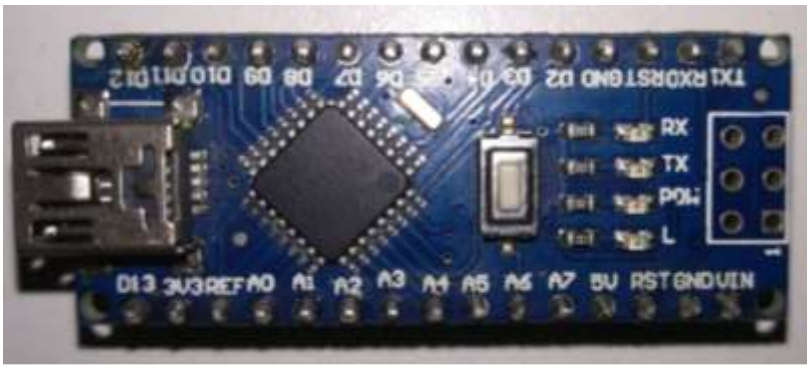


Figura 24-2: Arduino Nano
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

El Arduino Nano (Figura ***24***-2), se puede alimentar a través de la conexión USB Mini-B, 6-20V de fuente de alimentación externa no regulada (pin 30) o 5V de fuente de alimentación externa regulada (pin 27). La fuente de poder se selecciona automáticamente a la fuente de voltaje más alto, las características de este dispositivo se muestran en la Tabla 4-2.

El Arduino cumple la función de comunicar la Raspberry con los servomotores, por medio de comunicación serial, que es conectado con el cable de Arduino al puerto USB de la Raspberry. Para controlar los servomotores, la señal PWM se ha conectado a los pines del Arduino que viene

configurado para usar como tal, es decir, los pines 3, 5, 6, 9, 10, tal y como se muestra en la Figura 25-2.

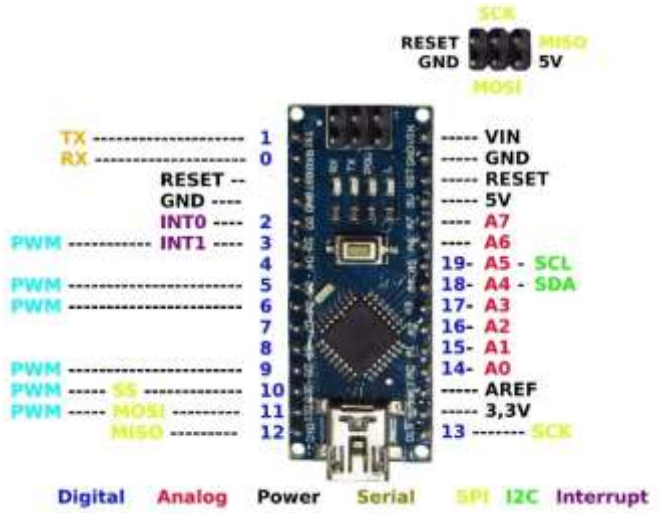


Figura 25-2: Esquema de los pines en el Arduino Nano
Fuente: https://goo.gl/DJsyhp

La apertura de cada servomotor depende de los ángulos que son recibidos de la Raspberry, después de que este haya determinado el valor de salida por medio de la lógica difusa que se empleó durante el desarrollo de este trabajo.

Tabla 4-2: Ficha Técnica - Arduino Nano ATMEGA328

Microcontrolador:	Atmel ATmega328
Tensión de Operación:	5 V
Tensión de Entrada (recomendado):	7 – 12 V
Tensión de Entrada (límites):	6 – 20 V
Pines Entradas/Salidas Digitales:	14 (6 pines proveen salidas PWM)
Entradas Analógicas:	8 Corriente máx por cada PIN de E/S: 40 mA
Memoria Flash:	32 KB de los cuales 2KB son usados por el bootloader
SRAM:	2 KB
EEPROM:	1 KB
Frecuencia de Reloj:	16 MHz
Dimensiones:	18,5mm x 43,2mm

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.4 Raspberry Pi 3 Modelo B

Figura 26-2: Raspberry Pi 3 Modelo B
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Este modelo de Raspberry posee características que sirven de utilidad para desarrollar proyectos de visión artificial entre otros, también se usa de manera personal, reemplazando un ordenador convencional por un microordenador, en este caso la Raspberry mostrado en la Figura ***26***-2.

Las características técnicas que posee este dispositivo se detallan en la Tabla 5-2.

Tabla 5-2: Especificaciones Técnicas Raspberry Pi 3 Modelo B

Procesador CHIPSET	Broadcom BCM2837 64bit
Velocidad Procesador	Quad Core @ 1.2GHz
RAM	1 GB SDRAM @ 400MHz
Almacenamiento	Puerto Micro SD para el S.O. y Datos
USB 2.0	4 USB, 2 Puertos
Consumo de energía / Voltaje	2.5A @ 5V
GPIO	40 pines
Puerto Ethernet	Puerto 10/100 LAN
WIFI	BCM43438 Wireless LAN
Bluetooth	Bluetooth Low Energy (BLE) 4.1
CSI	Puerto Pi Cámara Raspberry
DSI	Puerto Pi Touchscreen Display

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.5 Pantalla Táctil Raspberry

Raspberry dispone de una pantalla táctil oficial, las cuales varían de tamaño siendo de 3.5, 5 y 7 pulgadas. Para el desarrollo de este sistema prototipo se utiliza una pantalla de 7" (Figura 27-2), con la finalidad de visualizar el sistema operativo, desarrollar la interfaz y el funcionamiento final del sistema.

Figura 27-2: Pantalla LCD 7"
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Para el montaje de la pantalla LCD junto a la Raspberry es indispensable de un soporte (case), donde se inserte tanto la placa del microordenador y el display, por medio de un cable plano DSI se puede proporcionar la interfaz de pantalla entre la Raspberry y el conjunto LCD, el montaje se muestra en la Figura 28-2.

Figura 28-2: Montaje Final Raspberry Pi 3B + Pantalla LCD 7"
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.6 Pi Camera

La placa de esta cámara (Figura ***29***-2), se conecta al puerto CSI de la Raspberry Pi, ofrece una resolución de 5MP y captura de video HD de 1080p a 30fps. La cámara Raspberry cuenta con un sensor omnivision 5647 de 5MP (2592 x 1944 píxeles) dentro de un módulo de enfoque fijo.

Figura 29-2: Cámara Raspberry Pi
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

El módulo de la cámara se conecta directamente a la placa Raspberry Pi por medio de un cable de cinta de 15 pines y una longitud de 1.5m (ver Figura ***30***-2), el bus CSI es capaz de velocidades de datos muy altas y su transmisión de datos es de pixeles al procesador BCM2835.

Figura 30-2: Cable plano de 15 pines
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.7 Inversor de voltaje

Figura 31-2: Inversor de Voltaje 500W
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Ante la necesidad de una fuente de 110VAC dentro del auto, se optó por un inversor de voltaje de 12VDC a 110VAC - 500W, mismo que se muestra en la Figura 31-2, con la finalidad de conectar la fuente de la Raspberry (5VDC a 3A) y se encienda junto con la pantalla, posteriormente se realiza las capturas y pruebas necesarias del entorno de la carretera mientras se transita en la noche y con luces bajas, en la Figura ***32***-2 se muestra la manera de adquisición de las imágenes que sirvan de pruebas.

Figura 32-2: Encendido de la Raspberry con el inversor
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.8 Convertidor DC/DC Buck

Figura 33-2: Convertidor DC/DC Buck
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

El convertidor DC/DC (Figura 33-2), es utilizado para la alimentación de los servos, con la finalidad de proveer de buena corriente a los mecanismos y puedan funcionar con mayor eficiencia, las características eléctricas que tiene este convertidor se muestra en la Tabla 6-2.

Tabla 6-2: Especificaciones Técnicas - Convertidor Buck

Voltaje de entrada	12V / 24V (9V-35V)
Voltaje de salida	5V
Corriente de salida	5A
Potencia de salida	25W
Eficiencia de conversión	90%
Tiempo de arranque suave	Alrededor de 500ms
Aumento de temperatura a plena carga	50° Celsius
Rectificación	Síncrona

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.9 Módulo Relé

Figura 34-2: Módulo Relé de 1 canal
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

El módulo tiene 1 relay de alta calidad capaz de soportar cargas máximas de 250V/10A (Figura 34-2). Este módulo relé no posee opto-acoplador como los módulos de 2 más canales, la activación de este relé es mediante un transistor. Su funcionamiento es de la siguiente manera: activa la salida normalmente abierta (NO: Normally Open) cuando recibe un "0" lógico y al recibir un "1" lógico, desactiva la salida, las especificaciones técnicas se detallan a continuación, en la Tabla 7-2.

Tabla 7-2: Especificaciones Técnicas - Módulo Relé

Voltaje de Operación	5 VDC
Señal de Control	TTL (3.3V o 5V)
N° de Relays (Canales)	1 Canal
Capacidad Máx:	10A/250VAC, 10A/30VDC
Corriente Máx:	10A (NO), 5A (NC)
Tiempo de acción:	10ms/5ms
Activación de la Salida NO:	0 Voltios

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.10 Relé Automotriz

El principal uso de este relevador es para la conexión de lámparas halógenas o bocinas (claxon) en los automóviles, con la finalidad de no sobrecargar los interruptores ni los cables, en la Figura 35-2 se muestra el esquema de conexión de una lámpara en un vehículo.

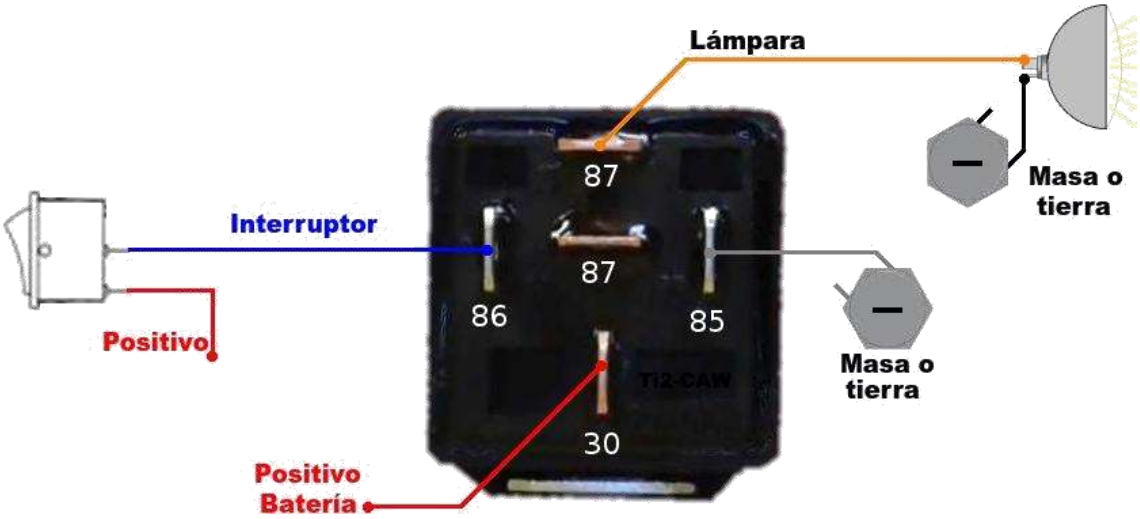


Figura 35-2: Esquema de conexión del relevador
Fuente: https://bit.ly/2Vy4vUe

A continuación, en la Tabla 8-2 se detalla las características eléctricas que posee el relevador automotriz.

Tabla 8-2: Especificaciones Técnicas – Relevador Automotriz

Voltaje Nominal	12VDC
Voltaje de Operación	9-15 VDC
Resistencia de contacto	≤30mΩ
Consumo de energía de la bobina	1.6W
Corriente Nominal	20A, 30A / 14V
Máxima Corriente de contacto	20A N/C 30A N/O
Corriente de conmutación máxima (ON)	25A N/C 80A N/O
Corriente de conmutación máxima (OFF)	20A N/C 40A N/O
Fuerza Dieléctrica entre contactos	50Hz 500V

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.2.11 Conector IEC

Conector IEC (International Electrotechnical Commision) es el nombre que utiliza para referenciar a los trece conectores de alimentación eléctrica y trece paneles de enchufe, establecidos en la especificación IEC 60320. Existen clases de conectores que se diferencian de acuerdo a su tipo de aislamiento eléctrico.

- Clase 0: no tienen toma de tierra y presentan un único nivel de aislamiento.
- Clase I: tiene su chasis conectado a la toma de tierra.
- Clase II: equipos con doble aislamiento que no necesitan una conexión con seguridad una conexión a la toma de tierra.
- Clase III: diseñados para ser conectados desde una fuente de alimentación SELV (Separated or Safety Extra-Low Voltage)

Los conectores se asignan por parejas, lleva una numeración desde el 1 hasta el 24, es decir, existen 12 diferentes parejas de conectores. En este trabajo se hizo uso del conector C13 y C14 (Figura 36-2) para la alimentación del sistema prototipo.

a) b)

Figura 36-2: Conector IEC. a) Conector de chasis C14 (entrada), b) Cable de alimentación C13
Fuente: https://bit.ly/2vtHCFL - https://bit.ly/2IV4Z3Z

2.3 Diseño de mecanismos para el sistema prototipo

2.3.1 Mecanismo para las luces frontales

Con el uso de la plataforma SolidWorks, se crea el mecanismo que permita la segmentación del haz de luz, dividida en cinco segmentos, cada segmento tiene una longitud máxima de 204mm y una altura total de 88.50mm (Figura ***37***-2).

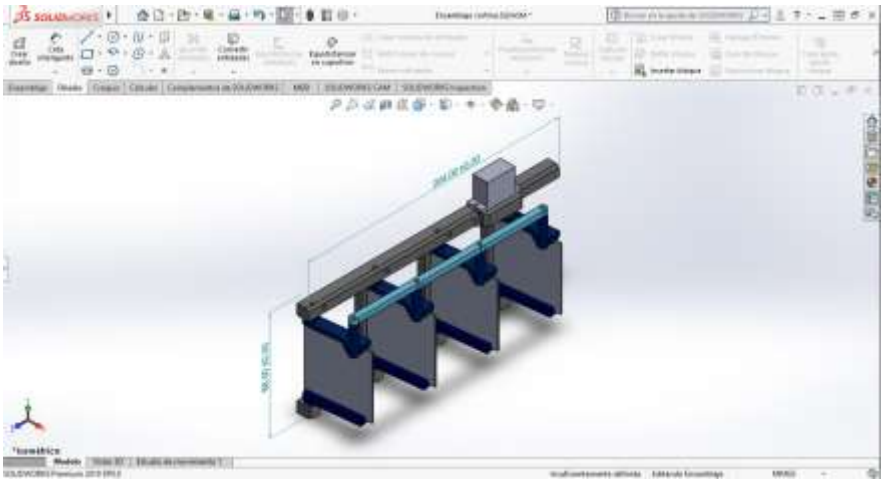


Figura 37-2: Ensamble del mecanismo tipo cortina - Solidworks
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

La apertura de este mecanismo depende del ángulo que reciba el Arduino, ante la dificultad que presenta los servomotores para realizar esta acción, debido a la fuerza del viento que se produce al instante de conducir el vehículo, se diseña soportes tanto en los bordes como en partes intermedias de la barra LED, estos soportes se usa para colocar una corte en acrílico transparente que sirva de cubierta y rompe vientos, permitiendo facilidad al servomotor de cumplir su respectiva función.

Los soportes extremos sobresalen del borde de la barra LED en una distancia total de 115mm (Figura 38-2), esta distancia es tomada en cuenta debido a la apertura que hace el mecanismo, brindando espacio suficiente para que este pueda accionar de mejora manera evitando inconvenientes futuros.

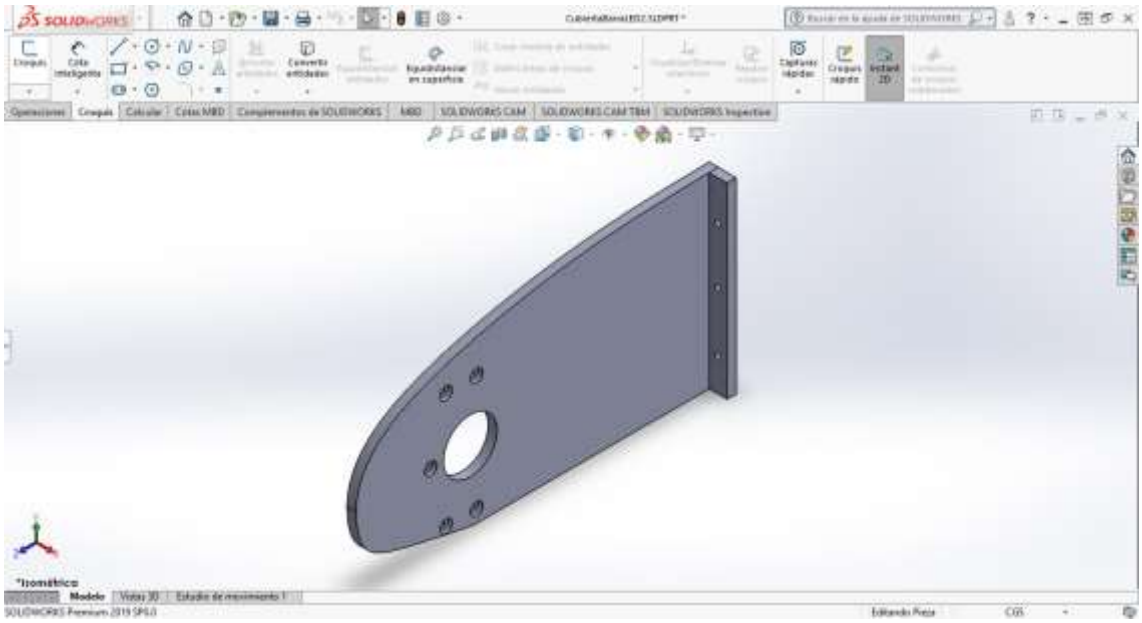


Figura 38-2: Soporte extremo para la cubierta LED - SolidWorks
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De igual manera se necesitaba de soportes intermedios en la barra LED, evitando que el material acrílico se rompa por la misma acción del viento que se genera en la carretera, en la Figura 39-2 se muestra el soporte que se usa en la parte inferior de la barra donde se encuentra el servomotor, se diseña el soporte de tal manera que cubra el este mecanismo incluyendo el servo.

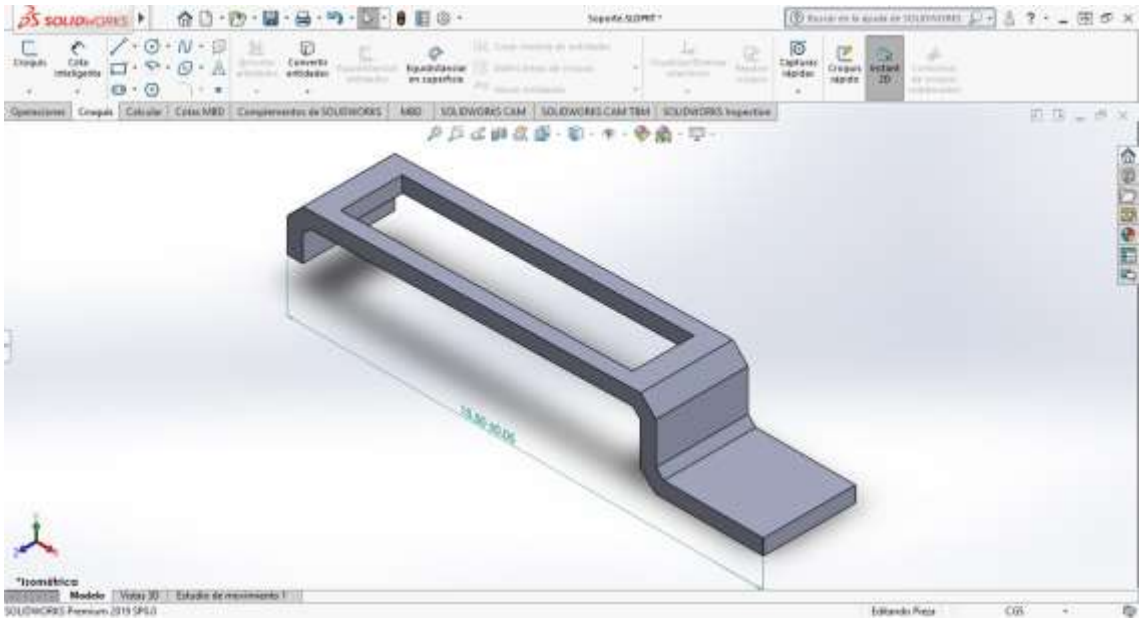


Figura 39-2: Soporte intermedio 1 - SolidWorks
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Dado que en la parte superior de la barra LED, el borde y mecanismo coinciden al mismo nivel, por lo que el soporte diseñado no debe sobresalir como el soporte mencionado anteriormente, se puede observar en la Figura 40-2 la diferencia que hay respecto al otro soporte.

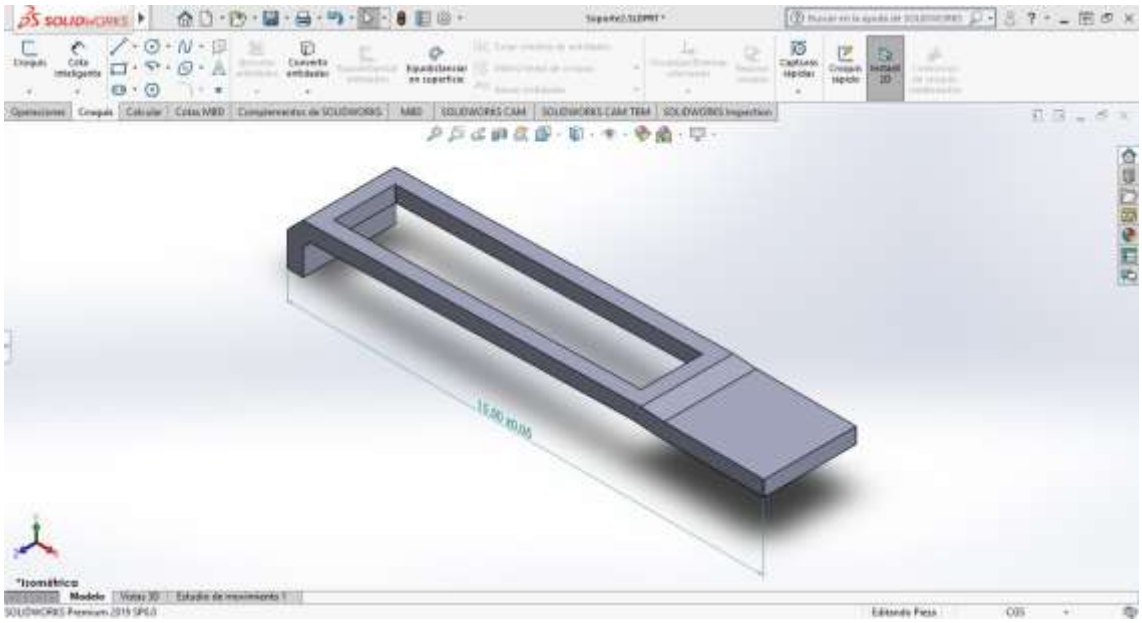


Figura 40-2: Soporte intermedio 2 - SolidWorks
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.3.2 Diseño del soporte para la cámara

De igual manera se realizó la base o soporte que se colocará encima del retrovisor interno del carro para situar la cámara de la Raspberry, de esta manera se obtiene la captura del entorno visible al conducir en carreteras que carecen de visibilidad, obligando al conductor a requerir luces largas para aumentar el campo de visión con la única desventaja de deslumbrar al conductor que transite por la misma vía.

Las dimensiones de este soporte, son de 46mm de ancho y 52.04mm de alto (Figura 41-2), con unas pequeñas muescas para que encaje perfectamente en el retrovisor y se pueda ajustar con tornillos milimétricos.

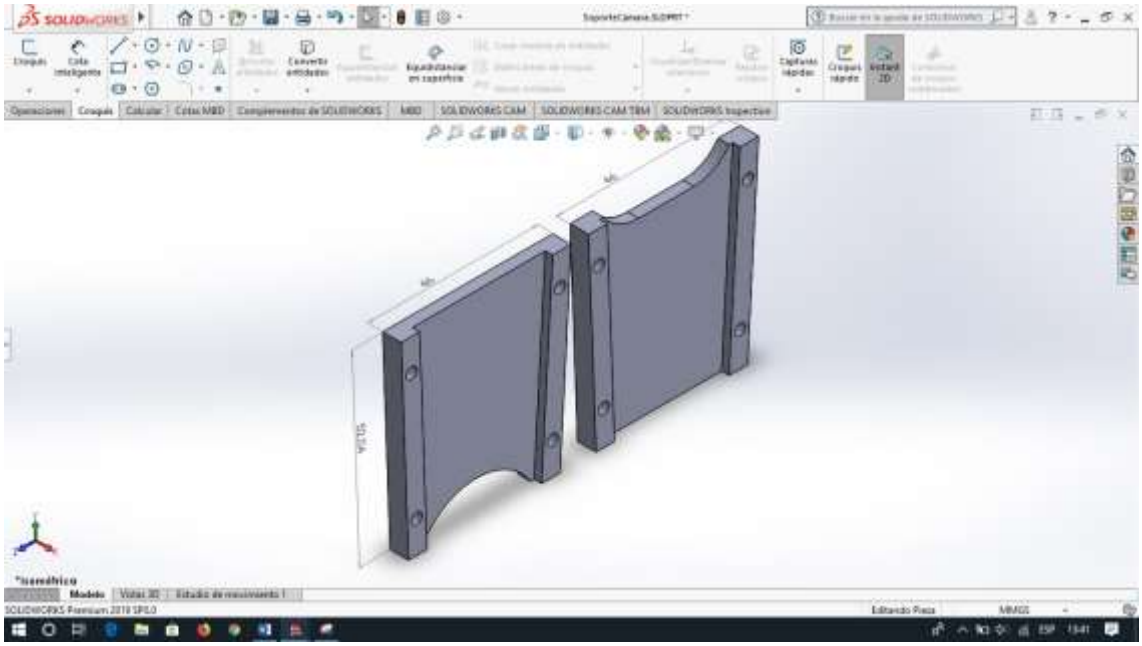


Figura 41-2: Soporte cámara para retrovisor - SolidWorks
Realizador por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.3.3 Diseño de la caja del sistema prototipo

Ante la necesidad de conectar todos los componentes y su transporte sea con facilidad, se hizo el diseño de una caja que albergue y proteja estos componentes.

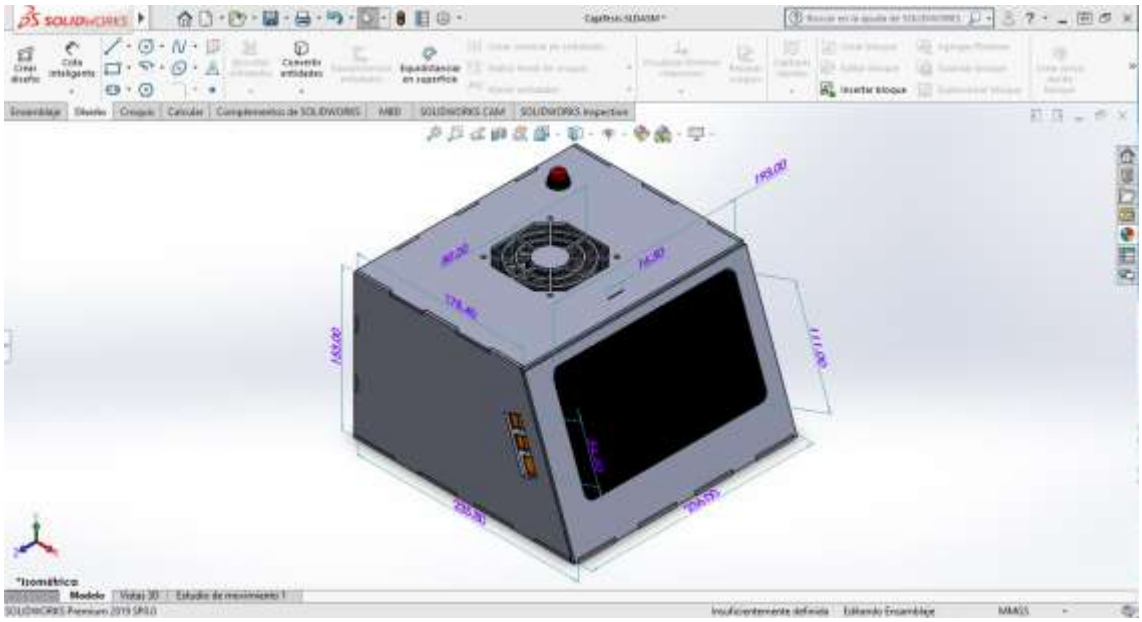


Figura 42-2: Ensamble de la caja del sistema prototipo - SolidWorks
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Figura 42-2 se muestra el diseño de la caja antes mencionada, y posee las siguientes medidas: 236mm de ancho, 235.80mm de profundidad y 153mm de altura, con estas dimensiones se tiene la capacidad de albergar todos los componentes y mantener todo en un solo lugar para poder ser transportado, de igual manera depende de las conexiones de electricidad tanto de la batería de 12V para la barra LED, y de un toma de 110VAC para la alimentación interna de la Raspberry, pantalla LCD, Arduino y demás elementos, esta fuente AC es proporcionada por el inversor que se menciona en la sección 2.1.7.

Dado que todo componente electrónico tiende a calentarse por el mismo hecho de estar en funcionamiento, se incluye al sistema un ventilador de PC con medidas de 80x80mm haciendo la función de propiciar un ambiente fresco de todos los elementos conectados, aportando en el rendimiento del sistema a pleno funcionamiento. El diseño de la caja posee un corte rectangular en un lateral permitiendo que los puertos USB y LAN queden con espacio suficiente para conectar dispositivos adicionales.

2.4 Desarrollo de la Lógica difusa

La lógica difusa se encarga de resolver problemas que no pueden ser resueltos por la lógica convencional, se establecen conjuntos de principios matemáticos basados en grados de membresía o pertenencia modelando así la información. Se basa en diferentes reglas lingüísticas (método de inferencia) que allegan a una función mediante la relación de entradas y salidas del sistema.

Esta lógica se emplea por medio de controladores difusos, entre algunos de estos sistemas se tiene, el control de calidad de agua, sistema automático operacional de trenes, elevadores, reactores nucleares, transmisiones automóviles y computadoras, entre otros.

Para el desarrollo del proyecto, se empieza con el conocimiento del problema a resolver, que en este caso trata de la detección de las luces de un vehículo que circule por la misma carretera con escasa iluminación, el sensor que se encarga de recibir las imágenes del entorno es la cámara, que se conecta a la placa Raspberry.

Las entradas del sistema hacen énfasis de acuerdo a la posición de los vehículos que son captados por la cámara, las posiciones están catalogadas respecto al eje X y eje Y (Figura 43-2), siendo el origen de coordenadas el punto superior izquierdo de la imagen capturada, el eje X proporciona entradas posicionales sea de lado izquierdo, centro o derecho, mientras que el eje Y provee entradas posicionales estando el vehículo lejos, cerca o muy cerca.

Figura 43-2: Identificación de las entradas del sistema prototipo
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

La función de membresía utilizada para representar las entradas y salidas del sistema es la trapezoidal debido a que la adquisición de video es continua, es decir frame a frame por lo tanto, se requiere un rango de pertenencia mayor en cada transición, a diferencia de las funciones triangular y sigmodal que el rango de pertenecían se reduce debido a la naturaleza de su forma respectivamente. En la Figura 44-2 se muestra la representación gráfica de la entrada en la posición X, siendo la línea de color azul la posición izquierda, de color verde la posición central y en rojo la posición derecha. El rango perteneciente a la posición izquierda comienza en 1 y se

mantiene hasta un 20% del ancho total de la imagen, luego decrece hasta 0 alcanzando el 40% del ancho, por último, se mantiene en cero hasta el final.

La posición central inicia en cero hasta el punto equivalente el 20% del ancho total, luego crece hasta el nivel más alto que se encuentra al 40% del total, de ahí se mantiene constante hasta el punto del 60%, posterior a ello decrece de constantemente hasta alcanzar el cero en el punto del 80% y finalmente se mantiene en cero hasta el final.

La posición correspondiente a derecha, empieza en cero hasta el punto del 60%, de ahí crece hasta alcanzar la unidad "1" que se ubica en el punto del 80% del total, y se mantiene en la unidad hasta el final del ancho de la imagen.

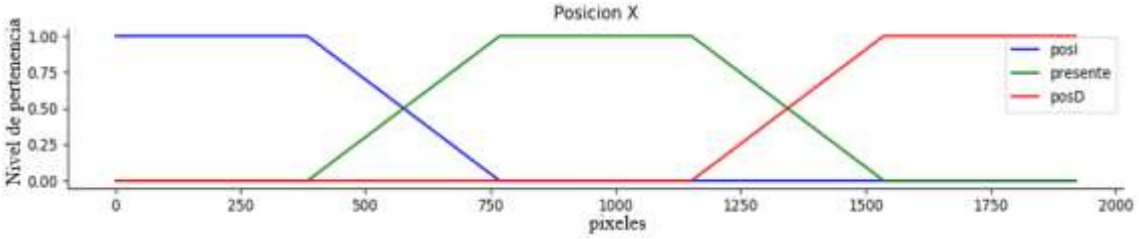


Figura 44-2: Representación de las entradas en la Posición X
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Las entradas que proporciona la posición Y, se representa en la Figura 45-2, la línea de color azul indica la posición de vehículo estando muy cerca, en color verde se representa la posición del vehículo estando cerca, y en color rojo la posición del vehículo si está lejos.

Para la posición lejana del vehículo, empieza con valor máximo hasta el punto correspondiente el 10% del alto, luego de ello alcanza el punto más bajo de forma constante ubicado en el punto del 20% de la altura de la imagen, mantiene su valor máximo hasta el final de la imagen.

La posición denominada "cerca" comienza en cero y se mantiene hasta el punto del 10%, desde ese punto empieza a tener comportamiento creciente hasta el punto 20% alcanzando el valor de uno, mantiene el valor hasta el punto del 50% y empieza a decrecer constantemente hasta alcanzar el valor de cero ubicado en el punto equivalente al 60% de la altura total, y mantiene el valor de cero hasta que finaliza.

Para la posición muy cerca se empieza en el valor más bajo siendo, se mantiene en cero hasta el punto localizado en el 50% del total de la altura que comprenda la imagen, luego crece hasta alcanzar el valor uno en el punto del 60% y se mantiene en ese valor hasta el final de la función.

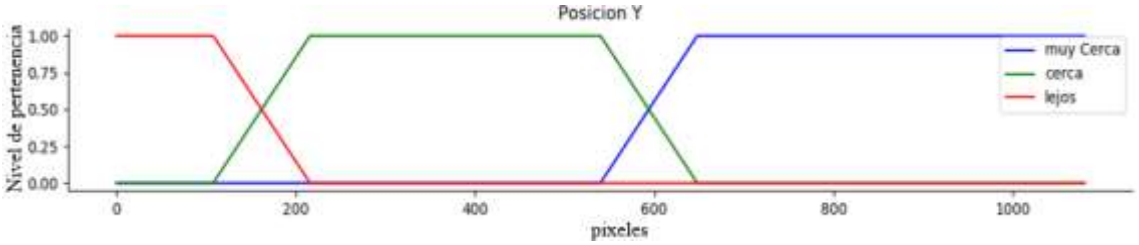


Figura 45-2: Representación de las entradas en la Posición Y
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

La salida se la representa con la función de membresía del trapezoide (Figura 46-2), en línea roja se presenta la salida cerrada, en verde medio cerrada y en azul la salida abierta, siendo la salida cerrada en 0° y abierta en 90°, debido a las reglas de la lógica difusa que serán descritas más adelante, el rango de valores en la salida tendrá un valor mínimo de 11° y 79 ° como valor máximo.

Figura 46-2: Representación de la función de membresía en la salida
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Figura 47-2**:** Identificación de vehículos-2 se muestra tres vehículos captados en un fotograma generado por la cámara, en recuadro amarillo el vehículo se encuentra cerca y de lado izquierdo, en rojo el vehículo se cataloga como cerca y en posición central y en recuadro verde el vehículo está lejos y de lado derecho.

Figura 47-2: Identificación de vehículos
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De acuerdo a las entradas establecidas se implementa reglas (Figura 48-2) haciendo del comportamiento una mejor opción al momento de segmentar el haz de luz proporcionado por la barra LED, estas reglas descritas sirven para obtener un ángulo de apertura en cada servomotor, las cuales se muestran en la Tabla 9-2.

Reglas de la lógica difusa para determinar las salidas.

- Si el vehículo está muy cerca y en la posición izquierda, entonces el sector está cerrado.
- Si el vehículo está muy cerca y en la posición central, entonces el sector está cerrado.
- Si el vehículo está muy cerca y en la posición derecha, entonces el sector está cerrado.
- Si el vehículo está cerca y en la posición izquierda, entonces el sector está cerrado.
- Si el vehículo está cerca y en la posición central, entonces el sector está medio cerrado.
- Si el vehículo está cerca y en la posición derecha, entonces el sector está medio cerrado.
- Si el vehículo está lejos y en la posición izquierda, entonces el sector está medio cerrado.
- Si el vehículo está lejos y en la posición central, entonces el sector está abierto.
- Si el vehículo está lejos y en la posición derecha, entonces el sector está abierto.

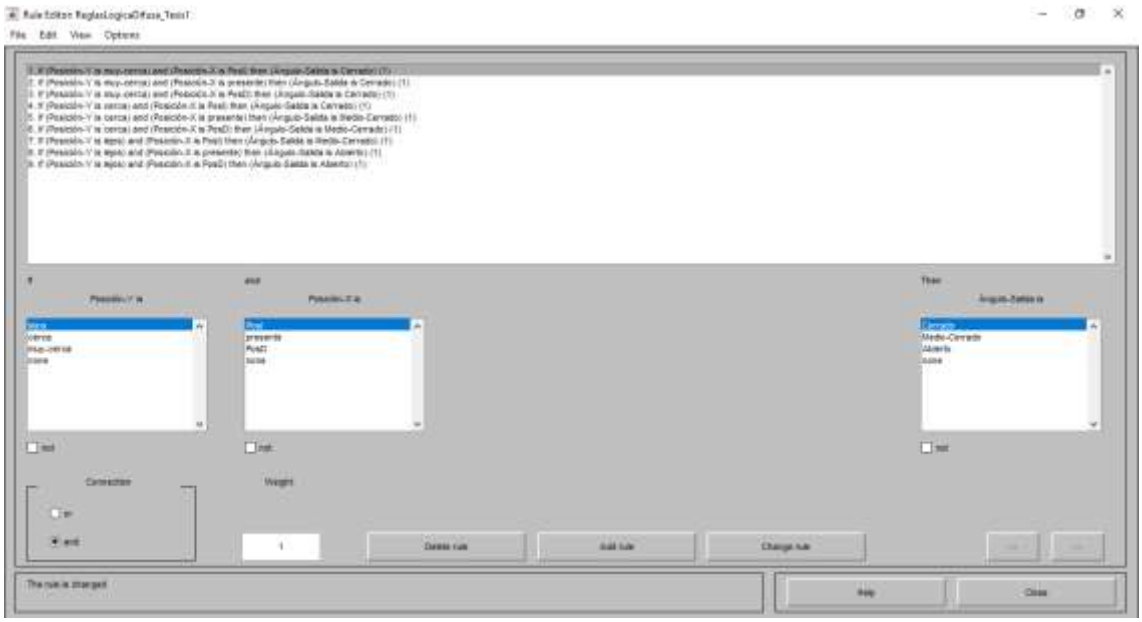


Figura 48-2: Reglas del control difuso establecidas en MATLAB
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Tabla 9-2: Relación Entradas-Salida de las reglas difusas

Entrada en X / Entrada en Y	Izquierda	Centro	Derecha
Muy Cerca	Cerrado	Cerrado	Cerrado
Cerca	Cerrado	Medio Cerrado	Medio Cerrado
Lejos	Medio Cerrado	Abierto	Abierto

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la plataforma de MATLAB se comprueba el funcionamiento de control difuso de manera simulada (Figura 49-2), haciendo cambios necesarios en las entradas, la salida muestra cambios cada vez que se manipula las entradas, el valor en la salida se debe al método del centroide el cual es uno de los métodos para desdifusificar el mecanismo de inferencia que se rigen a las reglas del sistema difuso y obtener de esa manera un valor concreto no difuso a la salida.

La Figura 50-2 muestra el comportamiento de la salida con respecto a las entradas, en una gráfica de 3 dimensiones y se puede apreciar notablemente los cambios que sufre la salida, cada que se incrementa o disminuye el valor de las entradas en la posición X y en la posición Y.

Figura 49-2: Simulación del comportamiento del sistema prototipo en MATLAB
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

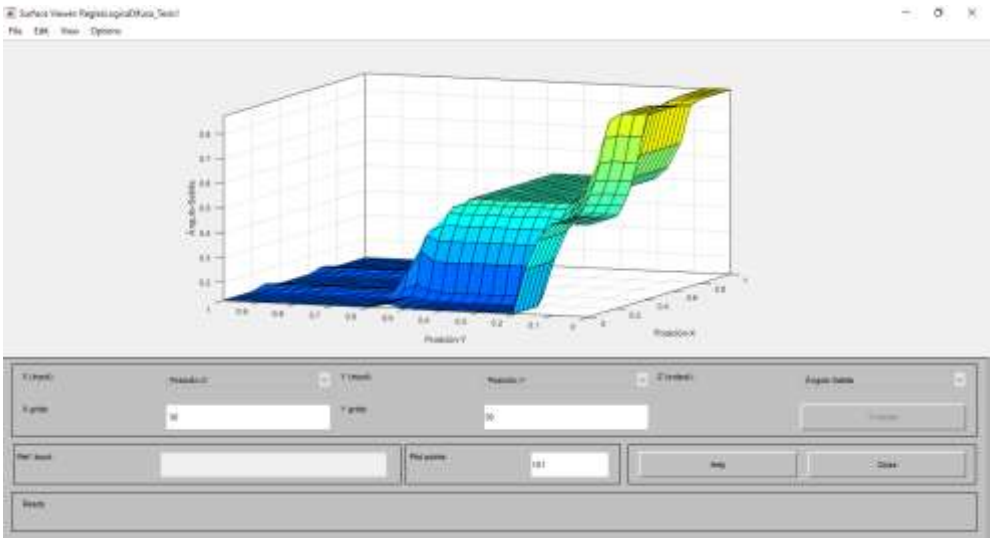


Figura 50-2: Comportamiento de entradas y salida – MATLAB
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

La Figura 51-2 muestra el esquema de funcionamiento del control difuso empleado en el sistema prototipo.

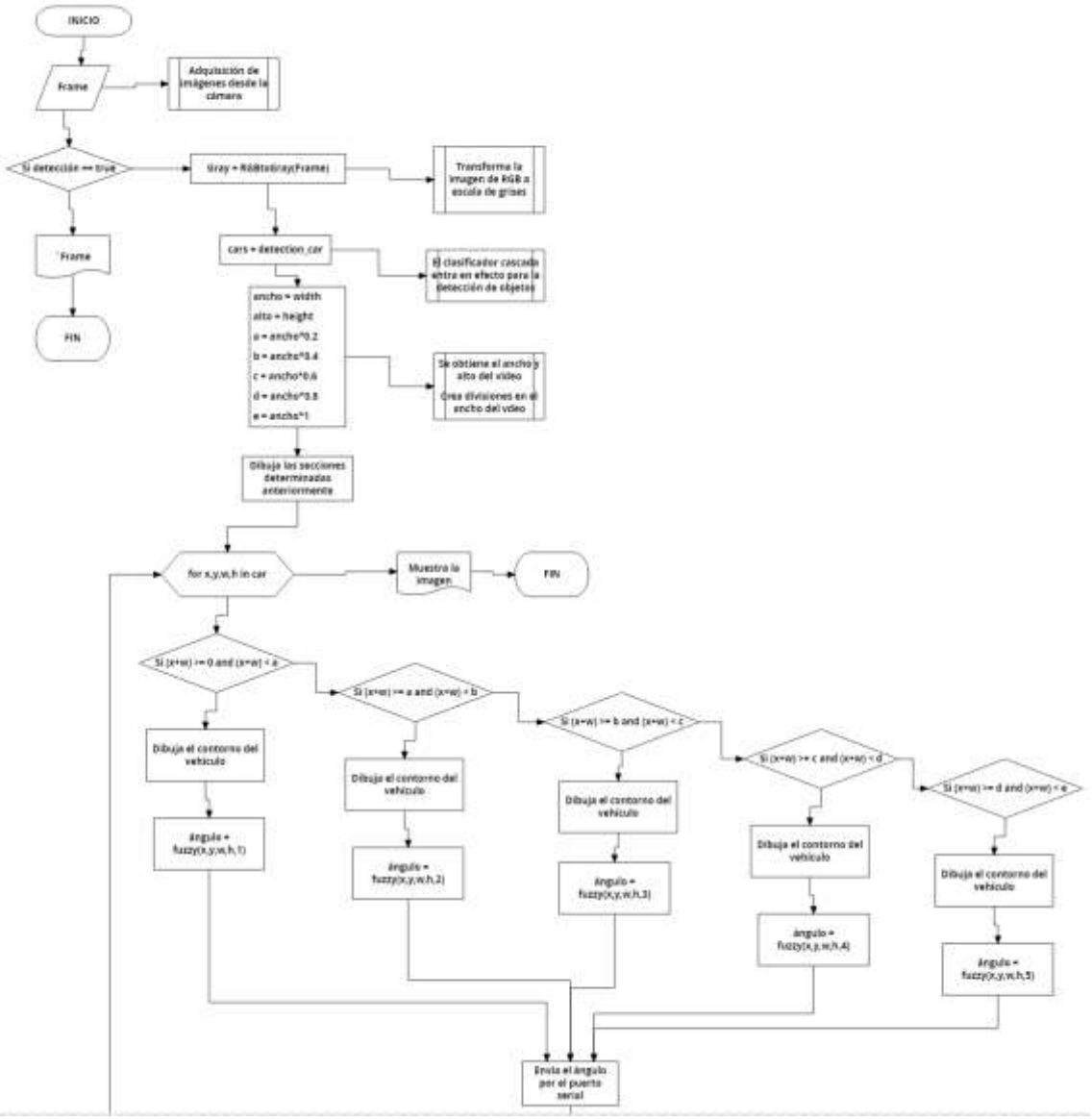


Figura 51-2: Diagrama de Flujo empleado en la Raspberry
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

El funcionamiento del control difuso empieza con la adquisición de la imagen, el sistema trabaja con los fotogramas que se descomponen instantáneamente de un video, al instante de ejecutar la detección de vehículos convierte cada fotograma a escala de grises para el análisis necesario, paralelamente dibuja recuadros que dividen el ancho de la imagen en cinco sectores con el objetivo de visualizar el sector activo cuando detecte el vehículo, tales sectores representan el comportamiento de los mecanismos tipo cortina que se implementan en la barra LED, como se muestra en la Figura 52-2, las condiciones a las que se rige el sistema para determinar el ángulo y el sector activo se detallan a continuación.

- Si el recuadro que encierra al vehículo se encuentra entre el rango de 0 a 0.2*ancho, entonces el sector uno se activa para ser accionado.
- Si el recuadro se encuentra entre los límites de 0.2*ancho hasta 0.4*ancho, entonces el sector dos se activa.
- Si el recuadro se encuentra en los límites de 0.4*ancho hasta 0.6*ancho, el sector tres se activa.
- Si el recuadro se presenta en los límites de 0.6*ancho hasta el 0.8*ancho, el sector cuatro se activa.
- Si el recuadro asoma en los límites de 0.8* hasta el final, entonces el sector cinco es el que se activa para las acciones requeridas.

Figura 52-2: Ejemplo de activación de un sector
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.5 Diseño del Circuito Electrónico

Es indispensable la conexión eléctrica de cada elemento, para ello se diseña una placa electrónica en la que todos los componentes se interconecten, se usa la plataforma EAGLE que permite la creación de PCBs.

Se empieza con el diseño esquemático de la placa en el entorno de trabajo de EAGLE (Figura 53-2), se inserta los siguientes componentes que se detalla en la Tabla 10-2, para luego realizar el ruteo respectivo.

Tabla 10-2: Elementos utilizados en el esquemático - EAGLE

Elemento	Cantidad
Bornera para circuito impreso	1 de 4 entradas, 3 de 3 entradas y 2 de 2 entradas
Fusible 12V	1 de 5A y 1 tipo cuchilla de 25A
Espadines machos	1 de 2 pines y 1 de 3 pines
Arduino Nano	1
Relay	1 Módulo de un canal y 1 Relay Automotriz
Convertidor DC/DC Buck 12/24VDC a 5VDC	1
Pulsador	1

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

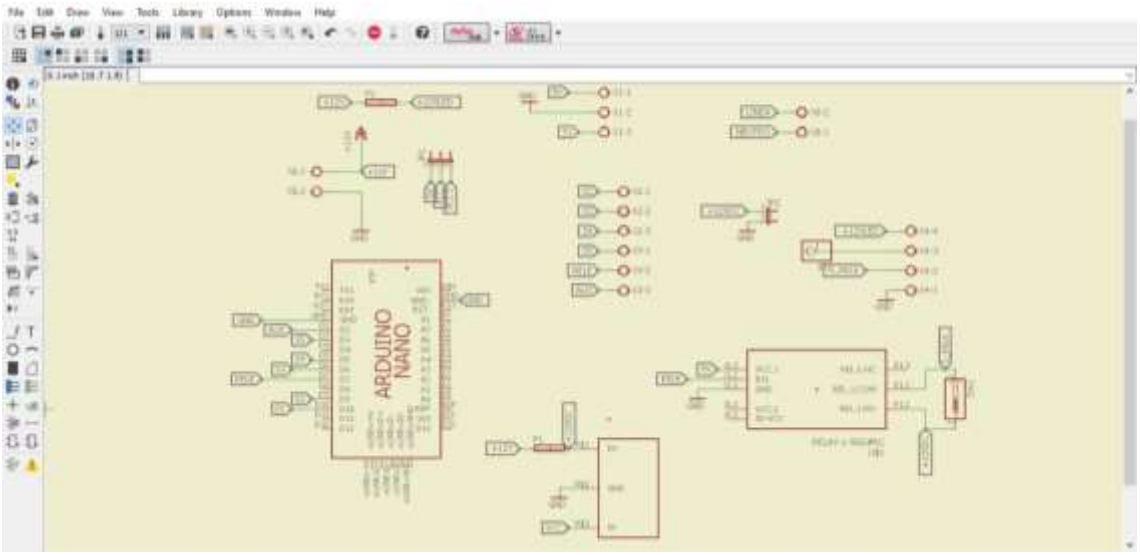


Figura 53-2: Diseño Esquemático de la placa electrónica
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

El diseño PCB se muestra en la Figura 54-2, para crear las rutas óptimas y minimizar puentes innecesarios, se coloca cada elemento de tal manera que la ruta respectiva sea la más corta, una vez realizado todo el ruteo, se hace la impresión de las pistas en papel couche, posterior a ello se transfiere el diseño a la baquelita por el método de la plancha, el tamaño de la baquelita debe abarcar todo el diseño PCB, una vez transferido se hace la eliminación del cobre innecesario de la placa, esto se logra con la mezcla de ácido férrico y agua caliente.

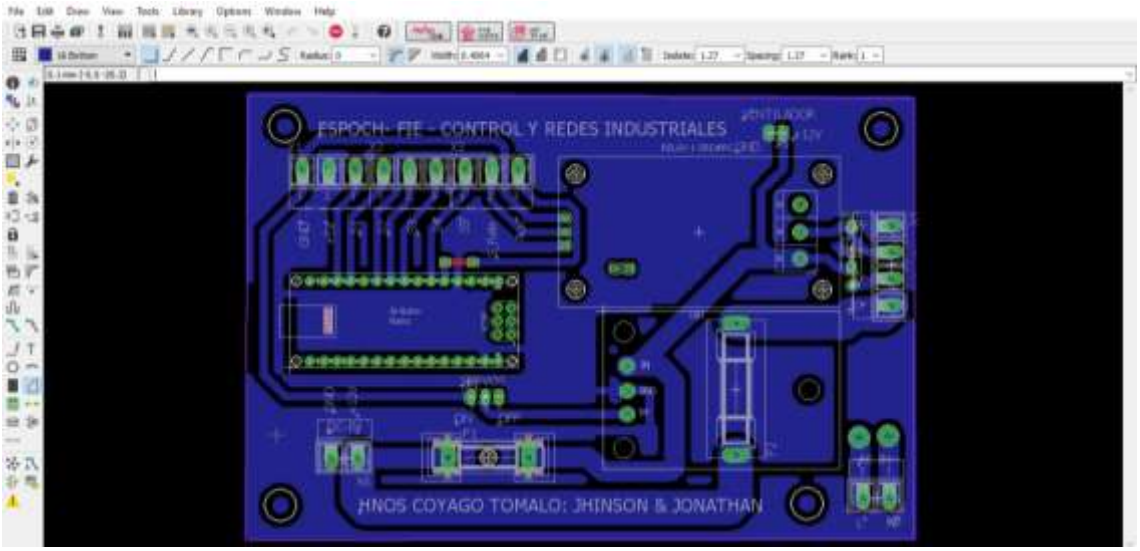


Figura 54-2: Diseño PCB de la placa electrónica
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

El diagrama de conexiones del circuito electrónico se presenta en la Figura 55-2, el sistema presenta dos circuitos, el de control y el de fuerza, en el de control como el nombre indica lo compone todos los elementos que interactúan entre sí para el correcto funcionamiento del sistema y que el voltaje y corriente no sean excesivas, en cuanto al sistema de fuerza lo compone desde el relé hasta la barra LED, debido a su consumo de corriente y voltaje en la que trabaja, en la figura se puede observar el sistema de fuerza dentro de líneas entrecortadas.

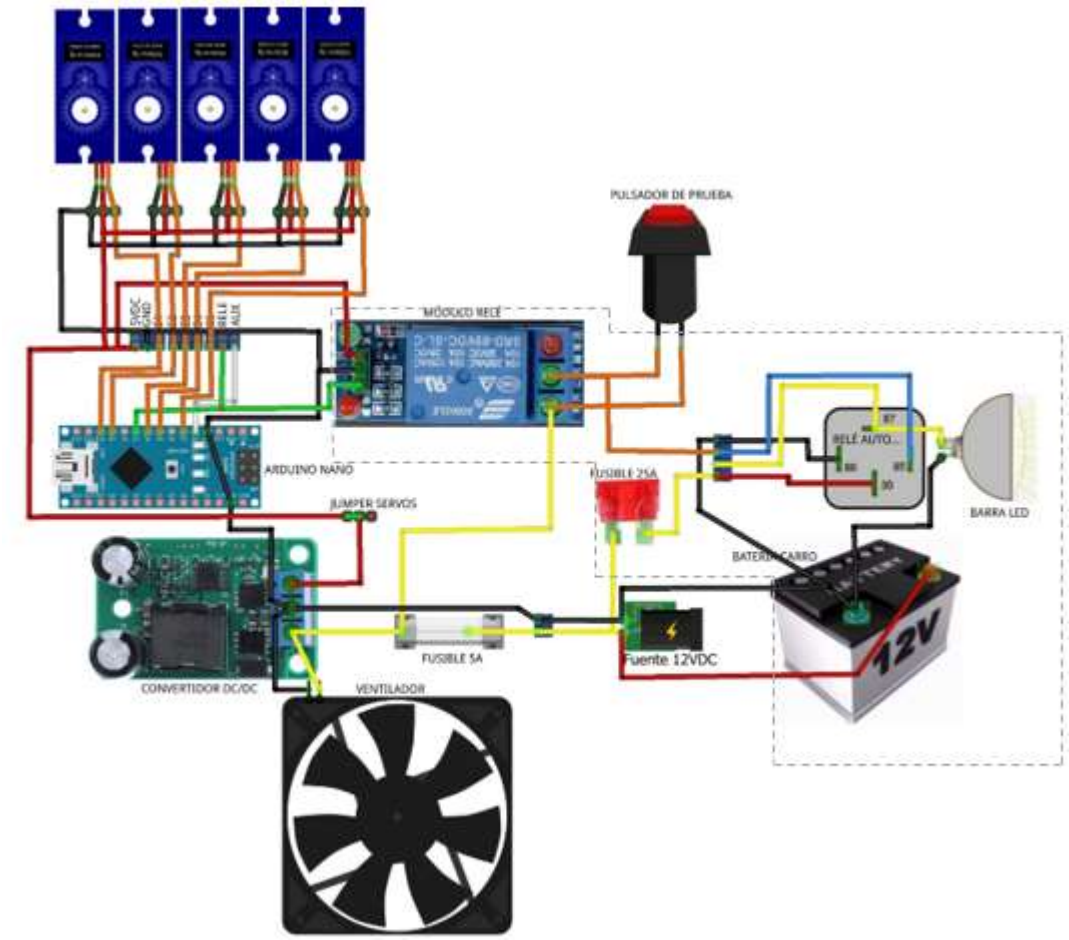


Figura 55-2: Esquema de cableado del circuito electrónico
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.6 Montaje del sistema prototipo

Para empezar con el montaje de todo el sistema prototipo de luces frontales, se empieza con la impresión 3D del mecanismo y los soportes, la colocación de los elementos en la placa electrónica, el cableado de todos los elementos y finalmente la sujeción de la barra en la parte frontal del vehículo a usar.

2.6.1 Impresión 3D

Con el diseño del mecanismo tipo cortina y de los soportes requeridos, se procede a imprimir las piezas necesarias para realizar el ensamble de cada sector, se observa en la Figura 56-2 el inicio y fin de la impresión de cada sector.

Figura 56-2: Impresión 3D del mecanismo
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Este mecanismo se ha incorporado a la barra LED (Figura 57-2) cubriendo todo el largo obteniendo una división de 5 sectores, cada sector cubre 16 de 80 leds en total y para cubrir el haz de luz se necesita de platinas que son colocadas en cada mecanismo.

Figura 57-2: Ensamble del mecanismo en le barra LED
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Para que el accionamiento del mecanismo tipo cortina sea de manera óptima se incorpora a la barra LED una cubierta de acrílico transparente (Figura 58-2 y 59-2), el mismo que cumple la función de romper el viento que es provocado al conducir el vehículo dirigido directamente al mecanismo, mientras mayor velocidad más dificultad presenta los servos para moverse libremente, por ello se añade el diseño que permite sujetar el acrílico a los soportes anteriormente creados en la plataforma SolidWorks e impresos en 3D.

Figura 58-2: Material Acrílico implementado en la barra LED
Realizado por: Coyago, Jhinson & Coyago, Jonathan.2019

Figura 59-2: Vista superior de la barra LED
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De la misma manera se imprime las respectivas piezas para la sujeción de la cámara en el retrovisor interno del vehículo, para fijar estos fragmentos fue indispensable usar tornillos milimétricos de pulgada y media los cuales son ajustados a conveniencia con sus respectivas tuercas, como se muestra en la Figura 60-2.

Figura 60-2: Fijación de la cámara en el retrovisor
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.6.2 Ensamble del gabinete del sistema

Antes de armar la caja del sistema se coloca los elementos necesarios en la placa electrónica elaborada anteriormente, soldando cada componente en el lugar correspondiente como muestra la Figura 61-2. Al finalizar dicho procedimiento, se comienza a ubicar cada elemento en la caja dejando espacio suficiente para optimizar la conexión cableada de todo el sistema, dentro de la caja se encuentra la placa antes mencionada, el relé automotriz con el fusible respectivo, el case de la Raspberry junto a la pantalla LCD y un ventilador.

Figura 61-2: Colocación de los elementos en la placa
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Para colocar la placa electrónica y demás elementos, se hace el ensamble de una caja protectora que abarque todos y cada de uno de los componentes además de las conexiones eléctricas, el diseño incluye un espacio especial para la pantalla LCD y espacio suficiente para encajar la placa electrónica (Figura 62-2). La alimentación del sistema prototipo requiere de una fuente de 110VAC y otra de 12VDC, el voltaje AC se usa para encender el micro-ordenador Raspberry junto a la pantalla mediante la propia fuente de 5V-3A, mientras que el voltaje DC sirve de alimentación para encender los servomotores y la barra LED, el paso de la alimentación en AC y DC es controlado por un interruptor cada uno, manipulando de mejor manera el encendido y apagado de todo el sistema.

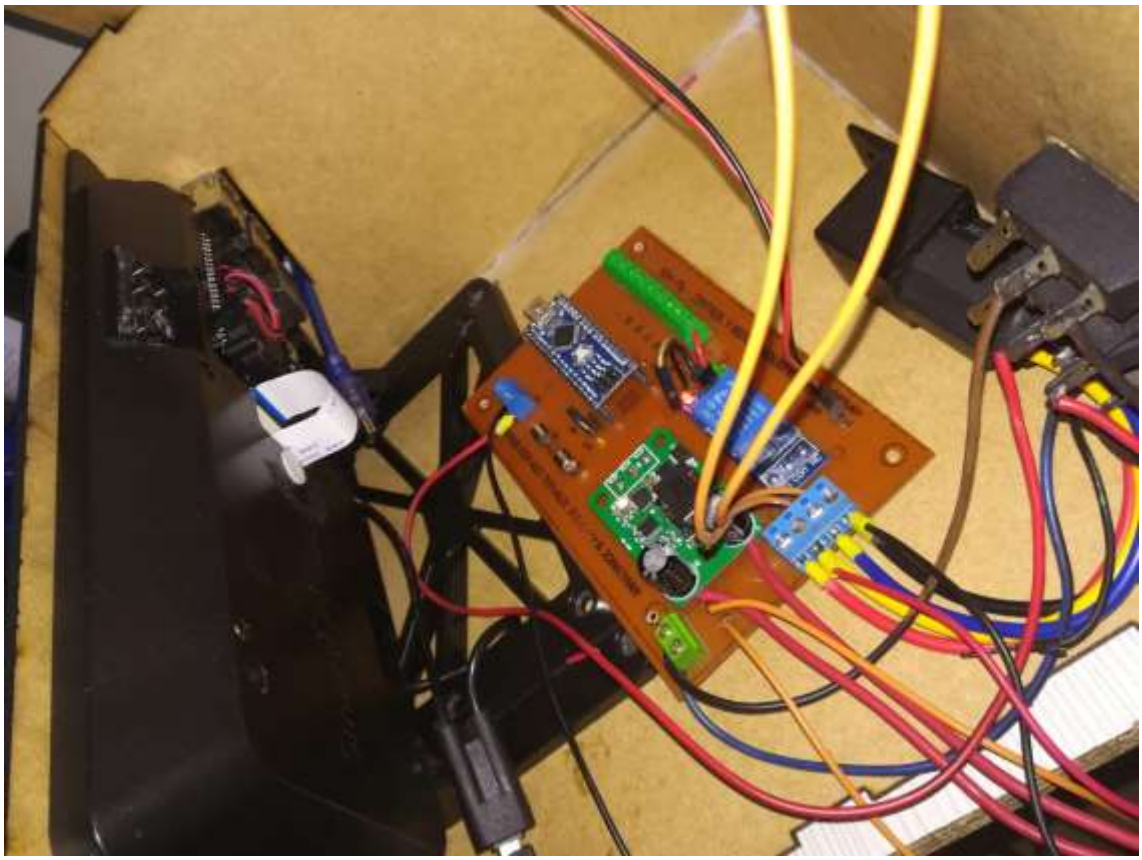


Figura 62-2: Armado interno de la caja del sistema
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Debido al calentamiento que generan los componentes electrónicos y por su espacio cerrado, se implementa un ventilador de PC con dimensiones 80x80x25mm manteniendo ventilado a todos los componentes que se encuentran internamente para evitar un sobrecalentamiento y hacer que desarrolle de mejor manera todo el sistema en sí, el pulsador que se encuentra en la parte superior de la caja hace la función de encender la barra LED cuando sea presionado, con esto se puede comprobar el funcionamiento correcto de las luces LED frontales descartando alguna falla, y en el caso de que encienda las luces y no funcione el sistema prototipo la falla se encuentra en alguno de los componentes electrónicos que conforma el sistema.

Figura 63-2: Montaje real de la caja del sistema prototipo
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Todos los componentes se encuentran sujetadas con pernos y silicona caliente, evitando que se muevan al momento de transportar el sistema y al momento de conducir por la vía, como se muestra en la Figura 63-2.

2.6.3 Montaje de la barra LED

Al realizar el montaje de la barra LED en la parte frontal del vehículo es indispensable optar por soportes que sean sujetas al chasis del carro y así colocar la barra, en la Figura 64-2 se observa como se ha colocado las luces frontales ubicada con una inclinacion prudente consiguiendo una mayor iluminación de la carretera.

Figura 64-2: Montaje de la barra LED
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.7 Captura de imágenes

Para las respectivas pruebas, es indispensable obtener imágenes del entorno en el que se desarrollará el sistema pototipo, las imágenes de pruebas han sido captadas en la vía Riobamba - Ambato, a la altura de San Andrés, al momento de realizar la trayectoria se filma videos con la camara oficial de Raspberry como se muestra en la Figura 65-2, con el objetivo crear una amplia cantidad de muestras que serviran de entrenamiento para los clasificadores en cascada generados por Cascade Trainer GUI.

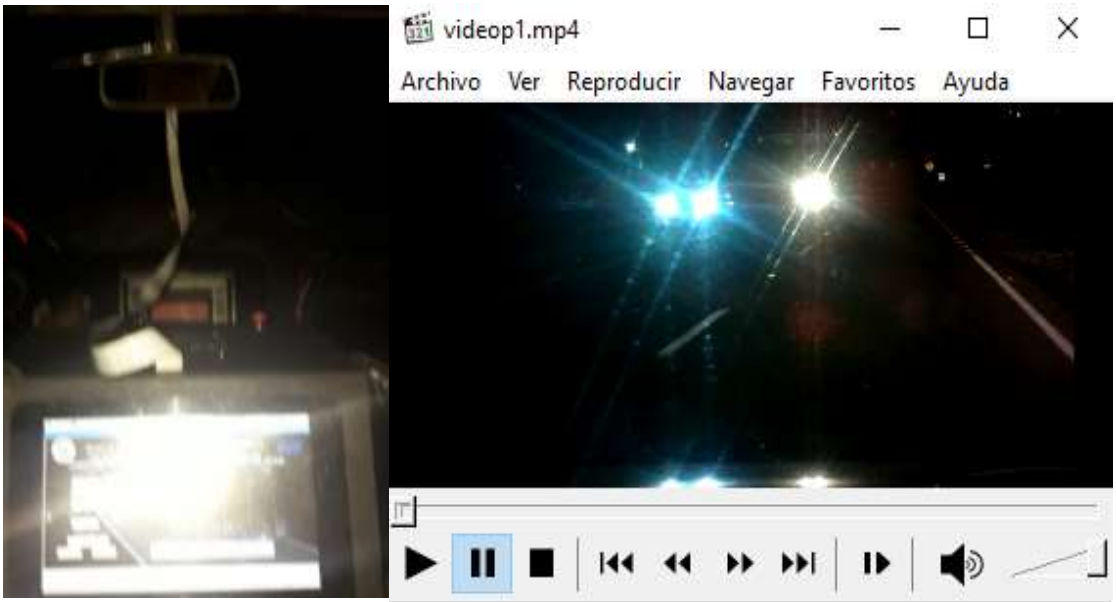


Figura 65-2: Captura de imágenes
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

2.8 Esquema de funcionamiento del prototipo

Luego de finalizar con las conexciones electricas del sistema prototipo montado en el vehículo comprobando el funcionamiento del mismo, en la Figura 66-2 se muestra el esquema explicativo del modo de operacion, se procede a encender el vehículo, inicializar el sistema, teniendo en cuenta que su espacio de trabajo debe ser en carreteras con carencia de visibilidad y donde sea o no indispensable el uso de luces largas, se procede a realizar las respectivas pruebas del sistema y que el mismo cumpla con los requerimientos propuestos inicialmente, ademas se pretende evitar deslumbrar o encandilar la vista de los otros conductores circulantes en la misma carretera.

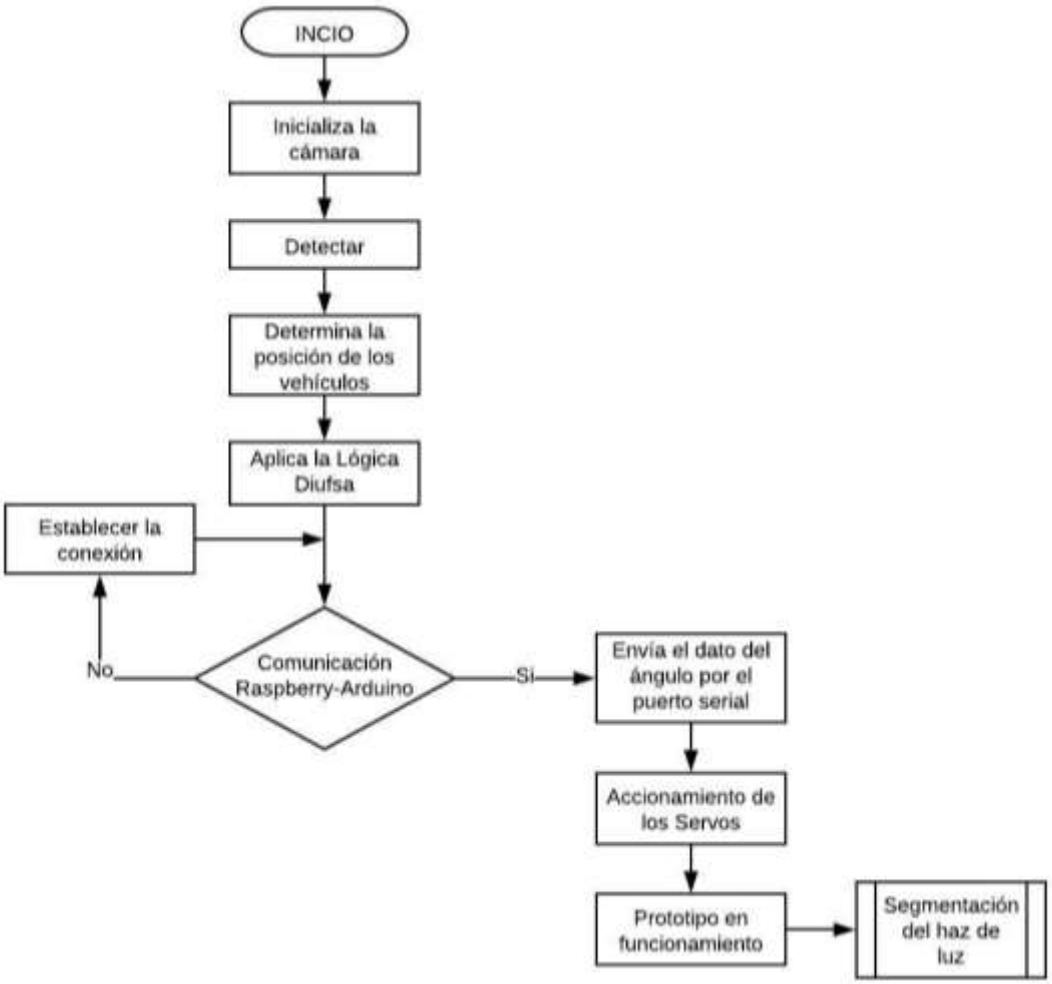


Figura 66-2: Esquema de funcionamiento del Sistema Prototipo
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

CAPÍTULO III

3. PRUEBAS Y RESULTADOS

Una vez determinado los parámetros que rige la lógica difusa para este sistema prototipo y el algoritmo de deteción de objetos, se procede a realizar pruebas de reconocimiento de vehículos que transiten en la carretera en condiciones climáticas aceptables y con escasa visibilidad, por ello se pone a prueba los diferentes clasificadores en cascada para determinar la efectividad de cada uno al identificar faros vehiculares.

3.1 Clasificadores en cascada en espacios controlados

El primer clasificador en cascada que se utilizó fue entrenado con un número de muestras específicas captadas por la cámara oficial de Raspberry, este entrenamiento posee una base de 330 muestras positivas y 413 negativas, con el archivo de extensión ".xml" generado, al final del entrenamiento se utiliza en la programación de python para que se ejecute al momento de la detección de los vehículos.

Antes de proceder a las pruebas del sistema prototipo en carreteras y en tiempo real, se hace el análisis respecto a la efectividad de detección, se realiza simulaciones de funcionamiento en espacios controlados con una serie de videos captados y almacenados por la Raspberry, dentro de los videos aparecen varios vehículos los cuales ponen a prueba el reconocimiento en base a las muestras positivas y negativas. Este clasificador presenta un gran margen de error ya que no es capaz de detectar la presencia de los vehículos como se muestra en la Figura 1-3, por lo cual se descarta inmediantamente.

Figura 1-3: Prueba primer clasificador
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En el segundo entrenamiento se añade más muestras tanto positivas como negativas, siendo un total de 435 muestras positivas y 679 muestras negativas. Se comprueba el funcionamiento del clasificador con los videos mismo videos antes mencionados como se muestra en la Figura 2-3, presenta un resultado con una mejor respuesta al momento de reconocer los vehiculos presentes en la carretera en comparación con el primer clasificador, pero se puede apreciar tambien que presenta falsas detecciones debido a que en la noche existen mucha iluminaria que puede ser detectada como si fuese un vehículo ya sean estas de casas, alumbrado público o reflejo de la señaleticas viales.

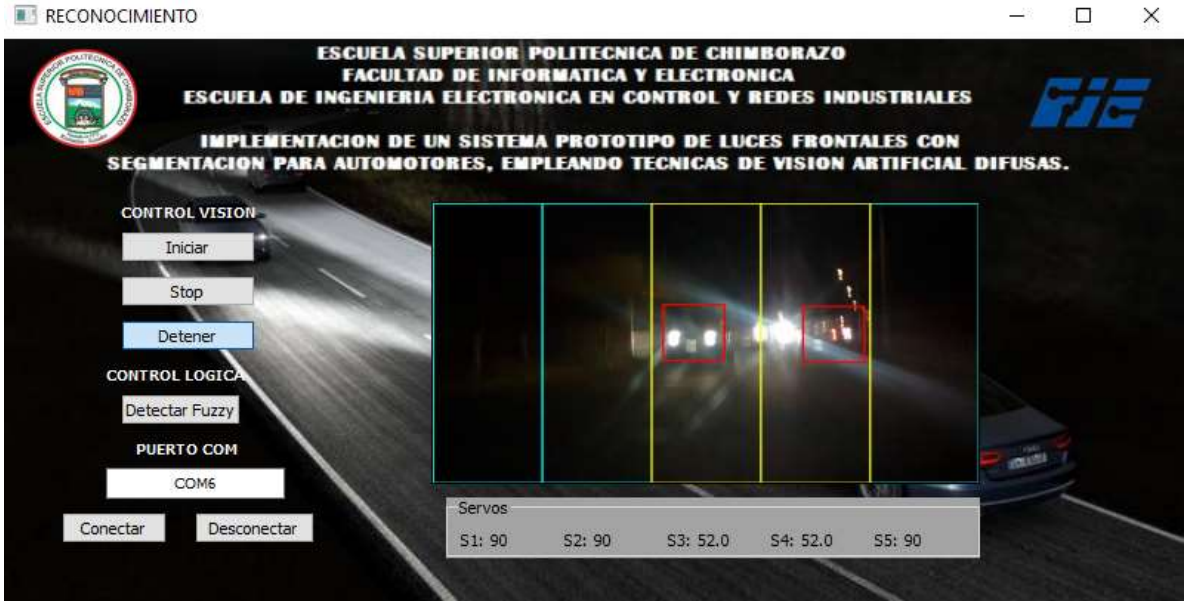


Figura 2-3: Prueba segundo clasificador
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Para mejorar la efectividad en la detección de los vehículos, se optó por renovar el clasificador en cascada, esto se logra nuevamente aumentando el número de muestras las cuales son tomadas de nuevos videos captados por la cámara, generando una base de datos de muestras en un total de 1000 muestras positivas y 4019 muestras negativas, al momento de probar dicho clasificador se obtiene un aproximado del 95% en detecciones positivas, como se muestra en la Figura 3-3. En comparación con el primer y segundo clasficador este ultimo llega a detectar más vehículos.

Para ser un sistema prototipo el porcentaje de efectividad del clasificador cumple con el requerimiento inicialmente propuesto, que es la capacidad de detectar los vehículos que transiten en la vía y con carencia de visibilidad nocturna. Se utilizó este último entrenamiento para realizar las pruebas necesarias en la carretera y en tiempo real, no se realizaron más entrenamientos debido al consumo computacional que requiere para crear un clasificador en cascada, la disponibilidad de movilización en carreteras fue un factor que provocó limitaciones en la adquisición de imagenes reales y para mejorar la efectividad al momento de detectar vehículos se requiere de

nuevas muestras positivas y negativas con una cámara de mayor resolución en donde se identifique de mejor manera los diferentes tipos de faros vehiculares.

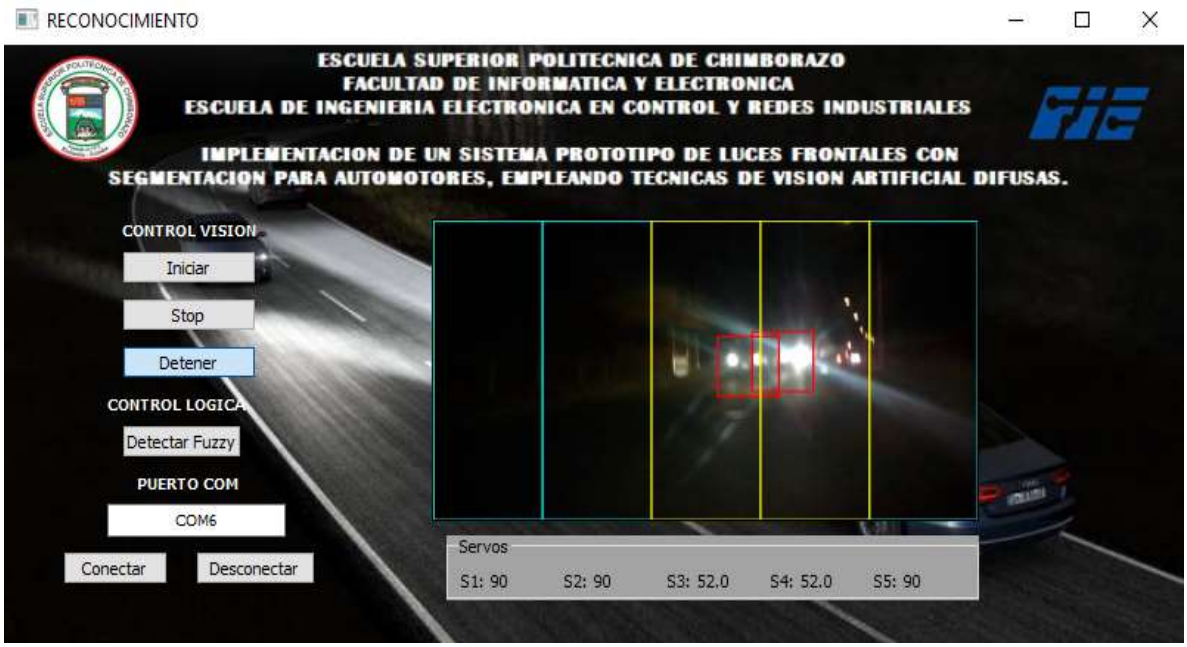


Figura 3-3: Prueba tercer clasificador
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Tabla 1-3, se muestra los resultados recopilados de los clasificadores en cascada puestos a prueba.

Tabla 1-3: Datos comparativos entre los clasificadores en cascada

COMPARACIONES DE CLASIFICADOR EN CASCADA							
Entrenamiento del Clasificador			**Resultados del Clasificador**				
Número de entrenamiento	**Muestras Positivas**	**Muestras Negativas**	**Verdaderos Positivos**	**Falsos Positivos**	**No detectados**	**Total de vehículos**	**Efectividad Verdaderos Positivos (%)**
1	330	413	2	5	77	79	2.53
2	435	679	52	8	24		65.82
3	1000	4019	75	3	4		94.94

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Para obtener el porcentaje de efectividad por cada clasificador se usa la siguiente relación (2), sin tomar en cuenta los falsos positivos porque el nivel de efectividad que se requiere está basado en la detección de los vehículos.

$$Efectividad(\%) = \frac{Verdaderos\ Positivos}{Número\ total\ de\ vehículos} \quad (2)$$

Las falsas detecciones se tiene debido al alumbrado público, casas o el reflejo de la señaletica vial presente en las carreteras, ya que en ocasiones particulares se asemeja a los faros de un vehículo, pero gracias a la robustez del entrenamiento las falsas detecciones no son a gran escala, para solucionar este tipo de errores se puede realizar un entrenamiento más riguroso, es decir, incrementar el número de muestras y que abarquen diferentes tipos de escenas considerando a los vehiculos circulantes que se lleguen a encontrar en la vía.

3.2 Tiempos de respuesta del sistema en espacios controlados

Para determinar el tiempo de respuesta del algoritmo de detección y lógica difusa, con la ayuda de un cronómetro de celular se procede a poner en marcha el sistema en ciertos casos, de igual manera se realizó las pruebas en dos computadoras que cuentan con mejor procesador y memoria RAM en comparación a la Raspberry.

Al inicializar el sistema se carga un video almacenado en la carpeta que contiene los archivos del mismo que tiene una resolución de 1920x1080 pixeles y un tiempo de duración de 2 minutos y 22 segundos, el cual es usado en la Raspberry y en las computadoras.
Las características que presenta las computadoras se detallan en la Tabla 2-3.

Tabla 2-3: Características de las CPU

CPU	PROCESADOR	RAM	GPU
Raspberry Pi 3	Quad Cortex A53 a 1.2GHZ	1 GB SDRAM @ 400MHz	VideoCore IV de doble núcleo
Sony Vaio – VPCEB15FM	Intel Core i3 – 2.13 GHz	4 GB, DDR2	Intel HD Graphics
Lenovo – G40-80	Intel Core i5 – 2.2 GHz	8 GB, DDR3	AMD - 2GB

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Los casos en los que se analiza el sistema son cuando: se muestra el video, empieza a detectar a los vehículos, comunicación con el Arduino y finalmente evitando mostrar el video, pero cumpliendo con las funciones de detectar y comunicar al Arduino.

Tabla 3-3: Tiempo de ejecución entre las CPU

Tiempo de ejecución			
Computadora / **Acción realizada**	**Raspberry Pi 3**	**Laptop Core i3**	**Laptop Core i5**
Mostrar Video	00:02:35.65	00:00:41.11	00:00:32.51
Mostrar Video + Detección	00:05:36.63	00:02:36.10	00:01:34.65
Mostrar Video + Detección + Comunicación Arduino	00:07:01.78	00:03:13.57	00:02:41.92
Detección sin mostrar Video	00:03:30.18	00:01:25.14	00:01:05.46
Detección + Comunicación Arduino sin mostrar Video	00:04:58.68	00:02:43.72	00:02:17.11

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Como se puede observar en la Tabla 3-3, existe una gran diferencia en lo que refiere a la ejecución del sistema entre la Raspberry y una Laptop, también se observa que el sistema tiene un mayor consumo de recursos computacionales al momento de estar en completo funcionamiento.

Para determinar el tiempo de retardo en la captura de video en tiempo real, se inicializa la cámara en la Raspberry y se procede a realizar movimientos controlados mientras la cámara está captando la imagen se toma el tiempo con ayuda del cronómetro, iniciando al instante en el que se realiza el movimiento y finalizando cuando en la pantalla se muestre dicho movimiento de acuerdo al análisis anterior se obtuvo los siguientes resultados.

Tabla 4-3: Tiempo de retardo en pleno del sistema en tiempo real

Tiempo de retardo			
Computadora	**Raspberry Pi 3**	**Laptop Core i3**	**Laptop Core i5**
Inicializar cámara	530ms	440ms	300ms
Inicializar cámara + Detección de vehículos	610ms	480ms	340ms
Inicializar cámara + Detección de vehículos + Comunicación Arduino	670ms	520ms	360ms

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Analizando los datos de la Tabla 4-3, se determina que el sistema prototipo empleado en la Raspberry Pi 3 tiene un retardo en la adquisición de las imágenes de tiempo real de unos 670 milisegundos en comparación a la computadora i5 que tiene 360ms. De acuerdo al criterio presentado en la Tabla 3-3, se puede mejorar el tiempo de respuesta del sistema si se evita mostrar el video que esté capturando en ese momento, pero empleando una señal de aviso visual cada vez que un sector es activado, como lo hacen en muchos sistemas comerciales, el tiempo que le toma a la Raspberry en ejecutar el sistema sin mostrar el video es de aproximadamente 300 milisegundos, lo que es mucho más favorable en la detección de vehículos al momento de transitar por la carretera en tiempo real.

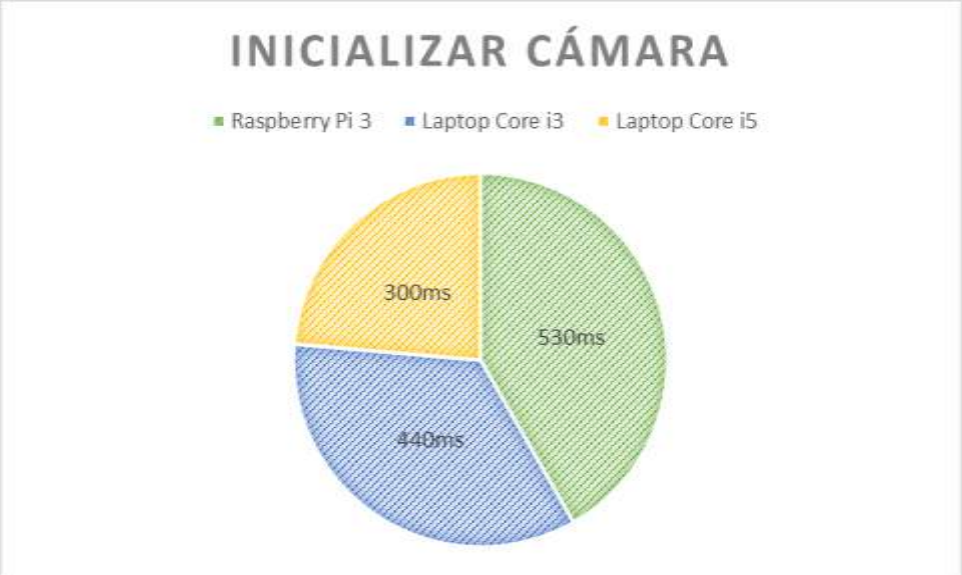


Gráfico 1-3: Tiempo de retardo al inicializar la cámara
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

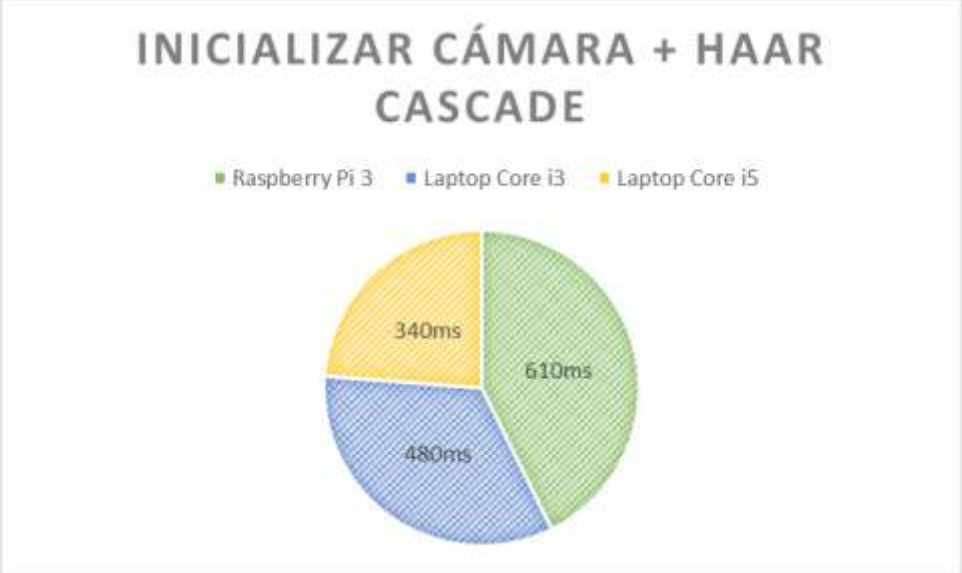


Gráfico 2-3: Tiempo de retardo al aplicar Haar Cascade
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Gráfico 3-3: Tiempo de retardo al poner a pleno funcionamiento el sistema
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Con las pruebas realizadas se obtiene que el sistema a pleno funcionamiento con la detección de vehículos y comunicación de Arduino se tiene un mínimo 360ms en un procesador Core i5, mientras que en la Raspberry se tiene un tiempo de retardo de 670ms, es decir aproximadamente se duplica.

3.3 Consumo de corriente

Con la ayuda de un multímetro "ingco - dm200", se realiza mediciones del consumo de corriente del sistema prototipo alimentado con una batería 12V – 18Ah, estando encendido y en reposo, al momento de iniciar la Raspberry y sin ejecutar el programa se tiene un consumo de corriente en DC de 0.03A (Figura 4-3) y el consumo de corriente en AC por parte de la Raspberry y Pantalla LCD se tiene un valor de 0.051A (Figura 5-3).

Tabla 5-3: Mediciones de corriente del sistema

Fuente	Encendido Raspberry	En reposo	Inicialización de la cámara	Detección de vehículos	Comunicación Arduino
12VDC	0.03 Amp	0.03 Amp	0.03 Amp	0.03 Amp	[1.71; 2.03] Amp
110VAC	0.051 Amp	0.012 Amp	0.071 Amp	0.071 Amp	0.071 Amp

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Figura 4-3: Medición de corriente en la Fuente DC
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Cabe recalcar que el consumo de corriente en la batería de 12VDC es el consumo de los componentes electrónicos colocados en la placa incluyendo a la bobina del relé automotriz y excluyendo el consumo de la barra debido a que el consumo del mismo se debe medir en serie al terminal 87 del relé automotriz, teniendo un consumo de corriente aproximado a 20A según las características de la barra led presentadas en el anterior capítulo.

Figura 5-3: Medición de corriente en la Fuente AC
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Para la medición de corriente en AC se usa una pinza amperimétrica TRUPER MUT202 en el momento de arrancar la Raspberry, al iniciar el sistema, inicializar la cámara, empezar con la detección de vehículos y al momento de conectar con el Arduino para el accionamiento de los servomotores, los datos recogido se tabulan en la Tabla 5-3.

La fuente de 12VDC no muestra cambios de corriente hasta que se haya establecido la comunicación con el Arduino y el sistema se encuentre detectando los vehículos, el microcontrolador manipula el encendido de la barra y el control de los servos sólo en el instante en que el sistema se encuentre en funcionamiento, mismo que es gobernado por los datos recibidos desde la Raspberry a través de comunicación serial.

3.4 Iluminación del entorno en espacios controlados

El vehículo usado para las pruebas pertinentes fue una camioneta Mazda BT-50 doble cabina año 2013, la barra es colocada en la parte frontal del vehículo adecuado a unas bases metálicas que se sujetan al chasis, en la Figura 6-3 se observa el montaje antes mencionado.

El objetivo primordial de estas pruebas de iluminación simplemente es para poder visualizar que efecto causa al obstruir el paso de luz en uno o varios sectores añadidos a la barra LED.

Figura 6-3: Montaje de la barra LED en el vehículo
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Se aprecia la vista frontal que es percibida por los conductores que circulan en la vía y en sentido opuesto como se muestra en la Figura 7-3, enumerando los sectores de derecha a izquierda como S1, S2, S3, S4 y S5.

Figura 7-3: Identificación de segmentos desde la vista frontal
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Con la barra situada en su lugar, se procede a verificar la visibilidad del entorno por parte del vehículo, comparando con las luces bajas, altas y con la ayuda de la barra LED.

a) b) c)

Figura 8-3: Iluminación del entorno. a) luces bajas, b) luces altas, c) barra LED sin cambios
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Como se observa en la Figura 8-3, la visibilidad del entorno mejora notablemente con la ayuda de la barra LED, en comparación con las luces bajas y altas originales del vehículo, se obtiene un campo de visualización con mayor iluminación para una conducción optima en la vía, permitiendo que el conductor se sienta seguro de su circulación y pueda tener mejor reacción ante eventos de riesgo o colisiones.

Cuando las persianas de la barra se encuentran cerradas, es decir, a su ángulo mínimo de 11° en todos los sectores, se puede apreciar que de ninguna manera llega encandilar, ya que la fuente lumínica está siendo obstruida casi por completo, ante lo descrito se puede observar en la Figura

9-3 el efecto producido quedando el campo visual con muy poca iluminación, por eso se debe mantener las luces bajas del vehículo siempre encendidas.

Figura 9-3: Vista frontal del vehículo con barra LED cerrada por completo
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

3.4.1 Vista posterior de la iluminación del entorno

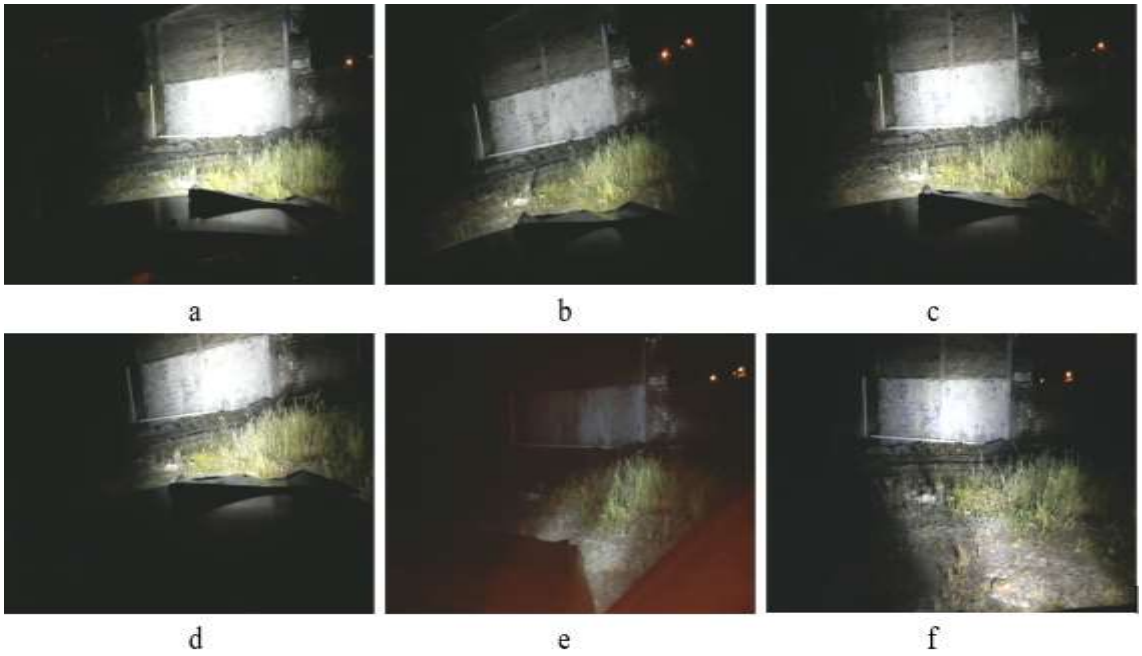


Figura 10-3: Vista posterior de las pruebas de iluminación
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Se realizaron las pruebas de iluminación que provee la barra LED en distintos casos como se muestra en la Figura 10-3, por lo cual se carga videos en los que se visualicen diversos eventos de acuerdo a la posición del vehículo y que cumpla con las reglas de la lógica difusa establecidas previamente, según los valores obtenidos se puede observar el cambio de visibilidad del entorno debido a variaciones en los servomotores según el sector correspondido, mismos que varían en un rango de 79° grados como máximo y un mínimo de 11° siempre y cuando se detecte un

vehículo para obstruir o dejar pasar el haz de luz, sin tener en cuenta las luces que posee el carro. Por otro lado, cuando el sistema no detecta la presencia de automotores circulantes durante una iteración de 15 fps o en su defecto aproximadamente 500ms debido a que la cámara trasmite video a una velocidad de 30fps, las persianas se abrirán completamente a 90° todos los sectores dejando de lado la lógica difusa.

En la Tabla 6-3 se presenta una serie de posibles sucesos con mayor relevancia observado durante las pruebas mostradas en la Figura 10-3, se marca como "X" a la ausencia de vehículos en el sector y con "✓" los sectores en los que detecta la presencia de vehículos y que el accionar se rigen al valor del ángulo emitido por el algoritmo.

Tabla 6-3: Presencia de vehículos en diferentes sectores

Sector / Ítem	S1	S2	S3	S4	S5	Lógica difusa	Visibilidad del entorno
a	X	X	X	X	X	No	Buena
b	X	✓	✓	✓	X	Si	Buena
c	X	X	✓	✓	✓	Si	Buena
d	✓	✓	✓	X	X	Si	Buena
e	✓	✓	✓	✓	✓	Si	Mala
f	✓	✓	X	✓	✓	Si	Buena

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

3.4.2 Vista frontal de la iluminación del entorno

Con los datos expuestos en la tabla anterior, se verifica el encandilamiento que será recibido por una persona que se encuentre en la misma vía con la vista de frente hacia la barra LED montada en el vehículo.

Tabla 7-3: Apertura de ángulos en cada sector – Vista Frontal

Sector / Ítem	S1	S2	S3	S4	S5	Lógica difusa	Encandila
a	X	X	X	X	X	No	Si
b	X	✓	✓	✓	X	Si	Si
c	X	X	✓	✓	✓	Si	Si
d	✓	✓	✓	X	X	Si	No
e	✓	✓	✓	✓	✓	Si	No
f	✓	✓	X	✓	✓	Si	Si

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la Figura 11-3 se observa el encandilamiento que presenta la barra hacia los demás conductores, en el primer caso es cuando no se detecta ningún vehículo por lo que los sectores se establecen en 90° y se aprecia un encandilamiento alto, para el segundo caso existe presencia de vehículos en los sectores centrales propiciando un encandilamiento medio hacia el conductor, de igual manera sucede en el tercer caso debido a que los sectores S1 y S2 son lo que más afectan al conductor que transita en sentido contrario y por ello el encandilamiento que reciben es medio.

En el cuarto y quinto caso el encandilamiento es bajo debido a que los sectores más propensos a encandilar se encuentran con presencia de vehículos, para el último caso presenta un encandilamiento medio para los conductores que se encuentren lejos, porque el sector medio no detecta vehículos.

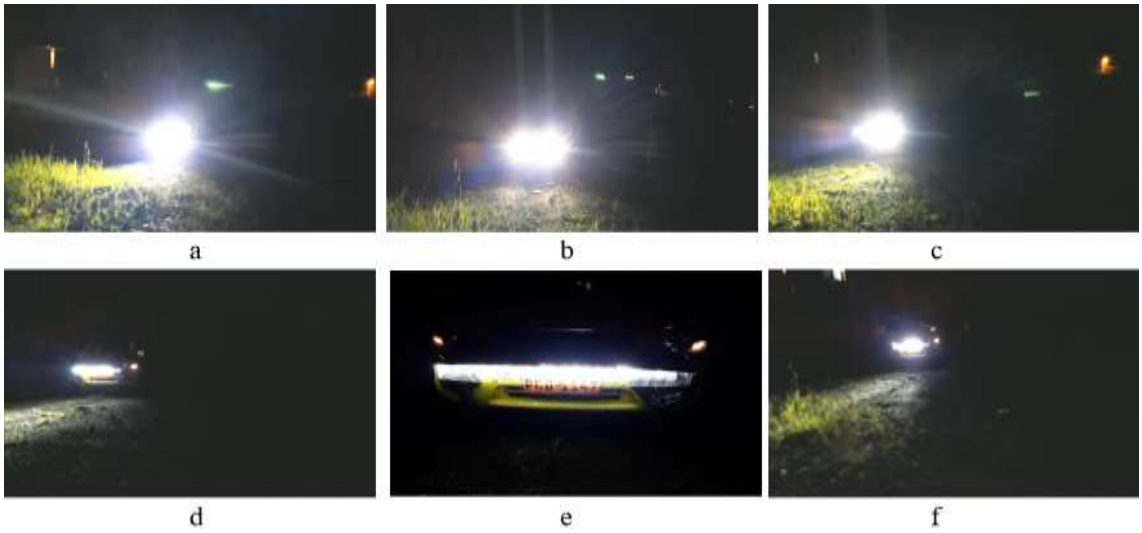


Figura 11-3: Vista Frontal de las pruebas de iluminación
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Los resultados presentados en esta sección no son los únicos que el sistema puede generar debido a que cada sector tiene 68 posibles valores de ángulos, de los cuáles dependerán de la posición en la que se encuentre el vehículo tanto en el eje X como en el eje Y, los ángulos varían entre los 11° y 79°.

3.5 Prueba de distancias en espacios controlados

Se realizaron pruebas de acuerdo a la distancia en la que puede detectar el clasificador en cascada las luces de los vehículos que llegue a ser captadas por la cámara, para ello se posicionó al sistema en un punto específico y partiendo de ahí se identificó referencias que se encuentren a 50, 100, 150 y 200 metros estas distancias fueron tomadas por una cinta métrica de 30m, como se muestra en la Figura 10-3, se toma como máximo los 200 metros debido a que ese es el valor límite en el que un conductor debe hacer el cambio de luces respectivos, según lo expuesto en el

REGLAMENTO GENERAL PARA LA APLICACIÓN DE LA LEY ORGÁNICA DE TRANSPORTE TERRESTRE, TRÁNSITO Y SEGURIDAD VIAL. Artículo 189.

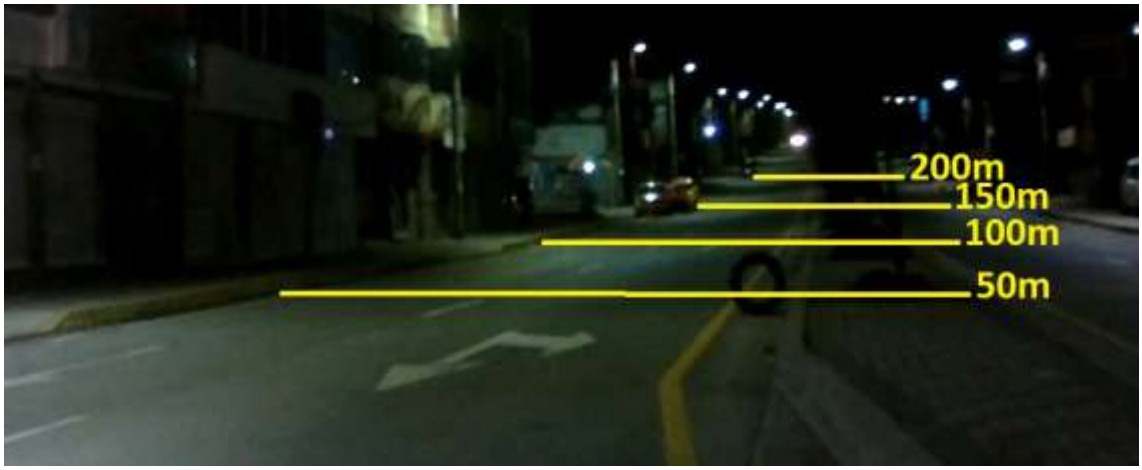


Figura 12-3: Distancias establecidas para las pruebas
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Se realizaron 3 pruebas para determinar el funcionamiento del clasificador elegido al momento de detectar vehículos que se encuentren a determinada distancia, cabe recalcar que cada una de las pruebas fueron realizadas en un mismo lugar de forma estática pero, la cantidad y velocidad de los vehículos circulantes son diferentes en cada una, en la primera prueba se presenció un total de 57 vehículos de los cuales fueron detectados en mayor parte cuando el vehículo se encontraba en el rango de los 150 metros, para validar el comportamiento del clasificador se realizaron dos pruebas adicionales presentando valores similares en la efectividad de detección de vehículos dentro del rango mencionado. Los datos obtenidos se muestran en la Tabla 8-3.

Tabla 8-3: Pruebas de detecciones a distancias

	Prueba 1		**Prueba 2**		**Prueba 3**	
Distancia (metros)	**Vehículos detectados**	**Vehículos no detectados**	**Vehículos detectados**	**Vehículos no detectados**	**Vehículos detectados**	**Vehículos no detectados**
50	56	1	35	0	46	2
100	54	3	33	1	40	8
150	50	7	26	9	38	10
200	24	33	20	15	30	18
Vehículos	57		35		48	
Total, Vehículos	140					

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De las pruebas realizadas se logra apreciar que para una distancia de 200 metros el sistema tiene mayor dificultad para detectar los vehículos, de la totalidad (140 unidades) el 47% no son detectados siendo un porcentaje por debajo de la mitad, estos datos pueden variar respecto a la calidad de imagen, velocidad de circulación o iluminación, ante lo expuesto anteriormente se puede afirmar que mayor efectividad presenta cuando los automotores entran en el rango de 150 metros, es decir, mientras más cercano se encuentran los objetos de interés, mayor número de detecciones tendrá el sistema prototipo.

En las Figuras 11-3, 12-3 y 13-3 se muestra las pruebas realizadas al detectar los vehículos a una distancia de 50, 100 y 150 metros respectivamente.

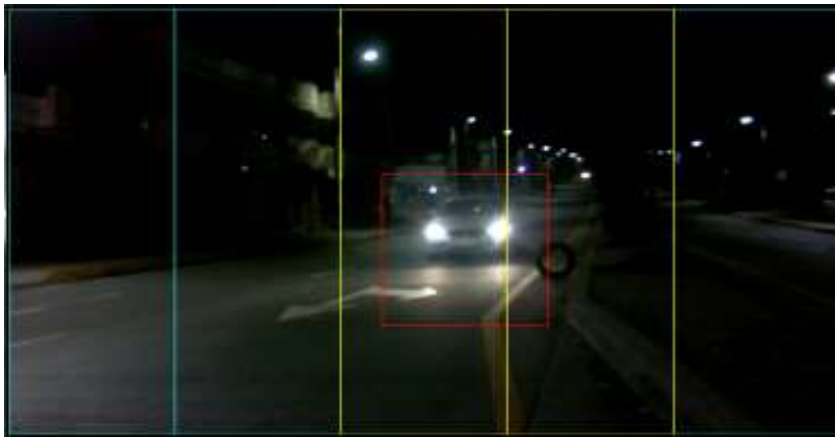


Figura 13-3: Vehículo detectado a 50 metros
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Figura 14-3: Vehículo detectado a 100 metros
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

Figura 15-3: Vehículo detectado a 150 metros
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

3.5 Pruebas de funcionamiento final del sistema prototipo en espacios no controlados

Las pruebas finales del sistema prototipo se las realizó en el trayecto de Riobamba-Ambato, en un ambiente despejado y sin quebrantar las leyes de tránsito circulando a la velocidad permitida y realizando maniobras seguras y permitidas, el tiempo que se demoró en llegar a la ciudad de Ambato fue de aproximadamente de 50 minutos, para el trayecto de ida se utilizó la cámara oficial de Raspberry de 1080p, y para el trayecto de regreso se cambió la cámara por una webcam con resolución de 640x480 pixeles de igual manera el tiempo de retorno es de 55 minutos, sumando un total de 1 hora con 45 minutos de funcionamiento del sistema prototipo en ambiente no controlado.

Figura 16-3: Trayecto recorrido para las pruebas finales
Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

En la siguiente Tabla 9-3 se muestra los datos recopilados durante la trayectoria hacia la ciudad de Ambato, los datos tabulados son de acuerdo a la velocidad en la que se circulaba arbitrariamente en diferentes tramos de la vía.

Tabla 9-3: Datos de vehículos detectados en espacios no controlados

Velocidad (km/h)	Vehículos en sentido contrario		Vehículos en el mismo sentido	
	Detectados	No detectados	Detectado	No detectados
50	18	3	7	2
55	17	6	5	3
65	20	5	9	8
70	12	11	12	6
75	12	5	8	13
80	10	16	6	8
90	6	14	5	4

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De los vehículos que circulaban en sentido contrario se lograron observar en la carretera un total de aproximadamente 155 vehículos, de los cuáles el 61% de su totalidad fueron detectados, la mayor parte de esos se detectaron cuando se circulaba a una velocidad moderada, mientras que a mayor velocidad se reducía el número de vehículos detectados debido a que el tiempo de respuesta del sistema no es lo suficientemente rápido para actuar ante este tipo de situaciones, los conductores que transiten en sentido contrario generalmente circulan a velocidades diferentes a la nuestra (por lo general mayores) y es un factor que tiene un impacto negativo en el sistema, aumentando el número de vehículos no detectados.

Para los vehículos que circulan en el mismo sentido es posible notar que las velocidades de circulación tras otro automotor son similares entre sí, se puede decir que la diferencia de velocidades son muy pequeñas, se llegaron a observar un total de 96 vehículos, los cuáles fueron detectados 52 vehículos pertenecientes al 54% de su totalidad, en este caso, el tiempo de respuesta del sistema se considera optimo, por estas razones la cantidad de vehículos detectados depende más del entrenamiento previo para obtener mayor efectividad en los resultados, considerando que se evitó en todo momento maniobras de adelantamientos.

En el retorno hacia la ciudad de Riobamba, se pone en marchar el sistema, pero ahora con un computador “Lenovo – G40-80” y una cámara web “LOGITECH C910” de 1920x1080 pixeles en comparación a la cámara de Raspberry usada en el presente trabajo. A continuación, se muestra

en la Tabla 10-3 los datos recopilados durante la nueva trayectoria, del mismo modo estos valores son obtenidos considerando los mismos parámetros de la Tabla 9-3.

Tabla 10-3: Datos de vehículos detectados en el trayecto de retorno

Velocidad (km/h)	Vehículos en sentido contrario		Vehículos en el mismo sentido	
	Detectados	**No detectados**	**Detectado**	**No detectados**
50	22	4	6	3
55	19	4	8	1
65	7	0	3	1
70	15	7	7	2
75	17	11	8	5
80	8	10	9	4
90	10	12	3	0

Realizado por: Coyago, Jhinson & Coyago, Jonathan. 2019

De los vehículos que circulaban en sentido contrario se lograron observar en la carretera un total de aproximadamente 146 vehículos, el 67% fueron detectados, en comparación al 61% de efectividad de la Raspberry, el computador "Lenovo – G40-80" presenta mejores resultados.

Los vehículos que circulan en el mismo sentido se registra un total de 60 vehículos, los cuáles se captaron positivamente 44 unidades que conforma el 73% de su totalidad, nuevamente el computador nos refleja mejores resultados en comparación a la Raspberry.

CONCLUSIONES

- De acuerdo a la investigación realizada se experimentó con varios tipos de algoritmos que permiten el reconocimiento de objetos, se eligió Haar Cascade porque permite desarrollar un entrenamiento clasificador de aprendizaje supervisado, destinado a detectar los patrones de faros vehiculares circulantes durante la noche, con el entrenamiento que se obtuvo el 95% de efectividad en las detecciones se procedió a realizar pruebas en espacios no controlados.

- Un método sencillo que permitió la segmentación de luz, fue obstruir el paso de iluminación que provee un faro como fuente lumínica, al realizar esta acción se aprecia de inmediato el cambio de visibilidad del entorno, se optó por una barra LED debido a su facilidad para añadir el mecanismo tipo persiana.

- En el desarrollo del algoritmo en Python usando OpenCV como librería de visión artificial, se debe determinar las variables de entradas y salidas que nos ayudó a establecer las reglas de lógica difusa para obtener el ángulo de salida en cada actuador y así el haz de luz emitido por la barra LED sea obstruido disminuyendo el encandilamiento hacia los demás conductores.

- La comunicación de todos los componentes depende del circuito electrónico diseñado para reducir el cableado excesivo de conexiones que permitió centralizar en un solo modulo el control de todo el sistema, está conformada por elementos que son fácil de encontrar en el mercado permitiendo la facilidad de cambiar el componente si deja de funcionar alguno de ellos.

- La implementación del sistema prototipo al estar conectados todos los elementos muestra un retardo de aproximadamente 670 milisegundos desde la adquisición de imagen hasta el accionamiento del servomotor correspondiente, el tiempo determinado varía de acuerdo a las tareas impuestas especialmente en el algoritmo, es decir, tarda menos si se evita mostrar imágenes en tiempo real.

- Las condiciones climatológicas y velocidades de circulación afectaron directamente el comportamiento del prototipo durante las pruebas realizadas, debido al cambio drástico en el campo visual de la cámara por ende las detecciones vehiculares disminuyeron y los actuadores no respondían correctamente en el rango de los 200 metros.

RECOMENDACIONES

- Para obtener un archivo Haar Cascade que permita la detección de objetos específicos con mayor calidad y aceptabilidad, en este trabajo los faros de vehículos circulantes en las noches, se debe aumentar el número de muestras positivas y negativas, además no se debe repetir o extraer sub-muestras de una misma imagen.

- Verificar el mecanismo de las persianas que se encuentren en buenas condiciones, es decir, que la movilidad de las persianas sea suave y ágil antes de poner en marcha el sistema, es necesario mantener el acrílico frontal que sirve de rompe vientos para que el trabajo del mecanismo no sea forzado provocando daños en los servomotores.

- Para trabajar adecuadamente con la librería OpenCV es necesario conocer los requerimientos de cada algoritmo, en este trabajo se usa el Haar Cascade para el reconocimiento de objetos por lo cual se debe instalar la versión 3.2 o superior de dicha librería.

- Para mejorar el tiempo de respuesta del algoritmo que gobierna el prototipo se debe optar por un ordenador más eficiente con el fin de cubrir los recursos computacionales necesarios para procesar el algoritmo destinado a la detección de vehículos y el accionamiento de los actuadores para segmentar el haz de luz. Por otro lado, se sugiere realizar pruebas reemplazando las Raspberry Pi 3 por una Raspberry Pi 4 b+ lanzada este año.

- Se puede reducir el uso de consumo computacional suspendiendo la visualización de la adquisición de imágenes en tiempo real para dejar de usar la pantalla, de tal manera que el modulo central se lo pueda rediseñar para que sea de menor tamaño y maniobrable.

- Para verificar el comportamiento del sistema en mejores condiciones, es necesario realizar varias pruebas en espacios reales seguros y controlados con dos o más vehículos disponibles para realizar maniobras que se puedan repetir, permitiendo recopilar imágenes con la cámara debidamente instalada que distribuya equitativamente el entorno que servirán para futuros entrenamientos.

GLOSARIO

Ababoost: Es un meta-algoritmo de aprendizaje automático supervisado.

Camshift: Es un algoritmo de segmentación de imagen en color.

Fuzzy Logic: Lógica difusa o lógica borrosa.

Javascript: Es un lenguaje de programación orientada a objetos.

Kriptón: Es un elemento químico de la tabla periódica cuyo símbolo es Kr y su número atómico es 36

Meanshift: Es una técnica de análisis de espacio de características no paramétrico para localizar los máximos de una función de densidad

Policarbonato: Es un grupo de termoplásticos, fácil de trabajar, moldear y termoformar, y es utilizado ampliamente en la manufactura moderna.

Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible

Wolframio: Es un elemento químico de número atómico 74 que se encuentra en el grupo 6 de la tabla periódica de los elementos.

Xenón: Es un elemento químico de la tabla periódica cuyo símbolo es Xe y su número atómico el 54.

Yuxtaposición: Es un procedimiento que refiere a la acción de yuxtaponer que implica la colocación de algo junto a otra cosa.

BIBLIOGRAFÍA

ALONSO, R; et al. *Análisis: Raspberry Pi 3 Modelo B+*. [en línea] Raspberry Pi 3 Model B+. 2018 [Consulta: 15 marzo 2019]. Disponible en: https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/

BERGER, W. *Deep Learning Haar Cascade*. [en línea] Haar cascade. 2018 [Consulta: 16 marzo 2019]. Disponible en: http://www.willberger.org/cascade-haar-explained/

CASTILLO, Y. & YANCHA, M. Implementación de un prototipo cortacésped inteligente, aplicando técnicas de visión artificial direccionando a escenarios deportivos [en línea] (Trabajo de titulación) (Pregrado) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Control y Redes Industriales. Ecuador, Chimborazo. 2017 pp. 23-25 [Consulta: 22 enero 2019]. Disponible en: http://dspace.espoch.edu.ec/bitstream/123456789/8964/1/108T0224.pdf

ETI. *Visión Artificial* [en línea] Conceptos Generales. 2010 [Consulta: 6 diciembre 2018] Disponible en: http://www.etitudela.com/celula/downloads/visionartificial.pdf

FIDALGO, R. *Tipos de faros en el automóvil* [en línea] Autocasión. 2013 [Consulta: 15 marzo 2019]. Disponible en: https://www.autocasion.com/actualidad/reportajes/tipos-de-faros-en-el-coche.

FLORES, P. & BRAUN, J. *Algoritmo SIFT: fundamento teórico* [en línea] Detección de extremos en el espacio-escala. 2011 [Consulta: 26 febrero 2019]. Disponible en: http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/FundamentoSIFT.pdf

GARCÍA, J. *¿Qué es Arduino?* [en línea] Arduino. 2018 [Consulta: 18 marzo 2019]. Disponible en: https://www.hwlibre.com/que-es-arduino/

GARZÓN, P. & SOLA, J. Diseño e implementación de un robot móvil provisto de visión artificial para reconocimiento de objetos y posicionamiento del móvil a los objetos [en línea] (Trabajo de titulación) (Pregrado) Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica. Ecuador, Quito. 2016 pp 1-11 [Consulta: 7 diciembre 2018] Disponible en: https://bibdigital.epn.edu.ec/bitstream/15000/15058/1/CD-6871.pdf

HERNÁNDEZ, J. Aplicación de la lógica difusa para detección de defectos en rodamientos [en línea] (Trabajo de titulación) (Pregrado) Universidad Carlos III De Madrid, Departamento de Ingeniería Mecánica. España, Madrid. 2010 [Consulta: 16 marzo 2019]. Disponible en: https://core.ac.uk/download/pdf/30043369.pdf

IBAÑEZ, P. *Sistemas de iluminación avanzados en coches* [en línea] Tipos de lámparas o emisores de luz. 2011 [Consulta: 14 marzo 2019]. Disponible en: https://www.xataka.com/automovil/sistemas-de-iluminacion-avanzados-en-coches

INSTITUTO ECUATORIANO DE NORMALIZACIÓN. *Vehículos automotores. Dispositivos para mantener o mejorar la visibilidad* [en línea] 2009. [Consulta: 7 mayo 2019] Disponible en: http://canfacecuador.com/normas/1155.pdf

JIMÉNEZ, J. *El sistema de alumbrado del vehículo* [en línea] El sistema de alumbrado permite ejercer la conducción con seguridad al aportar la iluminación necesaria para ver y ser vistos con claridad. 2018 [Consulta: 14 marzo 2019]. Disponible en: https://www.ro-des.com/mecanica/sistema-alumbrado-del-coche-que-es/

JULIO. *La imagen digital* [en línea] 2012. [Consulta: 8 mayo 2019]. Disponible en: http://cuartoinformatica.tecnojulio.com/wp-content/uploads/2012/02/Tema3.-Imagen_Digital.pdf

MONTOYA, D. & PACHACAMA, A. Diseño e Implementación de un Prototipo de Detección de Carril Mediante Visión Artificial [en línea] (Trabajo de titulación) (Pregrado) Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica. Ecuador, Quito. 2016 pp 5-15 https://bibdigital.epn.edu.ec/bitstream/15000/16751/1/CD-7347.pdf

OLMO, Á. Tutorial de Introducción de Lógica Borrosa [en línea] Aplicaciones de la Lógica Borrosa 2008. [Consulta: 16 marzo 2019]. Disponible en: http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/tutorial_fuzzy/introduccion3.html

PONCE, P. *Inteligencia Artificial. Con Aplicaciones a la Ingeniería* [en línea] Inteligencia Artificial. 2010 [Consulta: 16 marzo 2019]. Disponible en: https://lelinopontes.files.wordpress.com/2014/09/inteligencia-artificial-con-aplicaciones-a-la-ingenierc3ada.pdf

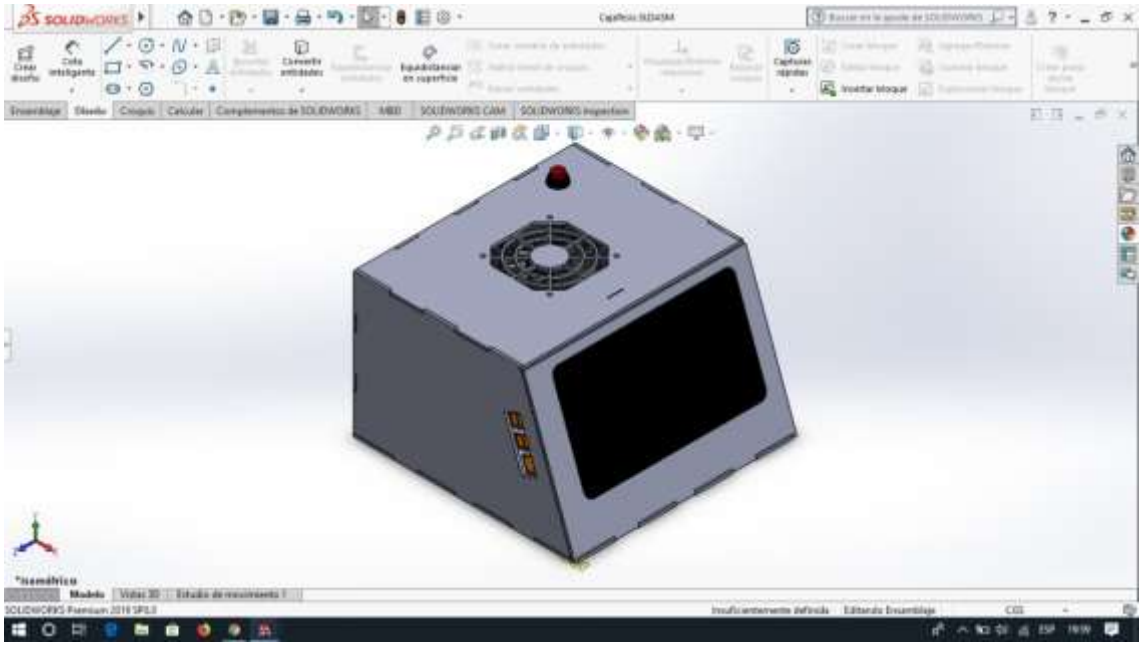
REBAZA, J. *Detección de bordes mediante el algoritmo de Canny* [en línea] Criterios del Algoritmo de Canny. 2014 [Consulta: 2 marzo 2019]. Disponible en: https://scholar.google.com/scholar?oi=bibs&cluster=4616170244461051737&btnI=1&hl=es

VELEZ, J. et al. *Vision por Computador* [en línea] 2003. [Consulta: 6 diciembre 2018]. Disponible en: http://www.visionporcomputador.es/libroVision/VisionPorComputador.pdf

YUBAL, F. *Qué es Arduino, cómo funciona y qué puedes hacer con uno* [en línea] Qué es Arduino. 2018 [Consulta: 18 marzo 2019]. Disponible en: https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno

ANEXOS

Anexo A Diseño del sistema prototipo en SolidWorks



Anexo B Diseño de la Interfaz Gráfica

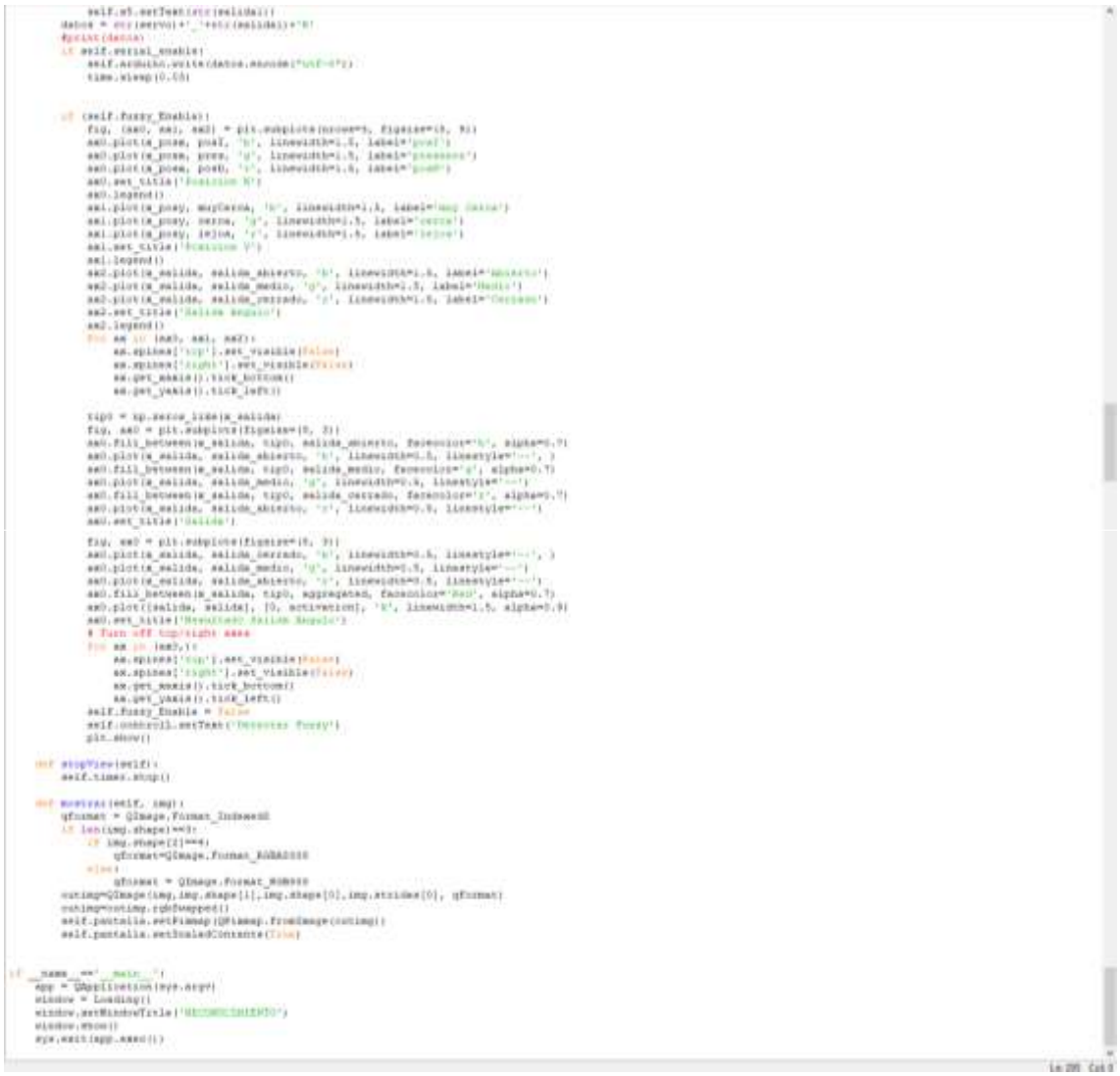
Anexo C Programación Sistema prototipo

```
"Deteccion_Fuzzy_Final - copia.py - C:\Users\CASA\Desktop\OpenCV-Dashcam-Car-Detection-master\Deteccion_Fuzzy_Final - copia.py (3.7.2)"
File  Edit  Format  Run  Options  Window  Help
import sys
import cv2
import math
import serial, time
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt
from PyQt5.QtCore import QTimer
from PyQt5.QtGui import QImage, QPixmap
from PyQt5.QtWidgets import QApplication, QMainWindow
from PyQt5.uic import loadUi

class Looking(QMainWindow):
    def __init__(self):
        super(Looking, self).__init__()
        loadUi('guiCar.ui', self)
        self.image=None
        self.imageFro=None
        self.startVideo.clicked.connect(self.startView)
        self.stopVideo.clicked.connect(self.stopView)
        self.detectar.toggled.connect(self.detectarView)
        self.detectar.setCheckable(True)
        self.detectar_Enable=False
        self.controll.toggled.connect(self.activarFuzzy)
        self.controll.setCheckable(True)
        self.Fuzzy_Enable = False
        self.conectar.toggled.connect(self.connect)
        self.conectar.setCheckable(True)
        self.serial_enable = False
        self.desconectar.clicked.connect(self.discon)
        self.CarCascade = cv2.CascadeClassifier('recursos_dir/cars2.xml')


    def detectarView(self, status):
        if status:
            self.detectar.setText("Detenar")
            self.detectar_Enable=True
        else:
            self.detectar.setText("Detectar")
            self.detectar_Enable=False
            if self.serial_enable == True:
                datos='b'
                self.arduino.write(datos.encode("utf-8"))

    def connect(self):
        self.serial_enable = True
        self.arduino = serial.Serial(self.puerto.toPlainText(), 115200, timeout=0.1)
    def discon(self):
        if self.serial_enable == True:
                datos='b'
                self.arduino.write(datos.encode("utf-8"))
        self.serial_enable=False
        self.arduino.close()

    def activarFuzzy(self, status):
        if status:
            self.controll.setText("Detener Fuzzy")
            self.Fuzzy_Enable=True
        else:
            self.Fuzzy_Enable=False
            self.controll.setText("Detectar Fuzzy")

    def startView(self):
        self.contador=0
        self.captura = cv2.VideoCapture(0)
        self.timer=QTimer(self)
        self.timer.timeout.connect(self.actualizar)
        self.timer.start(1)

    def actualizar(self):
        ret,self.image=self.captura.read()
        if (self.detectar_Enable):
            detected=self.detect_car(self.image)
            self.mostrar(detected)
        else:
            self.mostrar(self.image)
            self.n1.setText(str(90.0))
            self.n2.setText(str(90.0))
            self.n3.setText(str(90.0))
            self.n4.setText(str(90.0))
            self.n5.setText(str(90.0))

    def detect_car(self, img):
        gray=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        cars = self.CarCascade.detectMultiScale(gray, 4.0, 5)
        ancho = int(self.captura.get(cv2.CAP_PROP_FRAME_WIDTH))
        alto = int(self.captura.get(cv2.CAP_PROP_FRAME_HEIGHT))


        a = math.ceil(ancho * (0.2))
        b = math.ceil(ancho * (0.4))
        c = math.ceil(ancho * (0.6))
        d = math.ceil(ancho * (0.8))
        e = math.ceil(ancho * (1))
        img = cv2.rectangle(img, (0, 0), (a, alto), (255, 255, 0), 2)
        img = cv2.rectangle(img, (0, 0), (b, alto), (255, 255, 0), 2)
        img = cv2.rectangle(img, (0, 0), (c, alto), (255, 255, 0), 2)
        img = cv2.rectangle(img, (0, 0), (d, alto), (255, 255, 0), 2)
        img = cv2.rectangle(img, (0, 0), (e, alto), (255, 255, 0), 2)
        enable=False

        for (x, y, w, h) in cars:
            enable=True
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)

            if ((x + w) >= 0 and (x + w)<0.2) < a):
                cv2.rectangle(img, (0, 0), (a, alto), (0, 255, 255), 2)
                self.logicaFuzzy(0, a, (x+w), y,1)
            else:
                self.contador=0
                self.n1.setText(str(45.0))
                if self.serial_enable == True:
                    datos = str(1)+','+str(95.0)+'S'
                    self.arduino.write(datos.encode("utf-8"))

            if ((x + w) >= a and (x + w)<0.4) < b):
                cv2.rectangle(img, (a, 0), (b, alto), (0, 255, 255), 2)
                self.logicaFuzzy(a, b, (x+w), y,2)
            else:
                self.contador=0
                self.n2.setText(str(75.0))
                if self.serial_enable == True:
                    datos = str(2)+','+str(75.0)+'S'
                    self.arduino.write(datos.encode("utf-8"))
```

```
            if (ix + w) >= b and (x + w*0.5) < c:
                cv2.rectangle(img, (b, 0), (c, alto), (0, 255, 255), 2)
                self.logicaFuzzy(b, c, (x+w), y,3)
            else:
                self.contador=0
                self.a3.setText(str(79.0))
                if self.serial_enable is True:
                    datos = str(3)+'_'+str(79.0)+'%'
                    self.arduino.write(datos.encode("utf-8"))

            if (ix + w) >= c and (x + w*0.5) < d:
                cv2.rectangle(img, (c, 0), (d, alto), (0, 255, 255), 2)
                self.logicaFuzzy(c, d, (x+w), y,4)
            else:
                self.contador=0
                self.a4.setText(str(79.0))
                if self.serial_enable is True:
                    datos = str(4)+'_'+str(79.0)+'%'
                    self.arduino.write(datos.encode("utf-8"))

            if (ix + w) >= d and (x + w*0.5) < e:
                cv2.rectangle(img, (d, 0), (e, alto), (0, 255, 255), 2)
                self.logicaFuzzy(d, e, (x+w), y,5)
            else:
                self.contador=0
                self.a5.setText(str(79.0))
                if self.serial_enable is True:
                    datos = str(5)+'_'+str(79.0)+'%'
                    self.arduino.write(datos.encode("utf-8"))



        if self.serial_enable is True and enable is False:
            self.contador=self.contador+1
            if self.contador>=15:
                datos='A'
                self.arduino.write(datos.encode("utf-8"))
                self.a1.setText(str(90.0))
                self.a2.setText(str(90.0))
                self.a3.setText(str(90.0))
                self.a4.setText(str(90.0))
                self.a5.setText(str(90.0))
```

```
        if self.serial_enable is True and self.detector_Enable is True:
            datos='I'
            self.arduino.write(datos.encode("utf-8"))

        if self.serial_enable is False and enable is False:
            self.contador=self.contador+1
            if self.contador>=15:
                self.a1.setText(str(90.0))
                self.a2.setText(str(90.0))
                self.a3.setText(str(90.0))
                self.a4.setText(str(90.0))
                self.a5.setText(str(90.0))

        return img

    def logicaFuzzy(self, a, b, posicionX, posicionY,ancho):
        alto = int(self.captura.get(cv2.CAP_PROP_FRAME_HEIGHT))
        ancho = int(self.captura.get(cv2.CAP_PROP_FRAME_WIDTH))
        angulo = 90

        x_pose = np.arange(0, ancho+1, 1)
        x_posy = np.arange(0, alto+1, 1)
        x_salida = np.arange(0, angulo+1, 1)


        poscl = fuzz.trapmf(x_pose, [0, 0, (0.2 * ancho), (0.4 * ancho)])
        pose = fuzz.trapmf(x_pose, [(0.2 * ancho), (0.4 * ancho), (0.6 * ancho), (0.8 * ancho)])
        poscD = fuzz.trapmf(x_pose, [(0.6 * ancho), (0.8 * ancho), ancho, ancho])


        lejos = fuzz.trapmf(x_posy, [0, 0, 0.1 * alto, 0.2 * alto])
        cerca = fuzz.trapmf(x_posy, [0.1 * alto, 0.2 * alto, 0.3 * alto, 0.4 * alto])
        muyCerca = fuzz.trapmf(x_posy, [0.3 * alto, 0.6 * alto, alto, alto])

        salida_cerrado= fuzz.trapmf(x_salida, [0, 0, 0.167 * angulo, 0.333 * angulo])
        salida_medio = fuzz.trapmf(x_salida, [0.167 * angulo, 0.333 * angulo, 0.667 * angulo, 0.833 * angulo])
        salida_abierto = fuzz.trapmf(x_salida, [0.667 * angulo, 0.833 * angulo, angulo, angulo])


        data_poscl = fuzz.interp_membership(x_pose, poscl, posicionX)
        data_pose = fuzz.interp_membership(x_pose, pose, posicionX)
        data_poscD = fuzz.interp_membership(x_pose, poscD, posicionX)
```

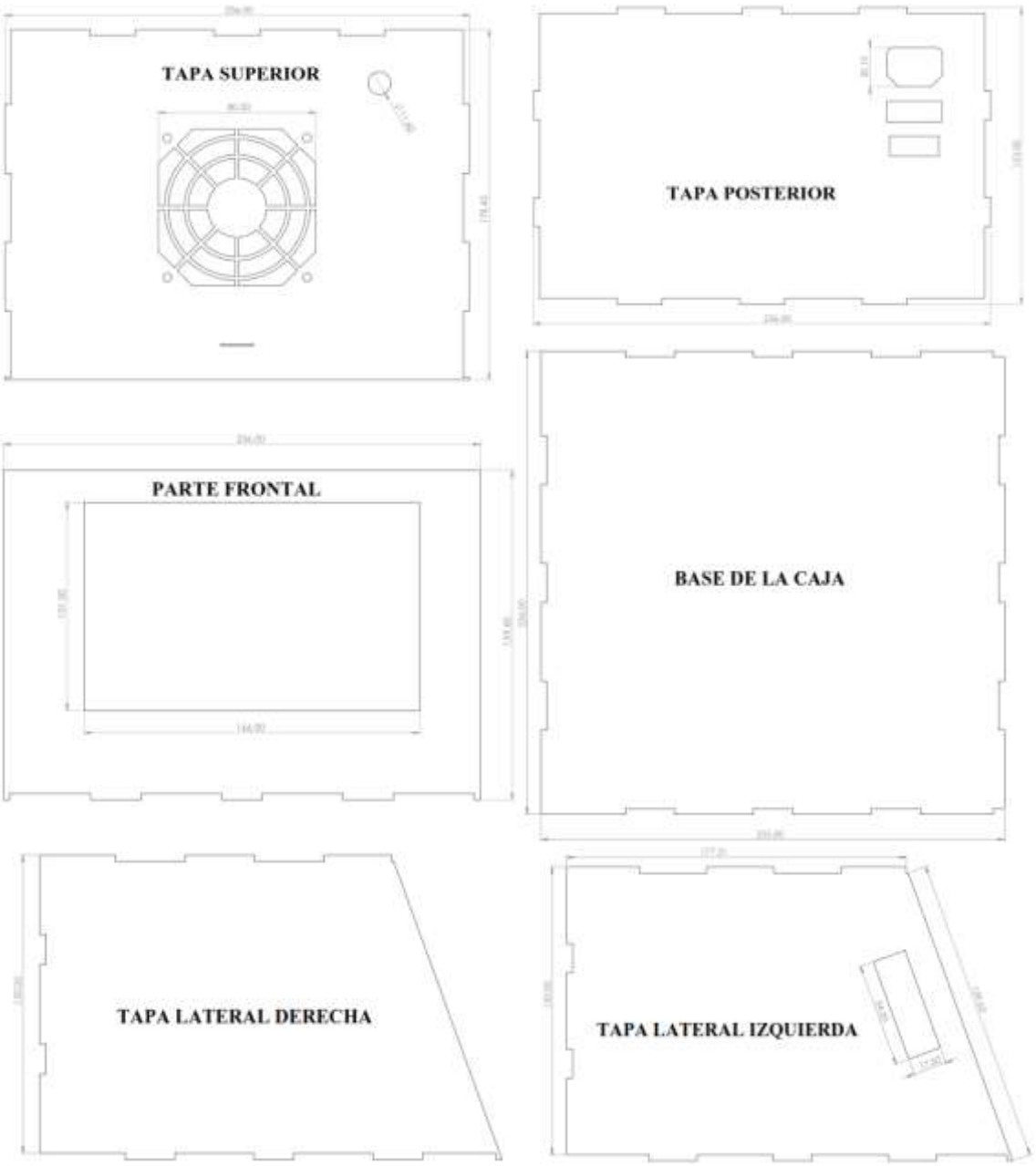
```
        data_muycerca = fuzz.interp_membership(x_posy, muyCerca, posicionY)
        data_cerca = fuzz.interp_membership(x_posy, cerca, posicionY)
        data_lejos = fuzz.interp_membership(x_posy, lejos, posicionY)
        # ============================reglas rules
        active_rule1 = np.fmin(data_muycerca, data_poscl)
        active_rule2 = np.fmin(data_muycerca, data_pose)
        active_rule3 = np.fmin(data_muycerca, data_poscD)
        active_rule4 = np.fmin(data_cerca, data_poscl)
        active_rule5 = np.fmin(data_cerca, data_pose)
        active_rule6 = np.fmin(data_cerca, data_poscD)
        active_rule7 = np.fmin(data_lejos, data_poscl)
        active_rule8 = np.fmin(data_lejos, data_pose)
        active_rule9 = np.fmin(data_lejos, data_poscD)
        rule1 = np.fmin(active_rule1, salida_cerrado)
        rule2 = np.fmin(active_rule2, salida_cerrado)
        rule3 = np.fmin(active_rule3, salida_cerrado)
        rule4 = np.fmin(active_rule4, salida_cerrado)
        rule5 = np.fmin(active_rule5, salida_medio)
        rule6 = np.fmin(active_rule6, salida_medio)
        rule7 = np.fmin(active_rule7, salida_medio)
        rule8 = np.fmin(active_rule8, salida_abierto)
        rule9 = np.fmin(active_rule9, salida_abierto)
        aggregated = np.fmax(rule1,
                             np.fmax(rule2,
                                     np.fmax(rule3,
                                             np.fmax(rule4,
                                                     np.fmax(rule5,
                                                             np.fmax(rule6,
                                                                     np.fmax(rule7,
                                                                             np.fmax(rule8, rule9)))))))
        salida = fuzz.defuzz(x_salida, aggregated, 'centroid')
        activation = fuzz.interp_membership(x_salida, aggregated, salida)  # for plot
        salida1=round(salida)
        if aa==1:
            self.a1.setText(str(salida))
        if aa== 2:
            self.a2.setText(str(salida))
        if aa==3:
            self.a3.setText(str(salida))
        if aa==4:
            self.a4.setText(str(salida))
        if aa==5:
```



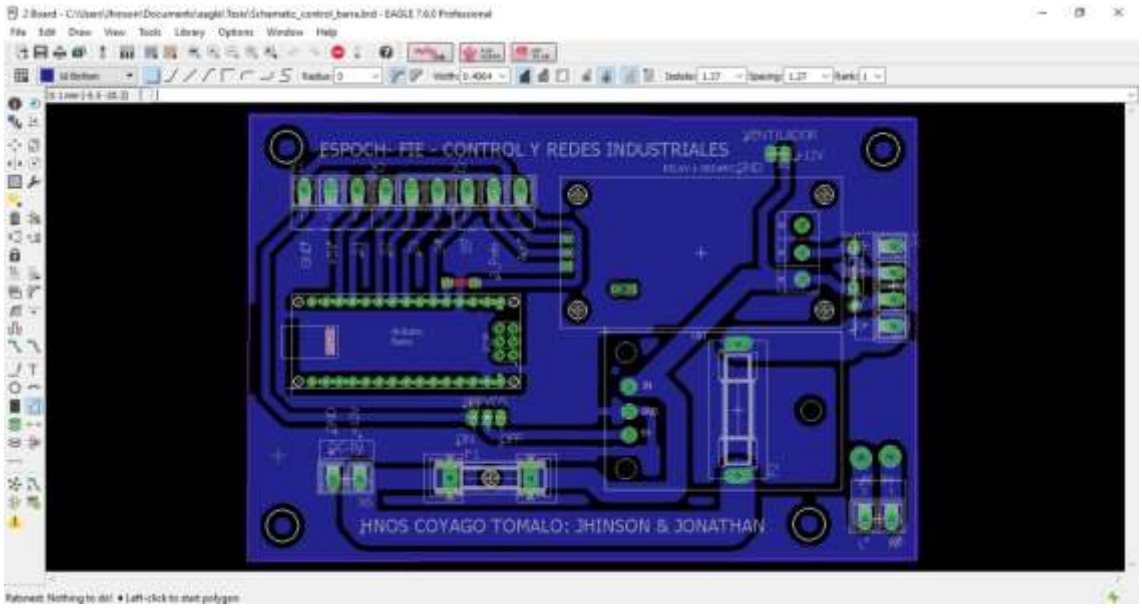
Anexo D Programación Arduino para los actuadores

```
void loop()
{
  if(Serial.available())
  {
    dato= Serial.read();
    Datos=Datos+d;
    Serial.println(dato);

  if(Datos=="7") //encender la barra led
  {
    digitalWrite(rele,LOW);
    digitalWrite(LED,HIGH);
    Datos="";
  }

  if(Datos=="1") // apaga la barra led y vuelve las posiciones
  {
    digitalWrite(rele,HIGH);
    digitalWrite(LED,LOW);
s1.write(0);
s2.write(0);
s3.write(0);
s4.write(0);
s5.write(0);
    Datos="";
   }
  if(Datos=="A")
  {
     s1.write(90);
     s2.write(90);
     s3.write(90);
     s4.write(90);
     s5.write(90);
    Datos="";
  }else{
    int posicion1 = Datos.indexOf("E");
    if(posicion1!=-1)
    {
    int posicion = Datos.indexOf("_");
    String comando = Datos.substring(posicion+1);
    int angulo=comando.toInt();
    char m= Datos[0];
    Datos="";
    if(posicion!=-1)
    {
    switch (m)
    {
        case '1':
        s1.write(angulo);
        delay(2);
        break;
        case '2':
        s2.write(angulo);
        delay(2);
        break;
        case '3':
        s3.write(angulo);
        delay(2);
        break;
        case '4':
        s4.write(angulo);
        delay(2);
        break;
        case '5':
        s5.write(angulo);
        delay(2);
        break;

    default:
    break;
    }
    }
  }
  }
}
```

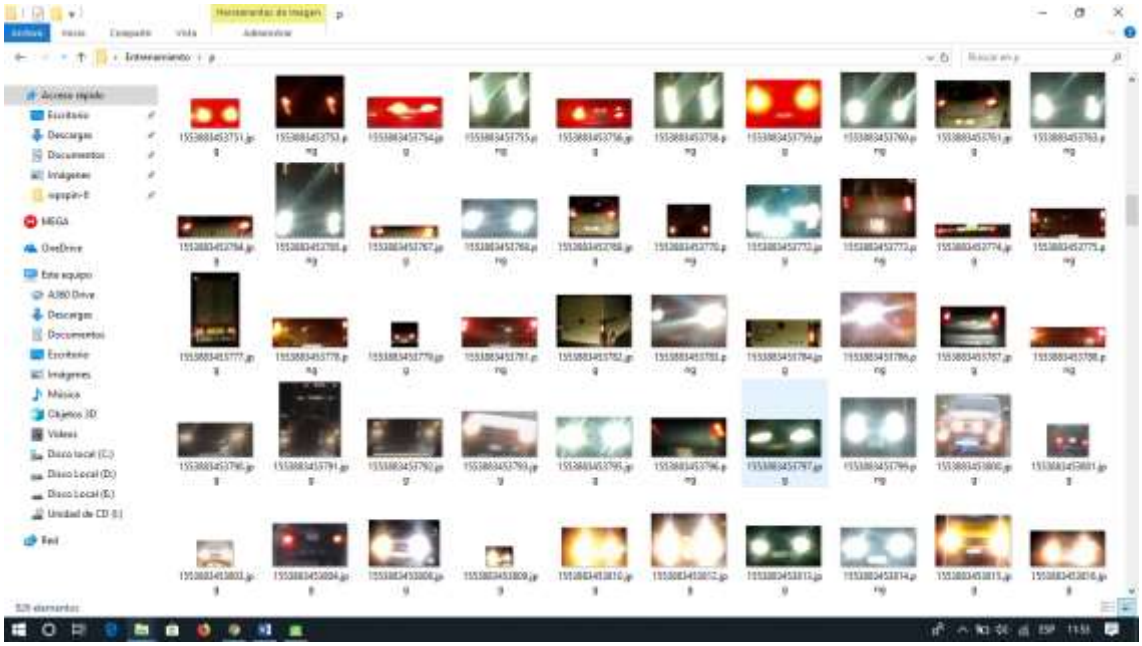
Anexo E Medidas de la caja prototipo



Anexo F Esquemático del circuito electrónico



Anexo G Muestras para el entrenamiento del sistema



Anexo H Prueba de funcionamiento en carreteras