UNIVERSIDAD POLITÉCNICA
SALESIANA SEDE GUAYAQUIL
CARRERA DE
TELECOMUNICACIONES

"DISEÑO DE UN PROTOTIPO PARA LA GESTIÓN Y ASIGNACIÓN DE PARQUEOS CON CAPTURA DE PLACAS VEHICULARES"

Trabajo de titulación previo a la obtención del

Título de ingeniero en telecomunicaciones

AUTORES: ALAN FERNANDO TERRANOVA VÉLEZ
ISAAC ANTHONY PÉREZ MALAVÉ

TUTOR: KLEVER CARRIÓN G, Msc

Guayaquil - Ecuador

2025

II. Certificado de responsabilidad y autoría del trabajo de titulación.

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.

Nosotros, Alan Fernando Terranova Vélez con documento de identificación N° 0955726666, e Isaac Anthony Pérez Malavé con documento de identificación N° 0953545878 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, a los 28 días del mes de agosto del año 2025.

Atentamente,

Alan Fernando Terranova Vélez
0955726666

Isaac Anthony Pérez Malavé
0953545878

III. Certificado de cesión de derechos de autor del trabajo de titulación a la Universidad Politécnica Salesiana.

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA.

Nosotros, Alan Fernando Terranova Vélez, con C.I. 0955726666 e Issac Anthony Pérez Malavé, con C.I. 0953545878, expresamos nuestra voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana del Ecuador la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Científico: "DISEÑO DE UN PROTOTIPO PARA LA GESTIÓN Y ASIGNACIÓN DE PARQUEOS CON CAPTURA DE PLACAS VEHICULARES", el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Institución facultada para ejercer plenamente los derechos concedidos.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, a los 28 días del mes de agosto del año 2025.

Atentamente,

Alan Fernando Terranova Vélez
0955726666

Isaac Anthony Pérez Malavé
0953545878

IV. Certificado de Dirección del Trabajo de Titulación.

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.

Yo, Klever Filiberto Carrión Gordillo, con documento de identificación N° 1102293402 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO DE UN PROTOTIPO PARA LA GESTIÓN Y ASIGNACIÓN DE PARQUEOS CON CAPTURA DE PLACAS VEHICULARES, realizado por Alan Fernando Terranova Vélez con documento de identificación N° 0955726666, y Isaac Anthony Pérez Malavé con documento de identificación N° 0953545878, obteniendo como resultado final el trabajo de titulación bajo la opción de Artículo académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, a los 28 días del mes de agosto del año 2025.

Atentamente,

Klever Filiberto Carrión Gordillo
1102293402

V. Dedicatoria

Dedico este trabajo de tesis a nuestros familiares, a nuestros padres y madres que a pesar de la enfermedad, la dificultad y la necesidad siempre estuvieron dando ánimos para poder terminar nuestra carrera, con dificultad y con poca motivación nos dieron un empujón y muchos ánimos para lograr concluir con esta etapa de la vida que se llama universidad, dedico este trabajo de tesis a nuestros hermanos que son una inspiración para cada uno de nosotros, aun siendo personas muy diferentes y con pensamientos distintos siempre nos hemos apoyado e inspirado mutuamente, dedico este trabajo de tesis a mis mascotas que siempre me han acompañado en cada mala noche en la que me he estado esforzando para realizar este documento y a lo largo de los 4 o 5 años de carrera que me he pasado mala noche ellos siempre estuvieron conmigo despiertos y por ultimo dedico este trabajo de tesis a mi primo Hugo Soledispa que fue un apoyo económico para estudiar estos 4 años de carrera, a pesar que estos momentos ya no se encuentre con nosotros busco que sus hijos me vean como inspiración y logren el mismo objetivo que yo estoy cumpliendo en este momento que, con esfuerzo, dedicación y sobre todo muchas ganas de no rendirse estoy logrando.

VI. Agradecimiento

Quiero expresar mi más sincero agradecimiento a mis amigos, Johnny, María Isabel, Sebastián, María Gabriela, que a pesar de que ya se han graduado o estemos en carreras distintas siempre estuvieron para nosotros para conversar o para tener un momento de alegría, quiero agradecer a los Ingenieros Pablo Echeverria, Holger Santillán y Klever Carrión que a pesar de todo han sido buenos tutores, profesores y amigos a lo largo de la carrera y quiero agradecer a la Universidad Politécnica Salesiana por darnos la oportunidad de utilizar sus instalaciones para ser mejores profesionales.

VII. Resumen

La cogestión vehicular es uno de los problemas más habituales en los parqueos privados como en plazas o centro comerciales. La falta de estacionamientos libres es uno de las principales razones para los atascos y esperas que rondan los treinta minutos a una hora de espera. Por lo tanto, esta investigación muestra un prototipo diseñado para asignar parqueos que usa Raspberry Pi, hecho para cuidar la reserva de espacios y mirarlos en tiempo real.

Este sistema tiene tres partes clave: la detección de proximidad del vehículo usando sensores que miden distancias con ondas ultrasónicas, el reconocimiento de caracteres de placa a través de una cámara que usa tecnología de OCR, y recogida automática de información. Al momento de realizar una reserva se almacena en un Excel en la Raspberry para posteriormente trasladarse al almacenamiento de la nube de Amazon web services.

Desde este lugar, la información se procesa en Amazon Athena y aparece en una página web creada con por un servidor Flask. Esta página deja ver una tabla de registros de puestos reservados. Los sensores ultrasónicos alcanzan una efectividad del 99.62 %, con una desviación aproximada de ±2 cm en la medición de distancias reales, lo que garantiza una detección precisa y confiable del estado de ocupación de los espacios.

1. Palabras Claves

a; Monitoreo b; Raspberry Pi c; Reserva de espacios d; OCR.

VIII. Abstract

Vehicle co-management is one of the most common problems in private parking spaces such as places or shopping centers. The lack of free parking is one of the main reasons for traffic jams and waits that range from thirty minutes to an hour. Therefore, this research shows a prototype designed to assign parking spaces that uses Raspberrand Pi, made to take care of the reservation of spaces and watch them in real time.

This system has three key parts: vehicle proximity detection using sensors that measure distances with ultrasonic waves, license plate character recognition through a camera that uses OCR technology, and automatic information collection. When making a reservation, it is stored in an Excel on the Raspberry and later transferred to the Amazon web services cloud storage.

From here, the information is processed in Amazon Athena and appears on a web page created by a Flask server. This page shows a table of reserved position records. The ultrasonic sensors reach an effectiveness of 99.62%, with an approximate deviation of ±2 cm in the measurement of real distances, which guarantees precise and reliable detection of the occupancy status of the spaces.

1. Keywords.

a; Monitoring b; Raspberry Pi c; Space reservation d; OCR.

IX. Índice de Contenido

X. Introducción

Según el artículo [1], un análisis presenta que cada año la cantidad de vehículos aumentara de forma alarmante tanto en escenarios controlados como agresivos debido al comercio y uso de los vehículos que han sido transformados a objetos indispensables. Como se indica en el artículo [2], el 30% de la congestión del tráfico se produce durante la búsqueda de un espacio de estacionamiento disponible afectando la fluidez y eficiencia para encontrar un lugar disponible.

El aumento de la demanda de vehículos amerita la implementación de soluciones automatizadas que faciliten la asignación de parqueos y evitar largos tiempos de espera. cómo sería el reconocimiento de caracteres de las placas vehiculares. Una tecnología que ha demostrado ser fundamental en el reconocimiento de matrículas es el reconocimiento automático de placas de matrícula (ANPR) [3].

La implementación de tecnologías como los sensores infrarrojos han aportado en la automatización de los estacionamientos [4]. El aporte general de este tipo de sensores es su tamaño reducido capaz de adaptarse a diferentes microcontroladores. La compatibilidad entre tecnologías es fundamental para la integración de los sistemas para asignación de parqueos y detección de placas vehiculares.

Una de las tecnologías mejor optimizada para la detección de caracteres de las placas es la arquitectura You Only Look Once (YOLO), la cual presenta un porcentaje de eficacia del 98.22%. En cambio, para la capacidad de identificación es del 78% en condiciones sin iluminación o números distorsionados [5].

A partir de los antecedentes, que resaltan la necesidad de soluciones tecnológicas que no solo cumplan la función de brindar un ticket de ingreso a un estacionamiento, sino que brinde información en tiempo real del procesamiento de la información. En este escenario, Se establece el diseño de un prototipo que concatene sensores ultrasónicos, interfaz de monitoreo y una cámara para capturas de imágenes.

La presente investigación gestiona el desarrollo de un prototipo con la capacidad de identificar y asignar puestos libres, capturar imágenes para registro de placas vehiculares, y almacena la información adquirida en la base de datos dentro de AWS (Amazon web services). Lo cual es presentado en una página web para el análisis de la asignación de estacionamientos.

Asimismo, busca reducir los tiempos de búsqueda de un lugar disponible dentro de los estacionamientos para controlar el congestionamiento en los parqueaderos privados. Al

desarrollar un prototipo tecnológico con almacenamiento de placa vehicular en una base de datos y asignación de puestos disponibles en el parqueo.

El prototipo planteado integra diversos módulos tecnológicos como lo son los sensores ultrasónicos HC-SR04, cámara de una laptop y una Raspberry Pi 4 como eje central de procesamiento de los datos y comunicación con la nube. Asimismo, los servicios de AWS como Bucket S3 y Amazon Athena garantizan un acceso seguro y escalable de datos.

Este procedimiento ayuda a tener un mejor control del espacio en los estacionamientos ya que la información del lugar sin uso será brindada al instante. De esta forma se da un aporte a la sociedad evitando el amontonamiento de vehículos y la disputa al buscar un lugar vacío para poder estacionar el vehículo [6].

1. Sensor ultrasónico HCSR04

Existen diferentes componentes tecnológicos en el prototipo de asignación de parqueos en los centros comerciales, tales como la figura 1 donde se muestra el sensor ultrasónico HC-SR04 el cual es de mucha importancia en la detección de objetos por medio de ondas ultrasónicas [7]. Existen artículos los cuales someten a prueba diversos componentes por lo que se tomó de referencia la investigación [8], en la cual se hicieron diez mil mediciones de distancia sobre un objeto fijo posicionado a 1 metro del sensor ultrasónico.

El nivel de ruido encontrado durante el proceso de medición fue muy pequeño, quedándose por debajo del 1% lo que muestra una buena precisión en la detección de objetos. Este sensor se ha usado en varias aplicaciones, tal y como lo expone el artículo [9] donde se utiliza para vigilar dónde llega el llenado de un contenedor de basura. En este caso, el umbral de alerta se pone cuando el 80% de lo que cabe del contenedor ha sido llenado, aun cuando este valor puede cambiarse por medio de codificación siguiendo las necesidades del sistema. Para más información de las características técnicas del Sensor ultrasónico HC-SR04 se detalla en el anexo 6.

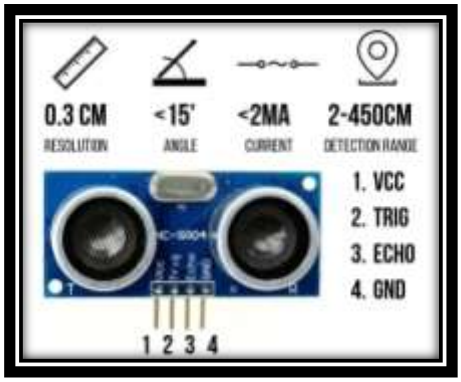

Figura 1: sensor ultrasónico.

2. Raspberry Pi

La figura 2 enseña cada uno de sus componentes internos del Raspberry Pi, se caracteriza por beneficiar a los profesionales de diversas áreas al momento de recolección de información o de ejecutar programas. El proceso de monitoreo remoto ayuda a que se reduzca las deficiencias experimentales [10] Los análisis presentan que esta placa tiene un rendimiento alto en velocidad, brindando una opción alterna económica en comparación a otros microcontroladores en el mercado [11].

Esta placa establece conexión a una red de internet, se presentan hojas de cálculo así mismo el procesamiento de textos y juegos [12]. Es una placa intuitiva con distintos pines de conexión y entradas para dispositivos externos. En la figura 1 se presenta la Raspberry Pi [13]. Para más detalle de la instalación del sistema operativo se describe en el anexo 1.

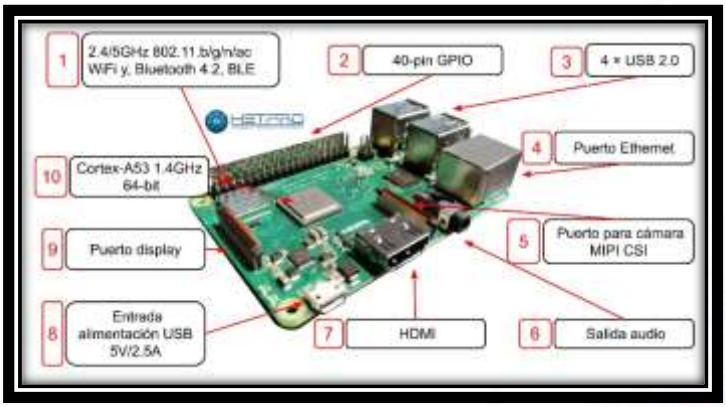

Figura 2: Raspberry Pi.

3. AWS (Amazon web servicies)

La figura 3 presenta el AWS que es un sistema de almacenamiento en la que forman una plataforma con diversas funciones como lo es el almacenamiento de información, base de datos, análisis de datos, tiene acceso remoto y muy seguro además que se puede integrar IOT para recibir datos en tiempo real, es usado en aplicaciones populares como Dropbox [14]. Para más información sobre la configuración de almacenamiento de AWS se especifica en el anexo 4.1.

Figura 3: AWS.

4. Diodos leds

La figura 4 muestra los diodos LEDs a su vez que nos ofrecen tipos de ventajas como el sistema de iluminación que ocupan menos espacios, ofrecen una mejor resolución y colores más nítidos y se logra bajar los costos de energía alrededor de un 40 % [15]. Para más información de las características técnicas del diodo LEDs se detalla en el anexo 6.

Figura 4: Diodos leds.

5. Thonny Emulator

En la codificación de los componentes tecnológicos, en la figura 5 utiliza Thonny emulador el cual trabaja como (IDE: entorno desarrollado integrado) con una programación de Python [16]. Adquiere una capacidad de tomar el código de forma específica, lo que permite a las personas visualizar lo que se está evaluando como las expresiones y cómo cambia el estado del programa en cada paso [17].

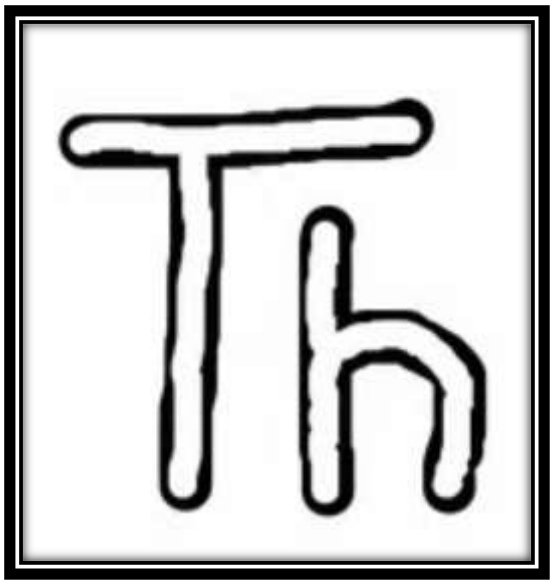

Figura 5: Thonny Emulator.

6. Amazon Bucket S3

La figura 6 expone el Amazon Bucket S3 que se trata de un servicio de almacenamiento ofrecido por AWS, Programado para almacenar información en diferentes formatos de archivos como csv o formato json. Presenta alta seguridad de acceso y escalable. Es posible direccionar los archivos por carpetas segmentadas para consultas desde otras herramientas de AWS [18]. Para más detalle sobre el almacenamiento de datos en Amazon Bucket S3 se explica en el anexo 4.2.

Figura 6: Amazon Bucket S3r.

7. Amazon Athena

La figura 7 nos presenta Amazon Athena, un servicio de análisis de datos que permite realizar consultas en lenguaje SQL de los datos almacenados en un Bucket S3 previamente diseñado para almacenamiento de información. Es de utilidad para realizar estudios de datos, generar reportes o implementar comunicaciones entre otras herramientas como lo sería una página web o presentaciones en dashboard [19]. Para mas detalle sobre la creación de la base de datos en Amazon Athena se define en el anexo 4.3.

Figura 7: Amazon Athena.

8. Resistencias de 220 Ohm

La figura 8 exhibe a la resistencia de 220 Ohm, su función es de limitar el paso de electricidad de un circuito, de esta manera asegurando la funcionalidad de los leds, evitando daños por exceso de corriente. Son comunes en los prototipos o proyectos como el planteado en esta investigación que utiliza un microcontrolador como la Raspberry.

Figura 8: Resistencias de 220Ohm.

9. Ecuaciones empleadas

9.1 Ecuación de distancia

Se muestra a continuación la distancia la cual es calculada por los sensores ultrasónicos donde la onda rebota en el objeto. El tiempo que transcurre de ida y vuelta se lo interpreta como t_{eco} y se multiplica por la velocidad del sonido sobre dos debido que la distancia real hasta objeto es la mitad del recorrido total de la onda, tal como se muestra en la ecuación 1.

$$D = \int_0^{t_{eco}} \frac{v}{2} dt \qquad (1)$$

En donde:

D:Distancia (cm).

v: Velocidad del sonido (cm/s).

t_{eco}: Tiempo que lleva en emitir y receptar una onda ultrasónica del sensor (s).

9.2 Ecuación de porcentaje de error de distancia

Utilizando la ecuación 2, se obtiene la divergencia entre las mediciones de los sensores y las medidas obtenidas con un flexómetro.

$$Porcentaje\ de\ error = \left|\frac{Ms-Mr}{Mr}\right| x100 \qquad (2)$$

En donde:

Ms: Medición simulada (cm).

Mr: Medición real (cm).

$\left|\frac{Ms-Mr}{Mr}\right|$: Corresponde al error absoluto. la divergencia entre la medición real y la medición simulada se realizó en Tinkercad.

XI. Artículos relacionados

En el artículo [20], presenta un sistema de reconocimiento de placas vehiculares con almacenamiento en la nube, usando IoT. Basándose en ANPR (Reconocimiento Automático de Matrículas) la cual mantiene una operación simultánea con una plataforma web en tiempo real teniendo un 98% de efectividad en el reconocimiento de placas en un entorno con buena iluminación y sin que tenga daños físicos la placa.

En el trabajo de investigación [21] se implementa la función de OCR con un 94.3% de efectividad en un entorno de igual forma controlado como lo seria con letras visibles y sin daños o decoloración. El artículo [22] presenta una plataforma web en el que se controla el ingreso y salida de los vehículos en los estacionamientos dependiendo de la disponibilidad existente en el momento.

Mantiene un 92% de efectividad en zonas urbanas con una conexión constante de internet. Este sistema permite generar reportes y estadísticas para un mejor análisis para futuras prevenciones de congestión vehicular. El artículo [23] estudia la manera de mejorar los sensores ultrasónicos de un costo bajo, centrados en la detección de proximidad de objetos. Las pruebas realizadas presentados una desviación estándar de 0.269 centímetros con un valor inferior al 1% de ruido.

Su principal limitante es en terreno inclinado ya que puede interferir con los datos que se reciben al retornar la onda. En el sistema propuesto en el artículo [24] combina sensores IoT y procesamiento de imágenes para asignar estacionamientos automáticamente. Esta implementación consigue un 90% de efectividad aplicando el uso de cámaras instaladas en cada puesto del estacionamiento.

XII. Metodología

Para la simulación del prototipo de estacionamiento autónomo para parqueaderos privados se usó una metodología experimental la cual comprende diferentes pruebas como las reales y las simuladas, que consisten en probarlos en varios escenarios, de esta manera, estando a prueba ayudan a comprobar la capacidad del sistema de parqueos en un ambiente ideal. Este proceso se fragmentó en diversas fases.

Lo primero es sobre el análisis de cuáles son los elementos compatibles con la placa Raspberry Pi. Entre los elementos tecnológicos a analizar fueron el sensor Ultrasónico, Cámara de una laptop, diodos led. Al tener una idea de cómo estos dispositivos son compatibles y se van a implementar en el prototipo.

Durante el desarrollo del prototipo, se realizó las conexiones de los componentes en la Raspberry Pi y diferentes codificaciones para su funcionamiento. El sistema utiliza de manera externa la cámara de una laptop configurada para la detección de placas y procesamiento de imágenes. También se utiliza un sensor de proximidad que realiza las respectivas validaciones de un espacio ocupado o libre.

La programación del software del prototipo utilizando Raspberry Pi, se ejecutó en Python, debido que es lenguaje ampliamente utilizado para integrar sensores junto con otros equipos anteriormente mencionados y también permite procesar datos en tiempo real. Python fue el lenguaje utilizado para que se habiliten las entradas de los sensores, de esta forma se generar la recopilación de datos.

La apariencia de la interfaz web es una parte fundamental del sistema, permitiendo un mayor control por parte de los guardias para tener la información en cualquier momento y lugar. La interfaz web es diseñada para la presentación del listado de las placas de los vehículos que ingresaron al estacionamiento. Al tener la información actualizada de los estacionamientos disponibles y ocupados, el prototipo asignará el lugar del estacionamiento disponible para su uso y no se volverá asignar hasta que se encuentre nuevamente disponible. Para mas detalle de la creación de la interfaz web se especifica en el anexo 5.

Los resultados obtenidos en la detección de proximidad definiendo una distancia de 10 centímetros es del 98% de efectividad al momento de que los sensores detectan un vehículo a una distancia de 2 centímetros, un escenario simulado en una maqueta. Los leds tienen un retardo de 1 segundo para encenderse y marcarlo como ocupado.

1. Esquema del prototipo para la gestión y asignación de parqueos con captura de placas vehiculares

Para el desarrollo de este prototipo se basó en el diseño de un sistema de asignación de parqueos en los centros comerciales, donde se integró varios componentes para la detección de espacios libres u ocupados, el cual permite mejorar la congestión de vehículos. En la figura 9 presenta un esquema grafico sobre las conexiones de los componentes en uso, como sensor ultrasónico para medición de distancia simulada de vehículos, diodos leds para visibilidad del estado del puesto, La cámara de la laptop se la utilizó para la detección de caracteres de la placa. Así mismo, presenta la ruta para cargar la información a la nube de AWS.

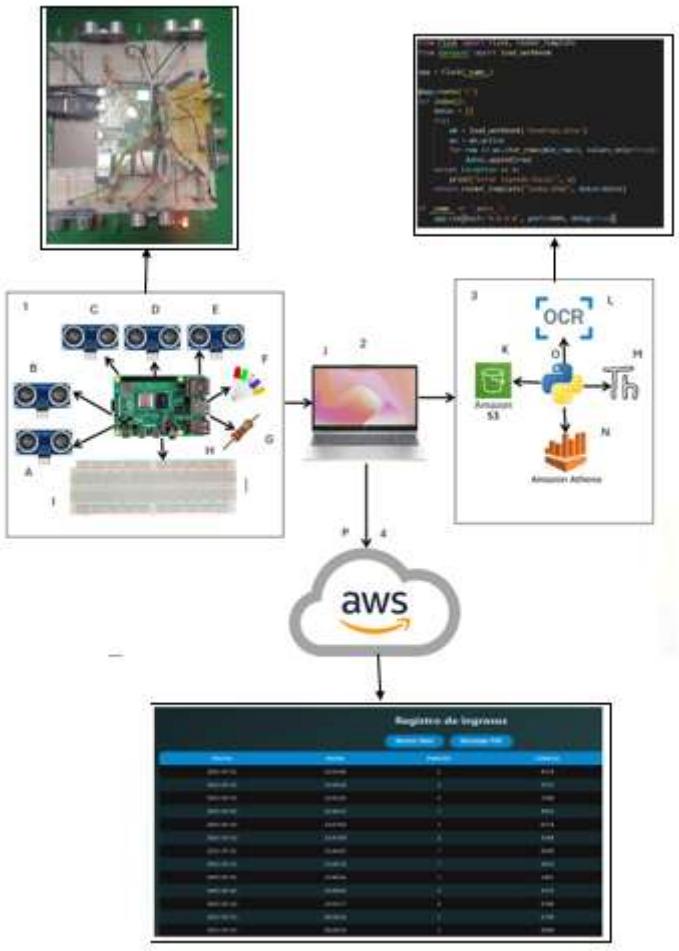

Figura 9 Esquema grafico del prototipo de asignación de puestos de parqueos.

Componentes:

a) Sensor ultrasónico 1.
b) Sensor ultrasónico 2.
c) Sensor ultrasónico 3.
d) Sensor ultrasónico 4.
e) Sensor ultrasónico 5.
f) 6 diodos leds.
g) 6 resistencias.
h) Raspberry Pi.
i) Protoboard.
j) Laptop.
k) Bucket S3.
l) OCR.
m) Thonny Emulator.
n) Amazon Athena.
o) Python.
p) AWS (Amazon web services).

En el primer punto se tiene la Raspberry Pi 4 la cual sirve para integrar las distintas funciones del prototipo como la codificación y conexión de componentes tecnológicos y las salidas físicas como los leds para identificar si el parqueo esta libre. En el segundo punto se usa la laptop para configurar de forma más cómoda y sencilla las parametrizaciones que tendrá el prototipo junto con su programación que será ejecutado en la Raspberry pi 4. La laptop es el centro de desarrollo, monitoreo y soporte del sistema.

En el tercer punto se usa el emulador Python ya que es totalmente compatible con la Raspberry pi 4 y tiene muchas librerías útiles a la hora de la captación de placas como OCR. Además, por si fuera poco, con las librerías RPI.GPIO o gpiozero se logra hacer que los leds y sensores tengan un correcto desenvolvimiento.

En el cuarto punto se usa el AWS para enviar la información de las reservas realizadas por el prototipo mediante un protocolo Message Queueing Telemetry Transport (MQTT). De tal manera permite crear un historial de los parqueos vehiculares que se asignaron, esto en conjunto con la cámara que la captura de caracteres de envía como notificación a Telegram al igual que la reserva para un monitoreo ordenado.

2. Etapas del Prototipo propuesto

Para que la elaboración del prototipo se realice de una forma correcta y con un buen porcentaje de efectividad se segmentó en etapas para optimizar el tiempo de ejecutar las actividades en curso. En la figura 10 se presentan Las etapas del desarrollo del prototipo que se tuvieron que cumplir para una correcta elaboración del prototipo.

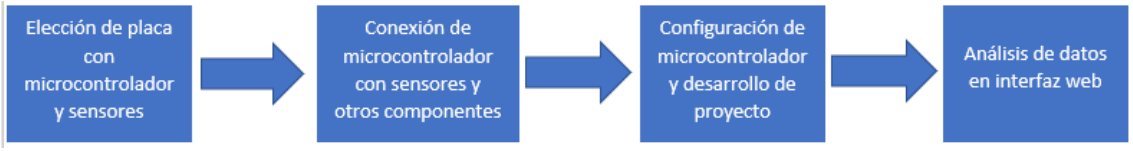

Figura 10 Etapas del desarrollo del prototipo.

En la primera etapa se selecciona el microcontrolador y los sensores dependiendo de la información sobre el proyecto que se va a realizar, de manera que, se diseña un prototipo que cumpla la función de asignar puestos en los parqueos. Este punto da inicio con un análisis destinado a las situaciones actuales en los países por el aumento de vehículos transitando en el día a día. Este prototipo propone una mejora en la búsqueda de un lugar disponible en poco tiempo.

En la conexión del microcontrolador y los sensores, se investiga arduamente los pines de conexión, además de la variedad de carpetas para el correcto funcionamiento del microcontrolador y sensores. También, se vincula con la cámara de una laptop con 9 megapíxeles y diodos led. Por lo tanto, se investiga y se deja conectado para una buena operación del prototipo. Para mas detalle del uso de las carpetas se explica en el anexo 2.

Esta herramienta tecnológica usa sensores de precisión que ayuda en la detección de movimiento del vehículo e interpretar el puesto del parqueo ocupado. Este componente funciona simultáneamente con los diodos Led para saber el estado del estacionamiento. El sensor ultrasónico, siendo relevante debido a su capacidad de medir distancias con la gran efectividad, presentando información confiable en tiempo real. Puede ser aplicado en pruebas en ambientes internos y externos.[17].

En la etapa de desarrollo y configuración del proyecto, se da uso a la placa base junto con sus librerías, las cuales hacen que los componentes tecnológicos tengan una correcta comunicación con la ejecución del programa en Thonny. Para la captura de caracteres se programa la cámara de una laptop por medio de codificación Python. Las imágenes son procesadas al instante permitiendo el reconocimiento de textos, siendo fundamental para captar la placa vehicular y poder ser enviadas a Telegram como notificación y seguimiento en conjunto con las reservas.

En la última etapa corresponde al análisis de los datos obtenidos, Este proceso permite revisar los datos obtenidos para asegurar que la implementación de los componentes esté generando datos reales y se mantenga una armonía con las demás conexiones en la placa base y asegurarse que no haya interferencia entre conexiones.

En la figura 11 literal a) se presentan las conexiones simuladas de los 4 pines del sensor ultrasónico los cuales ocupan las conexiones VCC y GND como alimentación y descarga a tierra. Los pines Trig y Echo mantienen la conexión en las conexiones GPIO 17 Y 27. El cátodo de los leds se conecta a la descarga a tierra y el ánodo se conecta al enlace en la resistencia para posteriormente ser concatenado a los pines de la Raspberry GPIO 23 Y 22. En el literal b) corresponde a las conexiones físicas de los componentes antes mencionados.

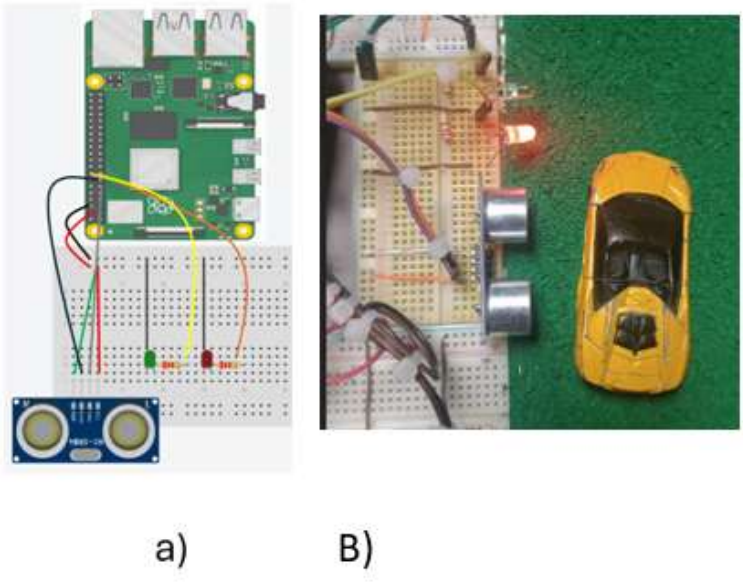

Figura 11 a) Conexiones simuladas del sensor ultrasónico 1, leds y resistencias a la Raspberry Pi 4. (b) Conexiones físicas del sensor ultrasónico 1, leds y resistencias a la Raspberry Pi 4

En la figura 12 literal a) se muestran las siguientes conexiones del sensor ultrasónico 2 el cual utiliza los pines GPIO 6 y 13 para la comunicación Trig y Echo. Asimismo, la conexión de los leds a su respectiva resistencia utilizando los pines de la Raspberry 16 y 26 GPIO. En el literal b) se presenta las mismas conexiones, pero ya en físico para gestionar los procesos respectivos.

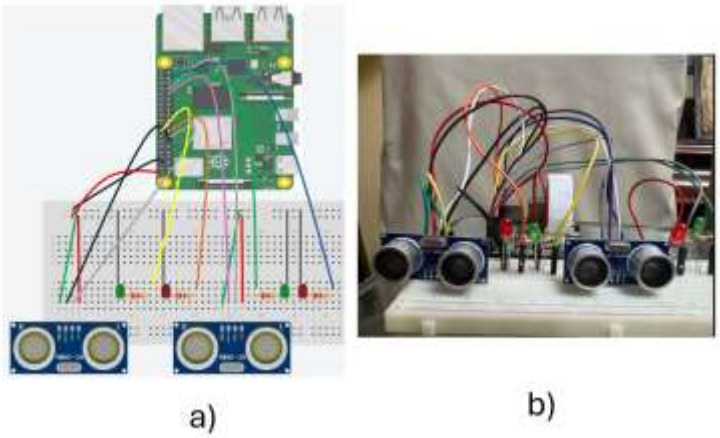

Figura 12 a) Conexiones simuladas del sensor ultrasónico 2, leds y resistencias a la Raspberry Pi 4. (b) Conexiones físicas del sensor ultrasónico 2, leds y resistencias a la Raspberry Pi 4.

En el literal a) de la figura 13 presenta la simulación completa de los sensores ultrasónicos en donde el tercer sensor usa los pines GPIO 5 Y 12 y los leds utilizan los pines 19 y 20 de la Raspberry Pi. El literal b) muestra las conexiones finales de los sensores en el prototipo.

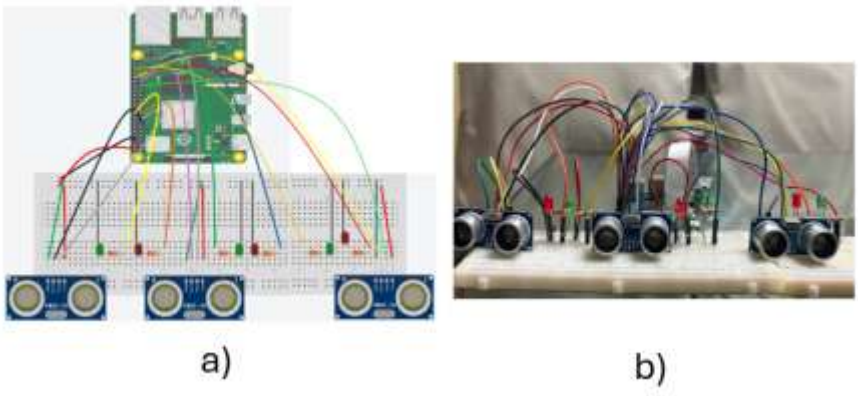

Figura 13 a) Conexiones simuladas del sensor ultrasónico 3, leds y resistencias a la Raspberry Pi 4. (b) Conexiones físicas del sensor ultrasónico 3, leds y resistencias a la Raspberry Pi 4

El proceso se repite 2 veces más para agregar sensores extras que mantendrán la misma función. El primer sensor extra estará usando los pines GPIO 14 y 15, en cambio sus leds trabajan con los pines 24 y 25. Asimismo, el segundo sensor extra trabaja con los pines 18 y 4, en cambio sus leds utilizan los pines 8 y 7. En la figura 14 se muestra como estas distribuido los cinco sensores ultrasonicos alrededor del microcontrolador Rasberry Pi, para más detalle del resultado final del prototipo se detalla en el anexo 3.

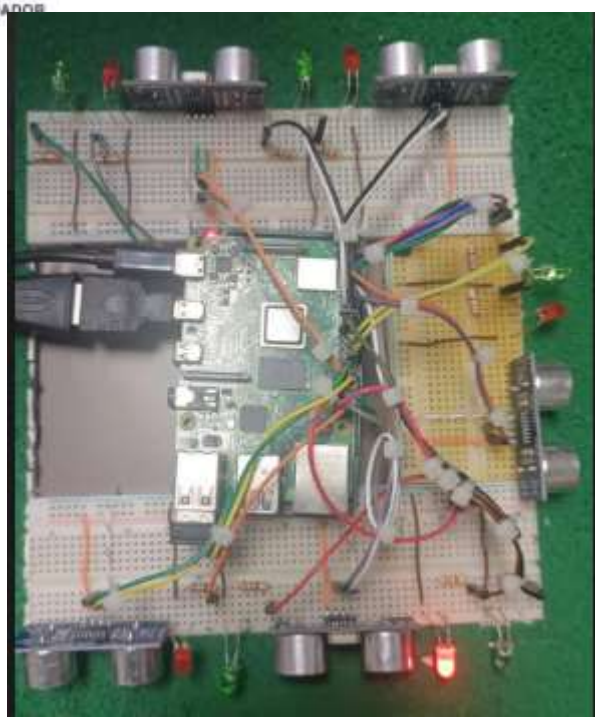

Figura 14 distribución de sensores ultrasónicos alrededor de la Rasberry Pi.

En la figura 15 se aprecia la codificación en Visual Studio y la ejecución del programa mediante el lenguaje de programación Python, el cual funciona para ejecutar un programa para la detección de caracteres de las placas por medio de la cámara integrada en la laptop.

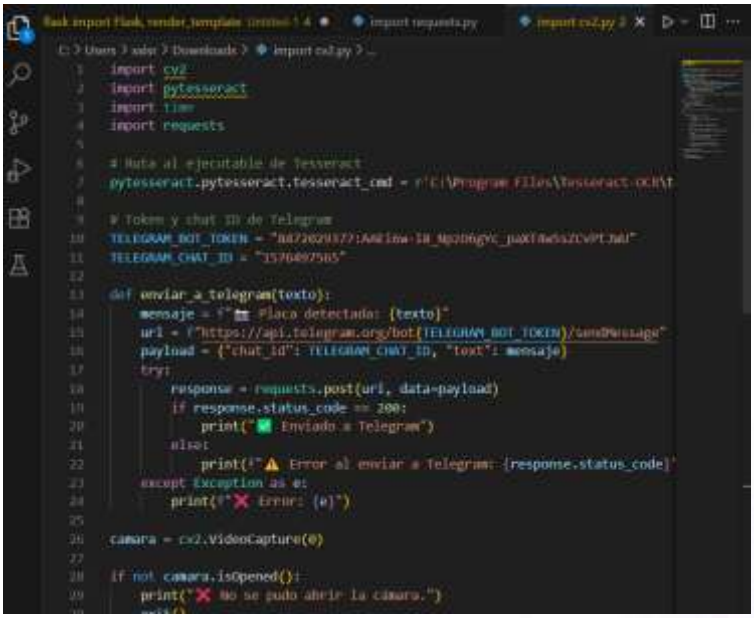

Figura 15 Codificación en Visual Studio para el uso de la cámara de la laptop para detección de caracteres.

3. Resultados

3.1 Datos simulados

3.1.1. Datos de la simulación con aproximación de vehículos a escala

En la figura 16 se observa la medición para obtener datos reales de los sensores ultrasónicos, se realizó cincuenta pruebas por cada sensor midiendo con una regla o flexómetro la distancia en la que se colocaba un objeto, en este caso un carro de juguete, en diferentes distancias midiéndolo en centímetros, de manera que se observa las diferentes posiciones del objeto y en que distancia se encontraba de los sensores ultrasónicos.

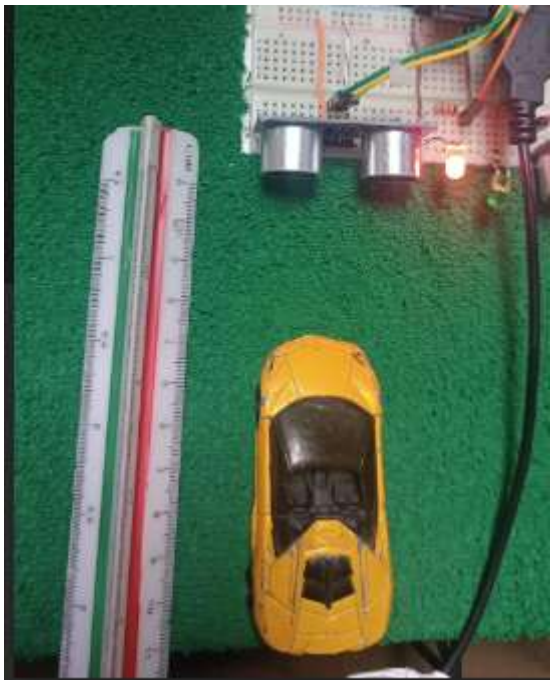

Figura 16 medición para obtener datos de los sensores ultrasónicos

La tabla 1 muestra la información adquirida en las pruebas individuales de los sensores de la detección de objetivos sin que el prototipo este operando al 100% de su capacidad teniendo una distancia de referencia de hasta 10 centímetros con los sensores ultrasónicos. Los datos representan los últimos cincuenta valores de reserva de estacionamientos. Las variaciones en los resultados indican que los sensores reaccionan de manera independiente en relación con las características de la superficie y del ángulo de incidencia.

Tabla 1 Tabla de medición de pruebas reales obtenidas con los sensores ultrasónicos 1, 2, 3, 4 y 5 en centímetros.

1er sensor en cm	2do sensor en cm	3er sensor en cm	4to sensor en cm	5to sensor en cm
9.72	9.66	9.91	9.75	9.30
9.34	9.85	9.76	9.62	9.84
9.64	9.48	9.51	9.94	9.51
9.85	9.59	9.37	9.42	9.76
9.02	9.78	9.55	9.10	9.98
8.96	9.21	9.23	8.98	9.22
9.67	9.47	9.44	9.86	9.41
9.30	9.65	9.31	9.70	9.33
9.55	9.24	9.27	9.59	9.88
9.21	9.37	9.42	9.27	9.92
9.28	9.52	9.26	9.38	9.71
9.63	9.18	9.33	9.53	9.69
9.84	9.73	9.59	9.87	9.94
9.76	9.68	9.50	9.45	9.60
9.20	9.15	9.16	9.22	9.38
9.42	9.66	9.34	9.65	9.17
9.91	9.70	9.73	9.91	9.79
9.38	9.94	9.86	9.36	9.88
9.11	9.58	9.75	9.11	9.92
9.79	9.81	9.90	9.78	9.70
9.95	9.72	9.93	9.94	9.84
9.49	9.67	9.89	9.52	9.98
9.36	9.63	9.61	9.35	9.66
9.10	9.20	9.34	9.10	9.23
9.00	9.10	9.29	9.00	8.94
9.03	9.05	9.26	9.03	9.09
8.95	8.90	9.10	8.95	9.20
8.85	9.00	8.99	8.85	9.48
9.12	9.28	9.22	9.12	9.54
9.09	9.11	9.15	9.09	9.33
9.18	9.14	9.05	9.18	9.12
8.90	8.89	8.92	8.90	8.84
9.01	9.08	9.01	9.01	9.07
8.75	8.93	8.95	8.75	8.93
8.91	8.96	8.97	8.91	9.02
8.86	8.75	8.82	8.86	8.73
8.79	8.85	8.86	8.79	8.98
8.80	8.70	8.79	8.80	8.95

8.68	8.64	8.70	8.68	9.06
8.94	8.66	8.61	8.94	8.99
9.03	8.55	8.54	9.03	8.80
8.67	8.53	8.48	8.67	8.95
8.95	8.45	8.47	8.95	8.93
8.85	8.38	8.32	8.85	9.04
8.93	8.20	8.30	8.93	8.97
9.01	8.15	8.12	9.01	9.11
9.12	8.08	8.04	9.12	8.92
8.99	8.01	8.06	8.99	9.06
9.03	7.98	7.92	9.03	9.13
9.08	7.91	7.89	9.08	9.22

En la figura 17, se observa los valores en una gráfica sobre los valores de las pruebas realizadas que mantiene relación a la tabla 1. Los datos presentados fueron adquiridos realizando pruebas independientes de cada sensor en su detección de proximidad y analizar si se cumple la función de encendido o apagado de los leds. Esta gráfica presenta en el eje de las Y la distancia en centímetro que fue detectada y en el eje de las X la cantidad de pruebas realizadas.

El color verde representa el puesto 1, el color azul corresponde al puesto 2, el color naranja es del puesto 3, el color amarillo es del puesto 4 y el color negro es del puesto 5. Cada prueba mantiene una divergencia de tiempo de 2 segundos para estabilizar la detección de objetos por los sensores, se aplica este método para obtener datos más certeros con el objeto de que se coloca al frente.

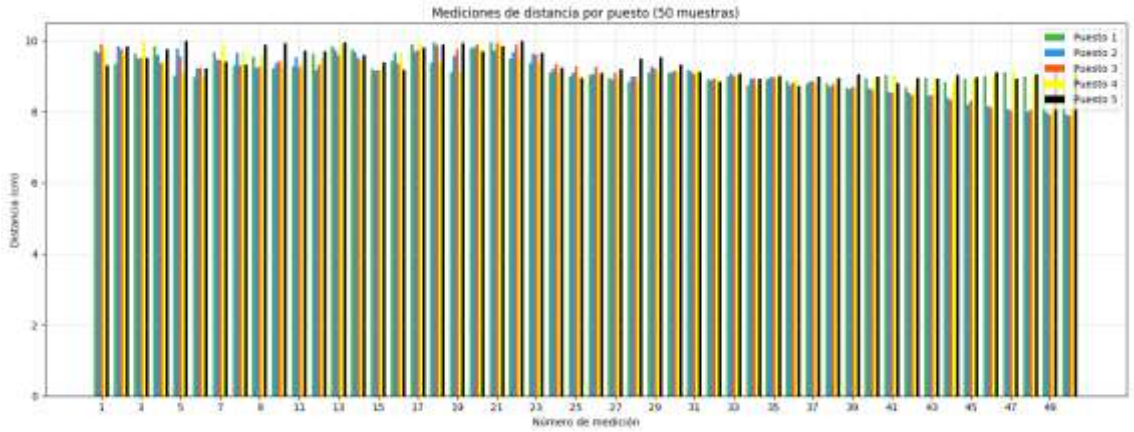

Figura 17 Gráfica de valores de las pruebas realizadas con los sensores que representan al puesto 1, 2 y 3.

3.2 Datos obtenidos del prototipo

3.2.1. Pruebas de reservas de puestos y comparativa de valores del prototipo y valores calculados con un flexómetro

La tabla 2 muestra las pruebas usando el 100% de la capacidad del prototipo usando la opción de reserva del puesto para indicar cuanto tiempo se demora en reservarse y en marcarse como ocupado, teniendo en cuenta la distancia de los objetos a los sensores al momento de ejecutar las funciones de prototipo, se aplicó esta función temporal para adquirir los datos en centímetros y analizar qué porcentaje de efectividad se mantiene. se marca como ocupado después de realizar la reserva, en el caso que se reserve y no se detecte la proximidad de 10 centímetros en un lapso de 30 segundos se marca como disponible el puesto y se enciende el led verde.

En el caso que se detecte una proximidad de 10 centímetros se enciende el led rojo. Se estableció esta distancia por las pruebas realizadas debido que al estar un objeto muy cercano causa interferencia en las ondas [25]. Al hacer esta modificación el porcentaje de error disminuyo a un 0.46% el cual se representa en el análisis de resultados. El lapso de los 30 segundos antes mencionados fue establecido en función de prueba y analizar que no exista alguna interferencia en la detección de objetos a una mayor distancia.

Tabla 2 Valores en centímetros de cada uno de los sensores ultrasónicos en la detección individual de proximidad.

Sensor 1 (cm)	Sensor 2 (cm)	Sensor 3 (cm)	Sensor 4 (cm)	Sensor 5 (cm)
4.00	5.32	4.00	8.38	9.46
4.00	4.00	4.00	9.67	8.23
4.47	4.63	4.88	7.19	8.91
5.98	4.00	4.00	8.90	6.33
4.00	4.00	5.14	9.35	8.74
4.00	4.00	4.00	8.18	9.93
4.00	4.00	4.00	9.88	9.65
5.12	4.00	4.73	9.53	8.75
4.00	5.00	4.00	8.46	9.55
4.00	4.00	4.00	9.45	9.80
4.00	4.00	4.62	9.17	8.56
4.00	4.00	4.00	8.60	9.14
4.00	4.29	4.00	9.94	9.13
4.00	4.00	4.00	9.02	9.82
4.00	4.00	5.91	8.36	9.51
4.38	4.00	4.00	8.62	9.67
5.73	4.00	4.55	8.74	9.02

4.00	4.00	4.00	9.73	8.99
4.00	5.31	4.00	9.87	8.40
4.00	4.00	4.87	8.48	9.28
5.66	4.00	4.00	8.71	9.94
4.00	4.00	4.00	9.38	9.75
4.00	4.00	5.00	8.05	9.36
4.00	5.41	4.00	8.80	9.17
4.00	4.00	4.12	9.30	9.70
4.00	4.00	4.00	8.01	8.88
4.00	4.95	4.00	8.65	9.63
5.00	4.00	5.47	9.18	9.85
4.00	4.00	4.35	8.79	9.09
4.00	4.00	4.00	9.71	9.78
4.00	4.00	4.00	9.21	9.26
4.00	4.00	4.00	8.27	9.44
4.69	5.56	4.00	8.66	9.76
4.00	4.00	4.00	9.38	9.29
4.00	4.00	4.00	8.70	9.00
4.00	4.00	4.00	8.95	9.38
4.00	4.77	4.00	8.77	9.52
5.21	4.00	4.00	9.27	9.59
4.00	4.00	4.00	9.05	8.92
4.00	4.00	4.00	8.88	9.39
4.84	5.12	4.00	9.60	9.91
4.00	4.00	4.00	9.12	9.74
4.00	4.00	5.66	8.97	8.63
4.90	4.00	4.00	9.49	9.57
4.00	4.42	4.00	9.20	9.33
4.00	4.00	4.00	9.61	8.95
4.00	4.00	4.26	9.44	9.86
4.00	4.00	4.00	9.79	8.60
5.89	5.67	4.00	9.37	9.12
4.00	4.00	4.00	9.00	9.25

En la figura 18, de igual manera, muestra una gráfica de valores adquiridos en el entorno de prueba y los valores obtenidos en centímetros de los sensores, en donde se observa las variaciones de los picos debido a la variación de proximidad de vehículo a escala u objeto. Las barras a la altura de 10 centímetros corresponden a los vehículos que han ocupado el lugar, los picos más altos corresponden a una detección de proximidad más lejana con un tope de 10 cm.

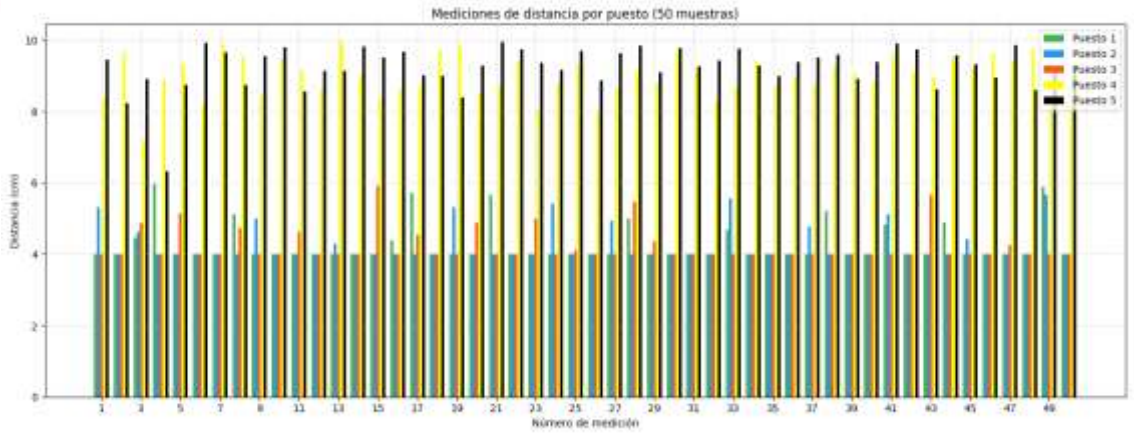

Figura 18 Gráfica de valores adquiridos en entorno de prueba de la proximidad detectada por lo sensores ultrasónicos.

En la figura 15 se realiza las mediciones manuales con la ayuda de un flexómetro los cuales se muestran en la tabla 2. En el instante que se realizaron las pruebas del prototipo para evidenciar que diferencia existía entre los datos que ofrece el prototipo y los datos manuales. Al realizar el proceso se evidencia una divergencia del 0.38%, este valor se obtiene utilizando una ecuación 2. Esto representa un valor aceptable para iniciar incorporaciones a mayor escala.

Tabla 3 Datos reales de detección de proximidad de objetos para generar comparativa con los valores que presentan los sensores ultrasónicos en modo de prueba.

Sensor 1 (cm)	Sensor 2 (cm)	Sensor 3 (cm)	Sensor 4 (cm)	Sensor 5 (cm)
4.12	5.20	4.05	8.30	9.50
4.05	4.15	4.10	9.50	8.30
4.50	4.60	4.95	7.25	8.85
5.90	4.05	4.10	8.80	6.40
4.05	3.95	5.10	9.40	8.70
4.00	4.10	4.05	8.20	9.90
3.95	4.00	4.00	9.85	9.60
5.10	4.10	4.70	9.50	8.70
4.10	5.05	4.00	8.50	9.60
4.00	3.95	4.10	9.40	9.75
4.05	4.05	4.60	9.10	8.60
4.00	4.00	3.95	8.60	9.10
4.00	4.30	4.00	9.95	9.10
4.00	4.00	4.00	9.00	9.85
4.00	4.00	5.90	8.35	9.50
4.40	4.00	4.00	8.60	9.70
5.70	4.05	4.50	8.75	9.00
4.00	3.95	4.00	9.75	9.00

4.05	5.30	4.00	9.85	8.45
4.00	4.00	4.90	8.50	9.30
5.65	4.00	4.00	8.70	9.90
4.00	4.00	3.95	9.40	9.70
4.00	4.00	5.05	8.00	9.35
4.00	5.40	4.00	8.75	9.20
4.00	4.00	4.10	9.35	9.65
4.00	4.00	4.00	8.00	8.90
4.00	4.95	4.00	8.60	9.65
5.05	4.00	5.45	9.20	9.80
4.00	4.05	4.35	8.75	9.10
4.00	4.00	4.00	9.70	9.75
4.00	3.95	4.00	9.20	9.30
4.00	4.00	3.95	8.25	9.45
4.65	5.55	4.05	8.65	9.80
4.00	4.00	4.00	9.40	9.25
4.05	4.05	3.95	8.70	9.05
4.00	4.00	4.00	8.90	9.40
4.00	4.75	4.00	8.80	9.55
5.20	4.00	4.00	9.25	9.60
4.00	4.00	4.00	9.05	8.90
4.00	4.00	4.05	8.85	9.40
4.85	5.10	4.00	9.55	9.85
4.00	4.00	4.00	9.15	9.70
4.00	4.00	5.65	8.90	8.60
4.90	4.00	4.00	9.50	9.55
4.00	4.45	4.00	9.25	9.35
4.00	4.00	4.00	9.60	8.95
4.00	4.00	4.25	9.40	9.85
4.00	4.00	4.00	9.80	8.60
5.85	5.65	4.00	9.35	9.10
4.00	4.00	4.00	9.00	9.20

La figura 19 presenta una gráfica de valores en centímetros adquiridos por un flexómetro, los mismos valores de la tabla 2 pero de forma más amigable, la información presentada en este apartado no muestra mucha diferencia en relación con los valores presentados en la tabla 2. Esto se debe a un correcto funcionamiento de los sensores ya que generan un valor cercano al real.

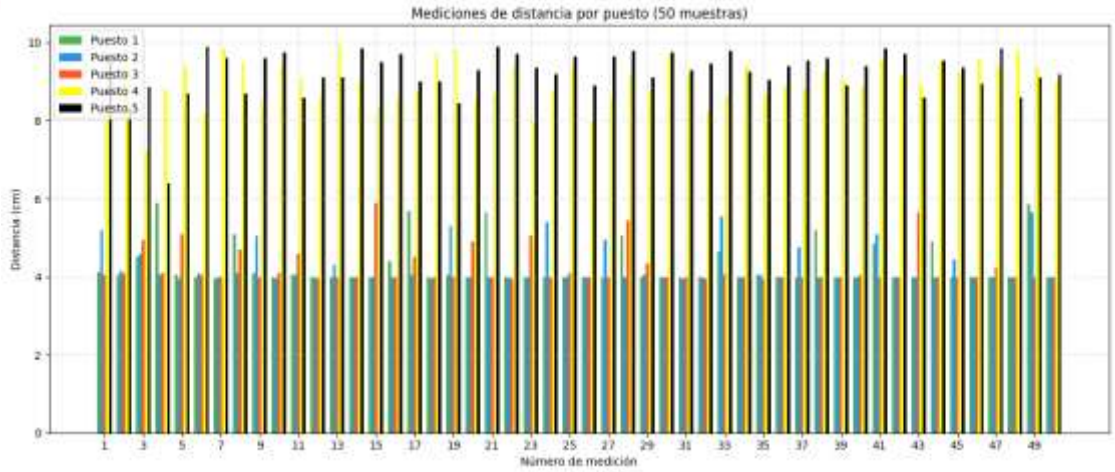

Figura 19 Gráfica de valores en cm adquiridos con un flexómetro.

3.2.2 Evaluación estática de detección de caracteres

En la figura 20 Se presenta la forma en la que la cámara captura la imagen al momento de generar la reserva de un estacionamiento. La distancia en la que se ubica es a 15 centímetros de la cámara para poder tener un mejor rendimiento en la identificación de caracteres. Se estableció esta distancia por las diversas pruebas realizadas a ciertas distancias para poder encontrar la más óptima.

Figura 20 Captura de imagen al momento de realizar la reserva del puesto en el estacionamiento.

La información que se capta al tomar la foto es enviada a Telegram en conjunto con la reserva. De esta manera es posible presentar en una interfaz web la información de las reservas de los vehículos y su asignación de estacionamientos y por medio de Telegram hacer la revisión de las placas de los vehículos ingresados con la reserva. Con este proceso se podrá llevar un control en el momento que se tenga que implementar a gran escala.

En la figura 21 se observa el momento en que se realiza la reserva del estacionamiento y se presenta en la interfaz de la asignación de parqueos del puesto que se asigna, el código aleatorio y la placa que corresponde. Estos datos son almacenados y presentados para revisión y monitoreo de los datos registrados.

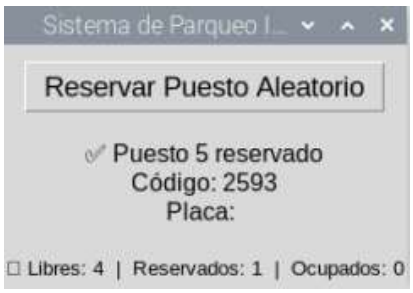

Figura 21 Presentación de reserva del puesto con la detección de la placa.

La interfaz web presenta los datos recopilados de la reserva de estacionamientos como lo son el día, hora, puesto y código. Esta información como antes fue mencionado es de utilidad para gestionar una revisión del flujo en la reserva de estacionamientos como muestra la figura 22.

Figura 22 Datos recopilados por el prototipo y presentados en la interfaz web.

4. Análisis de resultados

4.1 Cálculo de detección de proximidad de sensores ultrasónicos.

En el momento que se reciben los datos del prototipo que se presentan en la tabla 2, se coloca $\frac{1}{2}$ a la ecuación, esto por el motivo que la distancia del auto a escala corresponde a la mitad del desplazamiento de la onda ultrasónica. Se realiza el calculo de la medición de distancia implementando la ecuación 1.

Por lo tanto:

$3400\frac{cm}{s}$: Velocidad del sonido (cm/s).

$0.01\,s$: Tiempo que demora en emitir y recibir la onda ultrasónica del sensor (s).

$$D = \int_0^{t_{eco}} \frac{3400\frac{cm}{s}}{2} dt$$

$$D = \frac{3400\frac{cm}{s}}{2} \int_0^{t_{eco}} dt$$

$$D = 17150\frac{cm}{s} x(0.10s - 0)$$

$$D = 1.71\text{cm}$$

Al usar la ecuación 1 se tiene la posibilidad de conseguir la distancia entre el sensor y el vehículo a escala en función del tiempo del recorrido de la señal, esto con la finalidad de corroborar que los valores sean lo más próximo a la realidad. En la tabla 4, se muestran ciertos valores de prueba para corroborar que funcione en la detección de proximidad del vehículo a escala, estos datos son informativos para analizar el correcto funcionamiento de los sensores.

Tabla 4 Mediciones de pruebas físicas aplicando el cálculo de la ecuación 1 utilizando variables de distancia, velocidad y recorrido de la onda.

Mediciones de prueba en cm
1,71
4,56
2,63
5,35
8,84

4.2 Cálculo de porcentaje de error entre simulación y pruebas

En el momento que se tienen los valores del prototipo y las mediciones reales se procede a realizar las comparativas y sacar qué tan cercanos son los datos presentados. Para este proceso se implementa la ecuación 2. Los valores que se reemplazan son de "Ms" correspondiente a medición simulada y "Mr" correspondiente a medición real, el resultado se lo multiplica por cien para sacar el porcentaje de error de cada sensor.

$$Porcentaje\ de\ error = \left|\frac{Ms - Mr}{Mr}\right| x100$$

$$Porcentaje\ de\ error = \left|\frac{4cm - 4.10cm}{4.10cm}\right| x100$$

$$Porcentaje\ de\ error = 2.44\%$$

El proceso de la ecuación 2 se lo replica en cada valor de la tabla 2 y 3 para obtener un resultado final y analizar qué porcentaje de divergencia existe en promedio de los sensores. En la tabla 5 se presenta los valores individuales de divergencia de los sensores para obtener el valor general el cual corresponde al 0.38% de error entre los 5 sensores. Se procedió a sacar los valores de cada dato y al finalizar se sumaron los valores para sacar un porcentaje general por sensor.

Tabla 5 Porcentaje de error independiente de los sensores aplicando la ecuación 2 con cada valor obtenido.

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5
0,49%	0,43%	0,44%	0.34%	0.22%

4.3 Análisis de detección de placas

En el momento de realizarse las pruebas se analizaron cincuenta ejemplos de placas vehiculares de las cuales se determina que tiene un alcance de 40 centímetros en condiciones normales y luz directa. Esta distancia se obtuvo al analizar los resultados de la detección de caracteres en varias posiciones en conjunto de una buena iluminación directa para que la calidez de la imagen aumente.

En la figura 23 se aprecia como la cámara captura un texto y se envía a Telegram en el momento que se pone en frente una placa o algún texto. La lectura de caracteres debe ser estática debido que se estableció un rango de captura de imágenes de 5 segundos para evitar que se esté iniciando constantemente y se envié información no necesaria.

Figura 23 Presentación de detección de caracteres y envió a Telegram.

XIV. Discusión

A diferencia del prototipo propuesto en el artículo [26], se compara el uso manual con el automatizado dando como resultado una mejora de hasta un 23.83% dado que se utiliza sensores de proximidad con un costo más bajo y con una mejor eficiencia para identificar objetos a una distancia más corta. El artículo muestra una comparativa sobre el uso de cámaras, aunque muestra una mayor efectividad este valor depende de una buena iluminación y con el uso de cámaras costosas para obtener imágenes de mejores ángulos y resolución.

El prototipo planteado en este trabajo es más económico y menos susceptible a entornos de difícil visibilidad con una cámara estable y apuntando en una sola dirección, se realizó cincuenta pruebas para identificar la distancia correcta de la cámara que tiene que ser no mayor a 30 centímetros para un alto rendimiento.

El sistema del artículo [27] implementa tecnología RFID (Identificación por Radiofrecuencia) que facilita el control de ingreso y salida de los vehículos en conjunto con una aplicación móvil, pero no presenta un reconocimiento de placas para el control de ingreso y una asignación de parqueos, teniendo como dificultad la aglomeración de vehículos. Además, su efectividad depende de la correcta lectura de las tarjetas RFID y no tiene un interfaz web que almacena una base de datos con la hora, la fecha y la placa vehicular al momento del ingreso.

Este prototipo presenta la función de almacenamiento de datos en la nube de Amazon web services y control de estacionamiento mediante detección de proximidad y reservas, además de tener el uso de una cámara para la detección de placas vehiculares. El articulo [2] realiza procesos similares implementando redes neuronales lo cual requiere un entrenamiento previo y alto costo. A pesar de eso, mantiene un 95% de efectividad.

sin embargo, se centra especialmente en parqueos en zonas abiertas, debido a que no tiene una exactitud de los vehículos que están buscando parqueo utiliza predicciones de disponibilidad de espacios de estacionamiento en varias ubicaciones, en diferencia con nuestro prototipo propuesto que esta más centrado en parqueos privados tiene la exactitud de cual parqueo está disponible y cuando un vehículo lo necesita al momento de ser asignado.

En el análisis del artículo [28], presenta un análisis de identificación de caracteres de las placas de los vehículos el cual se obtuvo valor de efectividad entre un 96% y 98% en la identificación de la placa del automóvil mediante ANPR. Esto permite la segmentación de los estacionamientos.

XV. Conclusiones

En el diseño del prototipo de asignación de parqueos según los puestos disponibles ha tenido una operación correcta en base a la detección de proximidad de los sensores ultrasónicos presentando un 99.62% de efectividad entre los cinco sensores, los cuales funcionan en conjunto con los diodos leds para indicar en tiempo real el estado de cada puesto.

Por otra parte, La configuración para la detección de caracteres basado en la cámara de una laptop con una resolución de 1280x720 permitió identificar los textos en un rango máximo de 45 centímetros, lo cual demostró limitaciones en condiciones de baja iluminación o el texto no se encuentra bien enfocado.

El análisis determino que la distancia ideal es de 30 centímetros, dando como resultado que la detección de caracteres es eficiente únicamente en entornos controlados como la iluminación y el posicionamiento del texto, sin embargo, esta cámara mantiene 9 megapíxeles con un procesamiento de imágenes de 30 cuadros por segundo, lo cual afecta a la fluidez en la visualización de la imagen [29]. Por tal motivo se optó por realizar la detección de caracteres de forma estática.

XVI. Recomendaciones

Se sugiere que se analicen otro tipo de módulos que puedan unificar las operaciones de identificación de proximidad y que permita la integración con una inteligencia artificial sin configuraciones externas. Estas opciones permitirán que se reduzca el tamaño de elaboración y que se permita un mejor procesamiento al tener los módulos unificados.

El prototipo planteado puede tener mayor desarrollo si se usan cámaras en cada puesto de estacionamiento para que se pueda configurar inteligencia artificial para detectar características específicas de autos y así no generar falsos reconocimientos. Así mismo, implementar el proceso con un microcontrolador como la Raspberry pi pico y hacer un sistema individual por parqueo teniendo una comunicación en red.

Asimismo, se aconseja el uso de la inteligencia artificial ya que se puede entrenar con un sin número de fotos de letras para que así a pesar de estar en movimiento o a una gran distancia, esta pueda de igual manera identificar las letras. Para el desarrollo del prototipo es viable manejar una placa con una capacidad de rendimiento mayor para reducir la saturación de la operación de los programas de los componentes y de tal manera que no exista fallas en el funcionamiento de los dispositivos tecnológicos a implementarse.

XVII. Referencias

[1] N. Singh, T. Mishra, y R. Banerjee, "Projection of Private Vehicle Stock in India up to 2050", *Transportation Research Procedia*, vol. 48, pp. 3380–3389, 2020, doi: 10.1016/j.trpro.2020.08.116.

[2] G. Ali *et al.*, "IoT Based Smart Parking System Using Deep Long Short Memory Network", *Electronics (Basel)*, vol. 9, núm. 10, p. 1696, oct. 2020, doi: 10.3390/electronics9101696.

[3] A. Ditta, M. M. Ahmed, T. Mazhar, T. Shahzad, Y. Alahmed, y H. Hamam, "Number plate recognition smart parking management system using IoT", *Measurement: Sensors*, vol. 37, p. 101409, feb. 2025, doi: 10.1016/j.measen.2024.101409.

[4] P. G., H. R G., y D. R S., "Automated Parking System using IR Sensor", *IRO Journal on Sustainable Wireless Systems*, vol. 6, núm. 3, pp. 211–220, sep. 2024, doi: 10.36548/jsws.2024.3.002.

[5] Hendry y R.-C. Chen, "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning", *Image Vis Comput*, vol. 87, pp. 47–56, jul. 2019, doi: 10.1016/j.imavis.2019.04.007.

[6] H. Y. Moreno Agualimpia y R. S. Hernández Gómez, "Plataforma WEB para la automatización y control integral de parqueaderos Arparking", *Encuentro SENNOVA del Oriente Antioqueño*, vol. 10, núm. 1, pp. 5–18, dic. 2024, doi: 10.23850/es.v10i1.6441.

[7] S. G. Valenzuela-Ramírez, A. Contreras-Basurto, y E. A. Rivera-Landeros, "Práctica sensor ultrasónico y buzzer con Arduino", *Ingenio y Conciencia Boletín Científico de la Escuela Superior Ciudad Sahagún*, vol. 11, núm. 21, pp. 143–146, ene. 2024, doi: 10.29057/escs.v11i21.11728.

[8] D. Abreu, J. Toledo, B. Codina, y A. Suárez, "Low-Cost Ultrasonic Range Improvements for an Assistive Device", *Sensors*, vol. 21, núm. 12, p. 4250, jun. 2021, doi: 10.3390/s21124250.

[9] V. R. Ravi, M. Hema, S. SreePrashanthini, y V. Sruthi, "Smart bins for garbage monitoring in smart cities using IoT system", *IOP Conf Ser Mater Sci Eng*, vol. 1055, núm. 1, p. 012078, feb. 2021, doi: 10.1088/1757-899X/1055/1/012078.

[10] J. W. Jolles, "Broad-scale applications of the Raspberry Pi: A review and guide for biologists", *Methods Ecol Evol*, vol. 12, núm. 9, pp. 1562–1579, sep. 2021, doi: 10.1111/2041-210X.13652.

[11] C. A. Nugroho y A. Amiruddin, "A Low-cost Disk Imaging Device Using Raspberry Pi 4 for Digital Forensics", en *2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs)*, IEEE, ago. 2023, pp. 314–319. doi: 10.1109/ICoCICs58778.2023.10276963.

[12] H. Jasim Habil, Q. A. Al-Jarwany, M. Nema Hawas, y M. Jabbar Mnati, "Raspberry Pi 4 and Python Based on Speed and Direction of DC Motor", en *2022 4th Global Power, Energy and Communication Conference (GPECOM)*, IEEE, jun. 2022, pp. 541–545. doi: 10.1109/GPECOM55404.2022.9815716.

[13] S. Karthikeyan, R. A. Raj, M. V. Cruz, L. Chen, J. L. A. Vishal, y V. S. Rohith, "A Systematic Analysis on Raspberry Pi Prototyping: Uses, Challenges, Benefits, and Drawbacks", *IEEE Internet Things J*, vol. 10, núm. 16, pp. 14397–14417, ago. 2023, doi: 10.1109/JIOT.2023.3262942.

[14] W. Villegas-Ch, J. Govea, y I. Ortiz-Garces, "Developing a Cybersecurity Training Environment through the Integration of OpenAI and AWS", *Applied Sciences*, vol. 14, núm. 2, p. 679, ene. 2024, doi: 10.3390/app14020679.

[15] S. Flores Pérez, A. M. Castillo-González, L. A. Valdez-Aguilar, y E. Avítia-García, "Uso de diferentes proporciones de led rojos y azules para mejorar el crecimiento de Lilium spp", *Rev Mex De Cienc Agric*, vol. 12, núm. 5, pp. 835–847, ago. 2021, doi: 10.29312/remexca.v12i5.2607.

[16] A. Kurniawan, "Programming on Raspbian OS", en *Raspbian OS Programming with the Raspberry Pi*, Berkeley, CA: Apress, 2019, pp. 79–96. doi: 10.1007/978-1-4842-4212-4_3.

[17] A. Annamaa, "Introducing Thonny, a Python IDE for learning programming", en *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, New York, NY, USA: ACM, nov. 2015, pp. 117–121. doi: 10.1145/2828959.2828969.

[18] A. Kulkarni, "Amazon Redshift: Performance Tuning and Optimization", *International Journal of Computer Trends and Technology*, vol. 71, núm. 02, pp. 40–44, feb. 2023, doi: 10.14445/22312803/IJCTT-V71I2P107.

[19] A. Kulkarni, "Amazon Athena : Serverless Architecture and Troubleshooting", *International Journal of Computer Trends and Technology*, vol. 71, núm. 5, pp. 57–61, may 2023, doi: 10.14445/22312803/IJCTT-V71I5P110.

[20] A. Ditta, M. M. Ahmed, T. Mazhar, T. Shahzad, Y. Alahmed, y H. Hamam, "Number plate recognition smart parking management system using IoT", *Measurement: Sensors*, vol. 37, p. 101409, feb. 2025, doi: 10.1016/j.measen.2024.101409.

[21] A. Vaishnav y M. Mandot, "Template Matching for Automatic Number Plate Recognition System with Optical Character Recognition", 2020, pp. 683–694. doi: 10.1007/978-981-13-7166-0_69.

[22] H. Y. Moreno Agualimpia y R. S. Hernández Gómez, "Plataforma WEB para la automatización y control integral de parqueaderos Arparking", *Encuentro SENNOVA del Oriente Antioqueño*, vol. 10, núm. 1, pp. 5–18, dic. 2024, doi: 10.23850/es.v10i1.6441.

[23] D. Abreu, J. Toledo, B. Codina, y A. Suárez, "Low-Cost Ultrasonic Range Improvements for an Assistive Device", *Sensors*, vol. 21, núm. 12, p. 4250, jun. 2021, doi: 10.3390/s21124250.

[24] D. Sarwinda *et al.*, "Automatic Multi-class Classification of Indonesian Traditional Food using Convolutional Neural Networks", en *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, IEEE, sep. 2020, pp. 43–47. doi: 10.1109/IC2IE50715.2020.9274636.

[25] D. Yi, H. Jin, M. C. Kim, y S. C. Kim, "An Ultrasonic Object Detection Applying the ID Based on Spread Spectrum Technique for a Vehicle", *Sensors*, vol. 20, núm. 2, p. 414, ene. 2020, doi: 10.3390/s20020414.

[26] D. Sarwinda *et al.*, "Automatic Multi-class Classification of Indonesian Traditional Food using Convolutional Neural Networks", en *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, IEEE, sep. 2020, pp. 43–47. doi: 10.1109/IC2IE50715.2020.9274636.

[27] "Author Index", en *2021 4th International Conference on Information and Computer Technologies (ICICT)*, IEEE, mar. 2021, pp. 321–323. doi: 10.1109/ICICT52872.2021.00059.

[28] A. Vaishnav y M. Mandot, "Template Matching for Automatic Number Plate Recognition System with Optical Character Recognition", 2020, pp. 683–694. doi: 10.1007/978-981-13-7166-0_69.

[29] P. C. Madhusudana, X. Yu, N. Birkbeck, Y. Wang, B. Adsumilli, y A. C. Bovik, "Subjective and Objective Quality Assessment of High Frame Rate Videos", *IEEE Access*, vol. 9, pp. 108069–108082, 2021, doi: 10.1109/ACCESS.2021.3100462.

XVIII. Anexos

Anexo 1: Instalación del sistema operativo

Se tiene como objetivo la instalación del software de la Rasberry para que pueda ser utilizada en el desarrollo del prototipo. Para dar inicio se ingresa a la página de Raspberry y se instala el ejecutable como se observa la figura 24.

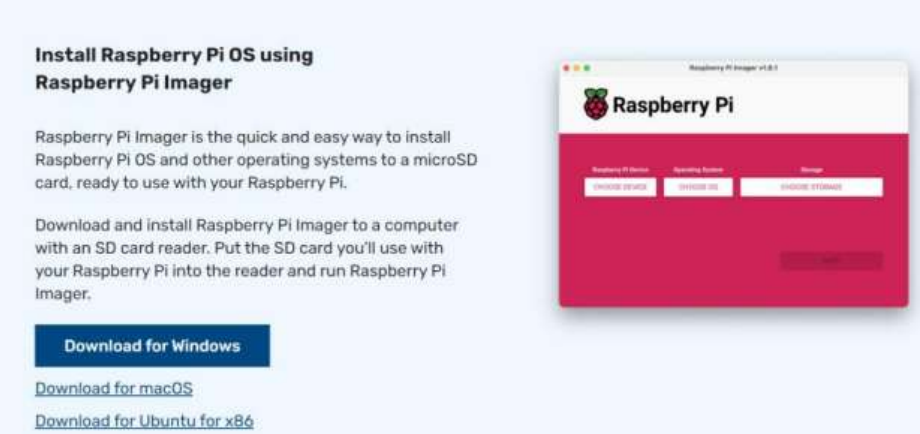

Figura 24 Página principal para instalar ejecutable.

En la figura 25 se muestra la instalación del sistema operativo. Esto facilita el proceso de descargar las versiones correspondientes y se pueda tener una correcta operación en el proceso.

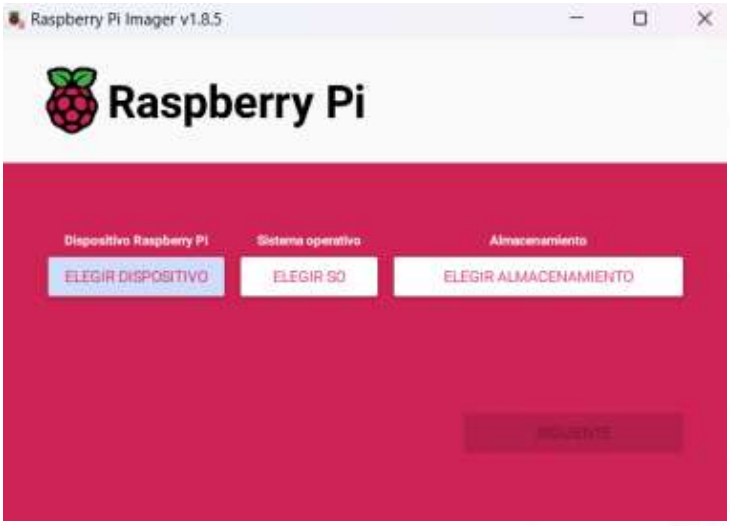

Figura 25 Ventana inicial para escoger el tipo de sistema operativo y su lugar de almacenamiento.

Luego de este proceso se debe seleccionar el tipo de placa que se está manejando para realizar el prototipo. Este paso es importante debido a que se instalaran las librerías indispensables para la correcta operación y ejecución de programas. Esta información se aprecia en la figura 26.

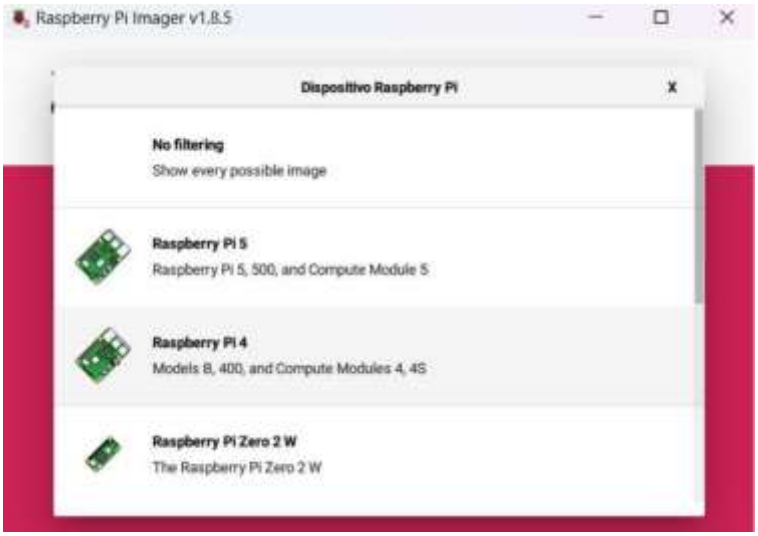

Figura 26 Opciones múltiples de placas disponibles.

En la figura 27 se realiza el proceso de instalación del sistema operativo, esto para que no consuma mucho almacenamiento y se pueda configurar para utilizar las librerías indispensables.

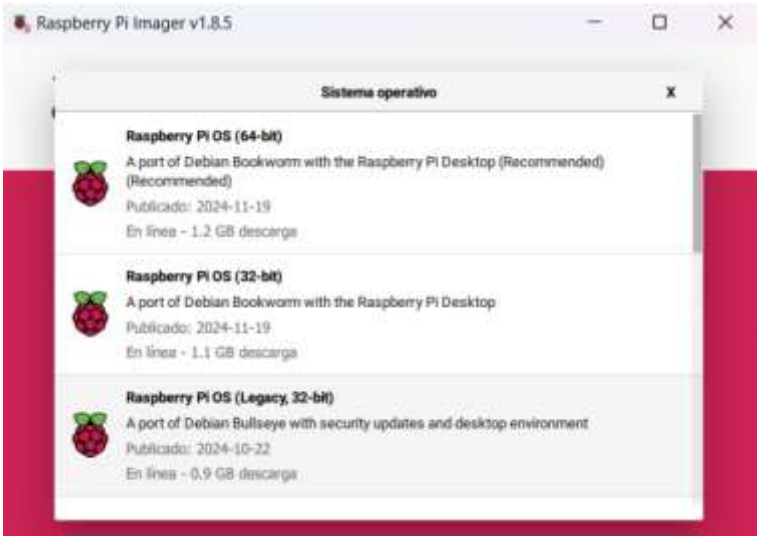

Figura 27 Instalación del sistema operativo de la Raspberry.

Posterior a eso de ingresa la tarjeta Sd para dar iniciación a la placa Raspberry y así poder hacer que se conecte al junto con los periféricos de entrada tales como teclado y ratón, así como se observa en la figura 28. Asimismo, cuando se termina este proceso se comienza con la programación de cada componente.

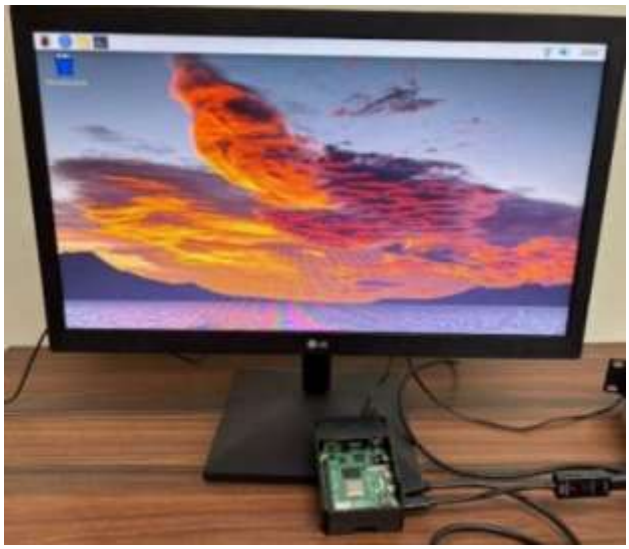

Figura 28 Conexión de la Raspberry al monitor.

De ahí se realizan las conexiones de los componentes a utilizar para iniciar la estructura del prototipo como se encuentra en la figura 29. Asegurarse que el suministro de energía sea óptimo para mantener operando a la Raspberry sin inconvenientes.

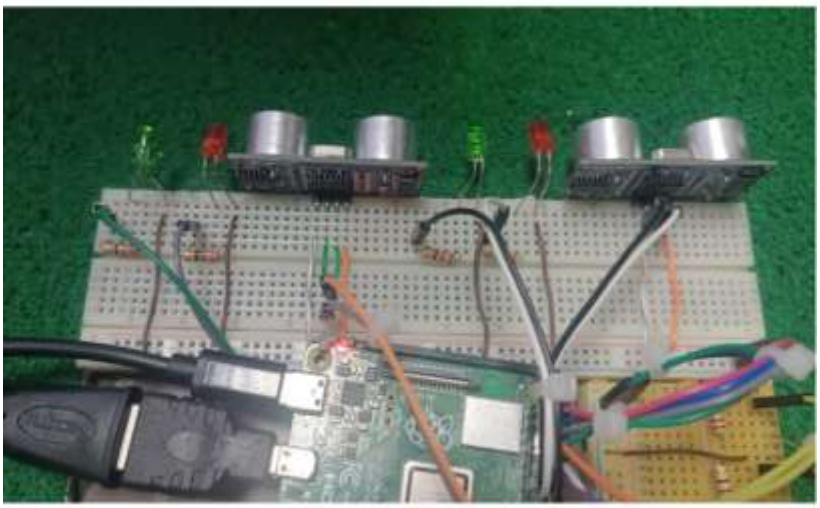

Figura 29 Conexiones iniciales de los componentes en el protoboard.

Anexo 2: Instalación de librerías

La función del anexo 2 es la instalación de las librerías y configuraciones de entrada de la Raspberry. Para la instalación de librerías se lo realiza mediante el terminal usando la codificación "Sudo apt get install (libreria)" como está en la figura 30.

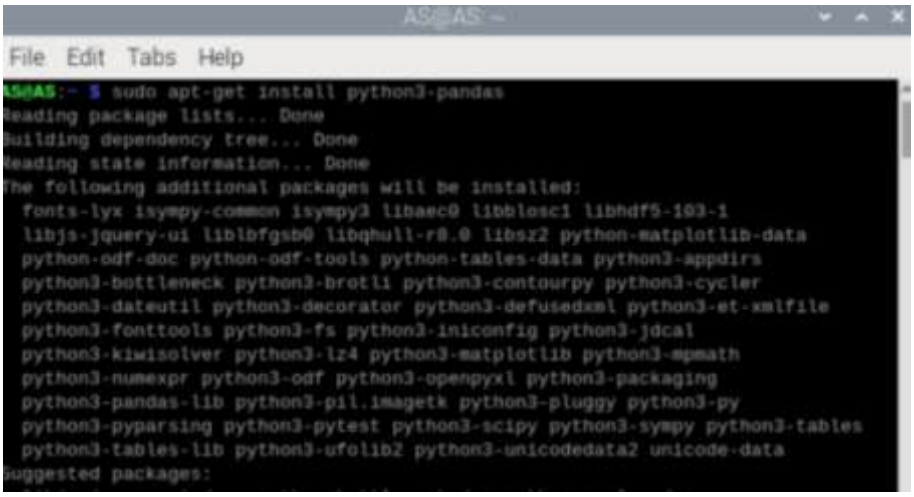

Figura 30 Instalación de las librerías por medio del terminal de la Raspberry.

Existen ciertos momentos no será posible realizar la instalación de la librería a través de un comando directo por lo que se deberá descargar el archivo de la librería desde internet para poder ejecutarlo desde el terminal de la Raspberry. Muchas ocasiones se necesita usar el comando pip pero debido a que se instaló el sistema operativo más básico no contiene tal función. Por tal motivo se descarga los archivos zip de las librerías a usarse directamente de Github o de Pypi, tal como se muestra en la figura 31.

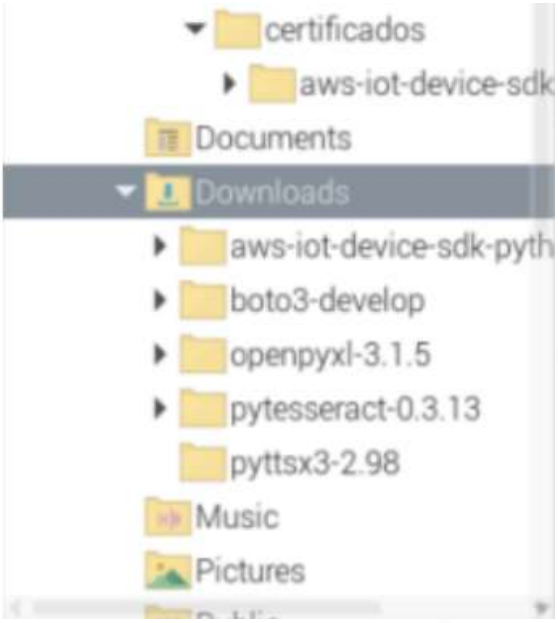

Figura 31 Visualización de archivos zip almacenados en la Raspberry.

En la figura 32 presenta la manera en la que se extrae el archivo zip para que se puedan realizar las instalaciones de las versiones de cada una de las librerías y poder operar con normalidad.

Figura 32 Extracción de información de las librerías en archivos zip.

Un paso extra en estas configuraciones de la Raspberry es habilitar la función VNC tal como se observa en la figura 33, para poder establecer una conexión remota sin requerir de un monitor y periféricos extras. Luego se desactiva la función de Serial Console y se activa el Serial Port para que se pueda usar solo las configuraciones realizadas y permita que los sensores operen con normalidad y las librerías cumplan su función.

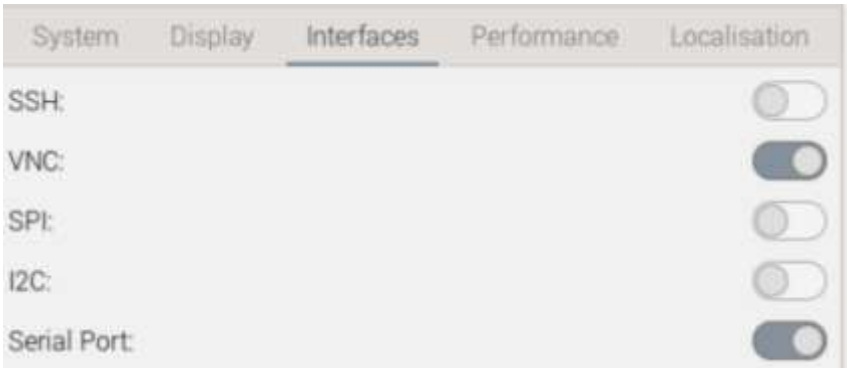

Figura 33 Configuración de interfaz de la Rasberry y acceso a puertos seriales.

Anexo 3: Resultados finales del prototipo

Después de haber realizado las respectivas configuraciones en la Raspberry y haber instalado las librerías necesarias para dar funcionamiento al prototipo se obtuvo como resultado final el armado y distribución de componentes. Esto se muestra en la figura 34.

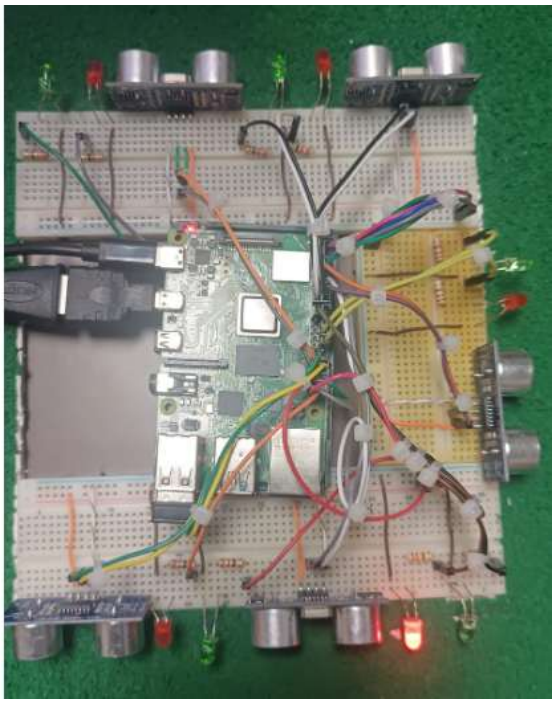

Figura 34 Distribución de los componentes en la estructura del prototipo de asignación de parqueos.

En la figura 35 presenta la estructura de la base que se realizó para dar soporte a la estructura y se pueda realizar las pruebas simuladas en un entorno controlado similar a un estacionamiento.

Figura 35 Diseño de la estructura de la base para dar soporte al prototipo.

Por último, se realiza la cubierta final del prototipo para dar las últimas modificaciones de la distribución de parqueos de la manera que se observa en la figura 36. Se hacen los ajusten para que los sensores y leds queden por fuera, de tal forma esto ayuda a que no se presenten desconexiones en la parte interior por movilización.

Figura 36 Diseño de la estructura superior del prototipo.

Anexo 4: Configuración de almacenamiento de AWS

El anexo 4 sirve para presentar la configuración del almacenamiento en la nube de los datos registrados de la reserva de puestos en los estacionamientos y luego se presenten en una interfaz que se ejecuta por medio de un servidor flask.

Anexo 4.1 Primera parte: Creación de llaves de acceso

Es necesario realizar la configuración de estas llaves utilizando la función IAM de AWS. Se debe crear el grupo de usuarios para después crear un acceso único dentro del grupo la cual cumplirá la función de comunicarse con las demás funciones para almacenar los datos. El resultado de este proceso se muestra en la figura 37.

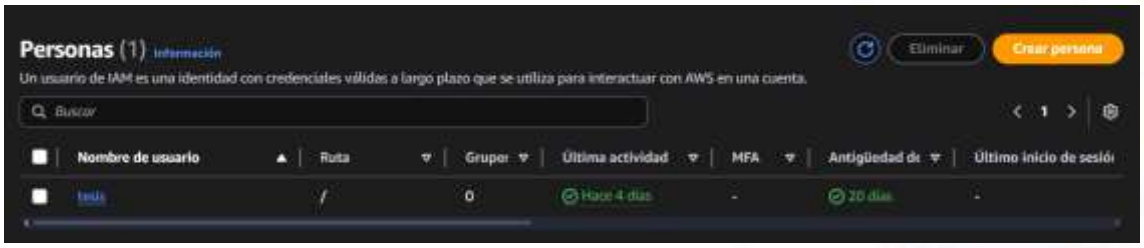

Figura 37 Creación de usuario para acceso a información de AWS.

Cuando ya se tiene creada la persona se procede a generar las claves de acceso, así como se presenta en la figura 38. La llave privada solo será posible observarse una vez al momento de hacer la configuración por lo que se la debe tener anotada para la configuración del código.

Figura 38 Configuración de llaves privadas para acceso general.

En la figura 39 se presenta las políticas de acceso configuradas para que se pueda almacenar de forma correcta los datos en el bucket y sea procesada a la base de datos en Amazon Athena.

Editor de políticas

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "athena:StartQueryExecution",
                "athena:GetQueryExecution",
                "athena:GetQueryResults"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::arqueo-datos",
                "arn:aws:s3:::arqueo-datos/*"
            ]
        }
    ]
}
```

Figura 39 Codificación para permisos de accesos generales entre el Bucket y Amazon Athena.

En la figura 40 se realiza la asignación de datos en los códigos de los sensores asignando la región en la que se tiene configurado AWS. Este punto permite que la información recopilada por el prototipo sea almacenada en la nube de Amazon configurada para poder procesarse a la base de datos y en la interfaz web.

```
}

tiempo_reserva = 30
directorio_fotos = "tesis"
os.makedirs(directorio_fotos, exist_ok=True)

# AWS S3
AWS_ACCESS_KEY = 'AKIAWO6L7I27PJ5YGVIP'
AWS_SECRET_KEY = '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'
AWS_REGION = 'us-east-2'
bucket_name = "arqueo-datos"
```

Figura 40 Asignación de llaves de acceso, región y ubicación del bucket para un correcto enlace de información.

Anexo 4.2 Segunda parte: Almacenamiento de datos en S3

En esta parte se realizan las configuraciones del almacenamiento del Bucket en Amazon S3. El Bucket cumple la función de recopilar y almacenar los datos y archivos que genere el prototipo, En la figura 41 se presenta la imagen del almacenamiento.

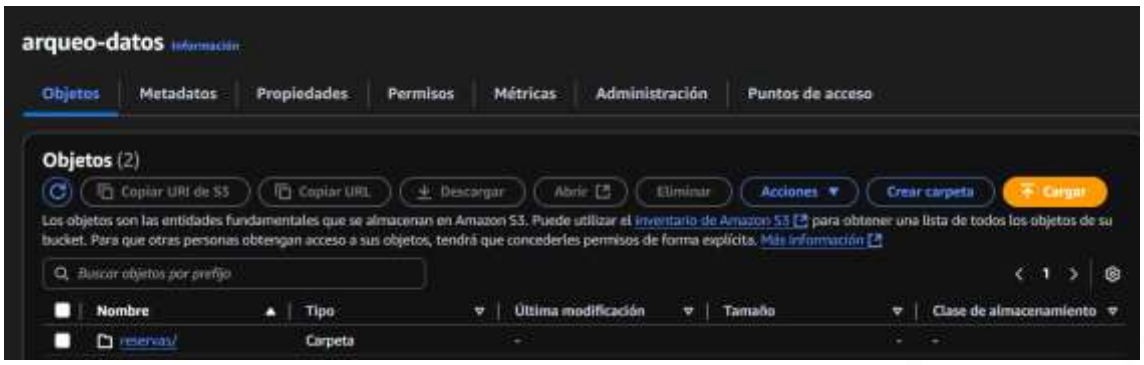

Figura 41 Creación de Lugar en donde se almacenará el archivo procesado por la Raspberry.

En la programación de prototipo se establece la configuración del almacenamiento en el bucket y la conversión del archivo Excel a csv así como se muestra en la figura 42. Esta parte del código es fundamental ya que toma la configuración inicial en donde se declararon las variables del acceso a la nube y así mismo la conversión del archivo Excel para que no se presente errores al generar los datos.

```python
def subir_a_s3(archivo_local, bucket, nombre_en_s3):
    s3 = boto3.client(
        's3',
        aws_access_key_id=AWS_ACCESS_KEY,
        aws_secret_access_key=AWS_SECRET_KEY,
        region_name=AWS_REGION
    )
    try:
        s3.upload_file(archivo_local, bucket, nombre_en_s3)
        print(f"✅ Subido a S3: {archivo_local} -> {bucket}/{nombre_en_s3}")
    except Exception as e:
        print(f"⚠️ Error al subir a S3: {e}")

def registrar_reserva_excel(puesto, codigo, placa_detectada):
    archivo_xlsx = "reservas.xlsx"
    archivo_csv = "reservas.csv"
    ahora = datetime.now()
    fecha = ahora.strftime("%Y-%m-%d")
    hora = ahora.strftime("%H:%M:%S")
```

Figura 42 Codificación para comunicación con el almacenamiento de la nube de Amazon.

En la figura 43 se muestra el almacenamiento del archivo csv que se estará actualizando con cada reserva que se genere al momento de tener operando el prototipo. Esto permitirá tener una información en tiempo real siendo actualizando constantemente.

Figura 43 Archivo almacenado correctamente en el Bucket.

Anexo 4.3 Tercera parte: Creación de base de datos en Amazon Athena

Se realiza la configuración de la base a través de Querys. En primera instancia se usa el método Create para la creación de la base y posterior a eso se realiza la creación de las tablas con cada una de sus estructuras y se define la Localia en donde se debe extraer la información tal como se muestra en la figura 44.

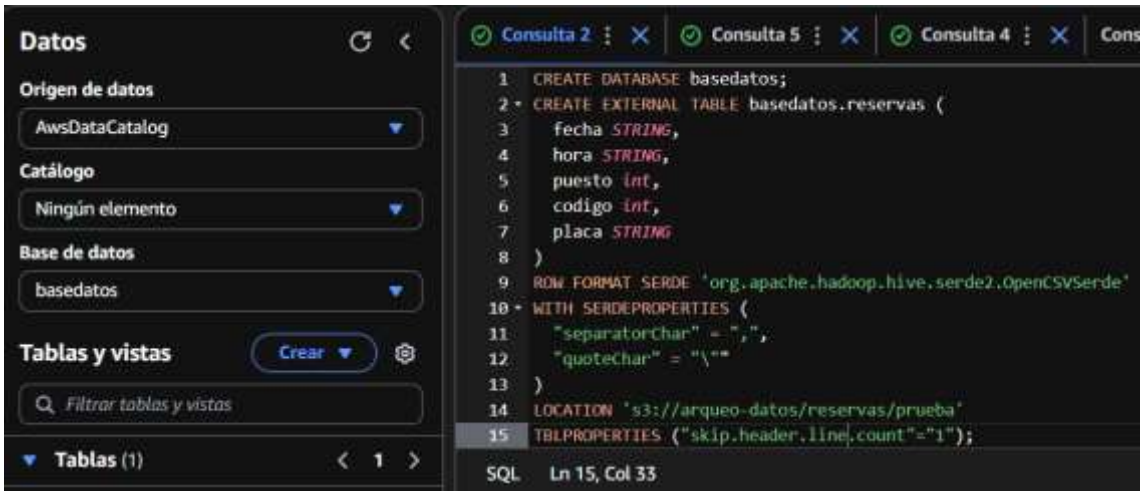

Figura 44 Creación de base de datos y tablas.

Teniendo este procedimiento completado se podrá realizar la revisión general de la información adquirida desde el bucket tal como se presenta en la figura 45. Esta información presentada se replicará en la interfaz web para gestión de monitoreos.

Figura 45 Consulta de las columnas de la tabla previamente creada.

Anexo 5: Creación del servidor flask

En este anexo lo que se busca es establecer la configuración de la interfaz web utilizando una ip local con el puerto 5000. Esta función permite presentar una interfaz web desarrollada por medio de HTML en la cual se muestra la información extraída del prototipo. Este procedimiento se muestra en la figura 46.

```python
from flask import Flask, render_template
from openpyxl import load_workbook

app = Flask(__name__)

@app.route('/')
def index():
    datos = []
    try:
        wb = load_workbook('reservas.xlsx')
        ws = wb.active
        for row in ws.iter_rows(min_row=2, values_only=True):
            datos.append(row)
    except Exception as e:
        print("Error leyendo Excel:", e)
    return render_template("index.html", datos=datos)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

Figura 46 Codificación del servidor flask para la interfaz web.

Anexo 6: Características técnicas de los componentes tecnológicos.

Sensor ultrasónico HC-SR04: Este componente es usado para realizar mediciones de distancia o detección de objetos. El sensor funciona por emisión de ondas ultrasónicas con un alcance de hasta 4 metros.

Tabla 6 Características técnicas del sensor ultrasónico

Características	Valores
Voltaje de operación	5V DC
Corriente de operación	15 mA
Rango de medición	2 cm a 400 cm
Frecuencia ultrasónica	40 kHz

Diodos leds: Estos componentes son de utilidad para dar advertencias en el prototipo diseñado. Son implementados para alertas visuales de disponibilidad de parqueos.

Tabla 7 Características técnicas de los diodos leds.

Característica	Valor
Tipo	LED 5mm estándar
Voltaje directo (Vf)	1.8 – 2.2V (rojo), 2.0 – 3.2V (verde)
Corriente recomendada	10 – 20 mA
Polaridad	Ánodo (+), Cátodo (–)

Raspberry Pi: Es usado como controlador principal para ejecutar los programas y realizar las conexiones de los diversos componentes tecnológicos compatibles. Es de gran aporte por su capacidad de procesamiento y compatibilidad con el lenguaje Python.

Tabla 8 Características técnicas de la Raspberry Pi.

Característica	Valor
Modelo	Raspberry Pi 4B
Procesador	Quad-core Cortex-A72
Memoria RAM	2GB / 4GB / 8GB
Puertos GPIO	40 pines
Puertos USB	2× USB 2.0, 2× USB 3.0
Alimentación	5V, 3A USB-C
Sistema operativo	Raspberry Pi OS