INSTITUTO TECNOLÓGICO SUPERIOR DE XALAPA

TÍTULO DEL PROYECTO

DISEÑO DE LA ARQUITECTURA DE HARDWARE Y CONECTIVIDAD PARA UN SISTEMA DE ESTACIONAMIENTO INTELIGENTE.

Opción de Titulación:

I Tesis profesional

Que como requisito parcial para la obtención de grado de

Maestría en Sistemas Computacionales

Presenta:

Posadas López Jesús

No. de Control:

207O00013

Director

Mtro. Manuel Prisciliano Ralero de la Mora

Xalapa-Enríquez Veracruz febrero de 20223

CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

DISEÑO DE LA ARQUITECTURA DE HARDWARE Y CONECTIVIDAD PARA UN SISTEMA DE ESTACIONAMIENTO INTELIGENTE.

INTRODUCCIÓN

En las grandes ciudades en crecimiento tecnológico y social, la congestión vehicular se ha convertido en una grave dificultad. Los problemas de congestión vehicular han aumentado exponencialmente con esto nos referimos a el problema de buscar un lugar de estacionamiento y todo lo que conlleva. Por ejemplo, hoy en día en algunas plazas comerciales es difícil encontrar un lugar de estacionamiento adecuado. Una de las dificultades es que muchas veces un usuario busca un espacio vacante cerca de la tienda a la que va a comprar para no caminar tanto. También por lo regular las personas buscan un lugar donde el vehículo tenga sombra, que tenga buena iluminación y que sea de fácil acceso. Dado lo anterior sabemos que uno de los problemas es el tiempo que gasta un usuario en encontrar un espacio vacío, la pérdida de tiempo en la búsqueda de un lugar libre conlleva a una mayor contaminación ambiental. En otras ocasiones el no encontrar una vacante donde estacionarse puede aumentar los niveles de estrés de las personas, esto genera en algunas ocasiones peleas por un lugar disponible.

Uno de los primeros trabajos tecnológicos enfocados en la problemática del congestionamiento vehicular es la de (jatuporn.chinrungrueng, 2007). Donde los autores propusieron una aplicación para ayudar a los conductores a encontrar un espacio de estacionamiento libre. La propuesta fue ayudar a sus usuarios para dirigirlos a un lugar disponible cerca de donde van a comprar, y así disminuir el tiempo de búsqueda de las personas y también proveer información en tiempo real de los lugares de estacionamiento vacantes. Para ello ocupaban sensores magnéticos o sensores ópticos conectados con un cable de fibra óptica que les

permitía enviar la información a una consola (Laptop o PC).

Hoy en día la tecnología ya es parte de la vida diaria en varias ciudades, por ejemplo, en el libro Smart Cities: Un primer paso hacia internet de las cosas (Ariel, 2011) nos dice que; en un entorno urbano con una demanda creciente de eficiencia, desarrollo sostenible, calidad de vida y sabia gestión de los recursos, las administraciones públicas han de plantearse una evolución en los modelos de gestión de las ciudades. Para ello la aplicación de las tecnologías de la información y las comunicaciones (TIC) se hace imprescindible y se traduce en el concepto de Smart City que adelanta con sus servicios lo que se ha denominado en internet de las cosas.

Con la ayuda de internet de las cosas, (loT, por sus siglas en ingles internet of things), se ofrecen más y mejores servicios a los usuarios. Estos servicios nos ayudan en muchas tareas cotidianas de la vida diaria y nos facilitan el día a día. Algunos investigadores definen a IoT como un modelo que abarca a las tecnologías de comunicación inalámbrica como lo son las redes de sensores inalámbricos, redes móviles y actuadores. Estos son denominados “objeto o cosa” y contienen una dirección única, y ayuda a formar una red de sensores inalámbricos (WSN). Estas han recibido una gran atención en los últimos años desde los puntos de vista académico e industrial gracias a los avances de la tecnología en estos ámbitos como lo es con los micro-sensores, redes inalámbricas y el procesamiento de dispositivos embebidos.

Uno de los ámbitos en los que se puede usar la definición de IOT es en los llamados estacionamientos inteligentes. Este tipo de estacionamientos se caracterizan en que permiten al usuario obtener información anticipada y soporte en la toma de decisiones oportunas al momento de buscar un lugar de estacionamiento. Acorde con (Giffinger, 2007) se presentan 4 características en este tipo de estacionamientos, las cuales son:

- Administración inteligente de vehículos: Permitir al usuario una visualización de los espacios disponibles.

- Movilidad inteligente: Una vez el usuario identifique el lugar que quiere ya podrá saber dónde se encuentra su espacio y se podrá dirigir a él directamente.

- Economía inteligente: El hecho de que el usuario pueda identificar y dirigirse al lugar de estacionamiento vacío significa un ahorro en combustible lo que es un beneficio económico para el usuario.

- Entorno inteligente: Significa que el estacionamiento debe estar adaptado con las tecnologías necesarias para el entorno que quiere decir, que tengamos una detección óptima y no impida en ningún momento que algún usuario no pueda llegar a dicho espacio.

El presente proyecto nace con la finalidad de brindar una solución a la búsqueda de un lugar de estacionamiento. Este pretende ofrecer un sistema basado en una red de sensores con la ayuda de un sistema basado en IoT. Esto ayudará a detectar la presencia de algún automóvil y utilizando un programa basado en IoT podremos ofrecer la opción de visualizar en una interfaz de computadora los espacios disponibles en los estacionamientos. Esto nos permitirá ofrecer al usuario una manera más rápida de encontrar un espacio libre en el cual estacionarse.

Con esto logramos un avance a la solución del molesto problema de buscar un lugar de estacionamiento, ahorrándonos tiempo a la hora de buscar un lugar vacío y resolviendo el problema de no encontrar un lugar de estacionamiento disponible.

CAPÍTULO I. DESARROLLO METODOLÓGICO

1.1 Planteamiento del problema

Los estacionamientos públicos o privados en muchas ciudades, son un servicio que tiene una gran demanda. Es común que los conductores busquen o tengan la necesidad de un espacio en el cual dejar su vehículo durante un periodo de tiempo ya sea al acudir a su jornada laboral o al realizar otro tipo de actividades. El desconocimiento de si hay o no lugares disponibles y la dificultad de encontrar un lugar vacío, es quizá la problemática más común a la cual se enfrenta un conductor. Esta situación ha ocasionado que se incremente la apertura de estacionamientos y uso de espacios libres con fines de convertirlos en estacionamientos públicos.

La búsqueda de un lugar de estacionamiento en el centro urbano de cualquier ciudad conlleva efectos que producen una pérdida o disminución de la calidad de vida de las personas. Los efectos más habituales asociados a la búsqueda de estacionamiento son: la contaminación acústica y atmosférica, la congestión vehicular, la inseguridad e indisciplina vial y la ocupación del espacio público por los vehículos (Recarey, 2014). La problemática que se ha detectado y que ha dado inicio al desarrollo de este proyecto es en parte la complejidad que enfrentan los conductores en la búsqueda de un lugar de estacionamiento. También por otro lado, se tiene la dificultad que tienen los dueños de un estacionamiento para la administración y control de espacios libres.

Por ello se propone diseñar un sistema de estacionamiento automatizado que apoyaría tanto a los dueños como a los usuarios del estacionamiento. A los dueños les puede proporcionar nuevas fuentes de ingresos ya que, gracias a la implementación de un estacionamiento inteligente, se pueden habilitar opciones de

pago por niveles que pueden depender del lugar donde sus clientes quieran aparcar. También el dueño podrá obtener los datos de los estacionamientos y esto ayuda para poder verificar violaciones a el estacionamiento u actividades sospechas. A los usuarios les facilitaría la búsqueda de un espacio de estacionamiento libre. Esto a su vez, conlleva varios beneficios, pues en muchas ocasiones el no encontrar un lugar disponible puede generar estrés, mal humor y algunas veces pudiera generar incluso un accidente al estar buscando un espacio vacío. Dicho lo anterior se tiene que los estacionamientos existentes no cuentan con un sistema capaz de guiar o mostrar al usuario donde se encuentra un espacio disponible.

1.1.2 Problema enunciado

Uno de los grandes problemas que enfrentan los encargados de los estacionamientos es el monitoreo eficiente y fiable en tiempo real del número de espacios disponibles que se pueden ofertar a los conductores. En este trabajo se propone un diseño de Hardware y Software que busca remediar dicha problemática.

1.2 Objetivo General

Diseñar diferentes módulos arquitectónicos basados en redes de sensores ultrasónicos que combinados integren un sistema modular que muestre la arquitectura adecuada de hardware y conectividad para un sistema de gestión inteligente de estacionamiento, basado en IoT, con la finalidad de administrar de forma eficiente el uso de los espacios de un estacionamiento.

1.2.1 Objetivos específicos

Los objetivos específicos son los siguientes:

- Determinar un modelado de hardware y conectividad para el diseño de la arquitectura.

- Realizar diferentes pruebas para evaluar las diferentes condiciones en las que pueden interactuar los sensores con el diseño.

- Crear la conectividad para la información a la nube con la ayuda de algún software que trabaje con IoT

- Determinar la mejor combinación de módulos para los diferentes modelos de estacionamiento y la implementación de una arquitectura.

- Desarrollar el programa que implemente una interfaz para detectar si un lugar está ocupado o vacío.

1.3 Justificación

Dado que el estacionamiento vehicular es un problema en algunos lugares ya sea por las condiciones de infraestructura, el tráfico, la congestión en horas pico y la poca disponibilidad en ciertas áreas metropolitanas para encontrar un espacio adecuado para estacionar el automóvil. Se propone la implementación de un estacionamiento inteligente y se espera que esto ayude en los siguientes ámbitos:

Ambiental: En este aspecto al proporcionar un espacio vacío y que se encuentre disponible se reduce el consumo de combustible, pues el conductor no tiene que andar dando vueltas o conducir sin una dirección fija. Esto nos ayuda a reducir los contaminantes que genera un automóvil.

Salud: Muchas veces los conductores se pueden estresar al buscar un lugar de estacionamiento ya que los tiempos de búsqueda pueden ser en algunas ocasiones muy largos. Esto genera estrés y frustración y algunas veces se pueden generar accidentes por la desesperación de querer encontrar un lugar vacío.

Seguridad: El sistema proporcionara al conductor o encargado del estacionamiento la ubicación de los espacios libres. Esto ayuda a evitar conflictos donde las personas se han llegado a agredir físicamente por un lugar de aparcamiento desocupado.

Social: Este sistema ayudará a contribuir a un mejor manejo de los lugares de estacionamiento y a una mayor fluidez vehicular dentro del mismo.

1.4 Alcances y limitaciones

A continuación, se describen aquellos aspectos que se lograrán alcanzar y algunos de los problemas que se pueden presentar y que limitarán nuestro trabajo.

Alcances

- El diseño de la plataforma incorporará una aplicación de soporte para el administrador.
- El sistema contara con conectividad a una plataforma en la nube, utilizando Azure IoT y bajo el protocolo HTTPS.
- El sistema estará basado en IoT con Arduino
- Los clientes del estacionamiento podrán conocer a través de un dispositivo que pueda conectarse a la red, si un lugar de estacionamiento está ocupado o desocupado.
- El diseño incorporará sensores ultrasónicos.
- Se realizarán pruebas de campo con el diseño para checar su funcionalidad con la ayuda de un software.
- La interfaz que utilizaran los clientes del estacionamiento será compatible con Mac, Windows y Linux a través de un software comercial.

Limitaciones

- El diseño propuesto no será capaz de distinguir vehículos de dos ruedas, si una moto ocupa el cajón de estacionamiento, no será detectada.
- El diseño solo detecta cuando el vehículo esta correctamente estacionado.
- El diseño propuesto considera que el Hardware siempre estará conectado a una fuente de energía.

- La tarjeta Arduino deberá estar protegida de las inclemencias del medio ambiente.
- No se tendrá apartado de lugares
- Todos los sensores ultrasónicos deberán estar protegidos de la lluvia.
- El diseño solo se pondrá a prueba en su primera etapa en un estacionamiento privado con una capacidad máxima de 10 vehículos

1.5 Pregunta de investigación

¿Cuál será el diseño de una arquitectura de Hardware que se adapte a la mayoría de los servicios de estacionamiento y que pueda ser administrada mediante un gestor a través de IoT?

1.5.1 Preguntas especificas

Como sub preguntas se consideran las siguientes:

- ¿Cómo se va a almacenar la información en la nube?
- ¿En qué software se trabajará para unir las tecnologías?
- ¿Los sensores podrán soportar los climas adversos o condiciones climáticas extremas?
- ¿Cómo se va a identificar en la interfaz si un lugar está vacío?

1.6 Hipótesis

La utilización de los sensores ultrasónicos, agregados en una red cableada o inalámbrica permite diseñar diferentes arquitecturas que se adapten en una misma plataforma de hardware para diferentes servicios de estacionamientos con distintos requerimientos, esto ayuda a la detección con mayor facilidad y eficiencia en tiempo real un espacio de estacionamiento vacío.

Las variables dependientes serán las siguientes:

- La detección o la ausencia de un vehículo: podrá ser medida o detectada bajo un sistema de sensores.
- El tiempo en el que tarda en detectar un vehículo y le avisa al encargado del estacionamiento

Las variables independientes son las siguientes:

- **La posición del vehículo:** podrá ser detectada gracias a la ayuda de los sensores ultrasónicos que marcaran si está ubicado correctamente en el espacio de estacionamiento.
- **La distancia que hay del sensor al vehículo:** Los sensores estarán ajustados de dos maneras, tanto puede ser arriba como una lámpara o enfrente del vehículo esto nos ayudará a medir la distancia que tiene el vehículo al sensor.

CAPÍTULO II. MARCO TEÓRICO

En este capítulo se hablará sobre el estado del arte de internet de las cosas y los sistemas de estacionamientos inteligentes junto con algunos tipos de sensores junto con los protocolos de comunicación que deben llevar, con ello también se abordara acerca de los softwares que se van a emplear. Después se abordarán algunos de los estacionamientos inteligentes basados en IoT y se mencionaran las ventajas que estos poseen.

Se iniciará por definir cuáles son las características para que un estacionamiento sea considerado inteligente, esto se puede dividir en dos categorías de estacionamientos inteligentes. La primera categoría estima el número de espacios ocupados e indica cuántos espacios hay disponibles. La segunda le dice a un conductor donde está el espacio disponible y lo guía a él. Las dos categorías son muy importantes para satisfacer el nombre de estacionamiento inteligente, esto quiere decir que para que un estacionamiento sea considerado inteligente debe cumplir con alguna de las dos características (Amin Kianpisheh, 2012).

2.1 Definición de internet de las cosas (IoT)

Internet de las cosas (IoT) se denomina como una red de objetos físicos que une vehículos máquinas y algunos electrodomésticos. Estos utilizan sensores para conectarse vía internet y con ello intercambiar o conectar información (Evans, 2011). IoT está diseñada para una serie integral de tecnologías como lo son las interfaces de programación (API) que conectan los dispositivos a internet. Una de las tecnologías clave que maneja IoT son las herramientas de gestión Big Data, así como también la recopilación de información para ser enviada a la nube bajo diferentes protocolos de comunicación.

Actualmente las personas requieren conectarse a internet por diferentes necesidades ya sean informativas, sociales, de entretenimiento, laborales e incluso hasta económicas. Esto conlleva a la evolución del internet y de las tecnologías, con

ello cada vez es más grande el crecimiento de IoT.

Existen diferentes áreas de aplicación donde puede utilizarse IoT, por ejemplo, se tienen tres aplicaciones de los objetos según (Alcaraz, 2021). Número uno, Smart appliances; son versiones de algunos dispositivos muy usados como lo son refrigeradores, microondas, estufas etc. Estos aprovechan los sensores que se les van integrando para un mejor funcionamiento o manejo.

Control y automatización: Eso se refiere al control inalámbrico de algunas aplicaciones y también sobre el control remoto de algunos objetos.

Monitorización del estado de la casa: Este tipo se refiere a las cosas que pueden ser automatizadas en el hogar, como lo son la iluminación, la temperatura, gas, etc.

2.1.1 Ventajas de internet de las cosas (IoT)

Una vez que se ha definido que es internet de las cosas, se puede decir que el objetivo principal de las empresas que usan IoT es hacer la vida más fácil y menos compleja. Dentro de los mayores beneficios que aportan el usar este tipo de tecnologías es la eficiencia tanto como en productividad como en innovación. Por ejemplo, en el sector industrial se requiere de innovación constante para poder tener una mejora en el negocio. Otra de las ventajas que nos brinda IoT es la comunicación que nos ofrece con el entorno de trabajo, esto quiere decir que es más fácil controlar el manejo y rendimiento de las maquinas.

IoT también ayuda también a que la información esté al alcance de cualquier persona en cualquier parte del mundo ya que gracias a que la información es llevada y almacenada a la nube les permite a ciertos usuarios monitorear sus trabajos a distancia. Otro de los beneficios que nos proporciona es la recolección de datos y el control de dispositivos. Esto nos ayuda a que las interfaces creadas a través de IoT sean más sencillas de manejar y entender.

Debemos comprender que la mayoría de las empresas tecnológicas que manejan IoT es esencialmente por el ahorro de tiempo. Las actividades al estar mejor automatizadas permiten un beneficio al momento de buscar un lugar vacante lo cual es muy importante y así se reducen los costos de personal. También el manejar tecnologías IoT se permite una mejor toma de decisiones, esto se debe a la gran cantidad de datos que podemos poseer, por lo que las decisiones serán más idóneas.

También como se menciona en el trabajo de (Seguí, 2014) las ventajas más notorias de IoT son:

- Incremento de un nuevo modelo y oportunidad.
- Ahorro de tiempo de cara al usuario.
- Gestión automática y eficiente de cara al usuario.
- Mejora del urbanismo y entorno.

2.2 Redes de sensores

Una red WSN (Wireless Sensor Network) de sensores inalámbricos es un conjunto de sensores que están interconectados. Estos pueden ser utilizados para diferentes mediciones, por ejemplo, están los de temperatura, sonido, vibraciones, presión y movimiento incluso algunos de ellos son capaces de medir algunos contaminantes. Los dispositivos son unidades autónomas que constan de un microcontrolador, una fuente de energía (casi siempre una batería), un radio transceptor y un elemento sensor (Aakvaag, 2006).

Las WSN se constituyen en base a diferentes componentes que incluyen:

Sensores: estos pueden ser de diferentes tipos y toman la información y la convierten en señales eléctricas.

Nodos de sensor: Toman los datos del sensor a través de puertas de datos y envían la información a un nodo base

Gateway: En su forma más simple, es un concentrador central inteligente para dispositivos IoT. Los gateway o puertas de enlace conectan dispositivos dentro de IoT entre sí y con la nube, traduciendo la comunicación entre los dispositivos y la nube utilizando el protocolo de comunicación TCP/IP.

Estación base y red inalámbrica: La estación sirve para recolectar los datos mientras que la red inalámbrica sirve para enviar la información por internet.

Otra de las características de las redes de sensores, es que suelen ser desatendidas y sin infraestructura preestablecida, muchas veces suelen operarse en entornos dinámicos y poco favorables. Cada sensor se distribuye de manera determinada donde las personas no puedan estar tan fácilmente, Una red de sensores se puede ver muy comúnmente como en la figura 1.

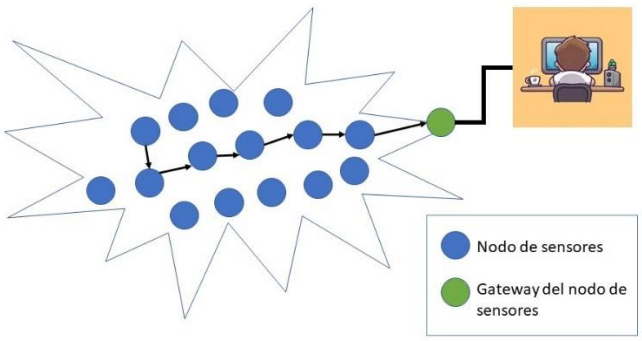

Figura 1: Red de sensores inalámbricos WSN
Fuente: Creación propia

Otra de las partes de los sensores son las WSAN (Wireless sensor and actuador networks) estos son sensores autónomos que al igual que las WSN se utilizan para monitorizar condiciones físicas. Las redes más modernas son bidireccionales y gracias a ello permiten el control de la actividad del sensor. Esto también permite

optimizar el rendimiento de la conexión de los sensores y el procesamiento de la información y gracias a esto tenemos un mayor control y análisis en tiempo real de la información que va a ser procesada. Las características principales de este tipo de redes según (Pérez, 2014) son las siguientes:

Topología dinámica: Los sensores deben adaptarse a los cambios que sufra la topología para poder transmitir los nuevos datos adquiridos.

Variabilidad de envió de información: Los canales de radio son muy variables, ya que existen una serie de fenómenos que los afectan, tales como: la atenuación, interferencias y desconexiones que alteran o interrumpen el envío de información y causando errores en los datos recabados y transmitidos por los sensores.

Infraestructura de la red: No se requiere de alguna infraestructura, ya que los nodos pueden actuar como emisores, receptores o enrutadores de la información. Esta información es recabada por el nodo principal y transmitida generalmente a un computador, el cual tiene la posibilidad de transmitir los datos ya sea de forma inalámbrica o por cable.

Tolerancia a errores: Una red de sensores es capaz de seguir funcionando a pesar de que algún sensor presente una falla.

Consumo energético: Un nodo debe tener un consumo energético bajo, el cual debe conjuntar autonomía y capacidad de procesamiento. Debe contar con un procesador de consumo ultra bajo, así como un transceptor de radio con la misma característica.

Limitaciones de hardware: Para poder tener un sistema autónomo y confiable es necesario que el hardware sea lo más simple posible, así como también el transceptor de radio, lo que deja una capacidad de proceso limitada. Por lo tanto, se requiere una buena compatibilidad entre la red de sensores y el microcontrolador.

Entonces una red de sensores es un conjunto de dispositivos autónomos (Nodos) que tienen una tarea en común trabajando de forma colaborativa que recolectan información de su entorno. Hoy en día las redes de sensores más utilizadas son las inalámbricas (figura 2) ya que en estas los nodos de los sensores son móviles, también se caracterizan por su facilidad al momento de configurarlos o instalarlos (Martinez, 2009). Otra de sus características por la cual los hace eficientes es por la gestión de energía que manejan la mayoría de estos sensores que les permite tener una tasa alta de autonomía y esto las hace altamente operativos.

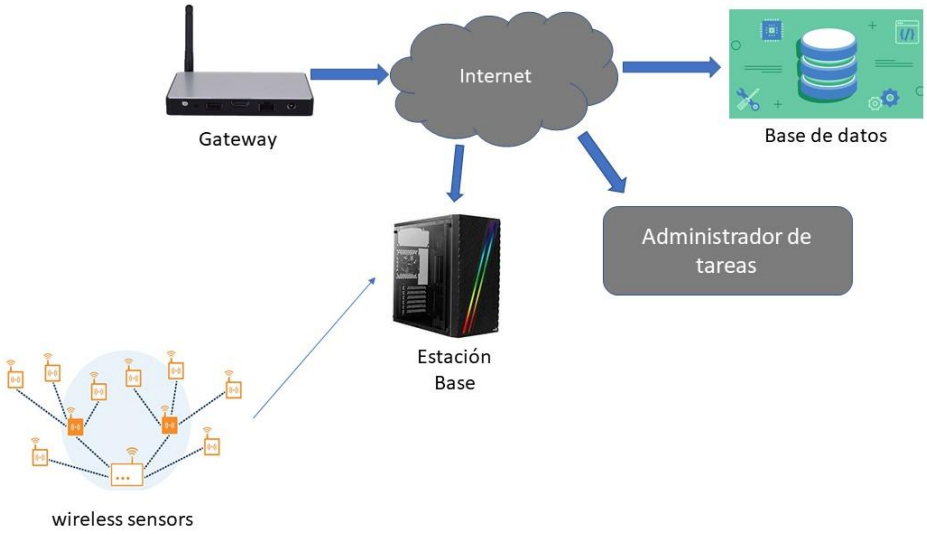

Figura 2: Demostración de cómo funcionan las redes inalámbricas
Fuente: Creación propia

El proceso a seguir para la realización de las aplicaciones es el siguiente: Se realiza una serie de mediciones o sensados, después dicha información en formato digital se transmite hasta el nodo principal (Martinez, 2009). Este nodo transmite la información fuera de la red de sensores hacia un Gateway, donde la información puede ser procesada. En la figura 3 se puede ver cuál es el flujo de información a través de los componentes de la *WSN*.

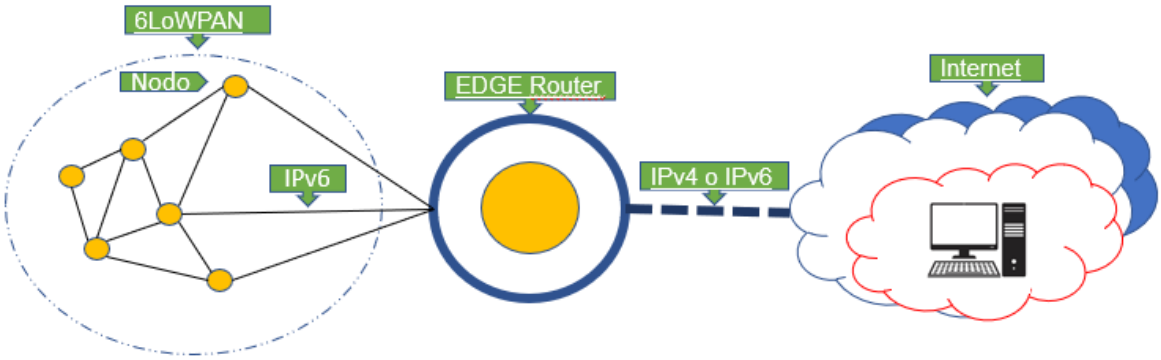

Figura 3: Envió de la información de los sensores a través de los nodos
Fuente: Creación propia

Por último, los componentes que debe tener una arquitectura WSN son los siguientes: Alimentación, comunicación, procesador, sensores y memoria como se ve en la figura 4.

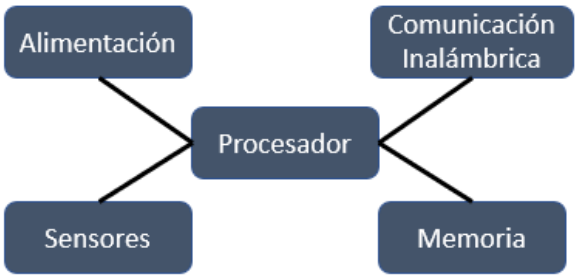

Figura 4: Arquitectura de un nodo inalámbrico
Fuente: Creación propia

2.3 Sensores digitales

Los sensores digitales son aquellos que emiten valores discretos, muchas veces están relacionados con la posición lineal o angular. Estos sensores se utilizan principalmente para detectar la presencia de un objeto o cuando algo está cerca de uno. Los sensores digitales más conocidos son los de proximidad, estos se encargan de detectar un objeto cercano sin hacer contacto con él, después se emite

una señal que puede ser de pulso o de voltaje.

Una de las ventajas que ofrecen los sensores digitales es que pueden aprovechar mejor las rutinas del software usado, por ejemplo, en técnicas como el sobre muestreó y el filtrado digital se mejora considerablemente la resolución del sensor en comparación con un sensor analógico. Los sistemas complejos de hoy en día impulsan más la demanda de que los sensores sean "*inteligentes*" (Córdoba, 2013). Entonces los sensores digitales envían datos a los microcontroladores a través de interfaces, lo cual ayuda a tomar decisiones basadas en la información recopilada.

Los sensores digitales utilizan un método muy conocido llamado modulación de duración de pulsos (PWM), ver figura 5, este permite que en la entrada se tenga una señal electrónica la cual es necesaria para los módulos de control (Fernández Martínez, et al., 2009)

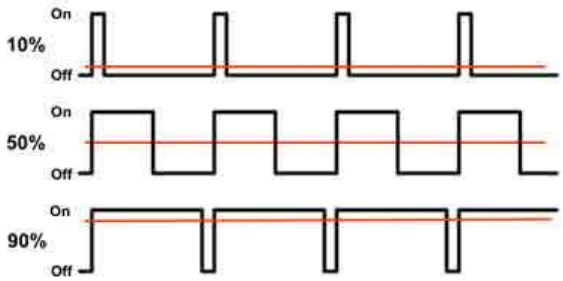

Figura 5: Pulsos PWM
Fuente: (Solectro, 2020)

La mayoría de los sensores digitales tienen 3 terminales las cuales están identificados por números o por letras. Por ejemplo, en el caso de un sensor ultrasónico las terminales vienen nombradas como *gnd*, *trigg* y *echo*. Normalmente *trigg* suele estar conectado a uno de los pines de voltaje que proporciona la energía al sensor. Echo viene conectado a las entradas digitales del microcontrolador y es el que recibe la señal del sensor. Por último, *gnd* está conectado a tierra.

Así, estos se conectan de acuerdo al proceso mostrado en la figura 6, tenemos la variable a medir que en el caso del sensor ultrasónico va a ser la distancia, nuestro elemento sensor va a ser nuestro ultrasónico que emitirá una señal analógica y la enviará a través de un circuito de acondicionamiento para así poder convertir la señal a la información solicitada por el administrador.

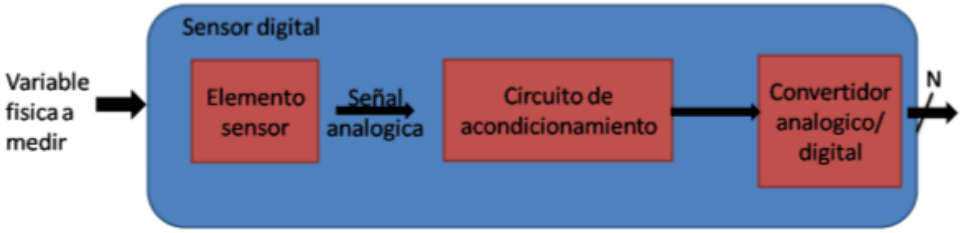

Figura 6: Bloques típicos de un sensor digital
Fuente: Creación propia

2.3.1 Evaluación de sensores

Para este proyecto se evaluaron diferentes tecnologías como son sensores de detección infrarroja, sensores magnéticos y sensores ultrasónicos. La siguiente tabla presenta un resumen de las principales características y muestra las ventajas y desventajas que nos proporciona cada sensor.

Tabla 1: Evaluación de diferentes tipos de sensores.
Fuente: Creación propia

Elemento	Ultrasónico	Infrarrojo	Magnético
Precio	$60	$30	$10,000
Distancia de detección máxima	6M	24M	60M
Precisión	A mayor distancia, menor precisión	A mayor distancia, menor precisión	Alta
Velocidad de respuesta	Media	Baja	Alta
Afectación contra el polvo y el agua	Inmune	El agua afecta su medición	Inmune

Rango de medición	Grande	Pequeño	Pequeño

2.3.2 Sensor SRF04

Se denomina sensor a un objeto capaz de variar diferentes magnitudes como lo son físicas o químicas, también conocidas como variables de instrumentación y se transforman con un actuador en variables eléctricas. Es decir, un sensor es un dispositivo que convierte una forma de energía en otra. Esto significa que es un dispositivo el cual a partir de la energía del medio donde se mide, da una señal de salida que es función de la variable medida. Sensor y transductor se emplean algunas veces como sinónimos, pero el sensor propone un significado más extenso.

La ampliación de los sentidos para adquirir un conocimiento de cantidades físicas que, por su naturaleza o tamaño, no pueden ser percibidas directamente por los sentidos. En cambio, el transductor, propone que la señal de entrada y de salida no deben ser homogéneas (Dhulipala, 2019).

Se denomina transductor a un dispositivo capaz de transformar las magnitudes físicas a otro tipo de magnitudes que podremos medir, registrar y analizar esto suele ser bajo pulsos eléctricos como, por ejemplo, temperatura, presión, humedad del aire, luminosidad, distancia, etc., (Leyva, 2012). Los transductores tienen parámetros que contribuyen al funcionamiento del mismo. Estos parámetros son los siguientes:

Exactitud: se refiere al valor verdadero de la variable que se va a detectar.

Precisión: Es la existencia de una pequeña variación en la medición de una variable y al igual que la exactitud debe ser lo más alta posible.

Rango de funcionamiento: Este solo debe responder a los cambios de los parámetros de la exactitud, la precisión y la amplitud

Velocidad de respuesta: Es la capacidad de identificar a los cambios en las

variables que deben ser detectadas en corto tiempo.

Para el desarrollo de este proyecto se utilizará el sensor ultrasónico **SRF04**, este es un sensor de distancias por ultrasonido capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 3 a 300 cm con un rango de apertura de 90º como en el que se muestra en la figura 7. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. Es de tamaño pequeño, y destaca por su bajo consumo de energía y gran precisión (Robotica, 2020). También este posee una tensión de alimentación de 5 Vcd con un consumo de 4mA y con un tamaño de 24 x 20 x 17 lo que lo hace muy práctico en zonas estrechas, otras de las características se pueden ver en la tabla 2.

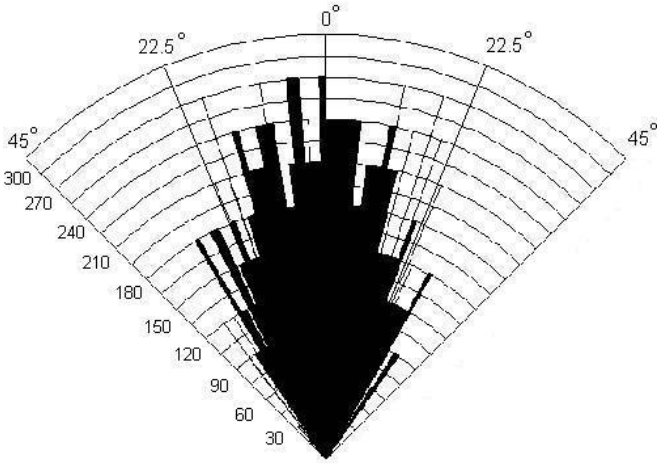

Figura 7: Rango del sensor ultrasónico

Tabla 2: Características del sensor SRF04
Fuente: (Julio, 2022)

Característica	Valor
Tensión	5V
Consumo	30 mA Tip. 50mA Max.

Frecuencia:	40 Khz.
Distancia Mínima:	3 cm.
Distancia Máxima:	300 cm.
Sensibilidad:	Detecta un palo de escoba a 3 m.
Pulso de Disparo	10 uS min. TTL
Pulso de Eco:	100 uS - 18 mS
Retardo entre pulsos:	10 mS Mínimo
Pulso de Eco:	100 uS - 18 mS
Tamaño:	43 x 20 x 17 mm
Peso:	10 gr.

2.4 Protocolos de comunicación

Los protocolos de comunicación son las reglas o normativas necesarias para la transmisión de información entre dos dispositivos. Un protocolo de comunicación de red de datos es un conjunto de reglas que norman el intercambio ordenado de información dentro de la red.

2.4.1 Transporte de la información

Para la comunicación y transportación de la información las tecnologías ocupan diferentes protocolos basados en mediciones ya establecidas. En el caso de la conexión entre un sensor y un microcontrolador se necesita un protocolo de comunicación que permita se lleve a cabo el proceso de transferencia de datos entre ambos dispositivos. La información en el caso de los sensores es digital y es transmitida como un flujo de bits. Donde cada bit puede tener un valor de 1 o 0 y la información que recibe el microcontrolador es de ocho bits.

2.4.2 Comunicación paralela

La comunicación paralela es un método para transmitir varios dígitos binarios (bits) formando bloques de una cantidad especifica de bits que se transmiten de manera simultánea. Esto quiere decir que este tipo de comunicación incluye muchos conductores eléctricos que son usados en la capa física y ayudan para transmitir una cantidad definida de bits. Por lo que, si se envía un dato de 8 bits a un mismo canal, se requerirán ocho líneas de datos y una de tierra. Esta cifra puede incrementarse dependiendo de la cantidad de bits que se necesiten enviar.

Este tipo de comunicación ayuda principalmente cuando se requiere de una comunicación rápida y precisa. Una de las desventajas que, de este tipo de comunicación, es que está limitada a distancias cortas. Aunque también esa característica se compensa dado que su aplicación en hardware no suele ser complicada.

2.4.3 Comunicación serial

La comunicación serial es un protocolo de comunicación estandarizado que permite el intercambio de información en forma de bits entre dos o más dispositivos. Realiza el envío de datos bit a bit de forma secuencial generando un flujo de bits. Este tipo de comunicación solamente requiere de una línea de comunicación, lo que es bastante práctico cuando se debe transmitir a larga distancia, a diferencia de la comunicación paralela que requiere de varias líneas de transmisión.

Los estándares de la comunicación serial fueron definidos en 1969 por el RS-232 (Recommended Standard 232) este establece los niveles de voltaje y la velocidad de la transmisión de los datos (Tozer, 2004). La desventaja de este tipo de comunicación es que el envío de la información se vuelve muy lenta ya que obliga a esperar a que un dato se termine de enviar para poder continuar con el siguiente dato, este proceso está diseñado para que no exceder la capacidad de la línea de comunicación de un canal y que la información sea enviada correctamente.

2.4.4 Protocolo UART

UART es el acrónimo *de Universal Asynchronous Receiver-Transmitter*, (transmisor-receptor asíncrono universal) y define un protocolo o un conjunto de reglas, para transferir datos en serie entre dos dispositivos. El UART transmite y recibe datos en ambas direcciones y consta de dos cables, (Transmision, Tx y Recepción, Rx) entre el transmisor y receptor ambos dispositivos también tienen una conexión a tierra (GND), como se ve en la figura 8.

Los datos se transmiten en forma de tramas La comunicación en el UART puede ser simplex (los datos se envían en una sola dirección), semidúplex (cada lado transmite, pero solo uno a la vez), o dúplex completo (ambos lados pueden transmitir en simultaneo). Entre las principales funciones que tiene la UART, son: el manejo de las interrupciones de los dispositivos conectados al puerto serie y de convertir los datos en paralelo a datos en serie, para que sean transmitidos por el puerto serie de los dispositivos.

La UART normalmente no genera o recibe directamente las señales externas entre los diferentes dispositivos, para ello normalmente se utiliza una interfaz separada para convertir las señales de nivel lógico del UART a los niveles de señalización externos, como puede ser la interfaz serial RS232. Así Ayuda a la transferencia de datos recabados y procesados por un microcontrolador. Este tipo de protocolo es asíncrono, esto significa que no existe un sistema de reloj que controla el envío de información que están procesando los bits.

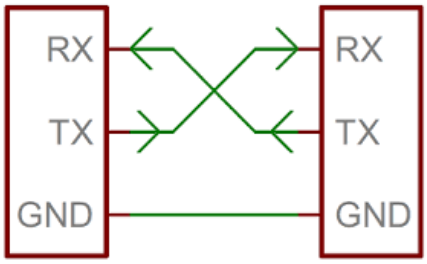

Figura 8: Transmisión de información UART

En el caso de Tx este suele tener un estado alto que suele ser de 0 a 1, para poder enviar la información en bits este estado se cambia a bajo para poder hacer la distinción de la información al final del envío. Cuando este llega a su destino final el valor vuelve a ser alto. En el caso de la velocidad de transmisión el valor inicial siempre es de 9600 baudios por segundos. Entonces si se calcula 1/9600 cada bit tiene una duración de 104 us (microsegundos) (Jin, 214). Este tipo de protocolo es llamado uno a uno y posee una transmisión de 20 kbps. Es muy utilizado en aplicaciones donde se necesita una muy buena precisión de la información y también es muy fácil de conectar con algunos microcontroladores como Arduino ya que sirve para subir el programa y también a interactuar con la información por el monitor serial.

2.5 Arduino®

“Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de utilizar”, las placas de Arduino están basadas en un microcontrolador ATMEL reprogramable y contiene una serie de terminales (pines) analógicos y digitales que permiten la lectura de entradas (sensores, pulsadores u otros dispositivos o sistemas) y enviar datos procesado a una salida, como puede ser un motor, encender un LED o enviar datos a otro dispositivo o sistema. Su programación consta de un conjunto de instrucciones que se pueden enviar al microcontrolador en la placa desde un computador. Para ello se utiliza el lenguaje de programación Arduino (basado en Wiring), y el Software Arduino (IDE), basado en Processing. (What is Arduino?, s.f.)

La tarjeta de Arduino es una placa PCB (Printed Circuit Board, Placa de circuito impreso) que contiene todos los componentes y elementos necesarios para permitirle al usuario desarrollar directamente una aplicación o sistema electrónico basado en microcontrolador.

La tarjeta de Arduino se utiliza principalmente para lo siguiente:

- Como un microcontrolador, programado desde un computador y que funciona de forma independiente de éste, puede controlar y alimentar dispositivos específicos y es capaz de procesar datos y tomar decisiones de acuerdo a su programación e interactúa con su entorno a través de sensores y actuadores.
- Como una interfaz entre un computador y otros dispositivos, como son sensores y actuadores que están conectados a la placa Arduino permitiendo que el computador ejecute una determinada acción.

Existen diferentes modelos de placas Arduino, cada una de las tarjetas tiene un propósito determinado y diferentes características como son: el tamaño físico, los números de pines, número y tipo de líneas de entrada y salida, el modelo del microcontrolador, entre otras. En la figura 9 se muestra una placa Arduino MEGA y en la tabla 3 se tienen las principales características. La selección del modelo no afecta al funcionamiento de alguna aplicación ya que todas contienen las mismas características de software como lo es la arquitectura, librerías e IDE de compilación.

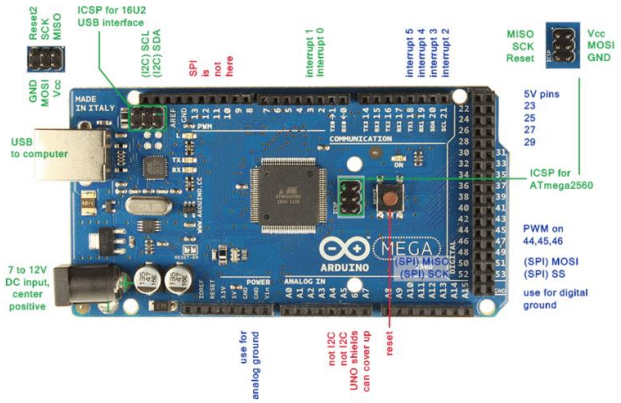

Figura 9: Arquitectura de la tarjeta Arduino Mega

Tabla 3: Características del Arduino

Microcontrolador	ATmega2560
Voltaje Operativo	5V
Voltaje de Entrada	7-12V
Pines digitales de entrada/salida	54(de los cuales 15 proveen salida PWM)
Pines analógicos de entrada	16
Memoria Flash	256kb (8kb usados por el bootloader)
EEPROM	8kb
Velocidad de reloj (Clock Speed)	16Mhz

2.5.1 Energía y Alimentación

Arduino puede ser alimentado mediante el puerto USB (5Vdc) o con una fuente externa de alimentación (rango de alimentación recomendado desde 7Vdc hasta 12Vdc). La alimentación es seleccionada de manera automática. Cuando se trabaja con una fuente externa de alimentación se debe utilizar un convertidor AC/DC de voltaje regulado en el rango de operación de la placa. De igual manera se puede alimentar la placa de Arduino mediante el uso de baterías.

2.5.2 NodeMCU ESP8266

Es una plataforma de desarrollo similar a Arduino basada en el microcontrolador ESP 8266 y está orientado al internet de las cosas. Esta placa tiene un núcleo SoM ESP-12E el cual está basado en SoC Wi-Fi ESP8266, además integra un conector micro-usb que se utiliza para la programación y comunicación en el computador (Cameron, 2021). En la figura 10 se puede observar el diagrama esquemático y los pines de la NodeMCU.

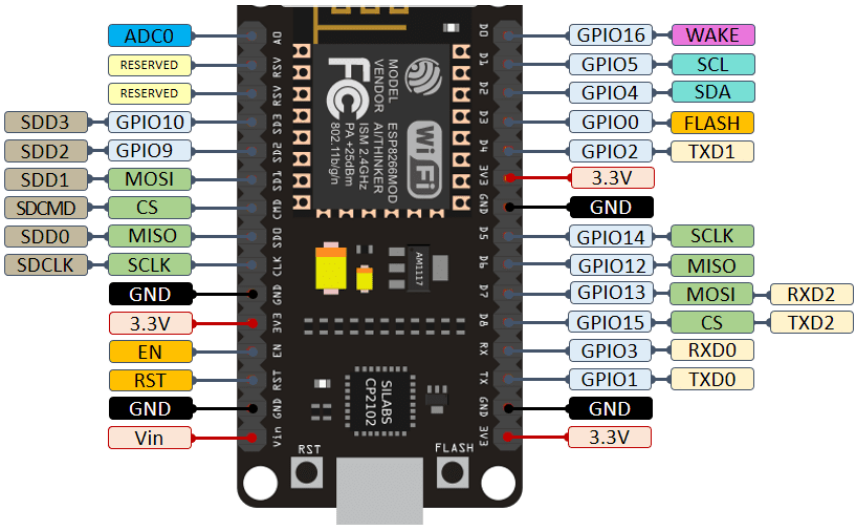

Figura 10: Diagrama de NodeMCU 8266

NodeMCU tiene un firmware que permite trabajar con diferentes lenguajes de programación como: Arduino, Lua, MicroPython, C/C++, Scratch entre otros. En el entorno de Arduino es posible utilizar el IDE para compilar los programas que serán enviados al microcontrolador de la placa. La tarjeta NodeMCU 8266 permite utilizar las librerías desarrolladas para Arduino que son de uso gratuito y se encuentran disponibles en internet.

2.6 Dispositivos y plataformas de IoT

Una plataforma IoT es un software que permite conectar diferentes dispositivos como sensores, actuadores y dispositivos de comunicación a un entorno digital, generando una red para que éstos puedan comunicarse y transferir información. Existen diferentes tipos de plataformas IoT: Plataformas cloud (también conocidas como plataformas de habilitación de aplicaciones), Plataformas de conectividad, Plataformas de dispositivos, Plataformas de análisis de datos, sin embargo, en cualquier caso, están integradas por los siguientes componentes:

- **Hardware**: Dispositivos y sensores con la función de captar información del entorno y actuar sobre él.
- **Software**: Se encarga de analizar la información enviada y tomar decisiones

al respecto. Normalmente, este software se encuentra en la nube, generando así las plataformas IoT cloud.

- **Conectividad**: Es la red que interconecta a los dispositivos y a través de la cual se efectúa la transmisión de datos y órdenes desde el hardware a la nube.
- **Interfaz de acceso:** Facilitan la interacción entre usuarios y sistemas IoT, son el punto de acceso del equipo humano a la información recolectada por el resto de elementos.

2.6.1 Arduino® e IoT

Arduino permite la comunicación de forma sencilla con diferentes tipos de dispositivos (sensores o actuadores), ya que contiene varios complementos los cuales permiten la conexión e interacción con internet, uno de estos complementos es un módulo de Ethernet y diferentes complementos para conectividad inalámbrica utilizando diferentes protocolos de comunicación, por ejemplo, WiFi, BlueTooth, Zigbee o LoRaWAN. Los controladores permiten recibir la información de los sensores y así manejar la información recopilada y enviarla a la nube. Para manejar la información recibida de los sensores se dispone de diferentes aplicaciones que estén basadas en IoT. Por ejemplo, la plataforma de *Temboo* es una plataforma de IoT que nos permite conectar fácilmente a la nube mediante una API (Appilcation Programming Interface, Interfaz de programación de aplicaciones) un Arduino para interactuar con los datos desde un navegador web e interconectar con servicios de terceros, también existen diversas plataformas que nos permiten la misma interacción y de forma gratuita como lo es Firebase.

El desarrollo de la tecnología digital en las fábricas y la industria de producción es lo que se conoce como Industria 4.0, de aquí se deriva otro concepto llamado M2M (Machine to Machine, máquina a máquina), esta es una forma amplia de referirse a la tecnología que permite a los dispositivos, los objetos y las máquinas intercambiar información y realizar acciones sin la ayuda de personas, ya sea que estén

conectados de forma inalámbrica o por cable. Gracias a esta tecnología, el intercambio de información puede darse tanto en una red privada como en una red. Esto permite que la industria esté conectada al exterior con todos los elementos que la rodean y así compartir o recibir la información de los dispositivos de manera remota.

Según (Picone, 2020) existen 3 preguntas básicas que se deben realizar al momento de implementar algún sistema con IoT, las preguntas esenciales son:

¿Qué se quiere medir?

Esta nos sirve para identificar el sensor a utilizar y si este se adaptara de manera correcta a nuestro microcontrolador. Para ello se necesita plantear el escenario donde será utilizada la tecnología como puede ser estacionamientos inteligentes, métodos de reducción de contaminación atmosférica, control de riego y control de procesos de producción.

¿Cómo se desea conectarse a internet?

Este apartado nos indica que ya seleccionada la tecnología que se va a utilizar es posible seleccionar el protocolo de comunicación que se utilizará para conectar los dispositivos a la red de datos y enviar la información al microcontrolador. La elección de la tecnología dependerá de las opciones de conectividad que contenga cada zona donde se implementará el sistema, así por ejemplo podría utilizarse Ethernet, WiFi o una red 4G/3G o GPRS.

¿Qué se requiere hacer con los datos?

Por último, una vez respondidas las preguntas anteriores es necesario elegir una plataforma de software que permita la conexión a la nube y así establecer la conexión e intercambio de información de manera remota. Para ello se deberá seleccionar la que mejor se adecúe al sistema ya sea por el aspecto económico, por la capacidad de mensajes que puede recibir, la seguridad de la información etc.

2.6.2 Plataforma de IoT Firebase

Como se vio anteriormente, una plataforma de IoT es un software o aplicación que permite la conexión a la nube y la recolección de datos generados por los sensores o actuadores, regularmente estos van conectados a un microcontrolador que procesa los datos recabados. Estas aplicaciones proporcionan una gran cantidad de servicios (Hong, 2020) como los son:

- Inserción de datos
- Transformación de datos
- Creación de una interfaz
- Gestión de reglas
- Gestión de dispositivos
- Servicios de seguridad
- Integración de la plataforma IoT

La integración de datos de varios sensores permite trabajar con la información conjunta y pude ser enviada por diferentes tarjetas Arduino. El enviar la información a la nube permite almacenar los datos recopilados por los sensores en una base de datos. Esta información puede ser visualizada a través de una interfaz gráfica que ayuda al análisis de información y el manejo de la misma de manera más eficiente.

Existe una gran variedad de plataformas que permiten la interacción de Arduino con dispositivos IoT. El problema que se presenta regularmente es que estas no son de uso libre, es decir son plataformas propietarias y tienen un costo o tarifa dependiendo de la cantidad de mensajes (datos) que se reciben al día y suelen ser de un alto costo, ya que este tipo de plataformas suelen ser utilizadas por industrias que recopilan una gran cantidad de datos. Por ello se investigaron y analizaron diferentes plataformas IoT, las cuales se abordarán a continuación y así seleccionar una plataforma adecuada con la que trabajará el sistema a desarrollar.

Una de las plataformas a analizar es Google firebase, esta nos permite visualizar información en la nube y así poder conectarnos a nuestra base de datos en tiempo real. Esta aplicación utiliza diferentes librerías que ayudan a la plataforma de Arduino a conectarse a Google. Las librerías ayudan a identificar el tipo de señal que es enviada por los sensores y así poder gestionar la información obtenida.

Las características principales de *Firebase* según (Giraldo, 2019) son:

- Es una herramienta que trabaja con diferentes plataformas móviles ya sea Android, IOS o páginas web.
- Permite ganar dinero mediante el uso de Admob, ya que habilita la opción de generar dinero mediante anuncios o publicidad
- Permite un desarrollo gratuito de nuestras aplicaciones y en caso de manejar niveles más avanzados de recopilación de información, se puede solicitar un plan de pago.
- Genera crecimiento y un mejor desarrollo de nuestras aplicaciones. Ya que ofrece diferentes servicios que nos ayudan a mejorar o analizar mejor nuestra app.

Para el *backend* de este proyecto, se utilizará Google Firebase. Este nos permite almacenar en la nube la información adquirida de los sensores y así poder visualizar la información de manera remota. Para el *frontend* de este proyecto se desarrollará una página web, diseñada en HTML y JavaScript.

2.6.2.1 ¿Qué servicios ofrece Firebase?

Los servicios que ofrece Google Firebase para el desarrollo se dividen en 3, que son desarrollo, crecimiento y análisis.

En el desarrollo tenemos diferentes utilidades que son:

- Base de datos en tiempo real: este nos ayuda para obtener información de una base de datos en tiempo real.
- Autentificación: este nos ayuda para identificar a los usuarios mediante un

email o también introduce la opción de registrarse por medio de alguna red social como puede ser Facebook, Twitter, Yahoo! entre otras.

- Nube de almacenamiento: se utiliza para el almacenamiento y envío de archivos
- Alojamiento: se utiliza para publicar la página web.
- Configuración Remota: Se emplea para modificar algunos aspectos de nuestra aplicación sin tener que actualizar nuestra app.
- Laboratorio de pruebas: se utiliza para probar la aplicación antes de publicarla
- Informe de incidentes: se utiliza para reportar los errores que puedan suceder en nuestra aplicación.

En la parte de crecimiento se tienen las siguientes características:

- Notificaciones: permite enviar notificaciones a los usuarios.
- Indexación de aplicaciones: permite mostrar las aplicaciones de una forma adecuada en los motores de búsqueda.
- Links dinámicos: proporciona una forma de acceder a la aplicación desde diferentes links, desde otras aplicaciones.
- Trabajos publicitarios: permite hacer publicidad en la aplicación
- Monetización: ayuda a monetizar nuestra aplicación mediante los anuncios y la publicidad.

En la parte de análisis se cuenta con una función llamada *Analytics* y esta muestra los resultados sobre el comportamiento de nuestra App. Esto nos ayuda a tomar mejores decisiones sobre nuestro sistema web y así también poder diseñar una estrategia de marketing que nos permita sacarle un mayor partido a nuestro sistema.

2.6.3 HTML5

Todo lo que se ve en internet está programado con un código interno, cuando los

usuarios acceden a un sitio web, el navegador recibe el código y lo transforma de forma visual para que así se pueda visualizar lo que el programador web a diseñado. Este código es el que decide la estructura que llevará una página web. A estos códigos se les llama lenguajes de programación y el lenguaje que se utiliza en la web mundial es el HTML. Sus siglas son HyperText Markup Language, que significa lenguaje de marcado de hipertexto (Gauchat, 2012).

En 1999 se lanzó el estándar de HTML4 y conforme fueron pasando los años y dada la actualización tecnológica de las páginas web se fue implementando el nuevo estándar de HTML5. Este es la última versión del estándar de HTML que se utiliza para la creación de páginas web e incorpora novedades notables como, por ejemplo; la reproducción de contenido multimedia, de forma que ya no tengas que recurrir a recursos de terceros como lo es Flash Player.

Gracias a la actualización a HTML5 se pueden programar aplicaciones web que pueden ser utilizadas como apps, es decir no se necesitará la instalación para poder ingresar cierta página. Es decir, la app se vuelve independiente en la PC o móvil ya que podrá ser utilizado desde el navegador. También se han añadido opciones de geolocalización, de manera que nuestra página web puede detectar la ubicación de los usuarios que ingresan a ella. Así se pueden ofrecer diferentes opciones de idiomas dependiendo del país donde se esté accediendo a la página web, o redirigirte a la página especifica que contenga de un país.

2.6.4 JavaScript

Es un lenguaje de programación que se utiliza regularmente para crear páginas web dinámicas. Es un lenguaje interpretado, por lo que no es necesario compilar el código para poder ejecutarlo, ya que se pueden probar directamente en el navegador (Pérez J. E., 2009).

2.6.4.1 ¿Qué es JavaScript?

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo con funciones de primera clase. Es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y usado en muchos entornos fuera del navegador, JavaScript es un lenguaje basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa. Permite añadir características interactivas a un sitio web.

2.6.4.2 Cualidad y desventaja

Desde la aparición de JavaScript la cantidad de usuarios que utilizan este lenguaje de programación para crear páginas web ha aumentado a través de los años. Con la aparición de aplicaciones AJAX programadas con JavaScript ha contribuido al aumento y la facilidad de utilizar para la creación de páginas web.

JavaScript presenta algunas limitaciones, por ejemplo, los scripts no pueden comunicarse con recursos que no pertenezcan al mismo dominio de donde proviene el script. Además, no pueden acceder a los archivos del ordenador, ya sea en modo lectura o en modo escritura. Tampoco pueden leer o modificar las preferencias del navegador (Pérez J. E., 2009).

2.6.4 Arduino y Firebase

La combinación de Arduino con Firebase se ha convertido en una opción comúnmente utilizada para desarrollar y realizar proyectos de IoT. Arduino cuenta con una biblioteca de trabajo y de comunicación que permite trabajar de manera conjunta con Firebase. En la tabla 4 se muestran los datos que Firebase permite administrar bajo el plan de acceso libre (sin costo).

Tabla 4: Limite de datos con el paquete gratuito
Fuente: (Firebase, 2022)

Nivel gratuito	Cuota
Datos almacenados	1 GB
Operaciones de lectura de documentos	50,000 por día
Operaciones de escritura de documentos	20,000 por día
Operaciones de eliminación de documentos	20,000 por día
Salida de red	10 GB por mes

Firebase utiliza una librería especial que permite establecer la comunicación con Arduino, esta se puede descargar desde la página oficial de Firebase/Google. Contiene una huella dactilar que se actualiza aproximadamente cada 6 meses esto proporciona seguridad de los datos que se envían a la nube. Los datos que son enviados a Firebase a través de Arduino tardan aproximadamente 2 segundos en recibir correctamente la información y mostrar el cambio en tiempo real de los datos enviados por el microcontrolador.

Firebase incluye varias acciones que permiten la interacción con la placa de Arduino estas acciones son las siguientes:

- **FirebaseHost:** Es la dirección del canal creado en la plataforma de Firebase.
- **FirebaseAuth:** Es la contraseña del sistema que es obtenida a través de la información de la base de datos creada en nuestro servidor.

CAPÍTULO III. METODOLOGÍA

El implementar diferentes tecnologías en una sola plataforma para detectar espacios vacíos u ocupados en estacionamientos públicos no es una tarea fácil. En los últimos años se han realizado diversas propuestas para la detección eficaz de espacios libres en estacionamientos. Sin embargo, esta es un área de conocimiento relativamente nueva en la cual el camino para lograr una arquitectura de hardware adaptable no está completamente definido. En esta sección se describe el proceso que seguiremos para diseñar dicha arquitectura, las herramientas y los materiales a utilizar. También se presentan los métodos para la conectividad del estacionamiento a través de IoT.

3.1 Materiales y Herramientas

El sofware que se va a utilizar es el siguiente:

- **JavaScript:** Es un lenguaje de programación que utiliza secuencias de comandos que permiten implementar funciones en páginas web. También puede ser utilizado para la creación de interfaces y comunicación con HTML5 para la programación de un frontend de una página web.

- **Firebase:** Es una plataforma en la nube para el desarrollo de aplicaciones WEB y móvil creada por Google cuya principal función es desarrollar y facilitar la creación de aplicaciones para dispositivos móviles que cuenten con una alta calidad a pesar de su rápida elaboración; esto con la finalidad de que se pueda incrementar la base de datos de usuarios. Esta plataforma permite gestionar, controlar y monitorear sensores a través de Arduino. Puede recibir 1 GB de información al día (en el plan sin costo), que permite mantener el sistema abierto a los usuarios por una gran cantidad de tiempo. Esta plataforma nos permite centralizar toda la información para posteriormente hacer la gestión de la información y recabar los datos en la

nube.

- **HTML5:** Es un estándar que sirve como referencia al software que conecta con la elaboración de páginas web en diferentes versiones, está definido por una estructura básica y un código que es conocido como HTML.

Los materiales que se utilizarán son los siguientes:

- Sensores ultrasónicos
- Computadora
- Borneras de instalación
- Base de sensores
- Cables de conexión
- NodeMCU 8266

3.2 Planteamiento del modelo a emplear

Tomando como base el trabajo de (Luna, 2016), se proponen las siguientes metodologías de diseño centrado en el usuario para el desarrollo del trabajo:

1.- Diseño: A través de un esquema desarrollado en un programa orientado a objetos como Visual Basic se validará el funcionamiento del estacionamiento. Se desarrollará una interfaz gráfica de usuario (GUI) donde se visualizará el estado en tiempo real de la ocupación de los espacios de estacionamiento, la detección de los vehículos que ocupan dichos espacios es usando los módulos de sensores ultrasónicos. En la figura 11, se muestra una pantalla de la interfaz gráfica donde se visualiza la ocupación de los espacios de estacionamiento.

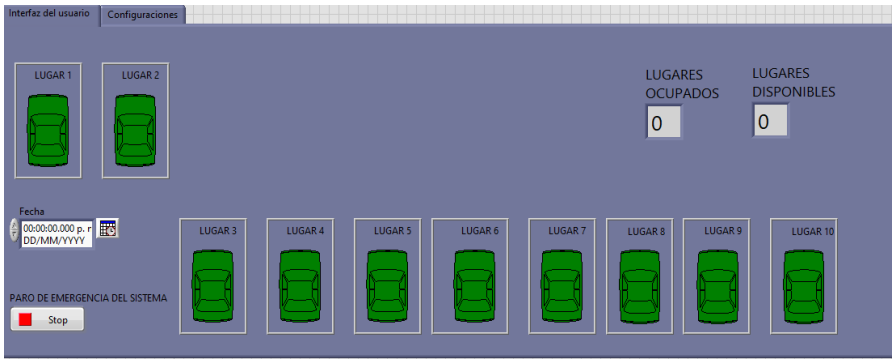

Figura 11: Interfaz para la monitorización de los espacios vacantes

2.- Evaluación: Se instalarán sensores ultrasónicos en un estacionamiento privado para validar la operación del diseño. Además, en una simulación se utilizarán valores reales para corroborar que la simulación hecha funcione correctamente. Además, se recabará información de métricas sobre:

a) Eficacia: Datos concretos sin pérdidas de conexión en los sensores, dando valores certeros de medición
b) Facilidad de aprendizaje: Tener un fácil dominio o control del número de estacionamientos disponibles y que sea entendible para los usuarios
c) Eficiencia: Tiempo promedio de ejecución del programa después de que el sensor detecte el automóvil
d) Cualidad de ser recordado: Tiempo promedio para recuperar la capacidad de ejecutar una tarea, después de un periodo de utilizar la plataforma

Otra de las metodologías que se utilizará será basado en la propuesta de (Mufaqih, 2020) que se divide en seis pasos como se ve en la figura 12.

1) Se evalúa la situación actual: Se trata de una revisión detallada que permite la comprensión de muchos factores que pueden presentarse en la realización del proyecto.
2) Recolecta la información del problema: Se obtienen los datos necesarios del problema y se plantea cual es la problemática a resolver.
3) Análisis y conclusión de la información: Consiste en someter los datos a la realización de operaciones para así obtener conclusiones precisas que nos

ayuden a alcanzar los objetivos. También nos ayuda para revelar ciertas dificultades que se puedan presentar en la realización del proyecto.

4) Definir los requerimientos: se especifican las características operacionales que tendrá el sistema. Se realizan entrevistas, investigaciones, observaciones y demás técnicas específicas para la recolección de la información. Con ello se describe el plan a seguir.
5) Revisión de tecnología: Se especifica cual es el tipo de tecnología que mejor se adapta a la problemática y bajo que protocolos van a comunicarse.
6) Propuesta de la estrategia a realizar: Ya que se ha recabado completamente la información se realiza la arquitectura para la implementación del proyecto.

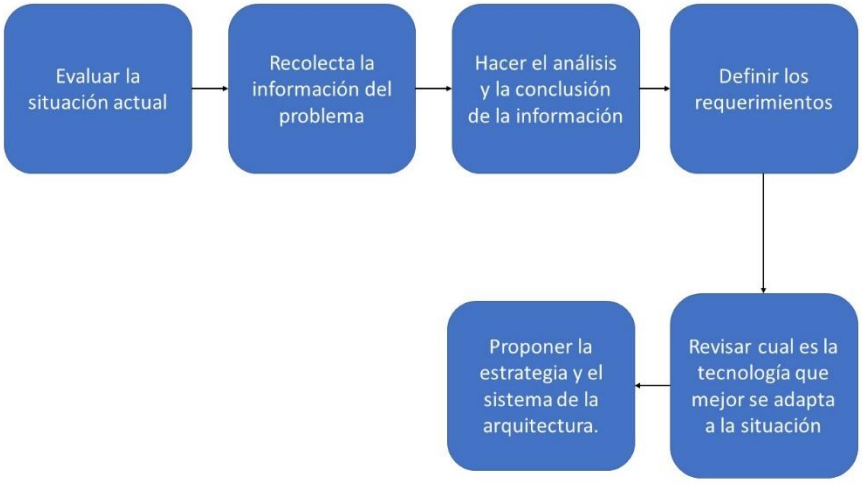

Figura 12: Pasos de la propuesta metodológica
Fuente: Creación propia

Para identificar la ocupación o no de los espacios vacíos se utilizarán sensores ultrasónicos HC-SR04 como los que se observan en la figura 13. Con estos sensores es posible detectar si algún vehículo está cerca, ya que estos sensores nos permiten medir la distancia hacia a algún objeto. El programa desarrollado va a detectar si un objeto esta frente al sensor. Para lograr esto programaremos el sensor de tal manera que solo detecte un objeto cuando el objeto pase de 120 cm de ancho.

Figura 13: Sensor ultrasónico HC-SR04
Fuente: (Julio, 2022)

3.3 Conexión y protocolos de las redes de sensores

Estos sensores van a ser conectados a través del estándar de comunicación IEEE 802.15.4. Este es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (Low-Rate Wireless Personal Area Network, LR-WPAN) (Romero, 2017). El grupo de trabajo IEEE 802.15.4 es el responsable de su desarrollo. También es la base sobre la que se define la especificación de ZigBee, cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de la pila de protocolos que el estándar no cubre como se ve en la figura 14.

En principio, el ámbito donde se prevé que esta tecnología cobre más fuerza es en los sensores que ocupen las tecnologías de IoT. La razón de ello son diversas características que lo diferencian de otras tecnologías:

- Su bajo consumo.
- Su topología de red en malla.
- Su fácil integración (se pueden fabricar nodos con muy poca electrónica).

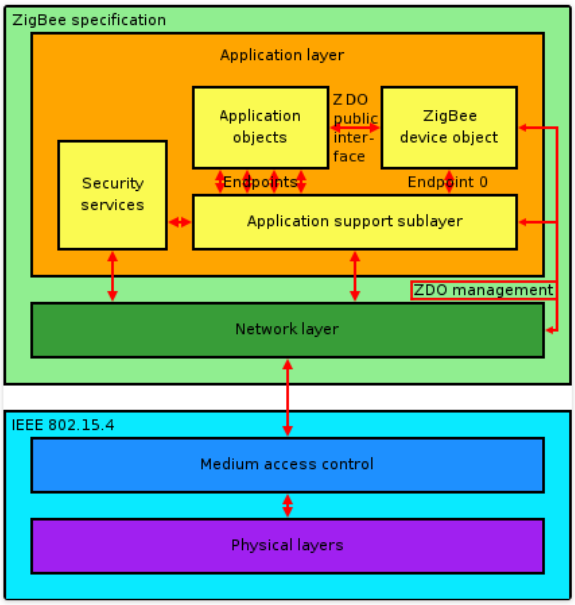

Figura 14: Especificaciones ZigBee

3.4 Selección de las variables de observación del sistema

El sistema desarrollado consta principalmente de 4 variables de trabajo que garantizan una buena operación, estas variables son las siguientes:

- Funcionalidad
- Eficiencia
- Fiabilidad
- Mantenibilidad

Las variables constan de unidades experimentales que permitirán evaluar de manera diferente cada una de manera individual, por ejemplo: en el caso de la funcionalidad se contempla la seguridad del sistema y la precisión del mismo. En la parte de eficiencia se tiene el desempeño del sistema. En la parte de la fiabilidad tenemos la estabilidad operativa y la respuesta del sistema ante ciertos tipos de contingencias. Por último, en la parte de mantenibilidad se tiene la seguridad del mantenimiento, la facilidad del mantenimiento y la realización de algunas modificaciones a nuestro sistema que ayude al mantenimiento del mismo. Cada uno

de ellos contiene diferentes tratamientos a incluir con diferentes tipos de métricas con las que se podrán comprobar las variables del sistema.

Seguridad del sistema: los usuarios que tienen acceso a la información importante o sensible serán categorizados como nivel de acceso privilegiado de "*super usuario*". Esto permitirá tener una mayor seguridad de la información que es manejada por el administrador y garantiza un mejor manejo de los datos y ayuda a la seguridad del sistema.

Precisión del sistema: este se evaluará minimizando los falsos negativos vs los falsos positivos bajo la fórmula vista en la figura 15.

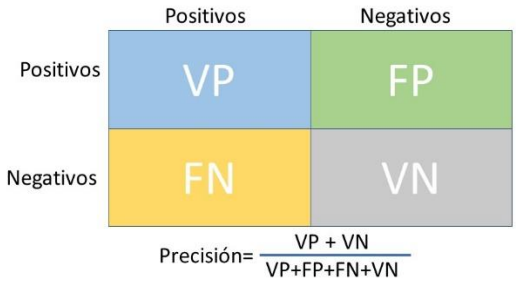

Figura 15: fórmula para el cálculo de la precisión del sistema
Fuente: (Creación propia)

Desempeño del sistema: Los indicadores de gestión de desempeño mejor conocidos como KPIs (key performance indicator) son métricas establecidas que ayudan a cuantificar y evaluar la calidad del sistema (D. Setijono, 2007). Existen 4 indicadores que nos ayudan a medir el desempeño del sistema que son:

- Indicadores de productividad
- Indicadores de calidad
- Indicadores de eficacia

- Indicadores de nivel de formación.

En este sistema para medir nuestro desempeño se calculará la productividad que tiene nuestro trabajo para ello se evaluará el beneficio por sensor. Este es un indicador que permite conocer que beneficio genera cada sensor dentro del sistema. En este caso un resultado alto es un buen indicador de desempeño del sistema.

Estabilidad operativa del sistema: La mayoría de los modelos de fiabilidad relativos a hardware van más orientados a fallos o desajustes que pueden ser provocados por el desgaste físico, efectos de temperatura, deterioro de los materiales y golpes causados por usuarios. En cuanto al software comúnmente los errores se producen por problemas de diseño o errores de implementación para ello se tiene una medida sencilla de fiabilidad donde se calcula la estabilidad operativa del sistema (JW. Gallagher, 2007). Una medida sencilla es TMEF (tiempo medio entre fallos) esta se calcula de la siguiente forma:

$$TMEF = TMDF + TMDR$$

Donde *TMDF* es el tiempo medio de fallo y *TMDR* es el tiempo medio de reparación del fallo. Esta medida resulta muy útil ya que el usuario se enfrenta a los fallos y no al número de errores que se presentan en el sistema

Respuesta a contingencias: para esto se propone un protector contra sobrevoltaje para que en caso de alguna falla eléctrica los dispositivos se mantengan seguros. También en caso de una falla eléctrica se deberá desenchufar el sistema para protegerlo de alzas de tensión.

Mantenibilidad: para medir esta variable se tiene el índice de mantenibilidad (IM) este mide la mantenibilidad del sistema, esta se divide en 3 etapas:

- Comprensión de los cambios que se hacen
- Realización de las modificaciones necesarias

- Pruebas de los cambios realizados

El índice de mantenibilidad fue creado para cuantificar la mantenibilidad de un sistema este se calcula de la siguiente manera:

$$IM = 171 - 5.2 * ln(aveV) - 0.23 * aveV(g) - 16.2 * ln(aveLOC) + 50 * sen(2.4 * perCM)$$

Donde $aveV$ es la cantidad de sensores que posee el sistema, $aveV(g)$ es la media de complejidad que posee cada sensor, $aveLOC$ es la media de datos que informa cada sensor y $perCM$ es la media porcentual de falsos negativos que posee el sistema. SI el IM es alto, este contendrá mayor mantenibilidad.

3.4.1 Método de validación de las variables

El método a utilizar para calcular o revisar las variables es ALMA (Architecture Level Modifiability Analysis). Este menciona los siguientes pasos, así como se observa en la figura 16 y permiten dar el seguimiento correcto al proyecto.

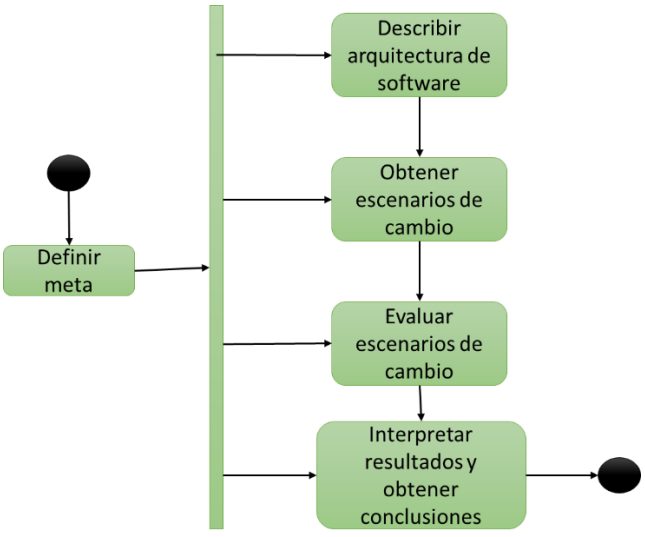

Figura 16: Pasos del método alma

- Se define la meta de evaluación: en este caso definiremos cual va a ser el

objetivo de la evaluación.

- Describir la arquitectura de software: en esta sección se colecta la información más relevante de la arquitectura, se incluyen sus principales componentes, las relaciones entre éstos, así como las relaciones con el entorno del sistema.
- Obtener escenarios: En esta parte se definen los escenarios de cambio y se agrupan en diferentes categorías. Esto se refiere a que se analizara como el sistema interactúa en diferentes situaciones y adversidades ya sean climáticas o de arquitectura. Así se podrá analizar la vida operacional del sistema
- Evaluar escenarios: Aquí se realiza un análisis de impacto, que consiste en identificar cuáles son los componentes afectados por los escenarios ya antes planteados y así determinar el efecto de cambio que tiene sobre los componentes o el sistema. Los resultados de este análisis se deben expresar ya sea de manera cuantitativa o cualitativa.
- Interpretar resultados: Por último, se interpretan los resultados, junto con las conclusiones del análisis del sistema.

3.5 Protocolos de comunicación para IoT

Existen diferentes protocolos de comunicación para interconectar una tecnología con IoT, la más utilizada es conocida como *LPWAN* (Low Power Wide Area Network) que contiene una serie de protocolos internos que ayudan a la comunicación del sistema con internet. Este se ocupa en redes de alta y baja potencia, es un protocolo para el transporte inalámbrico de los datos y es uno de lo más básicos a la hora de implementar IoT (Chachin, 2017).

El protocolo *LPWAN* tiene 3 características importantes que ayudan a los requerimientos que pide IoT los cuales son (Sheshalevich, 2017):

- Alcance geográfico: *LPWAN* está diseñado para transportar los datos de manera inalámbrica a largas distancias. En el caso de estar trabajando en un rango donde la distancia pueda medirse en metros, se sugiere mejor crear

un sistema cableado.

- Cantidad de datos enviados: *LPWAN* contempla un transporte de datos regular, es decir puede manejar una cantidad grande de datos, pero en caso de que los datos no sean enviados, el protocolo cierra su vía de transmisión. Esto es una forma de asegurar la información de los datos enviados por otros sensores y ayuda a el informe de errores, ya que al no recibir la información de un sensor se puede deducir que hubo algún error interno.
- Bajo consumo eléctrico: El protocolo no consume una cantidad excesiva de energía, ya que solo se encarga de la administración de la información y la recepción de la misma. El ancho de banda ocupado se mantiene al mínimo dependiendo la red que se utilice para la conexión a internet.

Este protocolo es compatible con el protocolo *Zigbee* para la conexión de sensores y con herramientas WiFi de Arduino que permiten el envío de la información de los sensores, también pueden manejar redes bluetooth o móviles como 3G. Este protocolo permite la transferencia de datos e información de forma inalámbrica de los sensores a internet. También es el más usado al momento de procesar información de una red de sensores o de dispositivos que están distribuidos geográficamente.

3.5.1 Protocolos LoRa y LoRaWAN

Estos protocolos inalámbricos de larga distancia y baja velocidad, que permiten la conexión de sensores y otros dispositivos con un mantenimiento y gasto de energía mínimo, haciendo que las baterías puedan durar años.

LoRa: Es una tecnología inalámbrica que emplea la modulación por radiofrecuencia que utiliza la tecnología de espectro ensanchado con una banda más ancha. Su chirp modulado por frecuencia emplea ganancias de codificación para una mayor sensibilidad del receptor. LoRa es desarrollada para permitir la comunicación de

datos de baja velocidad a larga distancia entre dispositivos para aplicaciones entre máquinas (M2M). También se utiliza para aplicaciones de IoT en las que se utilicen redes de sensores y actuadores que ayuden a la creación de ciudades inteligentes, actuadores para una mayor cobertura en lugares de explotación agrícola, etc.

LoRaWAN: Es un protocolo de red que utiliza la tecnología de LoRa, para redes de baja potencia y área amplia, *LPWAN* (Low Power Wide Area Network) utilizado para comunicar y administrar dispositivos LoRa. Se puede decir que LoRaWAN se encarga de unir diferentes dispositivos LoRa gestionando sus canales y parámetros de conexión, como son: canal, ancho de banda, factor de dispersión y tasa de codificación y cifrado. Este protocolo se compone de diferentes Gateways (pasarelas) y nodos que ayudan a la administración de la información.

- Gateway: Es el que recibe la información y se encarga de procesarla y enviarla a los diferentes nodos
- Nodos (dispositivos): Esta es la última parte del protocolo que se encarga de recibir la información envida por el Gateway y la envía hacia el usuario final a través de internet.

En la figura 17 se muestra la interconexión del protocolo LoRa y el LowRa para la transmisión de la información.

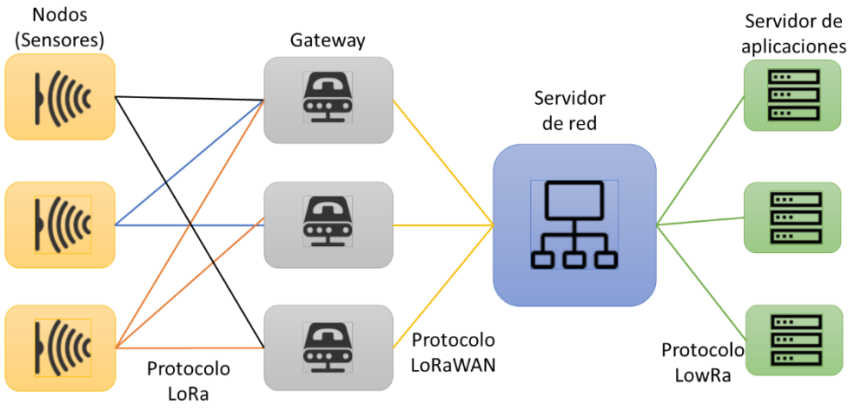

Figura 17: Comunicación de los protocolos
Fuente: Creación propia

3.6 Protocolos de conexión con la nube, protocolo TCP/IP

Actualmente la mayoría de los computadores están conectados a alguna red, ya sea Internet o bien una red local (Intranet) y casi en su totalidad lo hacen utilizando el modelo TCP/IP. Este modelo es una pila de protocolos para comunicación en redes con una arquitectura por capas, que permite que un equipo pueda comunicarse dentro de una red. En la figura 18 se puede observar el modelo TCP/IP, las capas y la suit de protocolos.

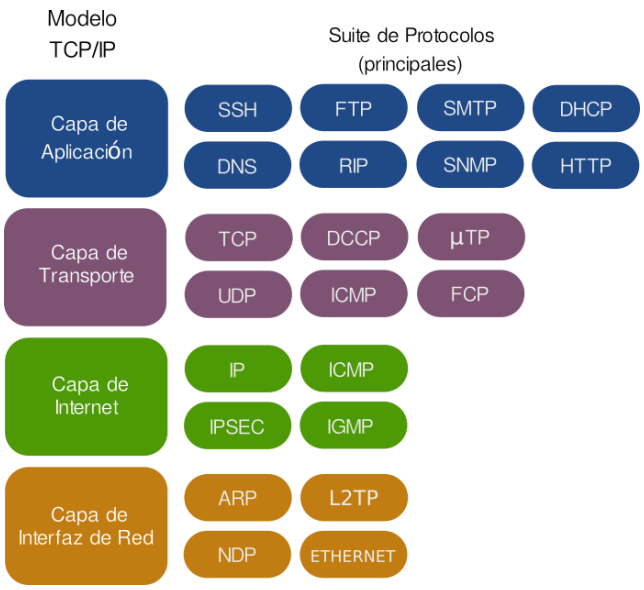

Figura 18: Capas del modelo TCP/IP

El modelo TCP/IP permite la transferencia de datos de forma fiable dentro y fuera de una red, especificando los pasos que se deben efectuar desde que se envían los datos (en paquetes) hasta que son recibidos. Para lograrlo utiliza una arquitectura de capas con jerarquías (capas adyacentes) que se comunican únicamente con su capa superior (a la que envía resultados) y su capa inferior (a la que solicita servicios).

Para conectar los dispositivos de la red de sensores y enviar información a la nube se utiliza el protocolo TCP/IP, permitiendo la conexión de la red de sensores a la nube, para el intercambio de información almacenada en los dispositivos de la red a la plataforma en la nube. TCP/IP es un conjunto o pila de protocolos. Las siglas TCP/IP se refieren a este grupo de protocolos:

- TCP es el Protocolo de Control de Transmisión que permite establecer una conexión y el intercambio de datos entre dos anfitriones. Este protocolo proporciona un transporte fiable de datos.
- IP o protocolo de internet, utiliza direcciones series de cuatro octetos con formato de punto decimal (como por ejemplo 75.4.160.25). Este protocolo lleva los datos a otras máquinas de la red.

La conexión de este protocolo se basa en las reglas de comunicación de direcciones IP, esta dirección se utiliza para enrutar los datos vía internet. El enlace a la nube es compatible con cualquier sistema operativo, sistema de hardware y software. La parte de TCP es donde los datos son segmentados en paquetes de datos para que la información sea manejable y son enviados individualmente a internet. El protocolo IP se encarga de encaminar los datos por la ruta correcta y se basa en una dirección específica (dirección IP del dispositivo) para entregar la información de manera correcta (j. Rut, 2016).

3.6.1 Conexión TCP

Una conexión TCP utiliza información segmentada para determinar si la información está lista para enviarse. Cuando se establece la conexión el protocolo envía un segmento llamdo *SYN* que utiliza la dirección IP de host para la recepción de la información. El protocolo TCP una vez que recibe la información devuelve un acuse de recibo del paquete, este acuse se conoce como *ACK,* el cual tiene como fin confirmar que el segmento se recibe correctamente. Este intercambio de información se le conoce como protocolo de tres vías (Hunt, 1987).

3.6.2 Envío de información a la Nube

Por último, para realizar el procesamiento de datos y subirlos a la nube se utilizará la plataforma Firebase. Este es un servicio que permite gestionar, controlar y monitorear dispositivos o unidades. Cada dispositivo puede tener conectados multitud de sensores, los cuales a su vez son objeto de monitorización. Este programa centraliza toda esa información y permite el tratamiento o gestor de dicha información.

Ahora se verá cómo se realiza la conexión de los sensores a la nube. Para ello usaremos la plataforma de Google llamada Firebase. Lo primero que se debe hacer es realizar los siguientes pasos:

- Ir a la página oficial de Firebase de Google https://firebase.google.com/
- Una vez dentro de la página se realiza el login con una cuenta de Google para así crear un nuevo proyecto, que nos permitirá almacenar información de nuestro sistema.

Una vez que se realizan los pasos anteriores se crea el proyecto en el portal de Firebase como se ve en la figura 19. Después de agregar proyecto aparecerá otra

ventana que da la opción de ingresar el nombre del proyecto y de seleccionar el modo de análisis de la base de datos. Es posible elegir el modo de análisis gratuito.

Figura 19: Muestra para crear el Hub en Azure

Para recabar la información de los sensores se utilizará RealTimeDataBase que se utiliza para recabar información en tiempo real de los sensores y para ello se deben configurar reglas en la base de datos como se ve en la figura 20. Esta configuración permite a la base de datos la entrada de información mediante librerías en Arduino u otras plataformas.

Figura 20: Configuración de reglas en Firebase
Fuente: Creación propia

Capítulo 4 Diseño y Resultados

4.1 Diseño conceptual del sistema

El sistema consta de tres bloques o módulos: 1) el módulo de sensores inalámbricos o alámbricos con los que se detecta la presencia de un vehículo en los espacios de estacionamiento; 2) el de control de sensores, con el que se tiene se tiene el control mediante un microcontrolador, en este caso se utiliza un Arduino Mega 2560, el cual recopila la información recabada por los sensores; 3) el módulo de recepción y gestión de datos para ser transmitidos a una plataforma en la nube donde se visualizará la información de manera remota.

Como parte del sistema se tiene una página web que proporcionará la visualización de los sensores de manera gráfica para que los usuarios puedan ver la información sobre la disponibilidad de los espacios de estacionamiento. En la figura 21 se puede ver el diagrama de los módulos del sistema.

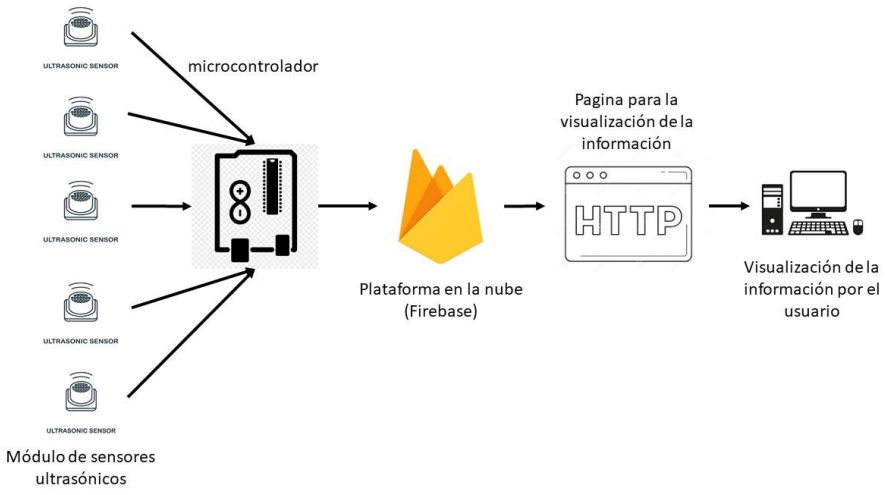

Figura 21: Diagrama de módulos del sistema
Fuente: Creación propia

En el diagrama anterior se pueden observar los módulos de sensores, en este caso sensores ultrasónicos inalámbricos conectados a través de los pines del

microcontrolador, para recopilar la información y almacenarla temporalmente en la memoria interna del microcontrolador. Una vez que se completa el proceso de almacenamiento de la información de los sensores, el microcontrolador la envía a una plataforma en la nube para que esta información se almacene en una base de datos y pueda ser procesada para posteriormente sea visualizada en linea por los usuarios del sistema. Una vez que la información es recopilada en tiempo real por la plataforma, se puede ver la información recopilada por los sensores, con lo que será posible visualizar si un lugar de estacionamiento se encuentra disponible u ocupado. Para que esta información sea más fácil de ver e interpretar se envía a una página WEB mediante la cual y utilizando un dispositivo móvil o una laptop los usuarios puedan visualizar en tiempo real la disponibilidad de lugares en un estacionamiento en particular.

4.1.1 Diseño del módulo de sensor ultrasónico inalámbrico

Para elaborar el diagrama del diseño electrónico del módulo inalámbrico de los sensores y su conexión con el microcontrolador se utilizó el software de diseño electrónico *Fritzing*, este ayuda a ver de manera gráfica la conexión de los componentes y también permite realizar pruebas de funcionalidad como se ve en la figura 22. En esta figura se puede observar que el sensor tiene dos pines digitales *trigger* (pin de disparo) y *echo* (pin de eco).

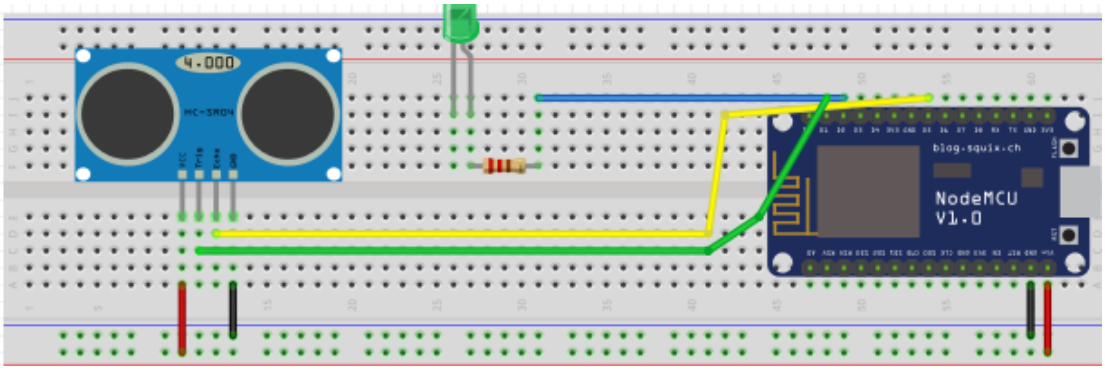

Figura 22: Conexión del sensor con microcontrolador

El pin *trigger* recibe un pulso de habilitación del microcontrolador, mediante el cual se le indica al módulo que inicie la medición de distancia mientras que en el pin echo el sensor devuelve al microcontrolador un pulso cuyo ancho es proporcional al tiempo que tarda el sonido en viajar del transductor al obstáculo y luego de regreso al módulo. La señal del pulso recibido en el pin *echo* se envía al microcontrolador, en este caso se utiliza una tarjeta *Nodemcu8266*, esta permite recibir y almacenar la información para enviarla posteriormente a la plataforma en la nube por medio del módulo WiFi de la misma tarjeta. En el caso mostrado en la figura 22 se utilizan los puertos *Rx* y *Tx* para hacer una interconexión entre microcontroladores utilizando los pines de Arduino en caso de requerirlos.

4.1.2 Interconexión de los sensores a la plataforma en la nube

Una vez que se tiene el esquema electrónico de conexión de los sensores y el microcontrolador se procede a realizar la conexión física de los componentes en una placa de PCB como se ve en la siguiente figura 23. En el caso de la tarjeta *Nodemcu8266*, esta se programó y configuró con la asignación de pines correspondientes para la interconexión, con lo que se evita tener alguna interferencia al enviar la información a la plataforma en la nube.

Figura 23: Conexión física de los sensores junto con el microcontrolador

Una vez que se realizaron las pruebas para comprobar el funcionamiento del dispositivo, este se coloca en una caja eléctrica para protegerla del entorno externo

y también tener un mejor manejo del dispositivo, como puede verse en las figuras 24 y 25. Esto ayudará a una mejor instalación de los sensores, así como una mejor mantenibilidad de los dispositivos. Con este diseño del módulo de sensores se tiene también una independencia de los mismos, con lo que se logra mayor fiabilidad ya que si por alguna razón uno de los módulos de sensor llegara a fallar, los otros pueden seguir trabajando sin ningún problema o afectación a la detección de los otros espacios de estacionamiento.

También permite una mayor flexibilidad, ya que estos módulos inalámbricos pueden ubicarse o reubicarse sin necesidad de un nuevo cableado para agregar nuevos espacios de estacionamiento (en caso de crecimiento o reacomodo del estacionamiento).

Figura 24: Sensor ultrasónico instalado en caja de seguridad
Fuente(Creación propia)

Figura 25: Conexión dentro de la caja de seguridad
Fuente(Creación propia)

Una vez instalados correctamente cada uno de los sensores inalámbricos en su caja, se verifica la conexión con la plataforma en la nube. Se comprueba el acceso correcto a la base de datos instalada en la plataforma en la nube y así teniendo un sensor ultrasónico inalámbrico funcional que nos ayude en la detección de lugares de estacionamiento disponibles en zonas complicadas de alcanzar.

4.2 Diseño de la interconexión de los sensores

Para crear el diseño de la arquitectura de interconexión de los sensores, primero se realizó una revisión sistemática de literatura, esto con el fin de investigar y analizar cuál es la tecnología más utilizada en la implementación de estacionamientos inteligentes. Para este fin se revisaron 43 artículos científicos, se encontró que la tecnología más utilizada es la de sensores ultrasónicos, en la tabla 5 se puede observar el resultado obtenido de este análisis.

Los sensores ultrasónicos ofrecen una serie de ventajas como son: su fácil acceso en diferentes tiendas, económicamente estos sensores son más baratos que los demás. Al ser muy utilizados estos sensores tienen una mayor tolerancia ante las contingencias ambientales y poseen una alta variedad de microcontroladores que

pueden acoplarse según las necesidades. También existen diferentes librerías que permiten la conexión de estos sensores con un microcontrolador.

En el caso de este trabajo el microcontrolador que recibe la señal de los sensores es un Arduino, este permite visualizar la información que llega de los sensores en un monitor serial en la misma IDE del dispositivo.

Tabla 5: Revisión sistemática de literatura de las tecnologías de sensores

Tecnología utilizada	Cantidad de veces utilizada
Ultrasónicos	19
RFID	9
Cámaras	5
Infrarrojos	12

Para la interconexión de los sensores se definen los pines *Trig* y *Echo* que envían y reciben las señales de control que van al microcontrolador y se encargan de enviar y recibir un pulso ultrasónico, así el microcontrolador calcula el tiempo que tarda el pulso en ir y regresar al sensor. El siguiente código en Arduino® es el utilizado para calcular la distancia a un objeto.

```
#define PIN_TRIG 7
#define PIN_ECO 6
#define PIN_LED 3

void setup() {
  // Inicializacion de la comunicacion serial
  Serial.begin (9600);
```

```
  // Inicializacion de pines digitales
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECO, INPUT);
  pinMode(PIN_LED,OUTPUT);
}

void loop() {
  long duracion, distancia;  // Variables

  /* Hacer el disparo */
  digitalWrite(PIN_TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(PIN_TRIG, HIGH);
  delayMicroseconds(10);        // Duracion del pulso
  digitalWrite(PIN_TRIG, LOW);

  /* Recepcion del eco de respuesta */
  duracion = pulseIn(PIN_ECO, HIGH);

  /* Calculo de la distancia efectiva */
  distancia = (duracion/2) / 29;

/* Imprimir resultados a la terminal serial */
  if (distancia >= 500 || distancia <= 0){
    Serial.println("Fuera de rango");
  }
  else {
    Serial.print(distancia);
    Serial.println(" cm");
  }

  if (distancia < 10){
    digitalWrite(PIN_LED,HIGH);

    }else{
      digitalWrite(PIN_LED,LOW);
```

```
      }

  // Retardo para disminuir la frecuencia de las lecturas
  delay(500);
}
```

Una vez que se hicieron pruebas para verificar que el microcontrolador recibiera la señal de un sensor se implementó el código para 10 sensores donde la respuesta de los sensores fue satisfactoria ya que el microcontrolador logro recibir la señal de los sensores y mostrarla en el monitor serial como se ve en la figura 26.

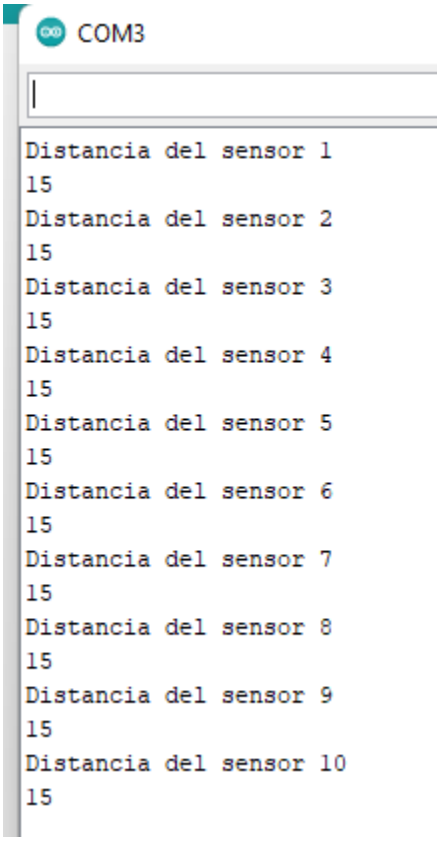

Figura 26: Muestreo de la distancia medida con Arduino
Fuente (Creación propia)

Una vez probados los sensores se procede a checar la conectividad de los sensores con la plataforma Firebase, para ello lo primero fue realizar pruebas utilizando un software de diseño electrónico (*Fritzing*) donde se hace la conexión de un sensor ultrasónico con el microcontrolador Node como se puede ver en la figura 27 y así checar que las librerías ocupadas para la conexión entre ambas plataformas sean correctas.

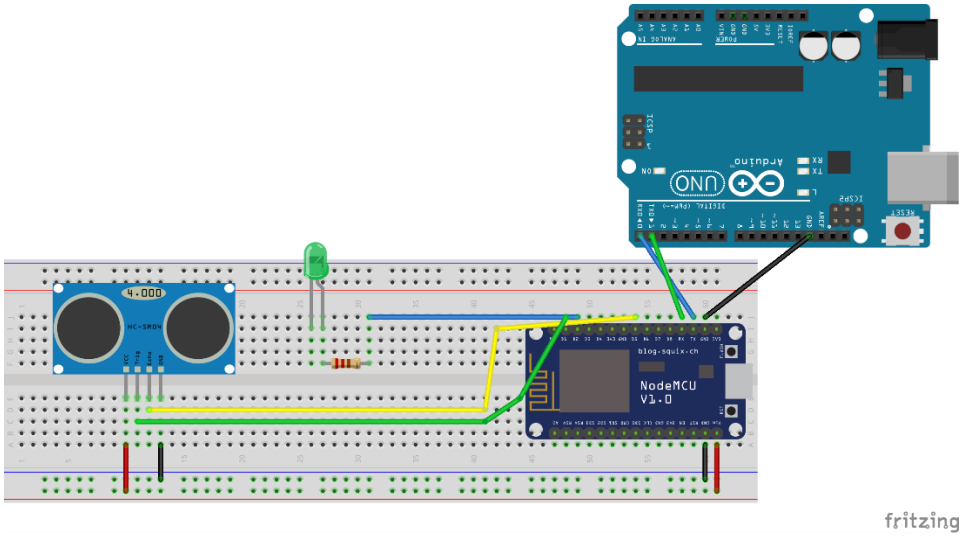

Figura 27: Diagrama de conexión de sensor y microcontrolador Node
Fuente (Creación propia)

Para corroborar que la conexión a la nube sea correcta se utiliza el monitor serial del IDE de Arduino, ya que permite obtener información en tiempo real sobre la transferencia de datos al microcontrolador.

4.3 Conexión a la nube

Para recibir información de los sensores y almacenarlos en una plataforma en la nube, se seleccionó *Google Firebase* ya que esta nos permite un mayor control de la información a diferencia de otras plataformas, por ejemplo, las características principales de Firebase son:

- **Desarrollo**: Firebase nos permite crear apps mejorando el tiempo de optimización y desarrollo, mediante diferentes funciones una de ellas es que nos permite detectar errores gracias a el sistema de testeo y error de Google. También nos permite configurar nuestra app o página web de manera remota y actualizarla al instante. Por lo que en la parte de desarrollo Firebase permite un manejo eficaz del sistema.

- **Analítica**: En este módulo de Firebase nos permite tener un control máximo del rendimiento de nuestra aplicación o página web, todo esto desde una consola de administración o panel llamado Google analytics (este es un servicio gratuito). Estos datos facilitan la toma de decisiones y están basados en los datos reales recabados por el sistema como se puede ver en la figura 28.

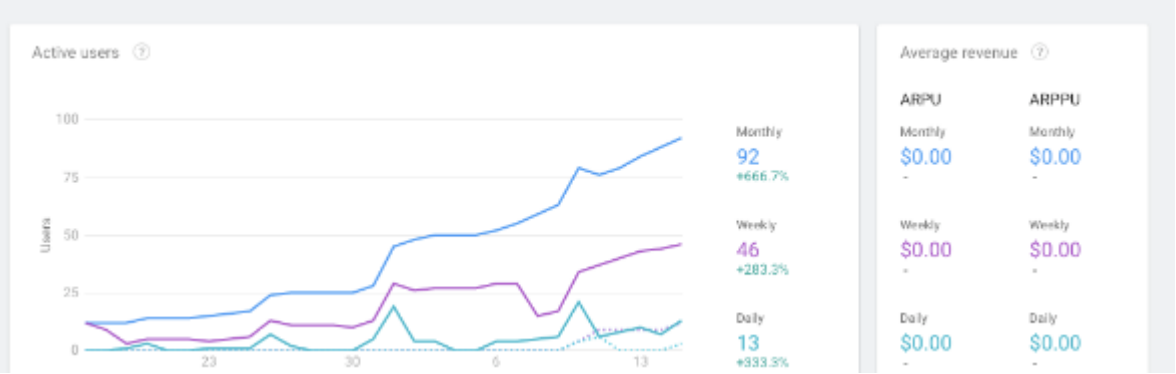

Figura 28: Datos analizados por Google analytics

- **Capacidad de crecimiento**: Esta característica de Firebase permite gestionar de manera más sencilla las aplicaciones web y páginas web. Ya que permite el ingreso de diferentes usuarios, ya sea a través de invitaciones o notificaciones.
- **Monetización**: Firebase permite monetizar la aplicación mediante AdMob, esta funciona de diferentes maneras: una de ellas es mediante *banners* o anuncios de videos, estos permiten el ingreso de publicidad pagada ya sea

a la aplicación o página web. La ventana de esta implementación es que Google no cobra algún porcentaje sobre el ingreso obtenido por la aplicación.

- **Agilidad**: Firebase ofrece desarrollo y gestión de Apps multiplataforma ya que dispone de API's integradas a SDK tanto para JavaScript y para diferentes sistemas operativos.
- **Multiplataforma:** es soportado por diferentes plataformas: IOS, Android y WEB

4.3.1 Planes de precio de Firebase

Firebase es una plataforma con diferentes planes y productos de uso gratuito, y planes de prepago, el plan gratuito es Spark, este plan incluye 1 GB de almacenamiento de datos, 20,000 escrituras por día, 50,000 lecturas por día y 20,000 eliminaciones por día. El plan de prepago es conocido como Blaze; esta versión utiliza el modelo de precios de pago de acuerdo a los servicios y productos utilizados (crecimiento a la medida).

Los costos de uso del servidor son de $0.18 dólares por GB y para el almacenamiento en la base de datos es de $0.026 por Gb. Este tipo de forma de pago utiliza también las cotizaciones en base al tipo de operaciones que el sistema realice. Los productos que se tienen de manera gratuita son:

- Análisis
- Mensajería proporcionada por la plataforma
- Entorno de pruebas
- Supervisión del rendimiento de nuestro sistema
- Enlaces dinámicos
- Configuración remota

En la tabla 6 se muestra una comparación de los planes que proporciona firebase.

Tabla 6: Comparación de planes

Base de datos en tiempo real	**Spark**	**Blaze**
Conexiones simultaneas	100	200k/base de datos
GB de almacenamiento	1	5
GB descargados	10 GB	100 GB
Cantidad de bases de datos por proyecto	1	Múltiples

Tabla 7: Comparación de planes en el área de almacenamiento

Almacenamiento	**Spark**	**Blaze**
GB almacenados	5 GB	0.026$/Gb
GB descargados	1 GB/día	0.12$/GB
Operaciones de carga	20k/día	0.05$/10k
Operaciones de descarga	50k/día	0.004$/10k
Cubos por proyecto	1	Múltiples

Tabla 8: almacenamiento y recepción en la nube

Cloud Firestore	**Spark**	**Blaze**
Gb almacenados (base de datos)	1 GB total por día	0.18$/GB

Escrituras	20k/día	$0.18/100k operaciones
Lecturas	50k/día	$0.06/100k operaciones
Eliminaciones	20k/día	$0.02/100k operaciones

Una vez analizados los planes de pago, nos damos cuenta que en para el desarrollo de nuestro sistema no se necesita el plan de pago ya que al estar conectados solo 10 sensores no se alcanza el límite máximo de uso de ninguno de los datos mostrados en las tablas 7 y 8.

4.3.2 Configuración de la plataforma Firebase

Una de las razones por la cual se seleccionó Firebase es por la seguridad que proporciona, ya que se requieren varios permisos para que algún otro usuario logre acceder al sistema. Por ello aun cuando el microcontrolador disponga de las librerías adecuadas para realizar la conexión la plataforma, si no se configura con las reglas adecuadas no será posible acceder. Para ello se configurarán las reglas dentro de Firebase y configuraremos las principales reglas de lectura y escritura como *true*. Esto permitirá al sistema un acceso mediante una dirección *url* proporcionada por la plataforma, de igual manera requeriremos una contraseña que será proporcionada por la plataforma. En la figura 29 se puede ver la configuración de las reglas que nos permiten acceder la plataforma.

Figura 29: reglas de entrada a Firebase

Una vez configuradas las reglas se procede a conseguir los accesos a la base de datos. Estos requerimientos son la *url* y la contraseña, para ello se realizan los siguientes pasos.

Para obtener la *url* accederemos a la base de datos en tiempo real (real time data base) en ella se encuentra la dirección *url* como se ve en la figura 30.

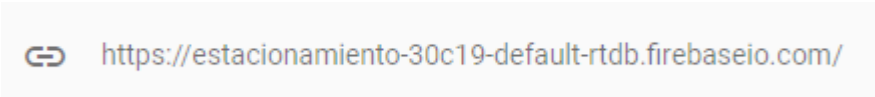

Figura 30: url de nuestro proyecto

Después de obtener la *url* se copia la contraseña que se obtiene desde la configuración general del proyecto, esta se encuentra oculta en una sección de la plataforma llamada *cuentas de servicio* y en el apartado de secretos de la base de datos encontraremos la contraseña necesaria para poder acceder a la base de datos por medio de otra plataforma. En este caso se utiliza esa contraseña para acceder por medio de nuestro microcontrolador. La contraseña será mostrada como se ve en la figura 31.

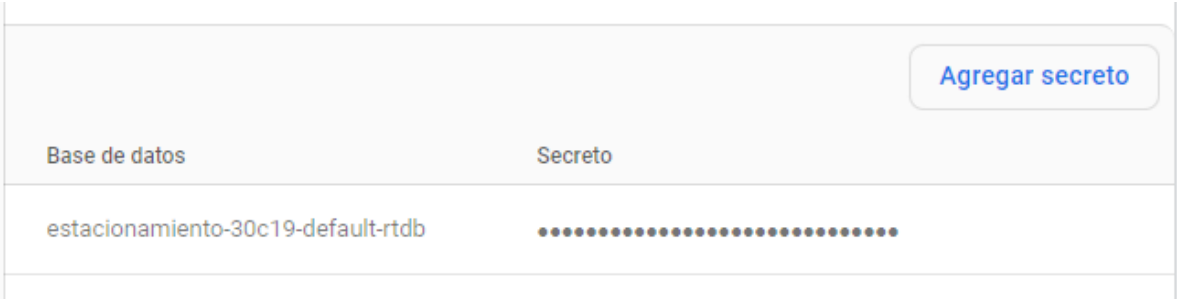

Figura 31: Contraseña de ingreso a nuestra base de datos

4.3.3 Conexión de microcontrolador con la plataforma en la nube

Para enviar información a la nube y almacenarla de manera correcta en la plataforma es necesario ingresar la dirección *url* y la contraseña antes mencionada. Pero antes de eso necesitamos instalar algunas librerías desde la IDE de Arduino ya que no se carga la información al Arduino, si no que la información se envía al

microcontrolador de la tarjeta *Node MCU*. Para ello se descarga desde el gestor de librerías de Arduino la librería *ESP8266* mostrada en la figura 32.

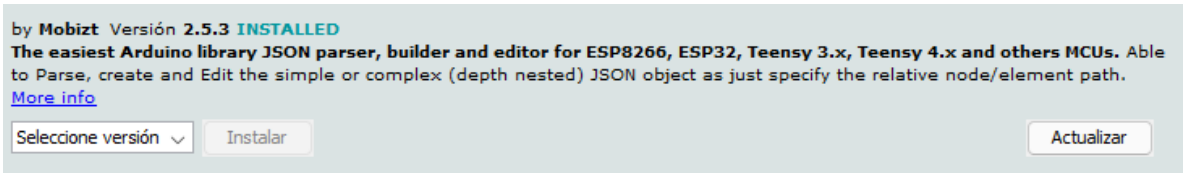

Figura 32: Librería requerida para trabajar con Esp 8266

Una vez que tenemos instalada la librería, es necesario de igual manera descargar una librería que le permitirá al microcontrolador conectarse a una red de internet inalámbricamente. Para ello se utiliza nuevamente el gestor de librerías para instalar la la librería ESP8266WiFi.h como se ve en la figura 33.

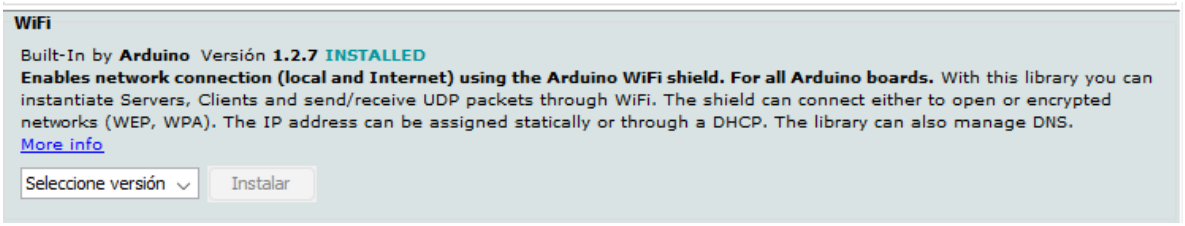

Figura 33: Librería para módulos wifi

Por último, es necesario instalar la librería de Firebase-ESP8266 que permitirá colocar la *url* y la contraseña para el ingreso a la plataforma, esta al igual que las demás la podemos obtener a través del gestor de librerías de Arduino como se ve en la figura 34.

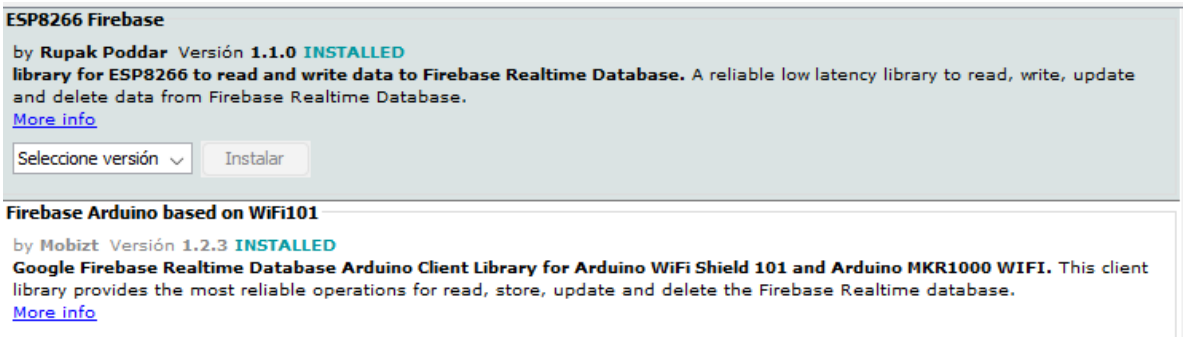

Figura 34: Librerías de Firebase para Esp 8266

Una vez que se tienen instaladas las librerías, estas pueden utilizarse desde el IDE de Arduino. En el siguiente código se puede observar que el *WIFI_SSID* se utiliza para colocar el nombre del modem o router al que se va a acceder. *WIFI_PASSWORD* permite ingresar la contraseña del modem para así acceder a internet. Una vez que el microcontrolador accedió a Internet necesitamos que ahora se conecte a la dirección *url* de la plataforma Firebase, para ello utilizamos *FIREBASE_HOST* en este podemos ingresar la url obtenida de Firebase y así direccionar el microcontrolador al sistema gestor de base de datos. Una vez ubicada la url debemos asignar la contraseña de la plataforma previamente obtenida por la opción *secretos de la base de datos*. Para ingresar la contraseña utilizaremos *FIREBASE_AUTH*. Esto nos permitirá tener una conexión exitosa a la base de datos.

```
#define WIFI_SSID "INFINITUM6977_2.4"
#define WIFI_PASSWORD "c5t1NumP0u"
#define FIREBASE_HOST "estacionamiento-30c19-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "z3zwdSwweAEikD97m6pcXv6xSdQWH3zcoRXItf9N"
```

Para corroborar que la conexión fue exitosa debemos ir a la base de datos en tiempo real y ahí se deberá mostrar la información que está enviando el microcontrolador a la nube como se ve en la figura 35.

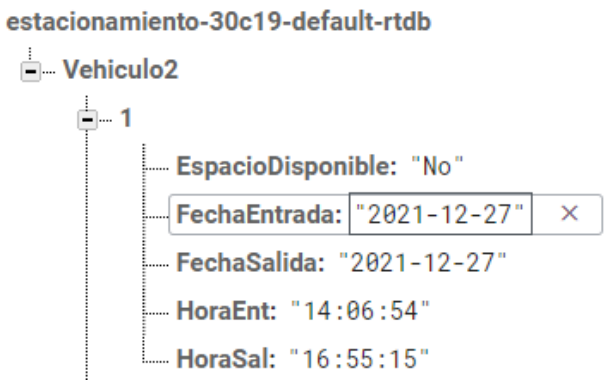

Figura 35: Vista del registro del sensor a la plataforma

4.3.4 Registro de super usuarios

Una vez que se tiene la configuración correcta para la plataforma en la nube y se ha verificado el acceso del microcontrolador a la nube se debe definir la autentificación del sistema. Existen 3 tipos de perfil de usuario para la autentificación: propietario, editor y visualizador. Cada uno contiene características específicas que permitirán a los usuarios tener un acceso a la aplicación como se ve en la figura 36.

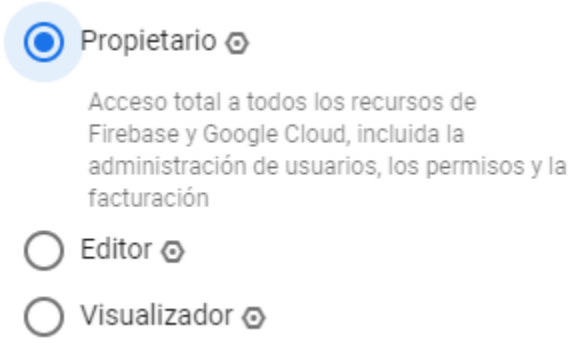

Figura 36: Tipos de registro en nuestra plataforma

En primer lugar, se tiene el perfil de propietario; este tipo de perfil concede al usuario un acceso total a toda la infraestructura del proyecto o aplicación en Firebase tanto

a los dados recopilados por los sensores como a la información de los usuarios, permite modificar la información recopilada, permite cancelar o conceder permisos dentro del sistema, así como facturación.

El perfil de usuario editor, le proporciona al usuario acceso y permisos para edición a todos los recursos de *Firebase* y *Google Cloud*. Por último, el perfil de visualizador permite al usuario acceso de solo lectura de los recursos de *Firebase* y *Google Cloud*.

En este proyecto se tienen dos tipos de usuarios. El usuario de propietario, quien se encarga de administrar toda la información recopilada en la base de datos para enviarla a una página web, donde los usuarios de un estacionamiento puedan visualizar en tiempo real la información de los sensores y de los lugares disponibles. Por otro lado, tenemos la parte de visualizador, este se encarga de ver que la información de los sensores sea correcta y no se pierda en ningún momento la visualización de los lugares de estacionamiento. En la figura 37 podemos ver a los usuarios que están dados de alta en la plataforma.

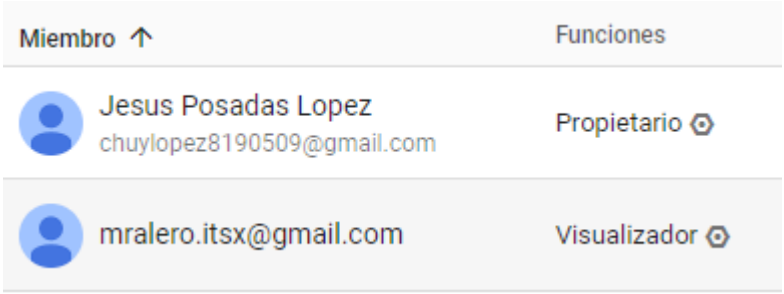

Figura 37: Usuarios registrados en nuestra plataforma

Para registrar a un usuario en la plataforma se puede hacer mediante un correo electrónico proporcionado por la persona que se quiere registrar, en la siguiente

imagen (figura 38) se puede observar cómo se agrega un nuevo usuario y también cuales son los datos requeridos por la plataforma.

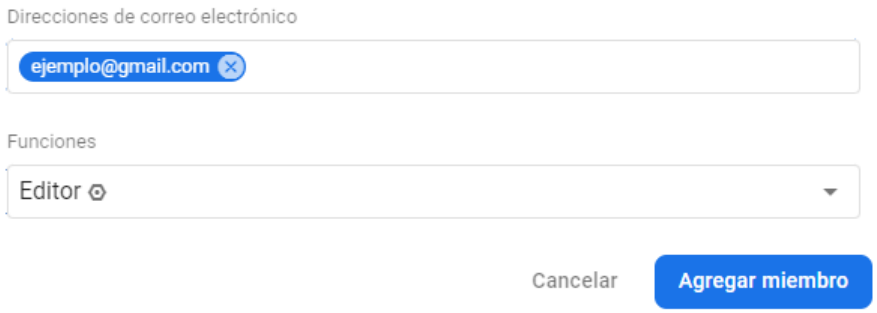

Figura 38: Ingreso de nuevo usuario a nuestra plataforma en la nube

4.4 Visualización de espacios disponibles por los usuarios

Para que los usuarios puedan visualizar en tiempo real los espacios disponibles del estacionamiento se implementó una página web, diseñada en HTML 5 para acceder a la información en tiempo real. Para ello se necesitó transferir la información recibida por nuestros sensores a nuestra página web. En esta sección explicaremos como se logró la conexión entre nuestra página web y nuestra plataforma en la nube.

Para crear la página web utilizamos HTML 5 y las librerías de *Bootstrap* para poder acceder a los diseños que la librería nos concede de manera gratuita. El diseño abarca 4 tipos de información: lugar, espacio disponible, hora de entrada y fecha de entrada del vehículo en cuestión. Para acceder a esa información que es recopilada por los sensores y almacenada en la base de datos de Firebase es necesario una comunicación a Firebase vía la implementación de un código, Firebase proporciona de forma gratuita un código que funciona como librería y conexión a la base de datos.

Existen diferentes maneras de conectarse a la información que tenemos almacenada en la plataforma. En este proyecto se utilizó *npm* (Node Package Manager), este sirve para compartir herramientas a través de JavaScript o HTML este ayuda a instalar varios módulos y administrar dependencias vía código.

En caso de algunos sistemas operativos podemos obtener el *npm* a través de la siguiente línea de comando (ver figura 39):

```
$ npm install firebase
```

Figura 39: Línea de comando de npm

Otra forma de conectarse a la plataforma en la nube es a través de la línea de código que proporciona *Firebase*, esta contiene la función de *Google Analytics* que permite gestionar la información recibida y almacenada de la base de datos. También proporciona un *API Key*, la cual es la contraseña de acceso de HTML a *Firebase*. La plataforma en la nube proporciona el nombre de un dominio a través del código y nos permite darlo de alta para que los usuarios puedan ver la información en línea.

En la figura 40 se puede ver el código proporcionado por Firebase para la integración de la base de datos a nuestro código.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyBhaX6meWMjeZqT3sEVJ1DK4noZ-Vh0p6I",
  authDomain: "estacionamiento-30c19.firebaseapp.com",
  databaseURL: "https://estacionamiento-30c19-default-rtdb.firebasei
  projectId: "estacionamiento-30c19",
  storageBucket: "estacionamiento-30c19.appspot.com",
  messagingSenderId: "824622742300",
  appId: "1:824622742300:web:b4c69cfac8fea38d5798d9",
  measurementId: "G-BW7TLH7BC2"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Figura 40: Código para integración con Firebase

4.4.1 Visualización del número de espacios disponibles

Para visualizar la cantidad de espacios disponibles que se tienen en un estacionamiento se utiliza el siguiente link http://127.0.0.1:5500/Tabla.html este nos dirigirá a una tabla que nos indica los lugares que se tienen disponibles y cuáles son los lugares ocupados. La figura 41 muestra el diseño de la interfaz para la visualización de los espacios disponibles.

Lugar	Disponible	Fecha de entrada	Fecha de salida
1	No	2021-12-27	2021-12-27
2	No	2021-12-27	2021-12-27
3	Si	----	----
4	No	2021-12-27	----

Figura 41: Visualización de espacios disponibles
Fuente: Creación propia

La figura 41 muestra una tabla que indica la disponibilidad de los espacios, en primer lugar, se tiene el número de los lugares activos del estacionamiento. Cada uno de ellos está monitoreado por un sensor que envía la información sobre la disponibilidad a la base de datos. Después se tiene la sección de espacio disponible que nos indica si el lugar actualmente se encuentra vacío u ocupado. Por último, se tiene la fecha de entrada del vehículo que proporciona al dueño del estacionamiento un conteo del número de vehículos que ingresaron en cierto día al estacionamiento.

4.5 Pruebas de eficiencia

Para verificar que la red de sensores trabaja de manera eficiente, se dejan los sensores conectados a una batería de 15 volts @ 2550 mAH para registrar el tiempo que los sensores pueden trabajar con la batería. Se obtuvo como resultado un funcionamiento útil de entre 7 a 8 horas, esto gracias al modo de *Deep sleep*, este modo consigue que el microcontrolador consuma menos corriente dando así más vida útil al dispositivo.

En estas 8 horas de trabajo se registró una eficiencia del sistema de 99.2% y una taza de error del 0.8%. Para calcular la eficiencia del sistema se utilizó la ecuación

de porcentaje de error, la cual determina que tan eficiente es el sistema de acuerdo a los datos registrados en base de datos. Para ello se toman 100 datos de registro de la medición de la distancia mandada por los sensores como se puede ver en la tabla 8.

Tabla 9: Mediciones para el cálculo de la eficiencia

Numero de medicion	Cm medidos	Numero de medicion	Cm medidos	Numero de medicion	Cm medidos	Numero de medicion	Cm medidos	Numero de medicion	Cm medidos	Numero de medicion	Cm medidos
1	12.35	18	12.35	35	12.35	52	12.35	69	12.35	86	12.35
2	12.35	19	12.35	36	14.13	53	12.35	70	11.01	87	12.39
3	12.35	20	12.35	37	12.35	54	13.12	71	12.35	88	12.35
4	12.35	21	12.12	38	12.35	55	12.35	72	12.35	89	12.35
5	12.35	22	12.35	39	12.35	56	12.35	73	12.35	90	12.4
6	12.35	23	12.35	40	12.35	57	12.35	74	11.55	91	12.35
7	12.35	24	12.35	41	12.45	58	12.35	75	12.35	92	12.35
8	12.57	25	12.35	42	12.35	59	12.45	76	12.35	93	12.35
9	12.35	26	12.35	43	12.35	60	12.35	77	12.41	94	12.35
10	12.35	27	13	44	12.35	61	12.35	78	12.35	95	12.43
11	12.35	28	12.35	45	12.35	62	12.35	79	12.35	96	12.35
12	12.35	29	12.35	46	12.36	63	12.44	80	12.35	97	12.35
13	12.75	30	12.35	47	12.35	64	12.35	81	12.49	98	12.35
14	12.35	31	12.35	48	12.35	65	12.23	82	12.35	99	12.55
15	12.35	32	12.35	49	12.35	66	12.35	83	12.35	100	12.35
16	12.35	33	12.01	50	12.35	67	12.36	84	12.38		
17	12.97	34	12.35	51	12.35	68	12.35	85	12.35		

La ecuación de el porcentaje de error se calcula bajo la siguiente formula:

$$\%E = \frac{|valor\ teórico - valor\ experimental|}{|valor\ teórico|} \times 100 \qquad Ec.1$$

Ecuación 1: Formula para el cálculo del porcentaje de error

Se tiene en cuenta el valor teórico es la moda de nuestras mediciones tomadas en la anterior tabla. En nuestro caso la moda de nuestras mediciones es 12.35, siendo este el valor teórico en nuestra ecuación de porcentaje de error.

Teniendo en cuenta que el valor experimental es la media aritmética de los datos registrados, da como resultado 12.25, para hallar la media se utiliza la ecuación 2.

$$Media\ aritmética = \frac{\sum_1^N x_i}{N} \qquad Ec.2$$

Ecuación 2: Media aritmética

Una vez obtenidos estos datos se calcula el error, dando como resultado:

$$\%\boldsymbol{E} = \frac{|\mathbf{12.35\%} - \mathbf{12.25\%}|}{\mathbf{12.35\%}} \times \mathbf{100}$$

$$\%\boldsymbol{E} = \mathbf{0.8097\%}$$

Con estos resultados podemos observar que la eficiencia del sistema es óptima y también asegura que el sistema cumpla con las necesidades y requerimientos de los dueños de estacionamiento para un monitoreo eficiente de espacios disponibles. También por la parte de los usuarios podemos observar que, dado que nuestro sistema detecta de manera correcta los espacios disponibles, ellos pueden obtener información en tiempo real sobre los lugares vacantes del estacionamiento y así ahorrar; tiempo, gasolina y reduce el nivel de estrés que se puede ocasionar muchas veces al no encontrar un lugar disponible donde estacionarse.

4.6 Pruebas de fiabilidad

Para medir la fiabilidad del sistema desarrollado en este proyecto se requiere de dos variables que son: el suministro eléctrico que se le proporciona a los sensores y la probabilidad de que un componente del sistema pueda fallar. Por ello se estiman los periodos de tiempo bajo condiciones definidas. Para este caso la disponibilidad se basa como el porcentaje de tiempo en el cual nuestro sistema puede trabajar sin interrupciones. Este es uno de los aspectos más básicos de la confiabilidad y puede ser medido por porcentaje o unidad y se puede evaluar bajo la siguiente formula.

$$Disponibilidad = \frac{Tiempo\ en\ servicio}{Tiempo\ en\ servicio + Tiempo\ fuera\ de\ servicio} \qquad Ec.3$$

Ecuación 3: Ecuación de disponibilidad

Los índices que se utilizan para evaluar la fiabilidad de un sistema están basados en interrupciones y la disponibilidad que tiene el sistema trata con cálculos determinar cuándo tendremos un estado de interrupción. Por ello existen diferentes índices que son de tipo determinista y estos reflejan el comportamiento mediante la continuidad de trabajo que posee un sistema.

La desventaja de esto es que no se consideran los valores aleatorios contenidos en algunos sistemas eléctricos, es decir no pueden determinar que un sensor o aparato eléctrico vaya a fallar por una descarga eléctrica o un fallo de energía provocado por una contingencia ambiental. Por ello las fórmulas o procesos para calcular la fiabilidad de un sistema son muy intuitivos y simples en cuanto a calculo y por ello requieren pocos datos para ser aceptados a diferencia de otras variables como lo puede ser la mantenibilidad, eficiencia, etc.

Por otro lado, se tienen los tipos de evaluación probabilista, estos consideran la aleatoriedad inherente en las operaciones de nuestro sistema y puede estar dividido en varios grupos como es, variaciones eléctricas, fallos del sistema o daños por usuarios.

4.6.1 Simulación de Montecarlo

Existen dos diferentes procesos de evaluación de fiabilidad, estos son: la simulación y la analítica. En el caso de las técnicas analíticas, estas están representadas con modelos analíticos y evalúan así los índices de fiabilidad del sistema usando soluciones matemáticas. Los métodos de simulación de Monte Carlo, estiman los

índices de fiabilidad y simulan el proceso actual y el comportamiento del sistema y tratamos al sistema como una serie de experimentos. Las ventajas de utilizar este método el cálculo de fiabilidad del sistema son las siguientes:

- Se pueden incluir todos los efectos del sistema o del proceso a evaluar.
- El número de muestras que se necesitan para un nivel de exactitud alto, es independiente del tamaño del sistema y este puede evolucionar con el tiempo o los ajustes que se le hagan a dicho sistema.
- Se pueden simular la probabilidad asociada con la falla de los componentes y las actividades de restauración lo cual en métodos analíticos no se puede.
- Se pueden simular factores del sistema de tipo no eléctrico, como pueden ser condiciones de operación de reserva de energía, cambios climáticos, etc.

Para este proyecto se realizó el análisis de la simulación de Montecarlo que nos indica que el porcentaje de fiabilidad que este sistema posee es de 96.6% proporcionando así un sistema con un gran porcentaje de fiabilidad. Para obtener el dato anterior se realizaron pruebas con los sensores durante 30 días, analizando así los fallos se encontraban en la detección de un vehículo. En la tabla 10 se observa el análisis realizado.

Tabla 10: Simulación de Montecarlo

Cantidad de fallos	Cantidad de dias	Frecuencia relativa	Frecuencia relativa acumulada	Intervalos		Numeros aleatorios	Posición de cada numero aleatorio
0	5	0.05208333	0.05208333	0	0.05208333	0.530119625	4
1	8	0.08333333	0.13541667	0.05208333	0.13541667	0.104753061	1
2	12	0.125	0.26041667	0.13541667	0.26041667	0.457605737	3
3	19	0.19791667	0.45833333	0.26041667	0.45833333	0.286535459	3
4	22	0.22916667	0.6875	0.45833333	0.6875	0.649238797	4
5	30	0.3125	1	0.6875	1	0.970792713	5

En la tabla anterior se puede observar varios puntos. El primero de ellos es la cantidad de fallos presentados conforme al número de días. Es decir, en 5 días no se presentó ningún error en la detección de un vehículo, a los 8 días se presentó un falso negativo donde nuestro sistema dejo de detectar la presencia de nuestro vehículo, a los 12 días se detectaron 2 errores, a los 19 días se detectaron 3 errores, a los 22 días se detectaron 4 fallas en el sistema y por último a los 30 días tuvimos un total de 5 fallas. Es decir, hay varios aspectos que pueden intervenir en la detección del sistema dado que pueden ser contingencias ambientales, mal uso del espacio de los sensores, afectaciones por polvo o agua a nuestro sistema, etc.

Para poder calcular nuestra fiabilidad se necesita obtener la frecuencia relativa y la frecuencia acumulada del sistema, Una vez obtenidas las frecuencias se generan los intervalos para tener un estimado de cuál sería la fiabilidad del sistema, si funcionará más días de lo que se probó, para ello, se generan 100 número aleatorios lo cual puede verse en la figura 42.

Numeros aleatorios				
	0.037415059	0.699282014	0.890333663	0.268399741
	0.26460113	0.403997761	0.028166439	0.204344165
	0.637254144	0.465067013	0.726885911	0.324809362
0.530119625	0.952931024	0.093118615	0.282309877	0.474959496
0.104753061	0.558677624	0.819570109	0.762745137	0.314953619
0.457605737	0.091599652	0.404115026	0.650037564	0.288138085
0.286535459	0.36269052	0.603184769	0.306127877	0.546206268
0.649238797	0.609945986	0.615252601	0.878405923	0.932096573
0.970792713	0.546982069	0.250648122	0.462030694	0.930403592
0.773096458	0.394035703	0.565337916	0.657932109	0.316628018
0.698780326	0.080098136	0.775367429	0.564751668	0.582854878
0.959458871	0.640625659	0.736663542	0.143663034	0.288032273
0.396689446	0.452763278	0.781796294	0.304092782	0.281136288
0.17143353	0.636288245	0.346382059	0.123390484	0.109837739
	0.766715739	0.341255883	0.62099967	0.137912562
	0.382767112	0.879007048	0.621771973	0.955845907
	0.260869079	0.571891477	0.770395195	0.50051518
0.438749447	0.870485466	0.346268074	0.730155283	
0.356691399	0.962640593	0.94736147	0.717036124	
0.716812161	0.417763088	0.167162214	0.340407887	
0.160989581	0.384264924	0.599840297	0.040493976	
0.143655057	0.794489943			

Figura 42: Números aleatorios generados para calcular el porcentaje de fallos

Una vez que se obtienen los números aleatorios cada uno de ellos se asignan en un ranking para observar, dependiendo al valor que probabilidad había de que los sensores presentaran fallos al momento de la detección. Dando como resultado un promedio de 3.50 que puede variar dependiendo los números aleatorios. De igual manera el rango de error se tiene es de 2.1 a 3.5, por lo tanto, el sistema maneja una fiabilidad de 96.6, lo que lo clasifica en un sistema con una alta efectividad y también proporciona confianza al momento de ser utilizado, ya que la tasa de falsos negativos o de detecciones incorrectas es muy baja.

4.7 Cálculos de mantenibilidad

Para poder abordar la mantenibilidad del sistema desarrollado se debe explicar de manera más concreta que es *mantenibilidad*. Esta es la característica que posee un elemento o sistema y se asocia a la capacidad que tiene un equipo de ser recuperado para realizar sus actividades o servicios, también se basan en diferentes reglas o normas conocidas como procedimientos o medio de instalación y prevención. Entonces para que un sistema pueda ser considerado mantenible o susceptible de ser mantenido deben cumplirse una serie de condiciones como lo podemos ver en la figura 43.

4.7.1 Cálculos de la mantenibilidad.

Para poder calcular la mantenibilidad de un sistema se necesitan diferentes variables, una de ellas es el tiempo con el cual el sistema se encuentra en reparación mayormente conocido como (TTR). Debido a que el TTR no es constante se utilizan diferentes medidas como lo son:

- Tiempo medio entre fallas (MTBF).
- Tiempo de recuperación (TTR)
- Tiempo medio de recuperación (MTTR)
- La función de la mantenibilidad
- La realización de la recuperación (t1, t2)

- Tiempo porcentual de recuperación (TTR%)

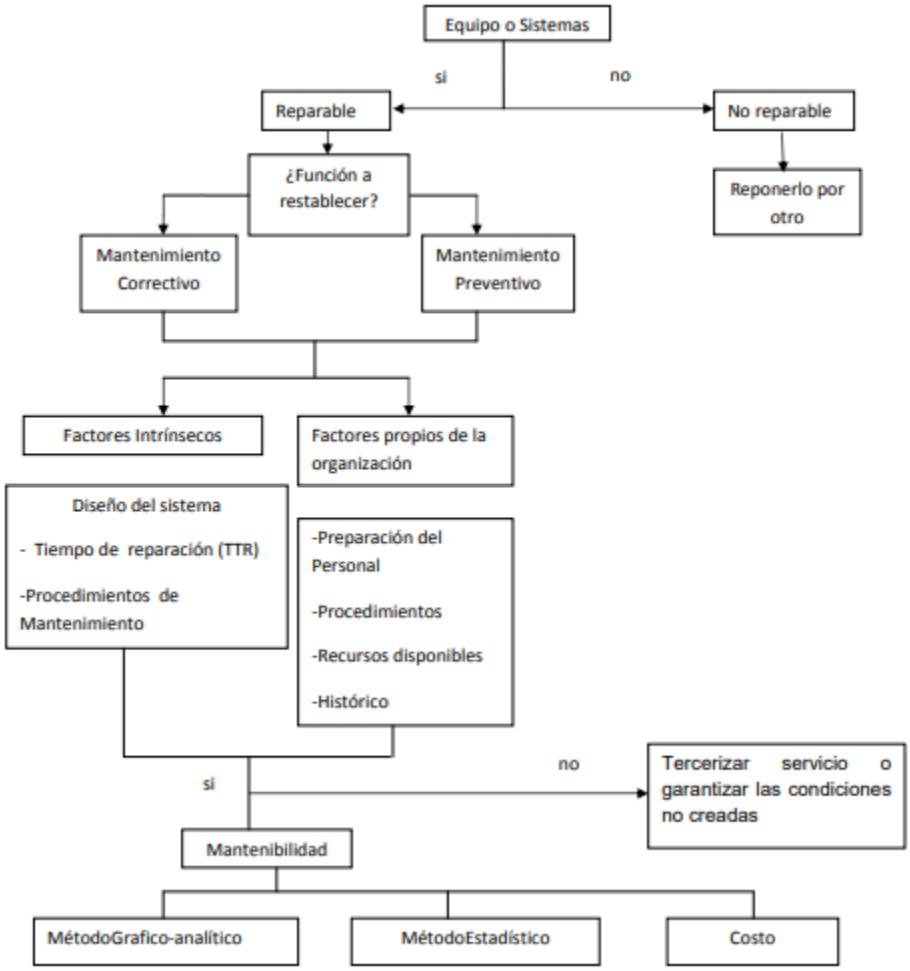

Figura 43: Condiciones para que un sistema sea mantenible

Las variables anteriores son consideradas aleatorias ya que solo pueden ser descritas de manera probabilística, es por ello que muchas veces para el cálculo de estas se utilizan herramientas estadísticas como pueden ser: función de densidad, función de distribución, tasa de fallos, etc.

Para la medición y el análisis de la mantenibilidad se desarrollaron diferentes etapas que componen la metodología del modelo universal CMD. Para esto se utilizó como base, los datos recopilados de los sensores y sus fallas desde el mes de noviembre de 2021 hasta el mes de mayo del 2022, la tabla 11 se muestran los tiempos operativos y de reparación o sustitución de los sensores ocurridos durante el periodo de evaluación del sistema.

Tabla 11: Tiempos de operación y reparación de los sensores

N.º de falla	Fecha	Tiempo de operación (H)	Tiempo de reparación (Min)
1	10/11/2021	216	62
2	14/11/2021	96	34
3	27/11/2021	312	49
4	5/12/2021	192	43
5	21/12/2021	384	39
6	19/01/2022	696	50
7	24/01/2022	120	33
8	11/2/2022	288	62
9	22/02/2022	264	26
10	9/3/2022	360	40
11	15/03/2022	144	27
12	17/03/2022	48	20
13	16/04/2022	720	38
14	25/04/2022	216	26
15	1/5/2022	168	50

Los datos presentados en la tabla anterior constituyen la base para poder determinar ß y α, mediante el método de mínimos cuadrados, esto con el objetivo de calcular los indicadores de mantenibilidad que posee nuestro sistema, mediante la distribución de Weibull.

4.7.1.1 Método de mínimos cuadrados

Las siguientes tablas muestran los tiempos de operación ordenados de menor a mayor, junto con la transformación doble logarítmica de la función de distribución acumulativa de los sensores. Mediante la transformación doble logarítmica se transformó la función de distribución acumulativa en una ecuación lineal de regresión, por los cuales se determinaron ß y α representados en la tabla 12.

Los sensores presentan valores de parámetro de forma ß de 1.495. Esto indica que se encuentran en el periodo de desgaste (etapa 3 de la curva de Davis). De acuerdo a la investigación realizada se puede fundamentar que los daños a los sensores son especialmente por cuestiones eléctricas o de calentamiento, ya que estos pueden producir un fallo al sistema (sensor) y provocar una falla en este.

Tabla 12: tiempos de operación y transformación del sistema

N.º de fallas	Tiempo de operación (H)	Rango de media RM(Ft)	Ln(t)	ln(ln(1/(1-Ft)))
1	48	0.06	3.87	-2.782632533
2	96	0.13	4.56	-1.971397744
3	120	0.16	4.78	-1.746671079
4	144	0.2	4.96	-1.499939987
5	168	0.23	5.12	-1.341838284
6	192	0.26	5.25	-1.20029593
7	216	0.3	5.37	-1.030930433
8	216	0.3	5.37	-1.030930433
9	264	0.36	5.57	-0.806792806
10	288	0.4	5.66	-0.671726992

N.º de fallas	Tiempo de operación (H)	Rango de media RM(Ft)	Ln(t)	ln(ln(1/(1-Ft)))
11	312	0.43	5.74	-0.576041853
12	360	0.5	5.88	-0.366512921
13	384	0.53	5.95	-0.281007617
14	696	0.96	6.54	1.169032176
15	720	0.99	6.57	1.527179626

En función de los datos obtenidos en la tabla 12, se presenta una ecuación lineal de regresión de los tiempos de operaciones de los sensores.

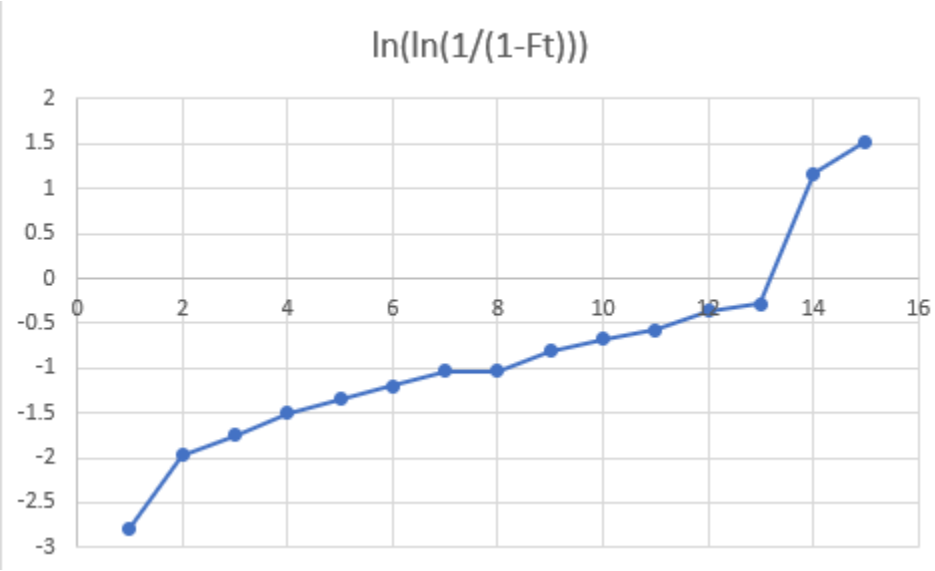

Figura 44: Ecuación lineal de los tiempos operativos de los sensores

Tabla 13: Resultados del cálculo de la mantenibilidad de nuestro sistema

Indicador de mantenibilidad calculado con t = 48 H		
Beta	TPPR(h)	Mantenibilidad
1.4595	30,598	78.39%

Con esto se puede asegurar que el sistema posee una mantenibilidad del 78.39% y garantiza que se pueda dar al sistema mantenimiento tanto preventivo como correctivo de manera eficiente.

Resultados finales.

Se probaron 10 sensores ultrasónicos en un ambiente controlado. Es decir, se conectaron 10 sensores al microcontrolador para medir la eficiencia, mantenibilidad y fiabilidad de nuestro sistema. En dichas pruebas se observó que la sensibilidad y el alcance de la detección de este sensor depende de la reflectividad acústica, el tamaño y la alineación del objeto que se trata de detectar, en este caso al ser objetos de un tamaño grande no se tuvo ningún problema en la detección de los automóviles. El modo más sencillo para detectar la presencia de un vehículo fue posicionando el sensor delante de los vehículos, en este tipo de medición la detección y la distancia del objeto se trasmiten tan pronto como el objeto este dentro del alcance del sensor.

En base a la Revisión Sistemática de Literatura se logró seleccionar el método más adecuado para los requistos del sistema para la detección de los vehículos, así como la elección más adecuada de los sensores. Una vez que se seleccionó la tecnología de detección, se procedió a determinar cuál sería el microcontrolador más adecuado para el cumplimiento de los requerimientos del proyecto desarrollado. En este punto se eligió el NodeMCU8266, que es una tarjeta de desarrollo de código abierto similar a Arduino, especialmente orientada al Internet de las cosas (IoT).

Incluye firmware que se ejecuta en el SoC Wi-Fi ESP8266 de Espressif Systems y hardware que se basa en el chip ESP8266EX, diseñado para cubrir las necesidades de un mundo conectado. Tiene 1 pin de entrada análoga y 13 pines de entrada/salidas digitales. Con esta tarjeta se recopiló la información y se envían los

datos a la nube. Una vez teniendo la información ordenada ya se puede enviar a la plataforma en la nube donde esta enviará la información a la página WEB para poder visualizarla de manera gráfica.

Con el desarrollo de este sistema basado en IoT, se puede confirmar que nuestro sistema posee una fiabilidad, eficiencia y mantenibilidad adecuada para ser utilizado en estacionamientos inteligentes, ya que se puede adaptar a cualquier tipo de entorno gracias a la movilidad de nuestros sensores ultrasónicos inalámbricos. Como resultado de esta investigación se puede concluir que el objetivo del trabajo fue cumplido. Gracias al desarrollo de este sistema se logró brindar información en tiempo real sobre la disponibilidad de los espacios vacíos de un estacionamiento.

Ya que hoy en día los estacionamientos existentes no pueden proporcionar ese tipo de información a los usuarios y los pocos estacionamientos considerados inteligentes no tienen la flexibilidad de instalación y la adaptación a los requerimientos y necesidades de los diferentes espacios o áreas dedicados a estacionamientos, tal como lo posee nuestro sistema gracias a la implementación del módulo de sensores inalámbricos y al reducido costo del mismo.

Conclusiones

Se puede, concluir gracias a esta investigación, que el conjunto de tecnologías como lo sensores ultrasónicos y las plataformas en la nube como *Firebase*, permiten lograr una fiabilidad, eficiencia y mantenibilidad correcta y funcional para la detección de espacios disponibles u ocupados en un estacionamiento a través del desarrollo de una aplicación que automatice este proceso de detección de disponibilidad de estacionamiento.

También se puede concluir que, gracias a la evaluación de las 5 tecnologías más utilizadas, analizando sus ventajas y desventajas, fue posible determinar cuál es la que más se adapta al sistema que se desarrolló en este trabajo de tesis. También, después del análisis de las tecnologías utilizadas en sistemas de estacionamientos inteligentes, fue posible determinar que un sistema basado en la el caso de tecnología de sensores magnéticos y sensores por radio frecuencia, estos elevan el costo de manera significativa, haciéndola poco redituable para los dueños de estacionamiento pequeños.

En el caso de los sensores fotoeléctricos, contienen un margen de error muy alto ya que su precisión se basa en la luz recibida. Por último, en el caso de las cámaras de videovigilancia su margen de error es amplio ya que se basa en el ángulo de visión recibida por la cámara. Gracias a esta investigación podemos afirmar que la tecnología con mayor adaptación son los sensores ultrasónicos. Estos nos permiten trabajar en diferentes entornos y temperaturas sin cambios drásticos a nuestras variables.

También podemos afirmar que esta tecnología es una de las de menor costo en el mercado y también es de las más utilizadas al momento de implementar un estacionamiento inteligente. Finalmente, una de las conclusiones más importantes es que la combinación de los sensores ultrasónicos con la plataforma en la nube *Firebase* permite monitorear y registrar los datos obtenidos por los sensores en tiempo real y también permite crear aplicaciones *WEB* para poder entregarle a los usuarios una forma de visualizar lo que los sensores registran por medio de una interfaz.

Trabajos futuros

El sistema podría mejorarse al incorporar mayores prestaciones en el análisis de información e incluir un mayor número de cajones de estacionamiento, lo que

implica que se tenga un mayor número de transacciones por día, por lo que sería necesario adquirir un plan de paga en la plataforma de *Firebase,* adicionalmente esto nos permitirá acceder a *Google Analytics* y así tener un mayor alcance para el análisis de la información que recopilen los sensores, que la obtenida por un plan gratuito de *Firebase*, como el que se utilizó en este proyecto. Ya que esta investigación fue realizada con la plataforma.

Otro aspecto importante que podría mejorar y crecer el sistema es contratar el servicio de un servidor en la nube, donde se podrá almacenar una mayor cantidad de datos recibidos de los sensores y con ello poder cubrir zonas de estacionamiento más grandes.

Al ser este un trabajo de IoT puede ser escalable en diferentes aspectos como en la creación de una aplicación móvil o se pueden crear diferentes apartados como pueden ser métodos de reserva para los espacios disponibles, lectores de placas por medio de cámaras de vigilancia para un mayor control de los vehículos que entran y salen del estacionamiento o una comunicación por radiofrecuencia para aumentar la seguridad dentro del estacionamiento.

Bibliografía

Alcaraz, M. (20 de Enero de 2021). *Internet de las Cosas.* Obtenido de http://jeuazarru.com/wp-content/uploads/2014/10/Internet-of-Things.pdf

Alvarez, M. A. (19 de Noviembre de 2003). *Desarrollo Web*. Obtenido de https://desarrolloweb.com/articulos/1325.php

Amin Kianpisheh, N. M. (2012). Smart Parking System (SPS) Architecture Using Ultrasonic Detector. *International Journal of Software Engineering and Its Applications*, 51-58.

Arduino. (s.f.). *What is Arduino?* Obtenido de https://www.arduino.cc/en/Guide/Introduction

Ariel. (2011). *Smart Cities: Un primer paso hacia internet de las cosas .* Madrid (España): Planeta .

Caïs, J. L. (2014). La Investigación Cualitativa Longitudinal. *Revista Latinoamericana de Metodología de las Ciencias Sociales*.

Cameron, N. (2021). *Electronics Projects with the Esp8266 and Esp32: Building Web Pages, Applications, and Wifi Enabled Devices.* Apress.

Carlos Alberto Vera Romero, J. E. (2017). La Tecnología ZigBee estudio de las características de la capa física. *Scientia et Technica Año*, 238-245.

César Augusto, S. S. (2011). Buses de campo y protocolos indistriales. *Universidad de Manizales Facultad de Ciencias e Ingeniería*, 83-109.

Chachin, P. (2017). LPWAN wireless technologies application for IoT market. *ELECTRONICS: Science, Technology, Business*, 140-144.

Chio Cho, N. T. (2011). REDES DE SENSORES INALÁMBRICOS. *Congreso Internacional de Ingeniería Mecatrónica-UNAB*.

Córdoba, D. M. (2013). ESTADO DEL ARTE DE LAS REDES DE SENSORES INALAMBRICOS. *Tecnología Investigación y Academia*, 4-14.

D. Setijono, J. D. (2007). Customer value as a key performance indicator (KPI) and a key improvement indicator (KII). *Measuring Business Excellence*, 44-61.

Dhulipala, G. R. (2019). Intelligence in IoT-enabled Smart Cities. En F. Al-Turjman, *Intelligence in IoT-enabled Smart Cities.* CRC Press.

Evans, D. (2011). Internet de las cosas Cómo la próxima evolución. *Cisco*, 2-12.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript.* Barcelona: marcombo.

Giffinger, R. y.-M. (2007). Smart Cities: Ranking of European Medium. *Vienna University of Technology*.

Giraldo, V. (16 de Abril de 2019). *Rockcontent.* Recuperado el 10 de Enero de 2022, de https://rockcontent.com/es/blog/que-es-firebase/

Hong, S. Y. (2020). AN EFFICIENT IOT APPLICATION DEVELOPMENT BASED ON IOT KNOWLEDGE MODULES. *Issues In Information Systems*, 72-82.

Hunt, C. (1987). TCP/IP network management server. *Computer Communications*, 158-162.

Implementation of Radio Frequency Communication System based Serial UART Communication. (2014). *Journal of Digital Convergence*, 257-264.

Ing. Alexander Laffita Leyva, M. S. (2012). Diseño, construcción y calibración de un transductor de fuerza tipo S. *Centro de Mecanización Agropecuaria, Universidad Agraria de La Habana, Mayabeque, Cuba.*

j. Rut, M. B. (2016). Visualization techniques for production processes using the communication TCP/IP protocol. *Mechanik*, 1732-1733.

jatuporn.chinrungrueng, u. s. (2007). SmartParking: anApplicationofopticalWirelessSensorNetwork. *International Symposium on Applications and the Internet Workshops* .

Jin, H. (214). Implementation of Radio Frequency Communication System based Serial UART Communication. *Journal of Digital Convergence*, 257-264.

JW. Gallagher, N. K. (2007). Repressed eIF2B-epsilon expression delays tumor growth in transformed mouse embryonic fibroblasts (TMEF). *The FASEB Journal*.

Luna, D. (2016). Validación del Diseño Centrado en el Usuario. 45-49.

María L. Luna-Gonzalez, S. M.-B.-D.-Q. (2020). Implementation of open-source technology and remote sensors for a biobank: the challenge of producing at low cost. *Información tecnológica*.

Mufaqih, M. S. (2020). Applying smart parking system with internet of things (IoT) design. . *IOP Conference Series: Materials Science and Engineering*, 725.

Pérez, J. E. (2009). *Introducción a JavaScript.* Creative Commons.

Pérez, J. U. (2014). Metodología para el diseño de una red de sensores inalámbricos. *Universidad Nacional Experimental Politécnica*.

Picone, M. (2020). IoT: A New Open Access Journal for Internet of Things. *IoT*, 145-146.

Recarey, L. E. (2014). Movilidad inteligente. Implantación de un sistema inteligente de aparcamiento en espacios libres. *Congreso nacional del medio ambiente*.

Fernandez Martinez, R., Ordieres Mere, J., Martinez de Pison Ascacibar, F. J., Gonzalez Marcos, A., Alba Elias, F., Lostado Lorza, R. y Pernia Espinoza, A. V. (2009). Redes inalámbricas de sensores: teoría y aplicación práctica. Universidad de la Rioja. Servicio de publicaciones. https://dialnet.unirioja.es/descarga/libro/377564.pdf

Robotica. (9 de Enero de 2020). *Sensor srf04*. Obtenido de http://www.superrobotica.com/S320110.htm

Seguí, P. (8 de Mayo de 2014). *Ovacen*. Obtenido de https://ovacen.com/internet-de-las-cosas/

Sheshalevich, V. (2017). LPWAN – Low-power Wide-area Network. Communication for the Internet of Things. *Bezopasnost informacionnyh tehnology*, 7-17.

Tozer, E. (2004). *Broadcast Engineer's Reference Book.* Routledge.

ubidots. (15 de 05 de 2021). *ubidots*. Obtenido de https://ubidots.com/

Vildósola, E. (24 de 11 de 20). *Soltex Chile S.A.* Obtenido de http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf