

SISTEMA INTELIGENTE DE SEMAFORIZACION BASADO EN PROCESAMIENTO DE
IMÁGENES/VIDEO PROGRAMADO EN RASPBERRY.

DIEGO FERNANDO LONDOÑO RUBIO

DANIEL ALVAREZ ARISTIZABAL

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍA Y ARQUITECTURA

INGENIERÍA ELECTRÓNICA

MONTERÍA

2023

SISTEMA INTELIGENTE DE SEMAFORIZACION BASADO EN PROCESAMIENTO DE
IMÁGENES/VIDEO PROGRAMADO EN RASPBERRY.

DIEGO FERNANDO LONDOÑO RUBIO

DANIEL ÁLVAREZ ARISTIZÁBAL

Trabajo de grado para optar al título de
Ingeniero Electrónico

Asesor

ANA MILENA LÓPEZ LÓPEZ

Ingeniera Electrónica

UNIVERSIDAD PONTIFICIA BOLIVARIANA

ESCUELA DE INGENIERÍA Y ARQUITECTURA

INGENIERÍA ELECTRÓNICA

MONTERÍA

2023

CONTENIDO

1. INTRODUCCIÓN:	8
2.MARCO TEÓRICO/ESTADO DEL ARTE:	9
2.1. Marco teórico	9
2.1.1. Semáforos	9
2.1.2. Teoría general del funcionamiento de los semáforos	10
2.1.3. Clasificación de los semáforos	11
2.1.4. Aplicación de los semáforos inteligentes	12
2.1.5. Diferentes tecnologías de semaforización inteligente.....	12
2.2 Tipo de software y hardware	16
2.2.1. Software para el procesamiento de imágenes	16
2.2.2 Sistema embebido (Hardware)	16
2.2.3 Cámara.....	17
2.3 Arquitectura del software	17
2.3.1 Protocolos	17
2.3.2. Lenguaje de programación	18
2.3.3. Servidor web	18
2.3.4. Base de datos	19
2.4. Situación del tráfico vehicular en la ciudad de Montería.....	19
2.5. Sistema del semáforo	24
2.5.1. Enlace a la base de datos	24
2.5.2 Implementación del semáforo en la Raspberry	24
2.6 Parámetros viales.....	25
2.6.1 Zona de estudio.....	25
2.7. Estado del arte.....	27
3.METODOLOGIA	29
3.1 Selección del hardware.....	29
3.1.1. Sistema embebido.....	29

3.1.2. Cámara.....	32
3.2 Selección del software.....	34
3.3. Descripción de datos para la duración del semáforo	34
3.4 Sistema de alimentación.....	36
3.5. Instalación de Raspberry pi OS	42
3.5.1. Instalación de OpenCV, Python y MySQL en Raspberry	44
3.5.2. Instalación Python	44
3.5.3. Instalación OpenCV.....	44
3.5.3. Instalación de MySQL	45
3.6 Ejecución del programa.....	46
4.RESULTADOS Y DISCUSIÓN:	50
5.CONCLUSIONES	53
6.RECOMENDACIONES	55
6.BIBLIOGRAFIA	56

LISTA DE FIGURAS

Figura 1. Posición de lentes de un semáforo.

Figura 2. Esquema de semáforo inteligente usando RFID.

Figura 3. Esquema típico del sistema de semaforización inteligente usando sensores inalámbricos.

Figura 4. Diagrama de bloques.

Figura 5. Resultados del procedimiento considerando análisis de red.

Figura 6. Resultados intersección avenida circunvalar por carrera 14.

Figura 7. Resultados intersección calle 41 por carrera 14.

Figura 8. Resultados intersección calle 44 por carrera 14.

Figura 9. Funciones de los pines GPIO de las raspberri pi 4.

Figura 10. Intersección de la calle 44 con avenida circunvalar.

Figura 11. Tiempo de espera del semáforo en rojo.

Figura 12. Tiempo del semáforo verde.

Figura 13. Opciones del mercado.

Figura 14. Comparación de modelos de cámara.

Figura 15. Sistema de información fotovoltaica.

Figura 16. Tabla general de las horas pico solar por mes.

Figura 17. Panel solar del proyecto de energía solar.

Figura 18. Batería del proyecto de energía solar.

Figura 19. Regulador de carga.

Figura 20. Convertidor DC-DC Buck XL4015 de 5 A.

Figura 21. Memoria micro SD de la raspberry.

Figura 22. Interfaz raspberry pi image.

Figura 23. Selección del sistema operativo.

Figura 24. Terminal de comandos.

Figura 25. Instalación OpenCV en raspberry pi.

Figura 26. Instalación OpenCV en raspberrY pi.

Figura 27. Instalación base de datos MySQL en raspberri pi.

Figura 28. Terminal de comandos.

Figura 29. Ejecución del código.

Figura 30. Carpeta contenedora.

Figura 31. Programa ya cargado.

Figura 32. Toma de captura de imágenes.

Figura 33. Conteo de vehículos.

Figura 34. Conteo de vehículos en grabación hecha por un dron.

Figura 35. Interfaz de la base de datos con resultados que arroja el sistema.

Figura 36. Parte interna del proyecto.

Figura 37. Parte principal del sistema

RESUMEN

En este trabajo se presenta el diseño de un sistema de semaforización inteligente a partir del procesamiento de imágenes. Para comenzar con el diseño se procedió a identificar varias estrategias para el análisis y la obtención de datos al sistema, a partir de esto seleccionamos que programas serían los más útiles y de fácil acceso para el desarrollo de este. Teniendo esto como base se empezó a desarrollar el sistema en base a librerías y un código desarrollado en Python instalada dentro de la Raspberry pi 4 B+ también se optó trabajar con MYSQL, este un sistema de gestión de base de datos el cual sirve para almacenar todos los datos arrojados por el programa y verse reflejados por medio de un servidor web gratuito llamado Hostinger

Una vez establecidas estas herramientas se procedió con la elección y compra de la cámara la cual es de gran importancia para la detección de vehículos y el sistema de paneles solares el cual va a ayudar para la sostenibilidad del sistema. Con todas las partes integradas, se comenzó con las pruebas para corroborar la lectura, interpretación y entrega de datos desde la toma de fotografías hasta la salida de datos que da el sistema a través del servidor web.

PALABRAS CLAVES: SEMÁFORO INTELIGENTE, PROCESAMIENTO DE IMÁGENES, BASE DE DATOS

ABSTRACT:

In this work, the design of an intelligent traffic light system based on image processing is presented. To being with the design, we proceeded to identify several strategies for the analysis and collection of data to te system, from this we selected which programs were the most useful and easily accessible for its development. Taking this as a basis, the system began to be developed based on libraries and a code developed on the Python platform installed within the Raspberry pi 4 B+, it was also decided to work with MYSQL, this is a database management system which it is used to store all the data thrown by the program and be reflected by means of a free web server. Once these tools were established, we proceeded with the choice and purchase of the camera, which is of great importance for the detection of vehicles and the solar panel system, which will help us for the sustainability of the system. With all the parts integrated, the tests began to corroborate the reading, interpretation, and delivery of data from the taking of photographs to the data output provided by the system through the web server.

1. INTRODUCCIÓN:

En la ciudad de Montería, se ha notado en los últimos años un alto crecimiento demográfico, por ende, crecimiento en el parque automotor, lo cual ha generado gran congestión vehicular en la ciudad. Según (razon, 2017) En los últimos 10 años este ha tenido un incremento significativo de vehículos y motos de aproximadamente un 193% esto ha generado mayores retos en el control de la movilidad y el tráfico de la ciudad.

Un método para evitar ciertas congestiones es la implementación de un sistema de semáforos inteligentes ligada al internet de las cosas, ya que los semáforos que se utilizan actualmente en la ciudad no cuentan con la tecnología necesaria de poder contabilizar o llevar un registro de un número de autos, estas congestiones traen consigo un problema que afecta directamente a los diferentes sectores de la población específicamente como obstrucción de avenidas por largas filas, acumulación de vehículos, además, el congestionamiento de tráfico vehicular potencializa otros problemas como son: desperdicio de combustible, emisiones prolongadas de Co2 que puede afectar al medio ambiente como también puede afectar al deterioro de la salud de la población.

En los últimos años la implementación del internet de las cosas y la utilización de los semáforos inteligentes se ha convertido en la solución más eficaz para la mejora de la fluidez en las grandes ciudades del mundo, ya que en estas el uso de los semáforos que están basados en ciclos de reloj quedó obsoleta como herramienta para el control de tráfico, por esta razón se implementó un sistema de semaforización inteligente para el control de flujo vehicular mediante el procesamiento de imágenes, este se basa fundamentalmente en el uso de cámaras para capturar las zonas de interés a través de la programación desarrollada en Python identificar la cantidad de vehículos existentes y así asignar el tiempo al semáforo, también se implementó una base de datos donde primero que todo se guarda dicha información para después estos datos mandarlos a un servidor web y cualquier persona tenga la disponibilidad de entrar y ver en tiempo real como lo son el número de automóviles detectados, la hora, el lugar y el estado del semáforo.

Buscando como objetivos:

- Diseñar un sistema sensor que permita detectar la velocidad, la cantidad de vehículos y el flujo vehicular en cada uno de los semáforos en la zona establecida.
- Diseñar un sistema que permita controlar el paso de los vehículos en cada uno de los semáforos en la zona establecida.

- Implementar un sistema de alimentación basado en paneles solares para la autosostenibilidad del sistema.

2.MARCO TEÓRICO/ESTADO DEL ARTE

2.1. Marco teórico

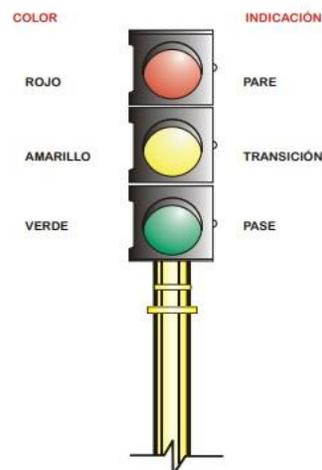
Hoy en día la congestión vial ha sido un tema que ha afectado a ciudades de todo el mundo principalmente a las capitales, estos sistemas con el tiempo han venido desarrollando diversos métodos de control que permiten la disminución de congestión vehicular como también de accidentes de tránsito. El control de tráfico está enfocado en detectar y organizar el flujo de vehículos que transitan por un mismo lugar con el objetivo de prevenir congestiones o accidentes. Hoy en día se pueden encontrar varias aplicaciones para dar una solución a este problema de movilidad.

2.1.1. Semáforos

Los semáforos son dispositivos de señalización mediante los cuales se regula la circulación de vehículos y peatones en las vías, asignando el derecho de paso de vehículos y peatones secuencialmente, por las indicaciones de luces de color rojo, amarillo y verde, operadas por un control de tráfico electrónico. (Gobernacion de medellin)

Figura 1.

Posición de lentes de un semáforo



Nota: adaptado de Semaforos capitulo 7 (pag 268), alcaldia de medellin,(s.f.), (https://www.medellin.gov.co/movilidad/documents/seccion_senalizacion/cap7_semaforos.pdf)

El semáforo ejerce gran influencia sobre el flujo del tránsito. Por lo tanto, es de vital importancia que la selección del punto de instalación del control semafórico, sea precedida de un estudio puntual y zonal de las condiciones del tránsito.

Los semáforos se usarán para desempeñar, entre otras, las siguientes funciones:

- Interrumpir periódicamente el tránsito de una corriente vehicular o peatonal para permitir el paso de otra corriente vehicular.
- Regular la velocidad de los vehículos para mantener la circulación continua a una velocidad constante.
- Controlar la circulación por carriles.
- Eliminar o reducir el número y gravedad de algunos tipos de accidentes, principalmente los que implican colisiones perpendiculares.
- Proporcionar ordenamiento del tránsito (Alcaldía de barranquilla, 2018).

2.1.2. Teoría general del funcionamiento de los semáforos

2.1.2.2. Semáforos de tiempos fijos o predeterminados

Un semáforo de tiempo fijo o predeterminado es un dispositivo para el control del tránsito que regula la circulación haciendo detener y proseguir el tránsito de acuerdo una programación de tiempo determinado o a una serie de programaciones establecidas. Las características de operación de los semáforos de tiempo fijo o predeterminado, tales como, duración del ciclo, intervalo, secuencia, desfase, etc. Pueden ser cambiadas de acuerdo un programa determinado. Los semáforos de control de tiempo fijo o predeterminado se adaptan mejor a las intersecciones en donde los patrones del tránsito son relativamente estables y constantes, o en donde las variaciones del tránsito que se registran pueden tener cabida mediante una programación pre-sincronizada sin causar demoras o congestión no razonables. El control pre-sincronizado es particularmente adaptable a intersecciones donde se desee coordinar la operación de semáforos con instalaciones existentes o planificadas en intersecciones cercanas en la misma calle o calles adyacentes o en intersecciones cuya capacidad vehicular esté en el límite (Semaforos, págs. 270-271).

Este tipo de semáforo se debe instalar y operar solamente si se satisfacen uno o más de los requisitos o condiciones siguientes:

Condición A: Volumen mínimo de vehículos que transitan por una intersección.

Condición B: Interrupción del tránsito continuo.

Condición C: Volumen mínimo de peatones.

Condición D: Movimiento o circulación progresiva.

Condición E: Antecedentes y experiencia sobre accidentes. (Semaforos, pág. 271)

2.1.2.3. Semáforos accionados por el tránsito

Un semáforo accionado por el tránsito es un sistema cuyo funcionamiento varía de acuerdo con las demandas del tránsito que registren los detectores de vehículos o peatones, los cuales suministran la información a un control local. Se usarán en las intersecciones donde los volúmenes de tránsito fluctúan considerablemente en forma irregular y en donde las interrupciones de la circulación deben ser mínimas en la dirección principal.

Los semáforos accionados por el tránsito se clasifican en tres categorías generales:

- **Semáforos totalmente accionados:** Disponen de medios para ser accionados por el tránsito en todos los accesos de la intersección.
- **Semáforos parcialmente accionados:** Disponen de medios para ser accionados por el tránsito en uno o más accesos de la intersección, pero no en todos.
- **Semáforos ajustados al tránsito:** Es un tipo de semáforo en el cual las características del despliegue de señales en los controladores locales para un área o para una calle, varían continuamente de acuerdo con la información sobre el flujo del tránsito suministrada a un computador maestro por detectores de muestreo ubicados en puntos de flujo típico en el área. (Semaforos, págs. 282-283)

2.1.3. Clasificación de los semáforos

2.1.3.1 Los semáforos peatonales

Los semáforos reservados para peatones podemos encontrarlos con una luz roja no intermitente en forma de peatón inmóvil. indica que estos no deben comenzar a cruzar la calzada.

Bajo la luz roja, se encuentra una luz verde no intermitente en forma de peatón en marcha. indica que pueden comenzar a atravesar la calzada. si es intermitente significa que el tiempo para cruzar está a punto de finalizar y que. En breve, se encenderá la luz roja (Race, 2020)

2.1.4. Aplicación de los semáforos inteligentes

(Anoroza, 2017, págs. 10-11) considera que algunas aplicaciones que se le pueden dar a los sistemas de semáforos inteligentes son:

- tomar decisiones propias para que sea más eficiente y fácilmente poder reemplazar a un policía de tránsito.
- Utilizan comunicación inalámbrica entre semáforos para optimizar el tráfico de vehículos.
- Semáforo con mando a distancia, que se active cuando una persona ciega valla a cruzar la calle y este a su vez emita un sonido para que esta persona pase.
- Comunicación con GPS para que las personas busquen rutas óptimas.
- Forzar semáforos en casos especiales. (Paso de ambulancias, policías, etc.)

Hoy en día el aumento del flujo vehicular hace que las autoridades competentes de la movilidad de tránsito estén tratando de crear nuevas ideas alternativas para dar una solución, por esta razón se han implementado semáforos inteligentes como lo ha mencionado (Anoroza, 2017, págs. 1-2) estas son capaces de tomar decisiones dependiendo de una serie de parámetros de entrada tales como:

- Flujo de vehículos
- Velocidad media
- Identificación de la calle, entre otros

Algunos problemas que se busca solucionar con este tipo de sistema serian:

- Congestión vehicular.
- Tiempos excesivos de viaje.
- Esperas innecesarias.
- Mayor gasto de combustible.
- Mayor contaminación en el medio ambiente.

2.1.5. Diferentes tecnologías de semaforización inteligente

Un sistema de semaforización inteligente está basado en aquel que por sí mismo puede tomar decisiones dependiendo de las variables que se quieran evaluar. Hoy en día existen diferentes sistemas de semaforización inteligente los cuales se enfocan en optimizar los tiempos del tráfico, minimizar los consumos de combustible de los vehículos como también mejorar la seguridad vial.

Existen diferentes sistemas de semaforización inteligente como lo son:

- Semáforo inteligente con RFID.
- Semáforo inteligente usando Redes de Sensores Inalámbricos.
- Semáforo inteligente mediante procesamiento de imágenes.

2.1.5.1. Semáforo inteligente con RFID

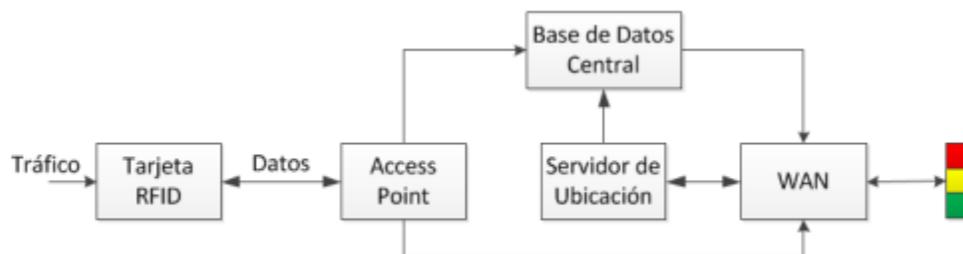
LA RFID es un sistema de identificación de productos que puede parecer similar al código de barras tradicional, pero cuenta con grandes ventajas. A diferencia del código de barras, que utiliza la imagen para identificar una etiqueta colocada en un producto, la RFID utiliza las ondas de radio para comunicarse con un microchip, que puede estar montado sobre gran cantidad de soportes, como por ejemplo un tag o etiqueta RFID, una tarjeta o un transpondedor. Se componen de una antena RFID, un lector RFID (Transceiver) y una etiqueta RFID (RF-tag). Dependiendo del tamaño del almacén y de su configuración también pueden ser necesarios instalar arcos de lectura o portales RFID. (RFID, 2021)

Este sistema de semáforo inteligente con RFID está dividido en 4 partes principales las que son:

- la tarjeta RFID
- un punto de acceso
- servidor de redes
- WAN

Figura 2.

Esquema de semáforo inteligente usando RFID



Nota. Adaptado de Anoroza, M, M, 2017, (<https://docplayer.es/21810495-Semaforos-inteligentes.html>)

todo esto con la finalidad de crear una base de datos la cual lea la cantidad, velocidad y el tiempo de cada vehículo en una intersección y pueda realizar los cambios en los tiempos del semáforo en base a la congestión que este en ese momento. La limitación que hace notar el autor es que este tipo de sistema debe implementar la tarjeta RFID en todos los vehículos que transiten por la ciudad. (Anoroza, 2017).

2.1.5.2. Semáforo inteligente usando red de sensores inalámbricos

Este sistema es capaz de controlar el tráfico y manipular en forma dinámica los tiempos de espera manejando las mismas ideas de los semáforos con RFID. este sistema cuenta con dos partes principales: la red de sensores inalámbricos y la estación base la cual se encargan de ejecutar el algoritmo de control. (Anoroza, 2017).

Figura 3.

Esquema típico del sistema de semaforización inteligente usando sensores inalámbricos



Nota. Adaptado de *Componentes de un sistema de semáforo inteligente* (pág. 756), 2010,

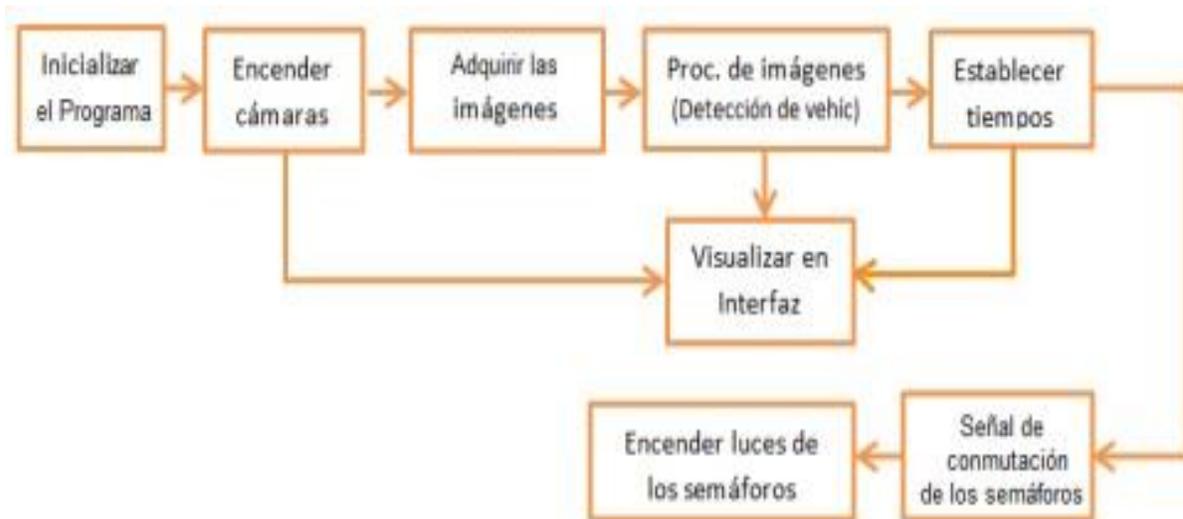
(https://www.researchgate.net/publication/220586947_Intelligent_Traffic_Light_Flow_Control_System_Using_Wireless_Sensors_Networks)

El funcionamiento básico es que a través de la red de sensores le proporcione datos a la estación base. La operación de cada sensor es obtener datos referentes a la velocidad, la cantidad, el número de vehículos y longitud de separación entre ellos, de manera de estar actualizando constantemente los datos que residen en la estación base y posteriormente ver si es necesario establecer algún cambio en la asignación de los tiempos de los semáforos controlados por la red de sensores.

2.1.5.3. Semaforización inteligente mediante el procesamiento de imágenes

Mediante el procesamiento de imágenes también es posible hacer un control de tráfico y de acuerdo con los datos obtenidos poder modificar los tiempos del semáforo, el proceso es el siguiente:

Figura 4.
Diagrama de bloques



Nota. Adaptado de *Diagrama de bloques del sistema por procesamiento de imágenes*, por carla Chávez, 2015, (https://repositorio.uta.edu.ec/bitstream/123456789/13061/1/Tesis_t1033ec.pdf)

Para poder trabajar con este tipo de imágenes se debe tener las herramientas necesarias para desarrollar el sistema como lo son:

- Acceder a los datos: esto se lleva a cabo adquiriendo las imágenes ya sea en vivo o por media de una grabación por medio de una cámara.
- Buscar algún tipo de software el cual ayude con el procesamiento de las imágenes que se logran obtener por medio de la cámara entre estos se encuentra como por ejemplo Matlab, OpenCV, LabVIEW.
- Buscar un sistema que ayude a establecer los tiempos que necesitamos.

2.2 Tipo de software y hardware

Un sistema de semaforización inteligente se da mediante el uso de herramientas los cuales deben de ofrecer un desempeño óptimo para el total funcionamiento de este, por esta razón es que es de vital importancia realizar una búsqueda sobre qué ventajas y desventajas pueden ofrecer cada una de las funciones a las necesidades del sistema.

2.2.1. Software para el procesamiento de imágenes

La IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano. Sin embargo, a diferencia de las personas, los dispositivos basados en IA no necesitan descansar y pueden analizar grandes volúmenes de información a la vez. (Rouhiainen, 2018, pág. 17)

Hoy en día existen varias plataformas las cuales tienen la capacidad de poder analizar imágenes y dar una respuesta sobre la cantidad de variables que se le solicite identificar. Algunas de estas plataformas son:

- OpenCV
- Microsoft visión API
- Matlab
- Amazon rekognition
- Google cloud visión api
- Clarifai
- Scikit-image

2.2.2 Sistema embebido (Hardware)

Un sistema embebido (también conocido como “empotrado”, “incrustado” o “integrado”) es un sistema de computación diseñado para realizar funciones específicas, y cuyos componentes se encuentran integrados en una placa base (en inglés. “motherboard”). El procesamiento central del sistema se lleva a cabo gracias a un microcontrolador, es decir, un microprocesador que incluye además interfaces de entrada/salida, así como una memoria de tamaño reducido en el mismo chip. (Luchetti, 2021).

Entre la variedad de sistemas embebidos que se encuentran en el mercado lo más a semejantes a nuestras necesidades encontramos los siguientes:

- Raspberry pi4 B+
- Raspberry pi Zero
- Banana pi BPI-M5
- Orange PI 3 LTS
- Arduino Mega 2560

2.2.3 Cámara

La cámara es un elemento bastante importante para el desarrollo del sistema debido a que está involucrado en todo el proceso de toma de imágenes para conteo de los vehículos, garantizando una buena definición de la imagen se puede asegurar que el procesamiento y detección de vehículos sea lo más fiable posible, hay que tener en cuenta que la cámara va conectada hacia el sistema embebido esta debe de contar ya sea conexión por cable USB o por alguna interfaz CSI dedicada dependiendo del sistema embebido de preferencia, en el mercado conseguimos las siguientes:

- Cámara web Logitech C270HD
- Raspberry pi cámara Rev.1.3
- Cámara web Genius 720HD
- Webcam Aukey Full HD1080p

2.3 Arquitectura del software

El software de este sistema está dividido en varias secciones donde cada una de ellas cumplen un papel importante para el desarrollo de este, a continuación, se explicarán detalladamente porque se eligieron y para qué sirven cada una de ellas.

2.3.1 Protocolos

Protocolo es el término que se emplea para denominar al conjunto de normas, reglas y pautas que sirven para guiar una conducta o acción. **Red**, por su parte, es una clase de estructura o sistema que cuenta con un patrón determinado. El concepto de **protocolo de red** se utiliza en el contexto de la **informática** para nombrar a las normativas y los criterios que fijan cómo deben comunicarse los diversos componentes de un cierto sistema de interconexión. Esto quiere decir que, a través de este protocolo, los dispositivos que se conectan en red pueden **intercambiar datos**. (Pérez Porto, 2013)

2.3.1.1 Protocolo HTTP

El *Protocolo de Transferencia de Hipertexto* se usa en todas las transacciones que tienen lugar en Internet, ya que cuenta con la definición de la semántica y la sintaxis que deben usar los servidores, los proxies y los clientes (todos componentes de la arquitectura web) para entablar una comunicación entre ellos. (Pérez Porto, 2013). esta herramienta es la base principal de cualquier intercambio de información en la Web, y un protocolo que se apoya en el esquema cliente-servidor. Quiere decir que el cliente realiza la petición a un programa y el servidor es quien le da la respuesta.

2.3.2. Lenguaje de programación

Se entiende por lenguaje de programación a un conjunto de instrucciones donde a través de ella el ser humano interactúa con las computadoras, este permite comunicar por medio de algoritmos e instrucciones escritas en una sintaxis y esta a su vez la computadora entiende e interpreta en lenguaje de máquina. (mendoza, 2020).

2.3.2.1. Python

Python es un lenguaje de programación donde puede ser usado en servidores web, desarrollo de software, la ciencia de datos y machine learning.

Se escogió este lenguaje ya que es gratuito, puede trabajar con varios sistemas operativos, hay muchos recursos de enseñanza.

Este es el lenguaje de programación que se va a instalar dentro de nuestro sistema embebido y será el encargado de toda la programación del sistema.

2.3.3. Servidor web

Un servidor web es una plataforma computacional muy potente que resguarda datos para ser consultados por diversos usuarios. Cuenta con un software que entrega la información solicitada por visitantes o miembros de un grupo de trabajo, a través de un dispositivo conectado a la red (coppola, 2023).

Para la selección del servidor web a nivel nacional se tuvieron en cuenta dos alternativas que eran Hostinger y Godaddy la diferencia entre estas dos es que Hostinger te permite hospedar otros dominios y aparte este provee o está ligada a una base de datos llamada MySQL, por lo tanto, el servidor web Hostinger so toma como referencia para el desarrollo del proyecto.

2.3.4. Base de datos

Para este proyecto es importante contar con un sistema que permita almacenar datos y que estos persistan a través del tiempo, por esta razón es que se implementan las bases de datos, por lo planteado en uno de nuestros objetivos fue de implementar un servicio que se encontrara dentro de las IoT por esta razón lo que buscamos fue alojar nuestros datos en una base de datos o nube. Esta va a permitir guardar y observar los datos que arroja el sistema las veces que se desee. Como se mencionó anteriormente en el capítulo 2.3.3 Hostinger es el servidor web seleccionado para el desarrollo del proyecto por ende la base de datos que va ligada con este será MySQL.

2.3.4.1 MYSQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Esto quiere decir que permite a pequeñas empresas y desarrolladores disponer de una solución fiable y estandarizada para sus aplicaciones. Por ejemplo, si se cuenta con un listado de clientes, una tienda online con un catálogo de productos o incluso una gran selección de contenidos multimedia disponible, MySQL ayuda a gestionarlo todo debida y ordenadamente. Este gestor de base de datos es utilizado para poder guardar los datos obtenidos por el sistema (robledano, 2019).

2.4. Situación del tráfico vehicular en la ciudad de Montería

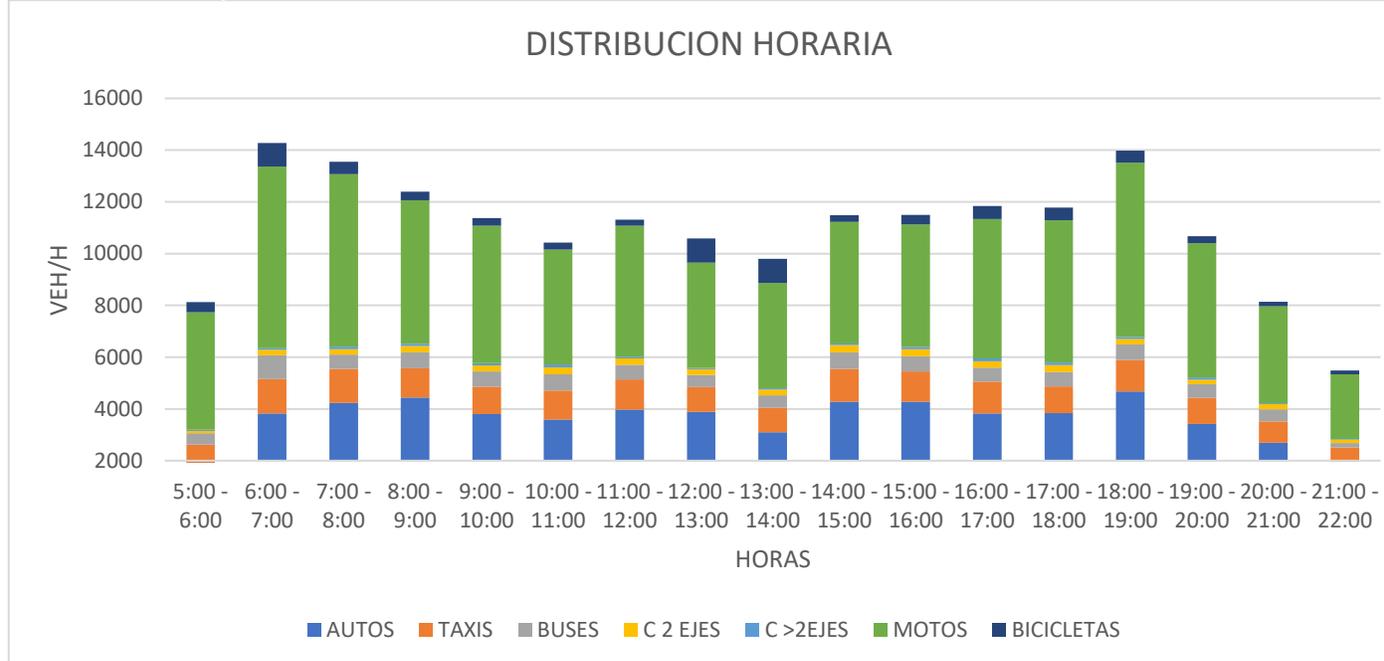
En la ciudad de Montería existen varios sitios y horarios donde se concentra la mayor afluencia vehicular. Los horarios considerados con mayor circulación vehicular se denominan horas pico. Estas horas pico son debido al ingreso de personas a distintas entidades públicas como privadas, estudiantes a los establecimientos educativos entre otras actividades. Según Gaviria et al. (2016, pág. 5) muestra mediante tablas que las horas pico de la ciudad de montería empieza en la mañana de 6:00/7:00 AM y en la tarde es de 18:00/19:00 PM. Asimismo, en su informe se analizan 3 calles donde se concentran el mayor flujo de vehículos de la ciudad como lo son:

- Intersección AV. Circunvalar con carrera 14
- Intersección calle 41 con carrera 14
- Intersección calle 44 con carrera 14

En las Figuras 5,6,7 y 8 se observa desde el número total de vehículos que circulan en todo un día como también el número de vehículos enfocados en las horas pico de cada una de las 3 intersecciones de la ciudad de montería.

Figura 5.

Resultados de procesamiento considerando análisis de Red



Nota. Adaptado a resumen volúmenes horarios (PAG 5), por Ipus Gaviria, O, N, 2016, (https://www.researchgate.net/publication/320820089_Evaluacion_tecnica_mediante_procesos_de_micro_simulacion_en_tres_intersecciones_en_Monteria_Cordoba)

Figura 6.

Resultados intersección Av. Circunvalar por Carrera 14

PICO AM										
PICO AM	AUTOS	TAXIS	BUSES	C 2 EJES	C >2 EJES	MOTOS	BICICLETAS	T MIXTOS	T EQUIV	% HORARIO
600 - 615	745	181	144	34	7	802	55	1968	1734	24%
615 - 630	602	276	197	43	8	799	80	2005	1823	26%
630 - 645	599	281	174	46	10	861	149	2120	1843	26%
645 - 700	690	224	144	17	9	881	118	2083	1743	24%
TOT PICO AM	2636	962	659	140	34	3343	402	8176	7143	100%
COMP VEH	32%	12%	8%	2%	0%	41%	5%	100%		
FACTOR DE HORA PICO AM									0,96	

PICO PM										
PICO PM	AUTOS	TAXIS	BUSES	C 2 EJES	C >2 EJES	MOTOS	BICICLETAS	T MIXTOS	T EQUIV	% HORARIO
1800 - 1815	851	158	92	33	6	999	52	2191	1806	27%
1815 - 1830	856	205	85	33	6	888	111	2184	1806	27%
1830 - 1845	713	152	70	32	8	784	69	1828	1518	23%
1845 - 1900	696	209	110	31	12	693	60	1811	1597	24%
TOT PICO PM	3116	724	357	129	32	3364	292	8014	6726	100%
COMP VEH	39%	9%	4%	2%	0%	42%	4%	100%		
FACTOR DE HORA PICO PM									0,91	

Nota. Adaptado a resumen volúmenes horas pico, (PAG 4), por Ipus Gaviria, O, N, 2016, https://www.researchgate.net/publication/320820089_Evaluacion_tecnica_mediant_e_procesos_de_micro_simulacion_en_tres_intersecciones_en_Monteria_Cordoba

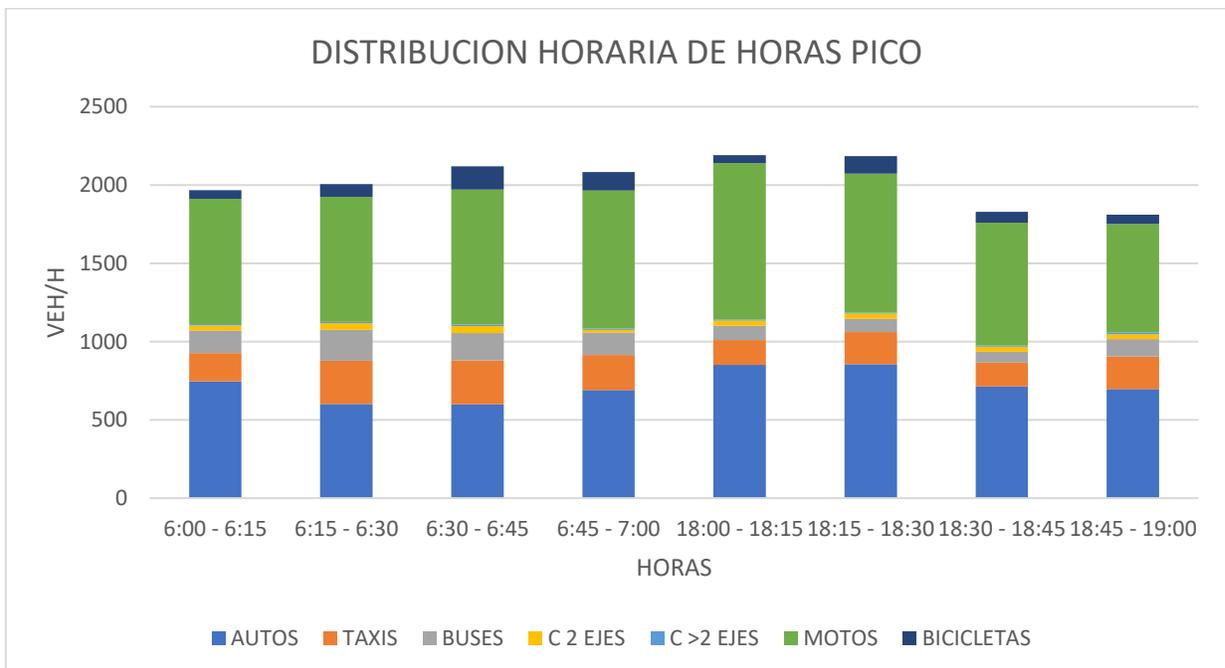


Figura 7.
Resultados intersección Calle 41 por Carrera 14

PICO AM										
PICO AM	AUTOS	TAXIS	BUSES	C 2 EJES	C >2 EJES	MOTOS	BICICLETAS	T MIXTOS	T EQUIV	% HORARIO
600 - 615	337	99	59	13	6	915	74	1503	1081	25%
615 - 630	335	71	53	21	6	946	111	1543	1086	25%
630 - 645	259	99	69	21	6	832	153	1439	1025	24%
645 - 700	263	87	82	14	21	981	171	1619	1143	26%
TOT PICO AM	1194	356	263	69	39	3674	509	6104	4336	100%
COMP VEH	20%	6%	4%	1%	1%	60%	8%	100%		
FACTOR DE HORA PICO AM									0,94	

PICO PM										
PICO PM	AUTOS	TAXIS	BUSES	C 2 EJES	C >2 EJES	MOTOS	BICICLETAS	T MIXTOS	T EQUIV	% HORARIO
1800 - 1815	461	120	60	29	7	920	35	1632	1262	27%
1815 - 1830	357	115	61	21	13	794	52	1413	1092	24%
1830 - 1845	403	128	70	20	20	838	40	1519	1202	26%
1845 - 1900	338	126	52	19	2	821	44	1402	1044	23%
TOT PICO PM	1559	489	243	89	42	3373	171	5966	4599	100%
COMP VEH	26%	8%	4%	1%	1%	57%	3%	100%		
FACTOR DE HORA PICO PM									0,91	

Nota. Adaptado a resumen volúmenes horas pico (PAG 5), por Ipus Gaviria, O, N, 2016,

https://www.researchgate.net/publication/320820089_Evaluacion_tecnica_mediante_procesos_de_micro_simulacion_en_tres_intersecciones_en_Monteria_Cordoba

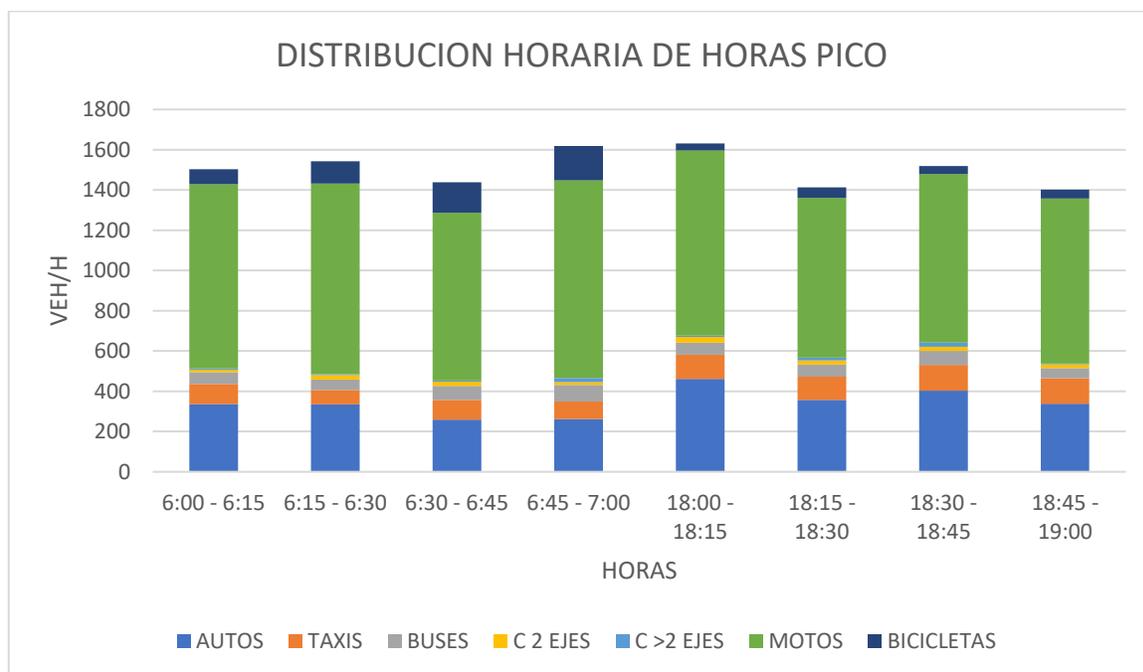
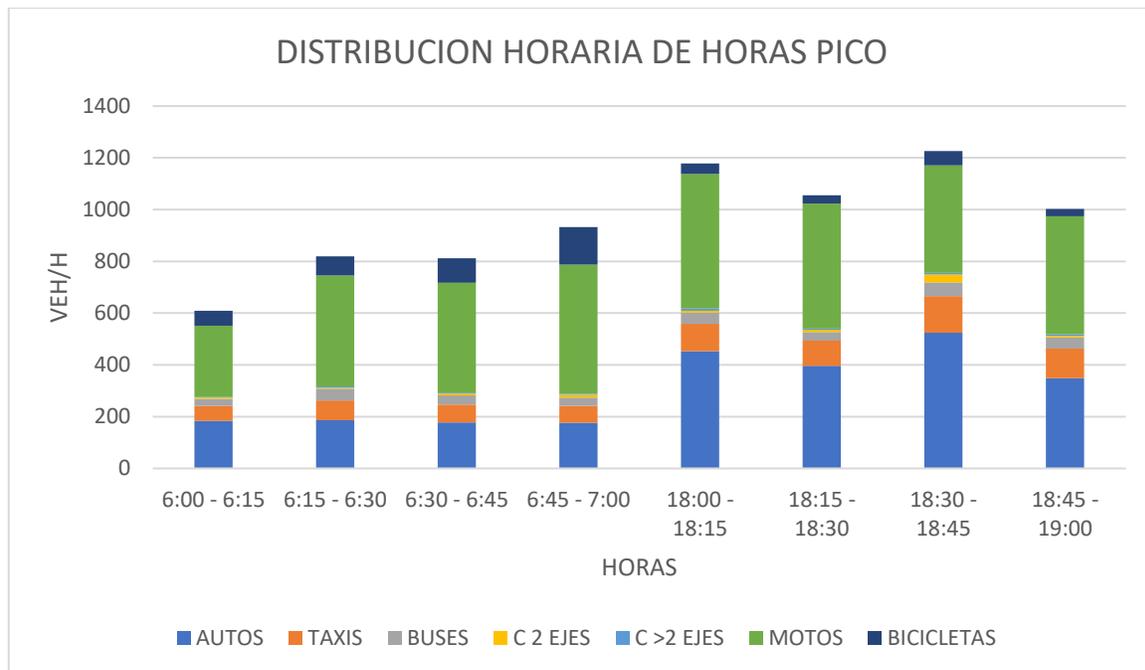


Figura 8.
Resultados intersección Calle 44 por Carrera 14

PICO AM										
PICO AM	AUTOS	TAXIS	BUSES	C 2 EJES	C >2 EJES	MOTOS	BICICLETAS	T MIXTOS	T EQUIV	% HORARIO
600 - 615	184	57	28	5	2	275	58	609	469	20%
615 - 630	186	76	46	4	4	430	74	820	611	27%
630 - 645	178	69	35	7	4	424	95	812	585	25%
645 - 700	176	65	32	12	3	499	146	933	636	28%
TOT PICO AM	724	267	141	28	13	1628	373	3174	2301	100%
COMP VEH	23%	8%	4%	1%	0%	51%	12%	100%		
FACTOR DE HORA PICO AM									0,85	

PICO PM										
PICO PM	AUTOS	TAXIS	BUSES	C 2 EJES	C >2 EJES	MOTOS	BICICLETAS	T MIXTOS	T EQUIV	% HORARIO
1800 - 1815	453	106	43	7	9	520	41	1179	957	26%
1815 - 1830	397	97	33	8	5	484	32	1056	844	23%
1830 - 1845	524	142	53	30	6	416	55	1226	1087	29%
1845 - 1900	349	114	43	6	6	456	29	1003	816	22%
TOT PICO PM	1723	459	172	51	26	1876	157	4464	3704	100%
COMP VEH	39%	10%	4%	1%	1%	42%	4%	100%		
FACTOR DE HORA PICO PM									0,91	

Nota. Adaptado a resumen volúmenes horas pico (PAG 5), por Ipus Gaviria, O, N, 2016,
https://www.researchgate.net/publication/320820089_Evaluacion_tecnica_mediante_procesos_de_micro_simulacion_en_tres_intersecciones_en_Monteria_Cordoba



2.5. Sistema del semáforo

Este sistema se ha realizado con la intención de gestionar los tiempos del semáforo obtenidos a través de nuestro sistema de detección, definir un tiempo en el cual el semáforo va a ejecutar dependiendo del número de vehículos que detecte nuestro sistema.

2.5.1. Enlace a la base de datos

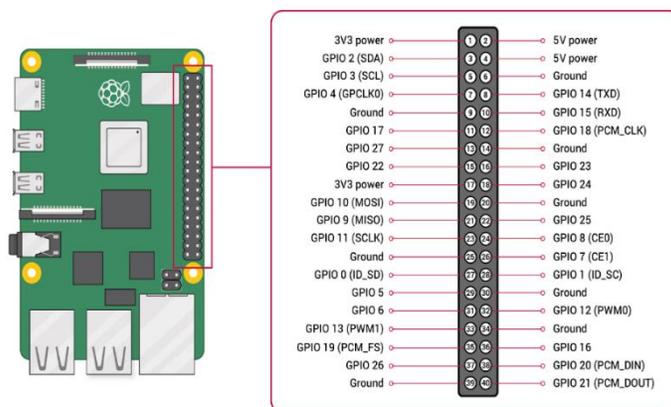
Se realizó una conexión a la base de datos en la que se usa una librería llamada pymysql esta cumple con la función de extraer los datos de la detección de vehículos estos serán cargados en la base de datos y así llevar un control de registro de cada instante mientras estuvo en ejecución nuestro sistema de detección y posteriormente estos datos serán usados para nuestro semáforo inteligente.

2.5.2 Implementación del semáforo en la Raspberry

Para conseguir que el circuito del semáforo funcione en la Raspberry fue necesario configurar sus puertos llamados GPIO, que son la interfaz física entre la Raspberry y el circuito del semáforo, los cuales trabajaran en base a la luz del semáforo a la cual se la va a asignar para su ejecución en el código. Para que el programa reconozca los pines de la Raspberry es necesario usar una librería dentro del entorno Python la cual es `import RPi.GPIO as gpio` esta ayudará con la comunicación entre los pines de la Raspberry y las salidas del semáforo.

Figura 9.

Funciones de los pines Gpio de la Raspberry pi4



Nota. Adaptado Esquema válido para la versión 4, por Isaac, software libre (https://www.hwlibre.com/gpio-raspberry-pi/)

2.6 Parámetros viales

2.6.1 Zona de estudio

La intersección que se empleará para la simulación del sistema de semáforo inteligente se encuentra ubicada en la AV circunvalar con calle 44 como se observa en la Figura 10.

Figura 10.

Intersección de la calle 44 con AV. Circunvalar



Nota. Zona de estudio donde se simulará el sistema de semáforo inteligente, Elaboración Propia,2022.

2.6.1.1. Tiempo de los semáforos en la zona de estudio

Como se observa en la Figura 11 y 12 se procedió a analizar los tiempos de duración del semáforo que se encuentra ubicado en la intersección de la Av. Circunvalar con calle 44, estos tiempos se tomaron en base a la duración en tiempo real del semáforo cuando se encuentra en estado verde como en rojo.

Figura 11.

Tiempo de espera del semáforo en rojo



Nota. representación del estado de tiempo de un semáforo ubicado en la AV. Circunvalar con calle 44, Elaboración propia,2022

Figura 12.

Tiempo del semáforo verde



Nota. representación del estado de tiempo de un semáforo ubicado en la AV. Circunvalar con calle 44, Elaboración propia,2022

Teniendo como resultado

Tiempo en verde= 37 Segundos

Tiempo en rojo = 100 Segundos

Tiempo en amarillo= este valor en general siempre es de 3 Segundos

2.7. Estado del arte

Los semáforos son capaces de medir el flujo del tráfico en la carretera, y según la carga, favorecen una u otra dirección, se miden el número de vehículos y la distancia entre ellos y se alargan las fases de verde o se reducen las fases de rojos. Esto está teniendo consecuencias positivas para la emisión contaminante, ya que los autos que circulan de forma constante consumen menos combustible (National Geographic , 2018).

Actualmente en el país se están tomando medidas con respecto a este problema el cual se puede ver reflejado en un proyecto que se está socializando en la ciudad de Bogotá. Con el cual se quiere reducir el tiempo de viaje del 15% y del 30%, también les permitirán a los peatones conocer el tiempo exacto de cruce de calle estos semáforos funcionarán a pesar de que haya apagones en la ciudad puesto que contarán con batería que les permitan operar un par de horas adicionales, además, estos estarán conectados en una central automática de operación asociada al centro de Gestión de tránsito de la ciudad. (El Espectador, 2017).

Por ejemplo en Bucarest, Rumania se registró el flujo máximo de vehículos en los cruces más importantes de las ciudades capitales y con los datos obtenidos, los cuales se analizaron y modelaron en la Universidad de Rutgers, Nueva Jersey, E.E.U.U, en estas modelaciones los semáforos tenían en cuenta la velocidad de todos los vehículos en las carreteras cercanas y se programan para calcular los cambios de color a fin de optimizar el flujo de tráfico, como resultado se pudo obtener que, el uso de estos semáforos inteligentes podía ayudar a mejorar los tiempos de viaje, reducir el consumo de combustible y mejorar la calidad del aire urbano y reducir el tiempo de espera en los semáforos hasta un 28% durante las horas pico (Glaskin, 2008).

Así mismo Investigadores de la Universidad de Aston en Birmingham (Inglaterra) han diseñado un nuevo sistema de inteligencia artificial que podría poner fin a las largas colas en los semáforos. Los desarrolladores construyeron un simulador de tráfico para entrenar el programa, enseñándole a manejar diferentes escenarios. El sistema es capaz de aprender, por lo que sigue mejorando cuando progresa, El sistema lee las imágenes de las cámaras en directo y adapta los semáforos en consecuencia para mantener el tráfico en movimiento de la mejor manera posible y reducir la congestión. Este sistema es el primero de su clase, según los científicos es tan eficaz porque utiliza el aprendizaje por refuerzo profundo, lo que significa que adapta sus procesos cuando no lo hace bien, intenta un curso de acción diferente, y sigue mejorando cuando progresa (Wang, 2022).

Del mismo modo en una ciudad alemana llamada Lemgo, situada cerca de Hannover, se ha puesto en marcha un proyecto piloto en el que se utilizan semáforos con inteligencia artificial, con el objetivo de mejorar el tráfico. Los primeros datos aseguran que este sistema funciona y que los trayectos son ahora más rápidos.

Estos semáforos inteligentes no son más que la reconversión de estos aparatos de una concurrida intersección de la ciudad. En ellos se han instalado cámaras de alta resolución y radares que estudian el volumen de tráfico que pasa por los mismos, la velocidad media de los vehículos y sus tiempos de espera. Mediante unos algoritmos diseñados para su propio aprendizaje, el semáforo mejora la regulación del tráfico con el paso del tiempo. Los primeros resultados del proyecto (que termina en verano) aseguran que se ha conseguido mejorar la fluidez del tráfico entre un 10% y un 15% sobre todo restando el tiempo de espera que los vehículos pasan parados esperando. Pero estos resultados tienen margen de mejora según los científicos, que esperan conseguir reducir el tiempo de espera hasta en un 30% (Torre, 2022)

3.METODOLOGIA

El desarrollo de un sistema de semaforización se da mediante la implementación de lógica programable, hardware y software los cuales ayudarán a lograr una buena simulación del sistema, para la solución de este proyecto se dividieron en diferentes partes las metodologías con el fin de cumplir los objetivos deseados.

3.1 Selección del hardware

Para este proyecto el hardware cumple un papel importante ya que es el causante de mostrar todos los resultados.

Para esto se tuvieron en cuenta 2 aspectos importantes para el hardware:

- Sistema embebido
- Tipo de cámara

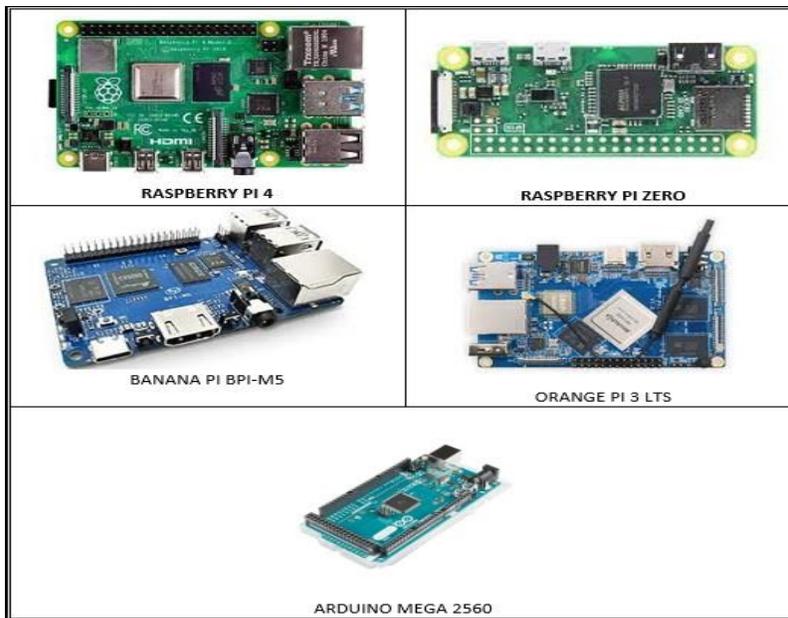
3.1.1. Sistema embebido

En este proyecto el sistema embebido debe tener unas características mínimas para el debido funcionamiento de este como lo son:

- Conexión a una cámara: esto es de gran utilidad para la obtención de imágenes.
- Conexión a internet: El sistema debe de estar en continua conexión a internet para estar comunicado con el servidor usando protocolos HTTP, MQTT.
- Capacidad de almacenamiento: debe de contar con un amplio almacenamiento donde se pueda guardar las imágenes y videos que se debe procesar.
- Lenguaje de programación: es de suma importancia que en el sistema embebido se pueda programar con un lenguaje de programación avanzado para el debido desarrollo de este.
- Maneje comunicación serial para poder hacer las respectivas conexiones al semáforo.

Se evaluaron cada uno de los sistemas embebidos establecidos en el **capítulo 2 sección 2.2** y se presenta la Tabla 1. Allí se observa el costo y las características de cada una de ellas dependiendo a los requerimientos mínimos para el desarrollo del sistema.

Figura 13.
Opciones del mercado



Nota. La figura muestra los sistemas embebidos que se evaluarán para el sistema de semaforzación inteligente, Elaboración propia.2022

Tabla 1.*Comparación de sistemas*

	Raspberr y pi4 B+	Raspberry pi Zero	Banana pi BPI-M5	Orange pi5 LTS	Arduino MEGA 2560
Conexión a internet	802.11 ac Wi-Fi/ Bluetooth 5.0, Gigabit Ethernet	Wi-Fi 802.11b/g/ n	10/100/1000 Mbit/s Ethernet Wi-Fi opcional USB dongle	Wi-Fi 6.0.2.4 GHz/5GH z dual- band Wi- Fi, support IEEE 802.11	Esta placa no cuenta con Wi-Fi, pero hay módulos Wi-Fi que se pueden agregar.
Protocolo UART	GPIO de 40 pines	GPIO 2x20 soldado	Cabezal de 40 pines	GPIO de 15 pines	Si, por los puertos GPIO
Almacenamiento	Máximo 256GB por micro SD	Máximo 256GB por micro SD	Máximo 256GB por micro SD	Máximo 256GB por micro SD	Necesita slot para expansión 256GB
Cámara	Con módulo de cámara web o USB	Con módulo de cámara web o USB	Cámara web USB	Con módulo de cámara web o USB	Con módulo de cámara web o USB
Lenguaje de programación	SI	SI	SI	SI	SI
Precio	380.000 COP	100.000 COP	540.000 COP	500.000 COP	113.000 COP

Nota. Esta tabla se muestra cada una de las características de los sistemas embebidos y su precio para determinar con cuál de ellas se trabajará.

En conclusión, todos estos dispositivos cumplen con los requerimientos mínimos, en este caso se opta trabajar con la placa de Raspberry pi 4 B+ debido a que es una tarjeta fácil de encontrar en el mercado local, cumple con los requerimientos técnicos, su precio la hace accesible al presupuesto, se presta para el tipo de sistema el cual se va a desarrollar, trabaja con sistema operativo Linux con el cual ya se tiene conocimiento y se tiene experiencia manejando este tipo de placa.

3.1.2. Cámara

El estudio de los diferentes tipos de cámaras realizados en el **capítulo 2 sección 2.3** se presenta en la Tabla 2. En el cual se observa el costo y las prestaciones de cada una de ellas

Figura 14.
Comparación de modelos de cámaras



Nota. La figura muestra cuatro tipos de cámaras el cual una de ellas que ayudará con la detección de vehículos, Elaboración propia.

Tabla 2.*Comparación de calidad de las cámaras*

	Logitech C270HD	Raspberry pi cámara Rev. 1.3	Genius Face 720HD	Aukey full HD 1080p
USB	Si	No, por interfaz CSI dedicada	Si	Si
Resolución	720p	720p	720p	1080p
Observaciones	Web cam De ángulo ajustable	Cámara para elementos fijos	Para elementos fijos	Web cam de ángulo 360°
Exteriores	No	Si	No	No
Precio	127.000 COP	35.000 COP	85.000 COP	153.000 COP

Nota. la tabla muestra el precio y cada una de las características que brinda cada cámara.

Se analizaron las 4 cámaras y se concluye que cada una de ellas sirven perfectamente para el proyecto ya que cada una cuenta con compatibilidad a la raspberry como también cuentan con una calidad óptima para la captura de fotografías, la cámara AUKEY por cuestiones que tiene soporte Linux el mismo sistema operativo que maneja la Raspberry, por su resolución y movilidad fue la más adecuada para el sistema.

3.2 Selección del software

Al evaluar los diferentes softwares para el procesamiento de imágenes, realizados en el **capítulo 2 sección 2.1** se presenta la Tabla 3.

Tabla3.

Software para el procesamiento de imágenes

Aplicación	Funcionamiento con raspberry	Uso Libre	Tiene entrenamiento robusto
Matlab	Si	No	Si
Open CV	Si	Si	Si
Microsoft visión API	No	No	No
Amazon rekognition	Si	No	No
Google cloud visión api	Si	No	No
Clarifai	No	No	No
Scikit - image	Si	Si	Si

Nota. En la tabla se puede observar cada uno de los softwares analizados con las características necesarios que necesita el sistema.

Al evaluar la anterior tabla se llegó a la conclusión de inclinarse por el uso de OpenCV ya que este es un software robusto, es un software libre para el uso institucional, cuenta con bastante información para su uso, es un sistema multiplataforma lo cual lo hace óptimo para trabajar en S.O. Linux el cual viene por defecto en la Raspberry, puede trabajar con diferentes lenguajes de programación que son de fácil acceso.

tanto Matlab como Scikit-image eran también muy buenas opciones, pero por la necesidad de trabajar en un ambiente Raspbian y que fuera un sistema compacto se eligió OpenCV.

3.3. Descripción de datos para la duración del semáforo

Para simular el tiempo en verde que va a durar el semáforo de este proyecto se realizó un estudio con los datos suministrados en el **capítulo 2 sección 4** donde se observa por medio de una tabla el número de vehículos que pasan por una intersección de la ciudad de Montería en las horas pico, también se tendrá en cuenta el número real de segundos de un semáforo cuando se encuentra en estado verde y rojo como observa en las Figuras 11 y 12. Esto se hará con el objetivo de simular

el tiempo que se le van a asignar a cada vehículo dependiendo del número de carros que detecte el sistema y a su vez este valor se repartirá en la duración del semáforo. Teniendo en cuenta que este proyecto se encuentra enfocado en un prototipo a escala de un sistema de semaforización inteligente ubicado en la intersección de la AV circunvalar con calle 44 sentido sur – norte, se tomaron los datos de las horas pico de la Figura 8 y se les hará una pequeña modificación en el sentido de eliminar los tipos de transporte que no cuenten como vehículos. cómo se observa en las Tablas 4 y 5.

Tabla 4.

Número total de vehículos

Pico AM	Autos	Buses	Taxis	C 2 ejes	C > 2 ejes	Total
6:00/6:15	184	28	57	5	2	276
6:15/6:30	186	46	76	4	4	316
6:30/6:45	178	35	69	7	4	293
6:45/7:00	176	32	65	12	3	288
Total, pico AM	724	267	141	28	13	1173

Nota. Esta tabla muestra el número total de autos que pasan en la intersección de la AV. Circunvalar con calle 44 en horas pico AM.

Tabla 5.

Número total de vehículos

Pico PM	Autos	Buses	Taxis	C 2 ejes	C >2 ejes	total
6:00/6:15	453	43	106	7	9	618
6:15/6:30	397	33	97	8	5	540
6:30/6:45	524	53	142	30	6	755
6:45/7:00	349	43	114	6	6	518
Total, pico PM	1723	172	459	51	26	2431

Nota. Esta tabla muestra el número total de autos que pasan en la intersección de la AV. Circunvalar con calle 44 en horas pico PM.

Para medir los tiempos en la intersección se debe primero calcular el número de ciclos que hace el semáforo en una hora para esto se debe tener en cuenta que son 37 segundos de verde más 3 segundos de amarillo más el tiempo que dura en rojo que son 100 segundos esto sumarian 140 segundos.

Se sabe que en 1 hora hay 3600 segundos entonces, realiza el cálculo de la cantidad de ciclos que hay en una hora:

$$C = \frac{3600seg}{140seg}$$

$$C = 25.71 \text{ ciclos} \approx 26$$

Ahora para hallar el número de autos que pasan en un solo ciclo de semáforo para las horas pico AM tenemos que el número total de autos son 1173 entonces:

$$N^{\circ}autos = N^{\circ}totalcarros/C$$

$$N^{\circ}autos = \frac{1173}{26}$$

$$N^{\circ}autos = 45,11 \approx 45$$

En un ciclo de semáforo pasaron 45 automóviles

Ahora se procede a hallar el número de segundos por vehículo:

T= es la sumatoria de los tiempos en verde y amarillo del semáforo

$$Seg = N^{\circ}autos/T$$

$$Seg = \frac{45}{40}$$

$$Seg = 1.125$$

Se aplicará la misma metodología para las horas pico PM

$$N^{\circ}Autos = \frac{2431}{26}$$

$$N^{\circ}Autos = 93.5$$

$$Seg = \frac{93.5}{40}$$

$$Seg = 2.3375$$

Entonces:

Para las horas pico AM son 1.125 segundos por vehículo

Para las horas pico PM son 2.33 segundos por vehículo

3.4 Sistema de alimentación

Este sistema de semaforización inteligente cuenta con una alimentación el cual está basado en energía solar y es autosustentable para su funcionamiento.

la Raspberry pi4 B+ es una placa que no utiliza la misma cantidad de energía de forma constante a lo largo del tiempo, todo dependerá del uso que se le esté dando a este según (Bugueados , 2020) dice que cuando la Raspberry está ejecutando un

código Python requiere un uso intensivo de la CPU de la Raspberry, con ayuda de un medidor de consumo de energía eléctrica (Amazon, s.f.) se ha visto un pico en

el consumo de 6.2 W pero en la mayor parte del tiempo estaba entre **5 y 5.5 W**. ante esto como el sistema no va a estar funcionando a tiempo completo se tomara como referencia que la media de consumo van a ser 5W por hora, entonces con este dato podemos calcular el consumo diario de nuestra Raspberry con la siguiente ecuación:

$$RasWh/dia = Cantidad * PRasp * h/dia$$

Donde:

Cantidad es el número de Raspberry con las que se va a trabajar
PRasp es la potencia media de consumo de la Raspberry por hora
día son las horas que va a trabajar la Raspberry en un día

$$RaspWh/dia = 1 * 5W * 2h/dia$$

$$RaspWh/dia = 10Wh/dia$$

El valor de dos horas diarias de trabajo de la Raspberry se basa en las horas pico establecidas en el **capítulo 2 sección 4** donde tomaran solamente 2 horas de trabajo que son las horas donde hay mayor flujo vehicular.

para poder determinar el sistema de alimentación basada en energía solar se debe tener en cuenta varios aspectos como:

1. Determinar el Angulo de los paneles y las horas solar pico de la ciudad de montería, para hallar estos valores se hizo uso de una base de datos llamada PVGIS.

Para determinar Angulo de los paneles tenemos la siguiente ecuación:

$$Inclinacion\ Optima = 3.7 + 0.69 * Latitud$$

El valor de la latitud se puede encontrar dentro del mapa de PVGIS como se observa en la Figura 16. esta variable es la latitud de la ciudad o el lugar donde se desee buscar el valor HSP.

Figura 15.
Sistema de información fotovoltaica

The screenshot shows the PVGIS web interface. At the top, the cursor is set to 8.755, 75.887. The elevation is 20 meters. The interface is titled 'DATOS MENSUALES DE IRRADIACIÓN'. The base of solar radiation data is set to 'PVGIS-NSRDB'. The start and end years are both set to 2005. Under the 'Irradiación' section, 'Irradiación global en ángulo' is selected, and the resulting value is 9.74164. At the bottom, there are buttons for 'Visualizar resultados', 'CSV', and 'json'. The 'CSV' button is circled in red.

Nota. Representación de una gráfica donde se determina el valor de las horas solares pico en determinado lugar, tomado de comisión europea, 2022, https://re.jrc.ec.europa.eu/pvg_tools/en/

Al obtener el valor de la latitud que es 8.755 reemplazamos en la ecuación

$$\begin{aligned} \text{Inclinacion optima} &= 3.7 + 0.69 + 8.755 \\ \text{Inclinacion Optima} &= 9.74164 \end{aligned}$$

El resultado de la inclinación optima se agregará en la casilla global irradiation. al tener este valor se descarga un documento de Excel como se observa en la Figura 16. Donde el valor de los HSP es simplemente el valor de la irradiación entre los días que tiene cada mes. Sacando una media entre los HSP de cada mes tenemos que nuestro HSP=5,33 solamente de la ciudad de montería.

Figura 16.

Tabla general de las horas de pico solar por meses

Latitude (decimal degrees):	8,756		
Longitude (decimal degrees):	-75.886		
Radiation database:	PVGIS-NSRDB		
year	month	Irradiacion	HSP
2015	Jan	171,5	5,71666667
2015	Feb	162,03	5,78678571
2015	Mar	189,02	6,09741935
2015	Apr	155,34	5,178
2015	May	155,37	5,01193548
2015	Jun	164,16	5,472
2015	Jul	154,78	4,99290323
2015	Aug	171,83	5,54290323
2015	Sep	154,85	5,16166667
2015	Oct	157,46	5,07935484
2015	Nov	141,54	4,718
2015	Dec	161,8	5,21935484
Total media			5,33141583
H(i)_m: Irradiation on plane at angle (kWh/m2/mo)			
PVGIS (c) European Union, 2001-2022			

Nota. La tabla nos define cada una de las variables para determinar el ángulo y las horas pico solar para un sistema de energía solar, tomado de comisión europea, 2022, https://re.jrc.ec.europa.eu/pvg_tools/en/

2. para este proyecto se utilizará un panel solar de 12W como se observa en la Figura 17. Se calculará cuanta energía producirá este panel solar por día, para esto se tiene la siguiente formula:

$$E_{Panel} = W_{Panel} * HSP * 0.9$$

Donde:0.9 es el rendimiento de trabajo del panel

$$E_{Panel} = 12 * 5.33 * 0.9$$

$$E_{Panel} = 57.56 \text{ Wh/dia}$$

El rendimiento del panel solar por día es más que óptimo para poder alimentar nuestro sistema.

Figura 17. Panel solar del proyecto de energía solar



Características del panel solar|

- Potencia: 12 W
- Modelo: M012P
- Máxima corriente de potencia (I_{mp}): 0.68A
- Voltaje de potencia máxima (V_{mp}): 17.89V
- Corriente de cortocircuito (I_{sc}): 0.75A
- Voltaje de circuito abierto (V_{oc}): 21.48V
- Dimensiones: 39 x 29 x 2.3 cm
- Tolerancia de potencia: $\pm 3\%$
- STC: Irradiancia 1000w/m², temperatura del módulo 25°C, AM=1.5
- Temperatura de funcionamiento: -40 +85 grados centígrados.
- Equipo para exterior, puede estar expuesto al sol y al agua.

Nota. Panel solar, artículos alta gama ,2022,
https://articulo.mercadolibre.com.co/MCO-602954445-panel-solar-jarrett-12w-12v-fotovoltaico-con-pinzas-_JM

3. Para calcular el amperaje hora de la batería se tiene la siguiente formula:

$$Ah = \frac{RaspWh}{dias * dias\ de\ trabajo / Pd * Vbateria}$$

Donde:

días de trabajo serán los días a la semana que la Raspberry va a trabajar

Pd: es la profundidad de descarga de la batería en este caso 0.7

$$Ah = 10W * 6 / 0.7 * 12$$

$$Ah = 7.14$$

Teniendo en cuenta el valor anterior se escogerá una batería de 7.5Ah de la Figura 18.

Figura 18.

Batería del proyecto de energía solar



Nota. Batería de litio de 12V 7.5Ah, 2022, Power computer

https://articulo.mercadolibre.com.co/MCO-893848544-bateria-12v-75-ah-75ah-o-7-ah-nueva-_JM

4. El regulador de carga es un punto importante para el sistema de energía solar ya que este controla el estado de carga de las baterías para alargar su vida útil. la fórmula para determinar la intensidad de un regulador de carga óptimo es el siguiente:

$$\text{Intensidad de carga del regulador} = \text{ISC del panel} * 1.25$$

Donde:

1.25 un factor de seguridad para evitar daños ocasionales en el regulador.

ISC es una característica que viene por defecto dependiendo del panel solar

$$\begin{aligned} \text{Intensidad de carga del regulador} &= 0.75A * 1.25 \\ \text{Intensidad de carga del regulador} &= 0.9375 \end{aligned}$$

Como estas fórmulas están basadas en sistemas de energía solar a gran escala y nuestro sistema trata de un simple prototipo algunos valores tornan a ser algo pequeños aun así el regulador que obtuvimos para el sistema se observa en la Figura 19.

Figura 19.

Regulador de carga



Nota. En la imagen se aprecia un regulador de carga PWM el cual por medio de su pantalla nos indicara el estado de la batería como también del panel solar, 2022, INELDEC SAS, https://articulo.mercadolibre.com.co/MCO-861098451-controlador-de-carga-regulador-solar-30a-pwm-30-amp-_JM#position=1&search_layout=stack&type=item&tracking_id=4f05f7e9-0572-4ca6-8d48-abfb3ec979c2.

5. Por último como se va a usar un sistema embebido el cual trabaja con voltajes DC se da la necesidad de investigar sobre algún dispositivo que ayudara a regular el voltaje ya que la batería arroja voltajes diferentes dependiendo de la radiación solar que está pasando por el panel solar, entonces lo que se implementara es colocar a la salida del regulador de carga un convertidor tipo DC a DC el cual convirtiera los 12V a 5V estables que son los necesarios para que la Raspberry pueda trabajar, el regulador se observa en la Figura 20.

Figura 20.
convertidor DC-DC Buck XL4015 De 5A



Nota. Representación de un conversor de voltaje de corriente directa,2022, vistrónica, Fuente:https://www.vistrónica.com/#q=CONVERTIDOR%20DC-DC%20BUCK&from_user=&page=0&refinements=%5B%5D&numerics_refinement_s=%7B%7D&index_name=%22prestashop1763_all_es%22

3.5. Instalación de Raspberry pi OS

Como se observa en la descripción de la Tabla 1. El almacenamiento de la Raspberry se basa en memorias micro SD, para poder instalar el sistema operativo el requisito mínimo de estas memorias es que sean mayor a 16GB de almacenamiento y tenga una velocidad clase 10 ya que posibilita la trasferencia de datos de forma rápida y efectiva, la memoria que usamos en nuestro proyecto se observa en la Figura 22.

Figura 21.
Memoria microSD de la Raspberry



Para la instalación del sistema operativo es necesario realizar la descarga de Raspberry pi imager como se muestra en la figura 22. esta es la herramienta encargada de poder instalar dentro de la memoria SD el sistema operativo.

Figura 22.
Interfaz Raspberry pi image

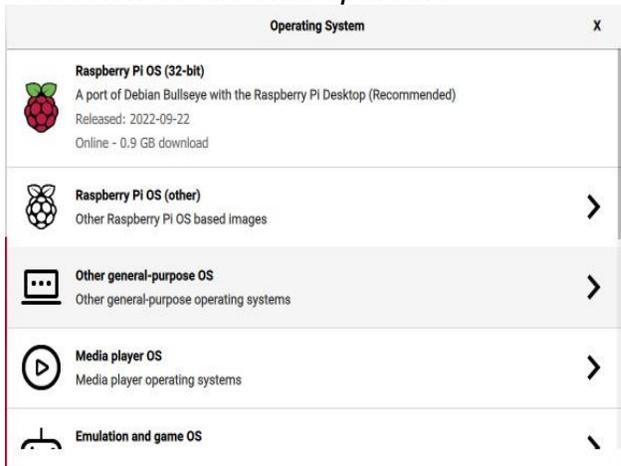


Al estar dentro de la interfaz se encuentran dos opciones Choose OS y Choose storage.

En Choose OS hay varios propósitos los cuales cada uno depende de la necesidad del usuario en el caso de este proyecto solo se necesitará instalar Raspberry PI OS (32-bit) como se observa en la figura 23.

Choose storage simplemente es la ubicación donde se encuentra conectada la memoria microSD.

Figura 23.
Selección del sistema operativo



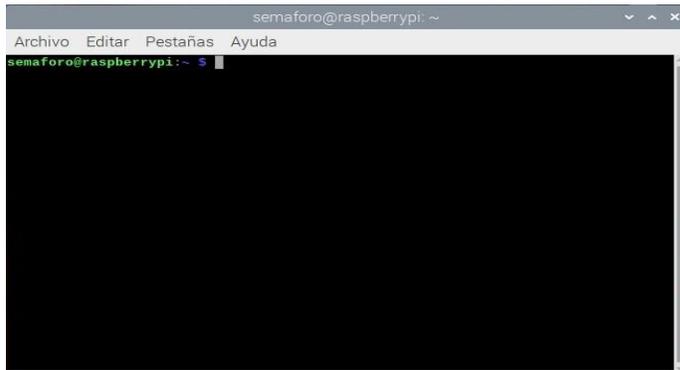
A partir de esto ya se tendrá instalado el sistema operativo Raspbian.

3.5.1. Instalación de OpenCV, Python y MySQL en Raspberry

Los sistemas operativos basados en Linux disponen de un intérprete de comandos u órdenes conocido como terminal como se observa en la Figura 24. Esta hace de interfaz entre el usuario y el propio sistema operativo. Es decir, mediante la terminal podemos acceder al sistema operativo sin utilizar una interfaz gráfica y realizar todo tipo de tareas en modo texto o mediante el uso de comando la mayoría de las veces.

Figura24.

Terminal de comandos



3.5.2. Instalación Python

Según (Tony, 2021) Los entornos virtuales son de gran beneficio ya que ayudan a crear un entorno aparte para poder ejecutar proyectos, por lo tanto, estos proyectos tienen sus propios conjuntos de dependencias y bibliotecas.

Para empezar con la instalación de un entorno virtual en la Raspberry, se ejecuta el siguiente comando dentro de su terminal.

sudo apt-get install python3-pip

al ejecutar este comando ya se debe tener instalado en la Raspberry el entorno virtual de Python (Tony, 2021).

3.5.3. Instalación OpenCV

Actualmente hay dos métodos para una respectiva instalación de OpenCV en Raspberry.

- Compilar OpenCV desde la fuente (instalación de Raspberry altamente recomendada)

- Instalando OpenCV usando Python PIP.

la diferencia entre estos dos es que instalando desde la fuente es un proceso largo y tedioso, pero es el más recomendable para cualquier tipo de proyectos y usando Python PIP son pocos comandos y mucho más rápido para su instalación (Tony, 2021).

3.5.3.1. Instalando OpenCV usando Python PIP.

Ejecutamos el comando **workon sbb_cv** para activar el entorno virtual

Figura 25.

Instalación OpenCV en raspberry pi

```
pi@raspberrypi:~ $  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $ workon sbb_cv  
(sbb_cv) pi@raspberrypi:~ $ █
```

Una vez dentro del entorno virtual ya se puede instalar OpenCV ejecutando el comando **pip3 install opencv-python**

Figura 26.

Instalación OpenCV en raspberry pi

```
(sbb_cv) pi@raspberrypi:~ $ pip3 install opencv-python  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Collecting opencv-python  
  Using cached https://www.piwheels.org/simple/opencv-python/opencv_python-4.5.1.48-cp37-cp37m-linux_armv7l.whl (10.5 MB)  
Requirement already satisfied: numpy>=1.14.5 in ./virtualenvs/sbb_cv/lib/python3.7/site-packages (from opencv-python) (1.20.2)  
Installing collected packages: opencv-python  
Successfully installed opencv-python-4.5.1.48  
(sbb_cv) pi@raspberrypi:~ $ █
```

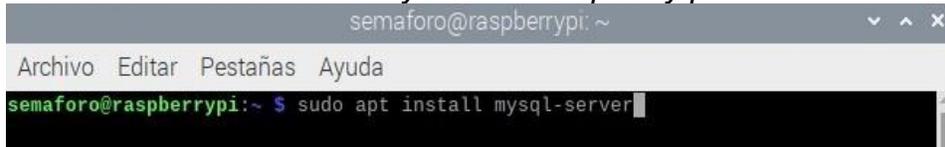
Con esto ya está instalado OpenCV dentro de la Raspberry de la forma rápida y sencilla (Tony, 2021).

3.5.3. Instalación de MySQL

Para cualquier proyecto de desarrollo la implementación de un gestor de base de datos es muy importante, MySQL es una de las opciones más recurrentes para proyectos caseros esto debido a que es una base de datos gratuita, es muy fácil de

usar y es muy rápida. Instalar MySQL en Raspberry Pi es muy sencillo, para esto se ejecuta la instalación con el siguiente comando que se observa en la Figura 27.

Figura 27.
instalación base de datos MySQL en Raspberry pi



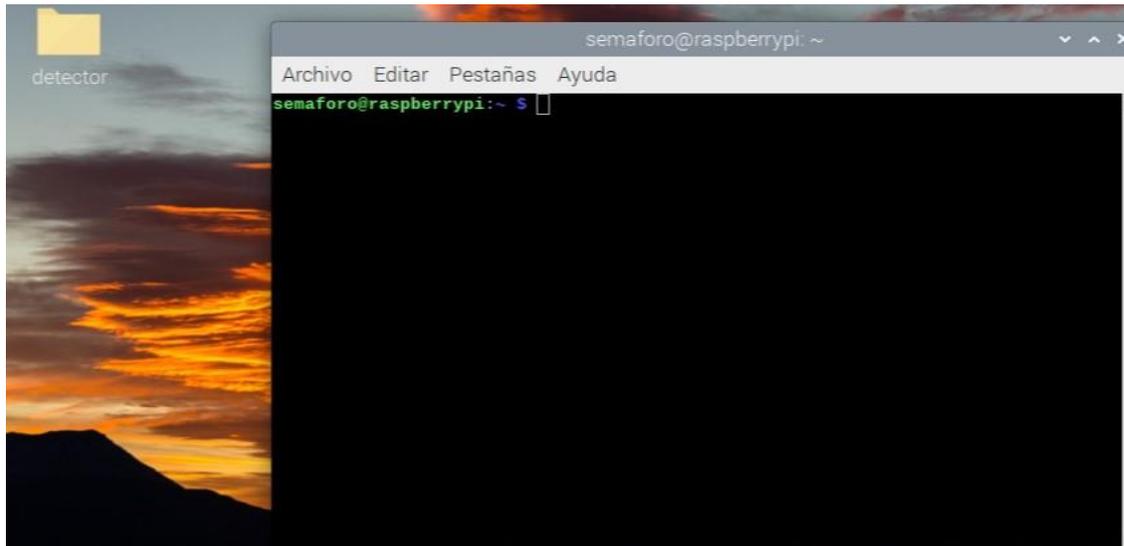
```
semaforo@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
semaforo@raspberrypi:~ $ sudo apt install mysql-server
```

Al ejecutar este comando MySQL estará instalado dentro de la Raspberry.

3.6 Ejecución del programa

Para comprobar el correcto funcionamiento del algoritmo diseñado en Python, se abre una terminal de comandos dentro de Raspberry.

Figura 28.
Terminal de comandos



Allí se puede observar la terminal de comandos que utiliza Raspberry y una carpeta llamada **detector** la cual es donde se encuentra almacenado todos los elementos del funcionamiento del sistema.

para ejecutar el programa escribimos lo siguiente en la ventana de comando

Figura 29.

Ejecución del código

```
semaforo@raspberrypi:~ $ cd /home/semaforo/Desktop/detector
semaforo@raspberrypi:~/Desktop/detector $
```

Nota. Como observa se dispone a hacer un llamado, este llamado hace referencia al lugar donde se encuentra ubicada la carpeta **Detector**, se ejecuta y la ventana muestra que **/desktop/detector** ya está disponible para ejecutar lo que se encuentre dentro de ella.

después de esto dentro de la carpeta hay un archivo llamado detector.py este es el responsable de la ejecución de nuestro programa.

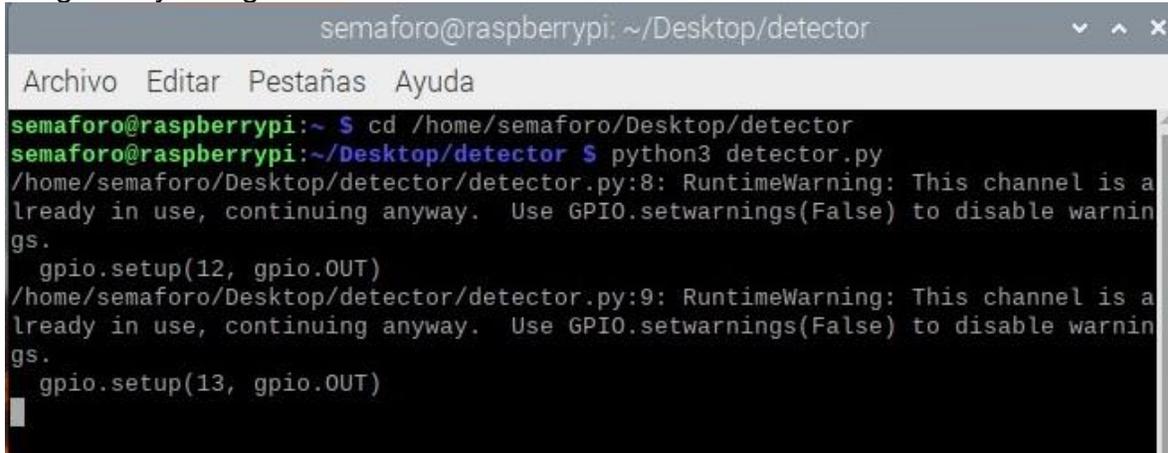
Figura 30.

Carpeta contenedora



Nota. Script en Python

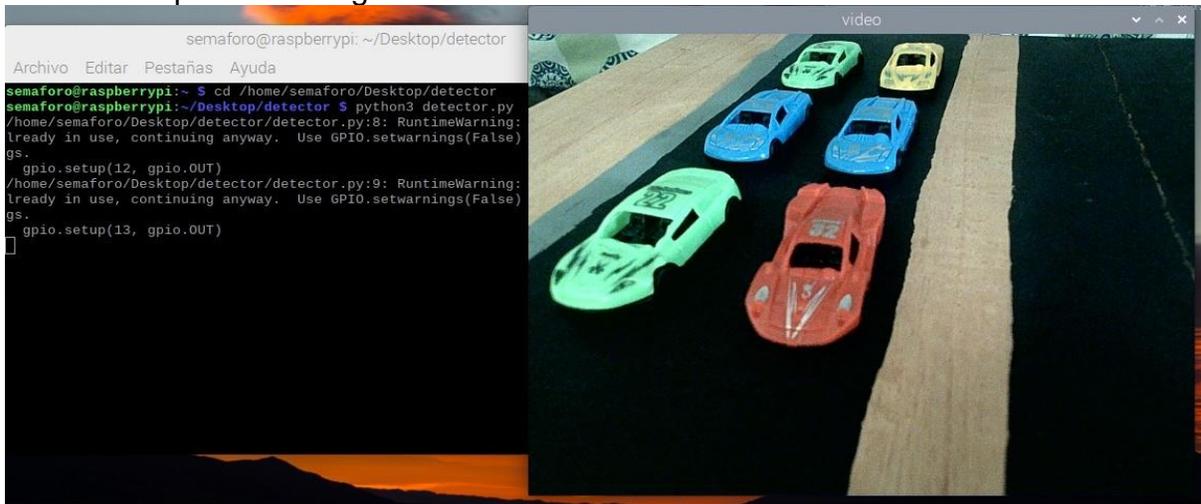
Figura 31.
Programa ya cargado



```
semaforo@raspberrypi: ~/Desktop/detector
Archivo Editar Pestañas Ayuda
semaforo@raspberrypi:~ $ cd /home/semaforo/Desktop/detector
semaforo@raspberrypi:~/Desktop/detector $ python3 detector.py
/home/semaforo/Desktop/detector/detector.py:8: RuntimeWarning: This channel is a
lready in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnin
gs.
  gpio.setup(12, gpio.OUT)
/home/semaforo/Desktop/detector/detector.py:9: RuntimeWarning: This channel is a
lready in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnin
gs.
  gpio.setup(13, gpio.OUT)
```

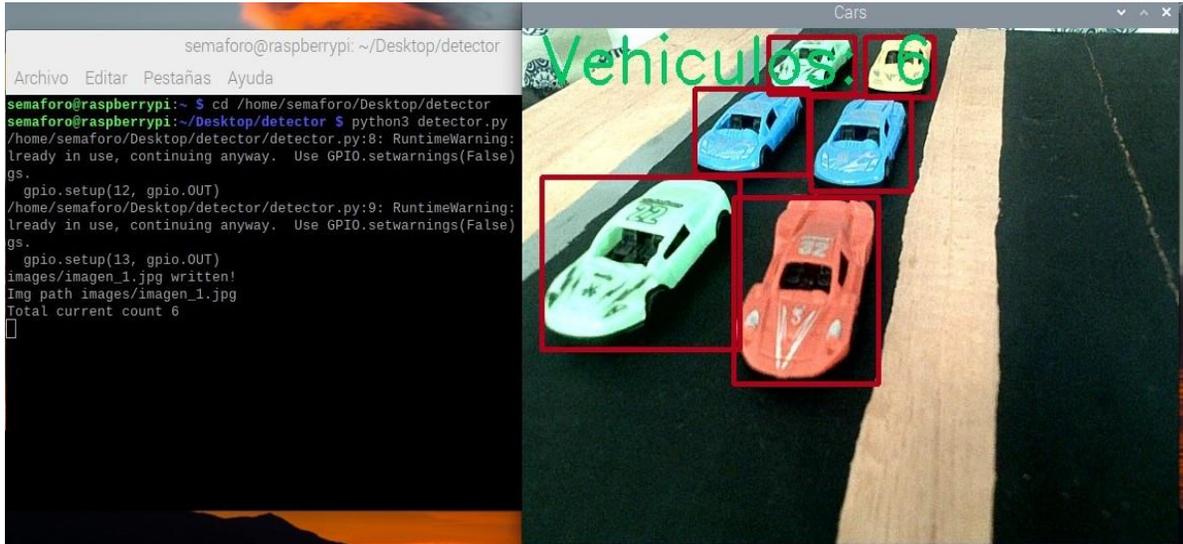
Al ejecutar la línea de comando **python3 detector.py**, lo que se hace es primero que todo ejecutar dentro de la Raspberry el programa de python3 y segundo, está haciendo referencia a la carpeta detector la cual es donde se encuentra el código de detección de los vehículos, al momento de ejecutar nos muestra también la salida de los pines la Raspberry donde está conectado el sistema del semáforo.

Figura 32.
Toma de captura de imagen



Al mismo tiempo de ejecutar el comando **python3 detector.py**, abre una ventana alterna donde se ejecuta el funcionamiento de la cámara en este caso ella queda habilitada a la espera de darle la instrucción la cual es capturar una imagen y después hacer el respectivo proceso de detectar el número de vehículos que se encuentran dentro del rango de la cámara este proceso es manual ya que se necesita de un tercero quien presione la tecla A y así pueda arrancar proceso de detección del sistema.

Figura 33.
Conteo de vehículos



Al momento de ejecutar el programa se observa en la ventana de comandos que la cámara toma la foto, la analiza y hace el reconocimiento de los automóviles que se encuentran dentro del Angulo de la cámara el de este caso fueron 6 automóviles.

Mediante el uso de un dron se realizó una grabación de una de las intersecciones de la calle 44 con AV. Circunvalar con la intención de probar el sistema en una calle con vehículos reales.

Figura 34.
Conteo de vehículos en grabación hecha por un dron



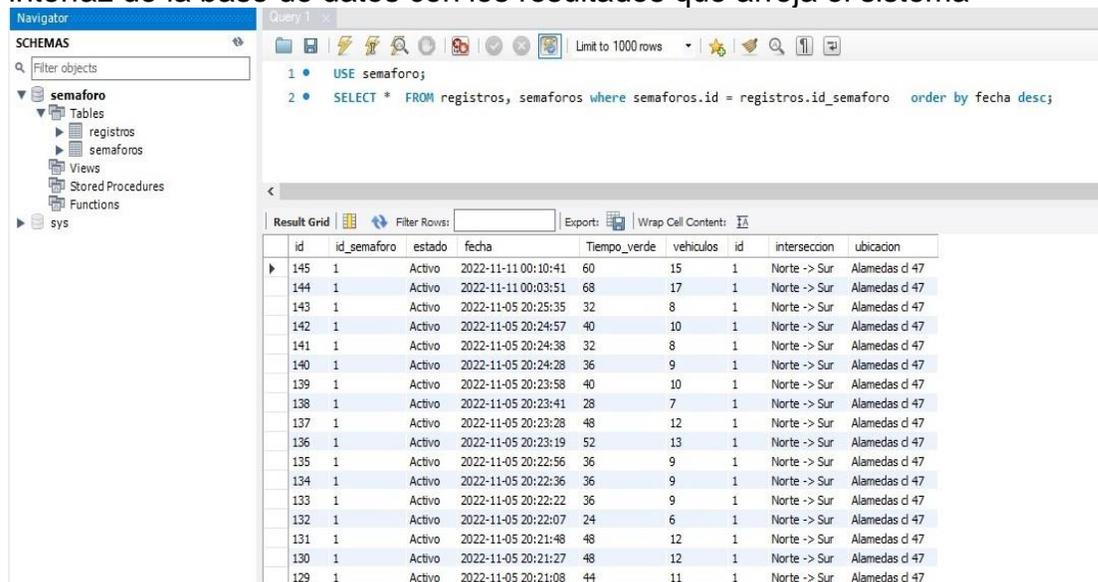
En la figura se aprecia que el programa también puede captar y detectar el número de vehículos que se encuentran sobre una vía en tiempo real de la ciudad de montería.

4.RESULTADOS Y DISCUSIÓN

Como resultado se tiene un prototipo de sistema de semáforo inteligente utilizando un sistema embebido, el cual nos permitió realizar una simulación de control de flujo de vehículos de una intersección por medio de la detección y conteo de vehículos. Además, este proyecto cuenta con un sistema de alimentación basada en energía solar el cual funciona eficientemente para su respectivo uso, como también, es posible conocer el estado del sistema debido a la implementación de un sistema de adquisición de datos que permitirá observar las variables de salida de la Raspberry y poder ser visualizadas por medio de un servidor web. en las Figuras 35,36 y 37 se observa el prototipo y la base de datos con las variables.

Figura 35.

interfaz de la base de datos con los resultados que arroja el sistema



The screenshot displays a database interface with a query window and a result grid. The query window shows the following SQL query:

```
USE semaforo;  
SELECT * FROM registros, semaforos where semaforos.id = registros.id_semaforo order by fecha desc;
```

The result grid contains the following data:

id	id_semaforo	estado	fecha	Tiempo_verde	vehiculos	id	interseccion	ubicacion
145	1	Activo	2022-11-11 00:10:41	60	15	1	Norte -> Sur	Alamedas d 47
144	1	Activo	2022-11-11 00:03:51	68	17	1	Norte -> Sur	Alamedas d 47
143	1	Activo	2022-11-05 20:25:35	32	8	1	Norte -> Sur	Alamedas d 47
142	1	Activo	2022-11-05 20:24:57	40	10	1	Norte -> Sur	Alamedas d 47
141	1	Activo	2022-11-05 20:24:38	32	8	1	Norte -> Sur	Alamedas d 47
140	1	Activo	2022-11-05 20:24:28	36	9	1	Norte -> Sur	Alamedas d 47
139	1	Activo	2022-11-05 20:23:58	40	10	1	Norte -> Sur	Alamedas d 47
138	1	Activo	2022-11-05 20:23:41	28	7	1	Norte -> Sur	Alamedas d 47
137	1	Activo	2022-11-05 20:23:28	48	12	1	Norte -> Sur	Alamedas d 47
136	1	Activo	2022-11-05 20:23:19	52	13	1	Norte -> Sur	Alamedas d 47
135	1	Activo	2022-11-05 20:22:56	36	9	1	Norte -> Sur	Alamedas d 47
134	1	Activo	2022-11-05 20:22:36	36	9	1	Norte -> Sur	Alamedas d 47
133	1	Activo	2022-11-05 20:22:22	36	9	1	Norte -> Sur	Alamedas d 47
132	1	Activo	2022-11-05 20:22:07	24	6	1	Norte -> Sur	Alamedas d 47
131	1	Activo	2022-11-05 20:21:48	48	12	1	Norte -> Sur	Alamedas d 47
130	1	Activo	2022-11-05 20:21:27	48	12	1	Norte -> Sur	Alamedas d 47
129	1	Activo	2022-11-05 20:21:08	44	11	1	Norte -> Sur	Alamedas d 47

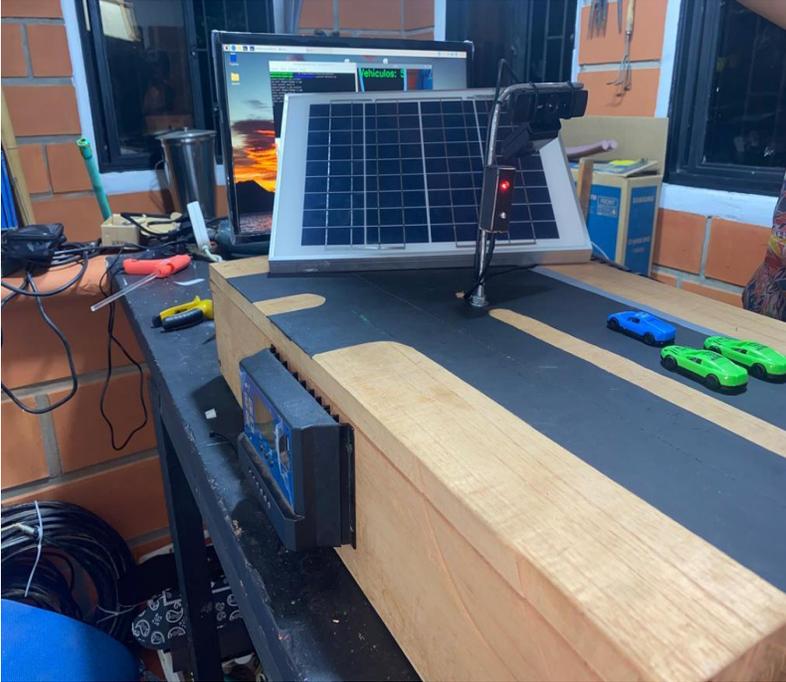
Nota. En la siguiente figura se puede observar dentro de la base de datos el estado del sistema cuando este encendido, se pueden apreciar las variables de salida como lo son el tiempo en verde, el número de vehículos, etc.

Figura 36.
Parte interna del proyecto



Nota. En la figura se puede observar la Raspberry donde los cables que están conectados en las salidas de los pines van directamente al sistema del semáforo y a su vez está siendo alimentada por el regulado DC que está conectada a la batería del sistema de energía solar.

Figura 37.
Parte principal del sistema



Nota. En esta figura se puede observar la parte principal el proyecto donde esta implementado todo el sistema desde el semáforo y la cámara para su respectiva detección hasta el regulador de voltaje de la batería y el panel solar.

5.CONCLUSIONES

En la investigación realizada se encontró que hay diferentes sistemas dedicados al control y monitoreo del flujo vehicular mediante sensores, fotos y videos.

Con el sistema propuesto se puede concluir que haciendo uso de un microordenador se pueden implementar muchas aplicaciones ya sea de uso institucional como también aplicaciones implementadas a la vida real para mejorar algún tipo de servicio o necesidad. Una de las grandes ventajas de estos microordenadores es que, al tener la oportunidad de ejecutar cualquier proyecto, estas placas son muy versátiles ya que ocupan un tamaño muy reducido, se caracterizan por ser una herramienta electrónica muy potente y su consumo de energía tiende a ser mínimo.

Muchos de estos sistemas ya se han implementado en grandes ciudades del mundo más que todo en las ciudades capitales ya que en estas ciudades tienen una mejor infraestructura vial óptima para el montaje de sistemas de detección de vehículos o de semaforización inteligente otro punto que influye mucho en estas ciudades es la cultura que se maneja.

Contar con un respaldo de energía solar permite que el sistema no pueda fallar en situaciones donde se puedan presentar fallas eléctricas ya que montería es muy habitual que pasen este tipo de inconvenientes. Por ende, el sistema puede evitar posibles accidentes o caos vehicular, incluir sistemas de energía solar en estos tipos de proyectos son muy beneficiosos para el medio ambiente ya que montería es una de las ciudades donde se puede aprovechar la radiación solar diaria por consiguiente nuestro sistema puede trabajar libremente sin necesidad de estar conectada a una red eléctrica.

Algunos de los aspectos negativos que se observaron al momento de estar trabajando en el proyecto es que para montar un sistema de semáforos inteligentes en la ciudad de Montería se puede dificultar por la cuestión de que en la ciudad diariamente circulan una gran cantidad de motocicletas que en la mayoría de las veces estas afectan o retrasa la movilidad como también que en Montería muchas personas no tienen cultura vial por ende no respetan las normas de tránsito. Por esta razón no se logró concretar uno de los objetivos propuestos que era medir la velocidad de los automóviles en la vía ya que por la simple razón de que si una motocicleta se atravesaba o si de pronto en la vía había un bache se perdía la detección del automóvil y a su vez no se podía calcular su velocidad.

Mediante se fue desarrollando nuestro sistema se presentó un inconveniente y fue que aunque nuestro sistema embebido es uno de los más actualizados en características su nivel de procesamiento para el código es algo lento ya que OpenCV maneja una base de datos muy extensa y por consiguiente atrasa un poco la respuesta final del sistema, se hicieron algunos ajustes al código para disminuir el tiempo de respuesta pero aun así no se pudo tener una respuesta inmediata para el sistema, se hizo la prueba por medio de una laptop con un nivel de procesamiento medio y este mostro los datos casi al instante, ante esto se recomienda trabajar con una placa que tenga mejores prestaciones de hardware .

6.RECOMENDACIONES

Ante la busca de una mejora para el proyecto se recomienda que al momento de seleccionar el sistema embebido este tenga un mayor nivel de procesamiento que el de la raspberri pi4 b+.

7.BIBLIOGRAFIA

- La razón. (2017). *Preocupa subregistro de parque automotor en Montería*.
<https://larazon.co/monteria/preocupa-subregistro-de-parque-automotor-en-monteria-sec-transito/>
- ¿Qué son los semáforos y cuál es su uso? (s.f.). Alcaldía de barranquilla
[Ultimo acceso: 2018] [https://www.barranquilla.gov.co/transito/que-son-los semáforos-y-cual-es-su-uso](https://www.barranquilla.gov.co/transito/que-son-los-semaforos-y-cual-es-su-uso)
- Alcaldía de Medellín. (s.f.). *Semáforos capítulo 7*.
https://www.medellin.gov.co/movilidad/documents/seccion_senalizacion/cap7_semaforos.pdf
- Race. (3 de septiembre de 2020). Tipos de semaforo que significa cada uno.
<https://www.race.es/tipossemaforos#:~:text=Franja%20horizontal%3A%20prohibido%20el%20paso,de%20una%20luz%20amarilla%20fija>
- Anorozo, M. M. (s.f.). *Semáforos Inteligentes*. [Proyecto de grado, Universidad católica nuestra señora de la asunción.]. Docplayer. <https://docplayer.es/21810495-Semaforos-inteligentes.html>
- Tecnipesa. (24 de marzo de 2021). *Que es y cómo funciona la tecnología RFID*.
<https://www.tecnipesa.com/blog/69-tecnologia-rfid-que-ventajas-tiene>.
- Yousef, K. M., Karaki, J., y Shatnawj, A. M. (20 de mayo). *Intelligent Traffic Light Flow Control System Using Wireless Sensors Networks. Research gate*.
https://www.researchgate.net/publication/220586947_Intelligent_Traffic_Light_Flow_Control_System_Using_Wireless_Sensors_Networks
- Chávez, C., (Julio 2015). *Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes*. [Proyecto de investigación, UNIVERSIDAD TÉCNICA DE AMBATO]. Repositorio Académico de la Universidad de Chile.
https://repositorio.uta.edu.ec/bitstream/123456789/13061/1/Tesis_t1033ec.pdf
- Luchetti, L. (1 de julio 2021). *Sistemas embebidos y sus características / Conceptos fundamentales*. Tribalyte technologies. <https://tech.tribalyte.eu/blog-sistema-embebido-caracteristicas>
- Pérez Porto, J., Gardey, A. (3 de septiembre de 2013). *Definición de protocolo de red - Qué es, Significado y Concepto*. Definicion.de . Recuperado el 16 de noviembre de 2022 de <https://definicion.de/protocolo-de-red/>

- Mendoza, M., L. (16 de julio de 2020). *Que es un lenguaje de programación* Openwebinars. <https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/>
- Gaviria, I., O., Pabon, C., S., y Caipe, R., M. (14 de diciembre de 2016). *Evaluación técnica mediante procesos de micro simulación en tres intersecciones en Montería – Córdoba*. https://www.researchgate.net/publication/320820089_Evaluacion_tecnica_mediante_procesos_de_micro_simulacion_en_tres_intersecciones_en_Monteria_Cordoba
- National Geographic. (9 de agosto de 2018). *los semáforos inteligentes*. <https://www.ngenespanol.com/el-mundo/los-semaforos-inteligentes/>
- El espectador. (29 de diciembre de 2017). *así funcionarían 1.500 semáforos inteligentes en Bogotá*. El espectador. <https://www.elespectador.com/bogota/asi-funcionarian-1500-semaforos-inteligentes-en-bogota-article-730880/#>
- Wang, F. E. (16 de mayo de 2022). *Este nuevo sistema de semáforos inteligentes con IA podría hacer que los atascos sean un recuerdo lejano*. DW made for minds. <https://p.dw.com/p/4BNgC>
- Glaskin, M. (12 de febrero de 2008). *Could smart traffic lights stop motorists fuming?*. Newsscientist. <https://www.newscientist.com/article/dn13306-could-smart-traffic-lights-stop-motorists-fuming/>
- De La Torre, A. (14 de febrero). *Menos tiempos de espera y más seguros: los resultados de aplicar inteligencia artificial a los semáforos*. Xataka. <https://www.xataka.com/movilidad/tiempos-espera-seguros-resultados-aplicar-inteligencia-artificial-a-semaforos>
- Rouhiainen, L. P. (2018). *Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro*. Editorial planeta. https://www.planetadelibros.com/libros_contenido_extra/40/39307_Inteligencia_artificial.pdf
- Bugeados. (18 de julio de 2020). *¿Cuál es el consumo de una raspberry pi 3/4?* <https://bugeados.com/raspberry/cual-es-el-consumo-de-una-raspberry-pi-3-4/>
- European commission. (1 de marzo de 2022). *Photovoltaic geographical information system*. https://re.jrc.ec.europa.eu/pvg_tools/en/
- Tony, G. (10 de mayo de 2021). *How to install OpenCV on Raspberry pi4*. <https://singleboardbytes.com/647/install-opencv-raspberry-pi-4.htm>
- Robledano, A. (24 de septiembre de 2019). *Que es MySQL características y ventajas*. <https://openwebinars.net/blog/que-es-mysql/>

Coppola, M. (20 de enero de 2023). *Que es un servidor web, para que sirve, como funciona y ejemplos.*
<https://blog.hubspot.es/website/que-es-servidor-web>

Hardware libre. (s.f). *GPIO: todo sobre las conexiones de la raspberri pi 4 y 3*
<https://www.hwlibre.com/gpio-raspberry-pi/>