

**REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL**

***Diseño y prototipado de un sistema de detección de lugares disponibles en un estacionamiento, mediante la implementación de tecnología inalámbrica y energía renovable.***

**Autor: Israel Díaz Ceballos**

**Tesis presentada para obtener el título de:  
Ingeniero en Mecatrónica**

**Nombre del asesor:  
Daniel Iván Pureco Niño**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar, organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





## ESCUELA DE INGENIERÍA MECATRÓNICA

Diseño y prototipado de un sistema de detección de lugares disponibles en un estacionamiento, mediante la implementación de tecnología inalámbrica y energía renovable.

TESIS

Que para obtener el título de:

Ingeniero en Mecatrónica

PRESENTA

Israel Díaz Ceballos

Yazek Martínez Bernal

ASESOR

Daniel Iván Pureco Niño

CLAVE:

ACUERDO:

MORELIA, MICHOACÁN

MARZO, 2020

## **Agradecimientos**

Este documento está dedicado a mi familia, a mis padres que me brindaron las bases para poder estudiar la carrera que yo deseaba, por su apoyo incondicional y por todas las veces que me han tenido que ayudar en esta vida, los amo.

A mis hermanos, que, aunque pueda llegar a tener problemas con ellos, siempre me han brindado la experiencia que me hace falta para afrontar diversos problemas, además de ser las personas con las que mejor me la paso.

A mis amigos hechos en la UVAQ, por hacer de la universidad la mejor etapa de mi vida, sé que triunfaran en un futuro y podremos volver a juntarnos para hablar, como si nunca hubiéramos dejado ese salón de clases.

Agradezco a mi compañero de tesis, uniendo esfuerzos pudimos completar el proyecto, por el buen trabajo en equipo que supimos llevar a lo largo del proceso, además de ser un gran amigo, fue divertido trabajar con él y le deseo mucha suerte en el futuro, aunque no creo que la necesite.

A mi asesor, que sin él no hubiéramos logrado cumplir con el proyecto, por los consejos y por orientarnos cuando no sabíamos continuar, por aguantar cuando flojeábamos y no enojarse mucho por ello, mejor asesor no pudimos tener.

Va dedicado a todas las personas que en mayor o menor medida me han ayudado a ser quien soy ahora y me han impulsado a ser mejor persona, espero responderles y no dejar de mejorar en el futuro.

Trabajen mucho por sus sueños, pero nunca dejen de disfrutar del proceso.

Israel Díaz Ceballos

Este proyecto profesional no habría sido posible sin la ayuda de muchas personas que formaron parte de mi vida ya sea por un corto o largo periodo de tiempo, agradezco que durante el proceso me guiaron y me brindaron su tiempo escuchándome y aconsejándome.

Agradezco a mi familia, quienes siempre han estado a mi lado, por siempre apoyarme, por el amor recibido y la paciencia, por enseñarme a ser un hombre y demostrarme que con esfuerzo y dedicación se puede lograr lo que sea, por apoyarme en cada decisión y proyecto y siempre ser los mejores guías, los amo.

Agradezco a mis amigos y compañeros por acompañarme durante la universidad y hacer de esta experiencia única, por darme la tranquilidad de siempre saber que puedo contar con alguien en las buenas o en las malas, espero que todos sus sueños tengan éxito.

A la Universidad Vasco de Quiroga por abrirme sus puertas y darme la oportunidad de realizar mi formación académica y profesional.

A mi amigo y compañero de tesis, que solo él y yo sabemos las horas de trabajo que necesitó, por apoyarnos siempre que uno o el otro no quería seguir y por impulsarnos para la realización de este proyecto.

A mi asesor Daniel Ivan Pureco Niño, que siempre mostró la disposición para resolver nuestras dudas y ayudó a darle pies y cabeza al proyecto.

Y a todas esas personas que pusieron su grano de arena para que me encuentre aquí, gracias a todos. La suerte favorece solo a la mente preparada.

Yazek Martínez Bernal

# Índice

Índice	4
Introducción	9
Justificación	10
Objetivo general	11
Objetivos particulares	11
Hipótesis	11
Descripción del proyecto	12
Capítulo 1 Marco teórico	14
1.1 Automatización	15
1.1.1 Proceso de automatización: definición	15
1.1.2 Objetivos generales y específicos del proceso de automatización	15
1.1.3 Sistema automatizado: Definición y fases de desarrollo	16
1.1.4 Parámetros característicos de los AS (Sistemas automatizados)	19
1.1.5 Conceptos básicos de automatización	22
1.2 Microcontroladores	23
1.2.1 Arquitectura de un microcontrolador	23
1.2.2 Unidad de control	24
1.2.3 Memoria	24
1.2.4 Entradas y salidas de propósito general	25
1.2.5 Temporizadores	26
1.2.6 Contadores	26
1.2.7 Registros	27
1.2.8 Puerto Serie	27
1.2.9 ESP8266	27
	4

1.2.9.1 Ventajas y desventajas de los módulos Wifi ESP8266	30
1.3 Sensores	31
1.3.1 Tipos de sensores	31
1.3.2 Sensores modulares y generadores	31
1.3.3 Sensores analógicos y digitales	31
1.3.4 Sensores por deflexión o comparación	32
1.3.5 Sensores ultrasónicos	33
1.3.5.1 Sensor Ultrasonico JSN-SR04T	34
1.4 Raspberry Pi	35
1.4.1 Historia de la Raspberry Pi	35
1.4.2 Hardware	36
1.4.3 SoC (System on a Chip, Sistema en un chip) (ARM vs X86)	36
1.4.4 CPU	37
1.4.5 GPU	37
1.4.6 RAM	37
1.4.7 Almacenamiento	38
1.4.8 Características técnicas	38
1.5 Sistemas informáticos	39
1.5.1 Base de datos	39
1.5.1.1 Tipos de bases de datos	40
1.5.1.2 Ejemplos de bases de datos	41
1.5.2 MySQL	42
1.5.2.1 SQL, la base de las aplicaciones de MySQL	43
1.5.3 Servidores	44
1.5.3.1 Terminos	45
1.5.3.2 Tipos de Servidores	47
1.5.3.3 Servidor web	48

1.5.3.4 Arquitectura	48
1.5.3.5 Funcionamiento	48
1.5.3.6 Servidor Web Apache	49
1.6 Aplicación móvil	50
1.6.1 Usabilidad de una aplicación	50
1.6.2 Efectividad de una aplicación	51
1.6.3 Eficiencia de una aplicación	51
1.6.4 Satisfacción en una aplicación	51
1.6.5 Adquisición de datos	52
1.7 Energías renovables	52
1.7.1 Tipos de energías renovables	53
1.7.1.1 Energía solar	53
1.7.1.2 Energía solar térmica	53
1.7.1.3 Energía eólica	53
1.7.1.4 Energía geotérmica	54
1.7.1.5 Energía hidráulica	54
1.7.1.6 Biomasa	54
1.7.1.7 Energía océanos	55
1.7.1.8 Hidrógeno	55
1.7.1.9 Energía térmica	55
1.7.1.10 Energía nuclear	55
1.7.2 Ventajas de las energías renovables	56
1.7.3 Sistema fotovoltaico autónomo	57
1.7.3.1 SFA de corriente directa	58
1.7.3.2 Energía Solar	59
1.7.3.3 Panel Solar	60
1.7.3.4 Funcionamiento de una celda solar	61

1.7.3.5 Panel Solar 10W y 12V de potencia	62
1.7.3.6 Controladores de carga	63
1.7.3.7 Controlador de carga solar PWM 20A 12V/24V	63
1.8 Baterías de ácido-plomo	64
1.8.1 Historia	65
1.8.2 Constitución	66
1.8.3 Batería CSB GP 1272	66
Capítulo 2 Metodología	68
2.1 Presentación del proyecto	69
2.2 Raspberry Pi 3	70
2.2.1 Instalando la raspberry Pi	70
2.2.1.1 Descargar el Sistema Operativo	70
2.2.1.2 Montando la tarjeta SD	71
2.2.2 Arranque de Raspberry	72
2.2.2.1 Conectando la Raspberry a Internet	74
2.2.2.2 Actualizando el sistema	75
2.2.2.3 Configurando IP estática	76
2.2.3 Instalar el servidor web	77
2.2.3.1 Instalación y configuración de MySQL	79
2.2.4 Red local	79
2.2.4.1 PHPMyAdmin	80
2.2.4.2 Crear la página para subir valores a la base de datos	84
2.3 Programación del microcontrolador	87
2.3.1 Drivers necesarios	88
2.3.2 Código interno	90
2.3.2.1 Void setup	92
2.3.2.2 Void loop	93



2.4 Desarrollo del prototipo	95
2.4.1 Tope de estacionamiento	96
2.4.2 Caja de alimentación	98
2.4.3 Alimentación del sistema	99
2.4.4 Caja de comunicación	100
2.4.5 Instalación	101
2.5 Aplicación Móvil	104
2.5.1 Diseño	105
2.5.2 Programación de la aplicación	105
Resultados	110
Conclusión	112
Bibliografía	114

## Introducción

El ser humano por naturaleza busca simplificar las actividades repetitivas y aprovechan los recursos y descubrimientos obtenidos a lo largo de su historia para mejorar el tiempo utilizado, pero aún existen muchos espacios por explorar.

Desde la llegada de los primeros automóviles, las personas han cambiado drásticamente el tiempo que utilizaban para llegar a su destino. Los automóviles como medio de transporte han sufrido diversas evoluciones durante el tiempo, cambios que van desde el combustible a utilizar, el tamaño y los diversos usos que se han encontrado para ellos, las distancias que recorren no son las mismas de cuando se inventaron.

Con todos los avances y perfeccionamientos que han tenidos los vehículos, hay un apartado que ha quedado rezagado y que también supondría una mejora en la vida de las personas, dando más tiempo en un ritmo de vida donde las personas lo necesitan más que nunca.

Que el desarrollo de estos nuevos sistemas vaya evolucionando de acuerdo a los avances tecnológicos que se viven ahora es algo necesario, por ende, es importante buscar nuevas y mejores alternativas para hacer un uso más sencillo y eficiente de los recursos, sin embargo, la aplicación de la nueva tecnología conlleva a una elevación en los costos.

En la actualidad, las personas necesitan destinar menos tiempo aún para llegar a su destino y una forma de conseguirlo sería disminuir el tiempo que se toma encontrar un lugar para estacionarse.

La intención de nuestro proyecto es diseñar un sistema de detección de lugares disponibles en un estacionamiento público mediante una aplicación con el fin de optimizar tiempos y hacer esta práctica más sencilla.

Para eso la gente al momento de llegar a un estacionamiento, pueda ver en una aplicación móvil los lugares que están disponibles y mejore un apartado que ha tenido muy pocos avances.

## Justificación

Encontrar un lugar donde estacionarse, por lo general, suele ser un suplicio para cualquier conductor que circule por las ciudades más congestionadas del mundo. De hecho, un estudio comisionado por Ford indica que se puede tardar hasta un día completo al año buscando estacionamiento.

Nosotros buscamos que los usuarios ahorren tiempo en situaciones así y, por eso, trabajamos en el desarrollo de una tecnología que sea capaz de reducir el estrés y te permita saber a tiempo real qué número de plazas hay disponibles en un estacionamiento de grandes dimensiones.

Utilizando el sistema se pretende llevar un control de los coches que están en el estacionamiento, gracias a la programación que desarrollaremos se sabrá la hora en que llegan y la hora en que se van.

En caso de robo, se podría verificar más fácil la hora en que sucedió y con esto determinar el momento en las cámaras de seguridad que pueda tener el lugar en que se instale el sistema.

El desarrollo del proyecto ayudaría a la optimización del tiempo de las personas, además que se reduciría el tiempo que el motor dura encendido ayudando a reducir la emisión de gases perjudiciales al medio ambiente.

El tráfico generado en los estacionamientos reducirá considerablemente después de la instalación del sistema, debido a que las personas no estarán dando vueltas alrededor del estacionamiento para encontrar el lugar deseado, si no que podrán visualizarlo antes de dirigirse al estacionamiento.

Se estima que hacia el año 2030, el 60% de la población mundial vivirá en grandes ciudades, y para 2050 el porcentaje se eleva al 70%. Muchas de las grandes ciudades se convertirán en 'megaciudades', y cada vez será más necesario incluir sistemas de control y de inteligencia.

La implementación de tecnología en los estacionamientos se ha visto muy estancada con el pasar de los años, introducir un sistema de control inteligente basado desde

una aplicación será un avance tecnológico que hasta ahora no se ha visto implementado.

## **Objetivo general**

Diseñar y desarrollar el prototipo de un sistema de detección de lugares disponibles en un estacionamiento, mediante la implementación de tecnología inalámbrica y energía renovable.

## **Objetivos particulares**

- Comprobar la viabilidad de la implementación de un sistema automático para la reducción de tiempo en la búsqueda de un lugar.
- Diseñar el sistema de topes integrando sensores, microcontrolador, el diagrama de los circuitos necesario y la interfaz de la aplicación móvil.
- Instalar el sistema prototipo.
- Desarrollar la programación, control de seguridad y una app que muestre al usuario de manera ordenada y a tiempo real los lugares disponibles en un estacionamiento.
- Crear una base de datos en MySQL, instalada en una Raspberry Pi.
- Desarrollar la aplicación funcional en la plataforma de AppInventor.
- Evaluar y seleccionar las tecnologías existentes para que los componentes que integran el sistema sean de bajo consumo, y así se tenga la proyección tecnológica adecuada.

## **Hipótesis**

La implementación de un prototipo de sistema automático que permita controlar los estacionamientos, tendrá un impacto positivo en lo relacionado a la búsqueda de un espacio libre de parte de un conductor, así como en el aspecto económico, tecnológico y social.

## Descripción del proyecto

La solución de estacionamiento consiste en un sistema de control y gestión de estacionamiento medido que monitorea de forma remota la disponibilidad de lugar en un lugar público. A través de la instalación de sensores en los cajones de estacionamiento, se podrá comprender si el lugar se encuentra ocupado o libre para estacionar. Esta información podrá ser consultada desde la app que desarrollaremos para cualquier dispositivo con sistema operativo Android.

El diseño del sistema y la programación de la aplicación será desarrollada por nosotros, así como la programación del microcontrolador con medidas de seguridad en casos de que el sensor detecte de manera errónea y también una actualización de los datos en tiempo real mostrados en la aplicación.

Se desarrollará un programa a partir de un Microcontrolador ESP8266 para la detección de lugares disponibles en un estacionamiento, mediante un sensor ultrasónico Jsn-sr04t-2.0 instalado en cada cajón, utilizar el microcontrolador ESP8266 tiene la ventaja de ser barato al momento de ampliar el estacionamiento debido a los módulos de expansión para el microcontrolador en caso de que se tecnifiquen más cajones de estacionamiento.

El sistema estará instalado en un tope de hule por cada cajón, cada cajón tendrá: microcontrolador ESP8266, sensor ultrasónico y un regulador de voltaje LM7805. Estos componentes estarán conectados a una batería situada en una caja de control junto a un regulador de carga que recibirá luz solar de dos paneles.



Figura 1 Imagen del tope.

Se utilizará un sensor de distancia ultrasónico a prueba de salpicaduras Jsn-sr04t-2.0 el cual tiene estas características: Cuenta con una medición de hasta 6 pm, además de compatibilidad con varios tipos de microcontroladores. Los sensores serán instalados en 4 cajones en una propiedad privada, estos estarán comunicados por Wi-Fi, el microcontrolador será programado por nosotros, teniendo en cuenta medidas de control de errores.

El microcontrolador recibirá los datos de los sensores a través de la programación que haremos en C++, una vez hecho esto será enviada como una variable booleana de manera inalámbrica a un servidor instalado en una microcomputadora Raspberry Pi, que almacena los datos para poder ser mostrados de manera gráfica e inalámbrica en una aplicación instalada en un dispositivo móvil.

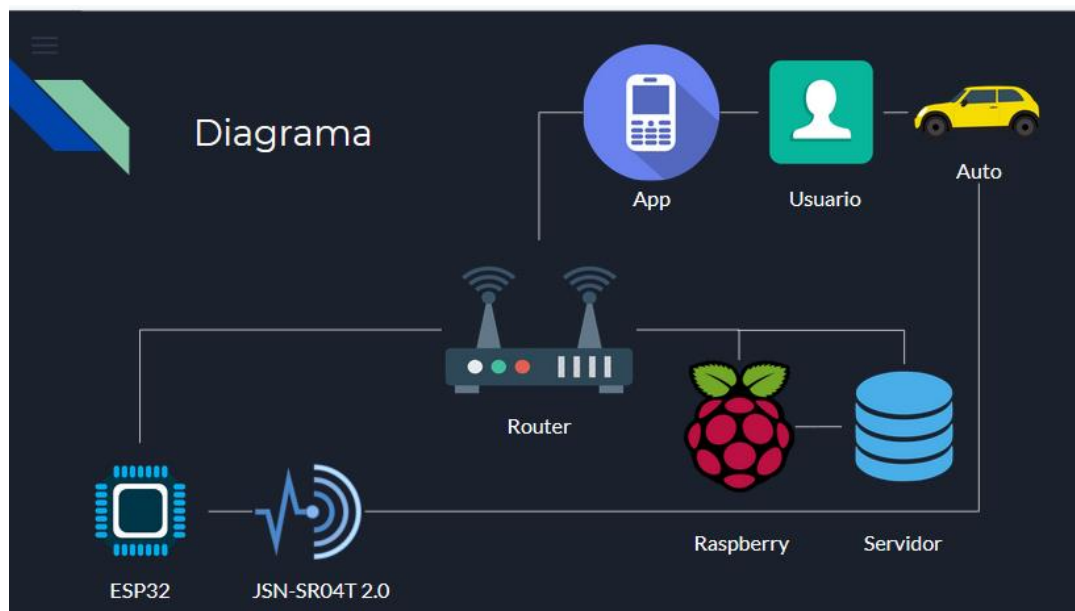


Figura 2 Diagrama del proyecto

La aplicación será diseñada en la plataforma App Inventor para dispositivos Android. Esta mostrará los 4 cajones de prueba, tendrá restricciones para el usuario, donde únicamente podrá observar los cambios en los cajones tecnificados. La distinción de los cajones ocupados y libres, serán mostrados gráficamente con un coche en el cajón o el cajón libre.

## **Capítulo 1 Marco teórico**

## **1.1 Automatización**

### **1.1.1 Proceso de automatización: definición**

La Real Academia de las Ciencias Físicas y Exactas, (2006) define la automática, “como el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas”. De esta definición original se desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales

Ponsa Asensio, (2001) define la automatización como “la incorporación de un conjunto de elementos y dispositivos tecnológicos que aseguren su control y buen comportamiento”.

La Industrial Systems Engineering (2011) lo define como, “un proceso que consiste en la sustitución de aquellas tareas tradicionalmente manuales por las mismas realizadas de manera automática por máquinas, robots o cualquier otro tipo de automatismo”.

De este modo, gracias al uso adicional de sensores, controladores y actuadores, así como de métodos y algoritmos de conmutación, se consigue liberar al ser humano de ciertas tareas. El ser humano siempre ha buscado la creación de nuevas herramientas y máquinas que le faciliten la realización de aquellas tareas más peligrosas, pesadas y repetitivas. En los últimos tiempos, la aparición de máquinas altamente sofisticadas ha dado lugar a un gran desarrollo del campo de la automatización y el control de las tareas, aplicado ya en muchas máquinas que se manejan diariamente.

Con esta información, podemos concluir que la automatización, es el proceso por el cual se pretende sustituir al hombre de ciertas tareas repetitivas y cambiarlo por máquinas o procesos previamente programados.

### **1.1.2 Objetivos generales y específicos del proceso de automatización**

Industrial Systems Engineering (2011) menciona que los principales objetivos del proceso de automatización son:

- Mejorar la productividad y eficiencia, reduciendo los costos de producción y mejorando la calidad y precisión del producto final.



- Optimizar la planificación y el control.
- Mejorar las condiciones de trabajo del personal, suprimiendo las tareas más tediosas e incrementando su seguridad.
- Realizar aquellas operaciones imposibles de controlar intelectual o manualmente.

El proceso de automatización, excepto para el caso más simple, requiere una aproximación compleja y sistemática para resolver los problemas. Las complicaciones que surgen a la hora de resolver estos problemas, de cara a la automatización se denominan sistema: automatización de sistemas de control y/o diseño, control automatizado de procesos tecnológicos, etc.

En un SI (Investigaciones científicas) teórico, generalmente, lo que puede llegar a automatizarse es el modelo matemático y los diferentes cálculos. En los SI experimentales, sin embargo, deben incluirse todos los procesos laboriosos no creativos, facilitarse la máxima creatividad al mismo tiempo que ser extremadamente eficientes y económicos.

En el proceso de automatización podemos distinguir distintos tipos de control dependiendo del grado de intervención humana. De este modo, denominaremos “control automático” cuando este se lleva a cabo sin ninguna intervención directa del ser humano, mientras que el “control automatizado” se referirá a aquel en el que existe una considerable intervención humana. Es necesario el desarrollo de sistema de regulación y el control automático para conseguir un buen control automático. Además, denominaremos Teoría del Control Automático (TAC) a aquella que se encuentra relacionada con los principios del desarrollo y los métodos de análisis y síntesis de los sistemas para la regulación y el control automático.

### **1.1.3 Sistema automatizado: Definición y fases de desarrollo**

La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos, como ya se mencionó en las secciones anteriores. Un sistema automatizado consta principalmente de dos partes: una de mando y otra operativa. Esta última es la que actúa directamente sobre la máquina haciendo que se mueva y

realice la operación deseada gracias a los actuadores y sensores que la componen. La parte de mando, sin embargo, suele ser un autómata programado que está en el centro del sistema y es capaz de comunicarse con el resto de constituyentes del sistema.

Las etapas de desarrollo de un Sistema Automatizado son las siguientes, Industrial Systems Engineering (2011):

- Falta de aparatos automáticos y manuales para la adquisición de los datos principales. Procesamiento de datos por medio de aparatos de medida semiautomáticos.
- Utilización de aparatos de registro y regulación automáticos para las mediciones indirectas.
- Empleo de sistemas con control centralizado y procesamiento de datos por medio de aparatos de conmutación programables con o sin conexión física directa.
- Utilización de: convertidores analógico-digital para datos analógicos, sistemas de medida de información con sistemas de computadores embebidos, automatización compleja para complejos indirectos y mediciones combinadas, automatización del diseño gráfico, etc.
- Uso de redes de ordenadores miniaturizadas, embebidas y con PC, desarrollo de sistemas para logística, así como de técnicas cibernéticas como la autorregulación, compresión automática de los datos, acumulación y sistematización de las bases de datos, etc.

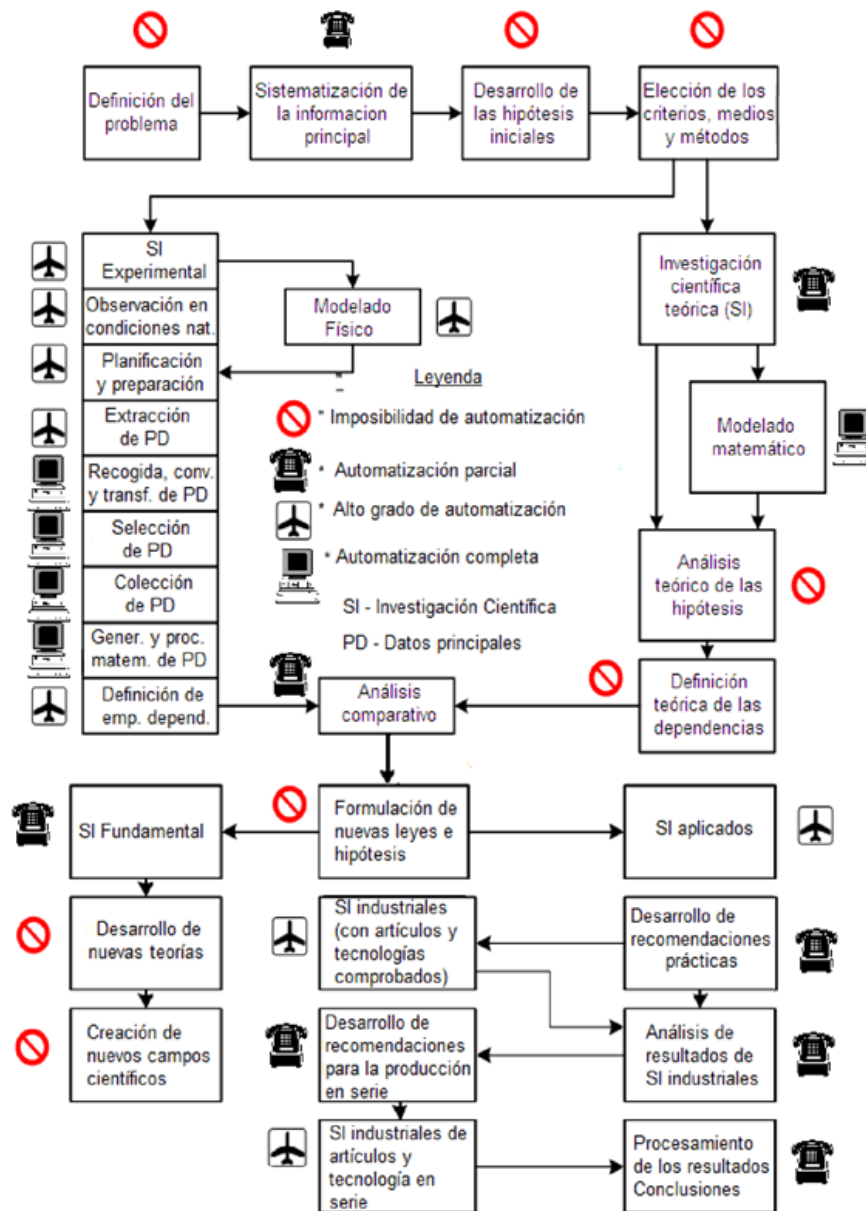


Figura 1.1 Principales etapas de los sistemas automatizados (Industrial Systems Engineering, 2011).

Hoy en día, para conseguir la automatización de un sistema, se utilizan principalmente métodos de control numéricos y, específicamente, métodos de control por computación. También han ido aparecido conceptos, tales como CAD (Diseño de Ordenadores Asistido), CAM (Manufactura gracias a los ordenadores) y, en general, Tecnologías asistidas con Ordenadores (CAx). Sin embargo, los principales modos de automatización son: Sistemas Neuronales Artificiales (ANN), sistemas de control distribuido (DCS), interfaz hombre-máquina (HMI), control de supervisión y adquisición de datos (SCADA), controladores lógicos programables (PLC), controladores programables automáticos (PAC) y robótica.

#### **1.1.4 Parámetros característicos de los AS (Sistemas automatizados)**

Los parámetros característicos a mejorar con la automatización son:

- Productividad
- Eficiencia
- Precisión
- Fiabilidad

La productividad es la relación entre la cantidad de productos obtenida por cierto sistema productivo y los recursos utilizados en su elaboración. Uno de los objetivos de los sistemas automatizados es conseguir un aumento de esta productividad y reducir la información redundante. Para este fin, se reemplazarán aquellos instrumentos más lentos por otros más rápidos; se utilizarán a la par ambos tipos de instrumentos (rápidos y lentos); se desarrollarán algoritmos óptimos de ejecución; se realizará una estricta selección de la información útil; y se extraerá la información que buscamos a partir de un cierto procesamiento preliminar. En algunos casos se podrá realizar también una reducción racional de la precisión con el fin de aumentar así las acciones rápidas. Además, se utilizarán aparatos con buffer (memoria de almacenamiento temporal) de tipo acumulativo, procesadores locales y métodos de motivación para aumentar la productividad de los operadores humanos (diálogo, simplificación de funciones, etc.). En resumen, los métodos organizativos utilizados para aumentar la productividad se basan en:

- Emplear medidas organizativas para uniformizar la complejidad y carga de los AS.
- Reducción del tiempo de preparación.
- Automatización de aquellas operaciones de ajuste, calibración, diagnóstico de fallos y reparaciones, etc.

La eficiencia se define como la relación entre los recursos utilizados en cierto proyecto y los logros conseguidos con el mismo. En el caso de la automatización de sistemas, la eficiencia está caracterizada por la relación entre el tiempo útil y el tiempo total del AS. Tiempo útil es aquel en el que realmente está trabajando el sistema, mientras que

el tiempo total es aquel durante el cual el AS se encuentra en condiciones de trabajo (aunque no siempre esté ejecutando una tarea). La carga normal de un AS tiene una eficiencia igual a 0,15-0,25 y raramente su valor es cercano a la unidad, excepto en estaciones automáticas para usos prolongados o en la automatización de procesos tecnológicos continuos. Sin embargo, en ciertos AS, muy raramente utilizados en experimentos científicos, la eficiencia puede llegar a ser menor a 0,03.

La precisión es el grado de repetitividad de un determinado resultado. La precisión de la reproducción depende del modo proporcionado, así como de la precisión en la medida de varias cantidades. Cuando tenemos medidas combinadas, el problema se resuelve tratando de obtener la misma precisión para cada una de las medias individuales. Además, la precisión de cada una de las medidas también influye en el sistema. El hecho de tener una medida muy exacta no mejora considerablemente la precisión del sistema total. Sin embargo, una medida inexacta sí puede llegar a devaluar el resto de las medidas, aunque sí hayan obtenido con alta precisión. De este modo el propósito para aumentar la precisión de una determinada medida será conseguir precisión en las siguientes mediciones. Los instrumentos de impacto también poseen una gran influencia en la presión del control de un proceso, a excepción de los instrumentos de medida. De hecho, se ha demostrado cómo la frecuencia de los errores en un control automático puede llegar a reducirse en un orden 3-4 en comparación con el control manual.

El diccionario de la RAE (2020) define fiabilidad como “probabilidad de buen funcionamiento de algo”. Así, en el caso de un sistema, fiabilidad es la probabilidad de que es sistema funcione o desarrolle una cierta función, bajo condiciones ya fijadas y durante un periodo de tiempo determinado. En el caso de los AS, queremos que la fiabilidad sea lo más elevada posible. Para ello, es necesario que no existan pérdidas en la precisión de los datos procedentes a las mediciones y control ni distorsiones intolerables. Además, no debe eliminarse nunca el objetivo de la investigación ni caer en desuso cualquiera de los equipamientos experimentales o tecnológicos utilizados. El incremento de la fiabilidad está relacionado con la resolución de tres problemas principales:

- Seguridad óptima.

- Control óptimo de la fiabilidad.
- Profilaxis óptima del trabajo con AS.

Además, los AS pueden clasificarse según el parámetro matemático principal y otros parámetros secundarios, como se indica en la siguiente figura:

	Parámetro matemático	Parámetro Secundario
AS	I. Automatización de objetos	Procesamiento industrial
		Construcciones
		Ingeniería de diseño
		Organización y planificación
		Investigación y entrenamiento
		Militar y objetos cósmicos
		Medicina y técnicas de diagnóstico
		Procesamiento de datos
		Programación
		Cálculos de ingeniería etc.
	II. Grado de universalidad	Universal
		Especializado
	III. Principio de la operación	Digital
		Analógico
		Digital-analógico
	IV. Modo de utilización	Continua
		Regular
		Irregular
	V. Grado de automatización	Automatico
		Semi-automatico
	VI. Estacionalidad e invariancia	Estacionario
Semi-estacionario		
Portátil		
Modular		
VII. Energía utilizada	Eléctrica	
	Hidráulica	
	Neumática	

Figura 1.2 Diferentes tipos de AS (Industrial Systems Engineering 2011)

### 1.1.5 Conceptos básicos de automatización

Valor de referencia: Es el valor ideal que se pretende obtener a la salida del sistema controlado. En un sistema más complejo, la salida es censada y comparada con el valor de referencia a fin de determinar la diferencia entre ambas para reducir el error de salida.

Controlador: Regula presiones, temperaturas, niveles y caudales, así como todas las funciones asociadas de temporización, cadencia, conteo y lógica.

Sistema: Es la combinación de componentes que interactúan para lograr un determinado objetivo. En este caso el sistema es el objeto a controlar.

Entrada del sistema: Es una variable que al ser modificada en su magnitud o condición puede alterar el estado del sistema.

Salida del sistema: Es la variable que se desea controlar (posición, velocidad, presión, temperatura, etc.).

Perturbación: Es una señal que tiende a afectar el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se la denomina interna, mientras que la perturbación externa se genera fuera del sistema y constituye una entrada.

Sensores o transductores: Captan las magnitudes del sistema, para saber el estado del proceso que se controla.

Hay de dos tipos como se menciona en el libro, ingeniería de control (2004):

- **Sistemas de Control de Lazo Abierto.** Los sistemas de control de lazo abierto son aquellos en los que la salida no tiene efecto sobre la acción del controlador, es decir, la salida ni se mide ni se realimenta para compararla con la entrada. Por lo tanto, para cada valor de referencia corresponde una condición de operación fijada. Así, la exactitud del sistema, depende de la calibración.
- **Sistemas de Control de Lazo Cerrado.** Un sistema de control de lazo cerrado es aquel en el que la señal de salida tiene efecto directo sobre la acción del controlador. La señal de error actuante, (que es la diferencia entre la señal de entrada y la de realimentación) entra al control para reducir el error y llevar la salida del sistema al valor deseado.

En otras palabras, el término “lazo cerrado”, implica el uso de acción de realimentación para reducir el error del sistema.

## **1.2 Microcontroladores**

Los microcontroladores son un tipo de elemento electrónico que, gracias a la integración de elementos de procesamiento, almacenamiento y periféricos, son capaces de automatizar tareas de control y monitoreo de dispositivos en un sistema. Siendo usados en un sinnúmero de dispositivos eléctricos como automóviles, semáforos, electrodomésticos, equipos de cómputo, equipos robóticos industriales e inclusive en juguetes.

Con la introducción de los microcontroladores serie TMS 1000 a principios de los setenta, los microcontroladores han sido una de las soluciones más eficaces y baratas para sistemas en donde una tarea debe ser realizada a través de varios procesos en donde los dispositivos deben tener una función dedicada, es decir, en donde los dispositivos deben ser monitoreados o controlados en tiempo real.

### **1.2.1 Arquitectura de un microcontrolador**

La arquitectura de un microcontrolador puede ser diferenciada dependiendo del bus por el que los datos e instrucciones del microcontrolador sean procesados. Las dos arquitecturas más usadas son:

- **Arquitectura Von Neumann.** Se caracteriza por manejar una única memoria, por lo que existe un bus común que comunica la información del programa con el procesador, así como también los datos del programa. Por lo que carecen de velocidad al no diferenciar las instrucciones del programa con los datos de éste.
- **Arquitectura Harvard.** A diferencia de la anterior, ésta tiene una memoria especial para los programas del microcontrolador, además de contar con una memoria que sólo almacenará los datos que el programador desee, haciendo la ejecución más fluida y por ende un acceso a los datos más eficiente.





Figura 2.1 Arquitectura de Von Neumann.



Figura 1.3 Arquitectura Harvard

### 1.2.2 Unidad de control

La unidad de control de un microcontrolador, comúnmente está compuesto por una unidad central de procesamiento (CPU) pequeño, siendo basados en la tecnología semiconductora de los transistores que, a diferencia de los CPU's de nuestras computadoras, tienen un número limitado de transistores, que reducen no sólo el costo sino también la capacidad de procesamiento. La unidad de control se encarga de direccionar las señales eléctricas de todo el microcontrolador a través de la unión de la Unidad de Lógica Aritmética y la Memoria de éste.

### 1.2.3 Memoria

En el libro escrito por Valdés, F. y Areny, R. P. (2007) definen la memoria como un elemento muy importante en la estructura de un microcontrolador, ya que se encarga no sólo de contener los programas que realizará el micro, sino también contener datos almacenados al realizar mediciones y cálculos cuando se está ejecutando el programa.

La memoria RAM (Random Access Memory) es un tipo de memoria que permite al microcontrolador almacenar información volátil del programa, que es usada para la ejecución de tareas en donde el proceso necesita el almacenamiento de datos por un tiempo limitado, siendo común en cálculos y operaciones lógicas.

A pesar de que la mayoría de las memorias de los microcontroladores pueden ser reescritas, existen también otros tipos de memoria usadas en los microcontroladores en donde no es posible reescribirlas ni inclusive leerlas. Siendo comúnmente usados los siguientes tipos.

- Memoria PROM: (Memorias programables de sólo lectura) son un tipo de memoria usada para microcontroladores en donde las actualizaciones no son requeridas, resultando en una memoria en donde sólo se puede programar una sola vez.
- Memoria EPROM: (Memorias programables borrables de sólo lectura) permiten a los usuarios borrar la memoria, siendo está a través de la exposición de ésta a luz ultravioleta.
- Memoria EEPROM: (Memorias programables borrables eléctricamente de sólo lectura) a diferencia de la mencionada anteriormente, permiten borrar los datos de la memoria eléctricamente, siendo innecesario sacar el microcontrolador para borrar los datos como se acostumbraba con las memorias EPROM.
- Memoria Flash: Proveniente de la tecnología EEPROM, este tipo de memorias a diferencia de las EEPROM puede ser programada sin ser necesario borrar información antes de su escritura, resultando en una memoria más rápida para la escritura y lectura. Resultando un éxito comercial no sólo en su implementación en microcontroladores sino también en almacenamiento USB o SSD.

#### **1.2.4 Entradas y salidas de propósito general**

Comúnmente nombradas GPIO (del inglés General Purpose Input Output o Entrada/salida de propósito general) son un tipo de periféricos usados en los microcontroladores, en donde el uso no está especificado inicialmente, permitiendo al usuario programar cómo se comportará dicho pin del microcontrolador, como entrada o como salida según éste lo desee.

Los pines GPIO, al permitir la lectura de las conexiones aplicadas al microcontrolador, suelen ser usados para conocer los valores de entrada de un dispositivo, siendo en algunos microcontroladores posible configurarse para cambiar los voltajes lógicos en

donde se podrá leer el valor de entrada. Por otra parte los pines de escritura permiten realizar tareas desde encender un LED indicador, hasta activar un circuito para encender un motor trifásico, resultando en un tipo de periféricos muy versátiles.

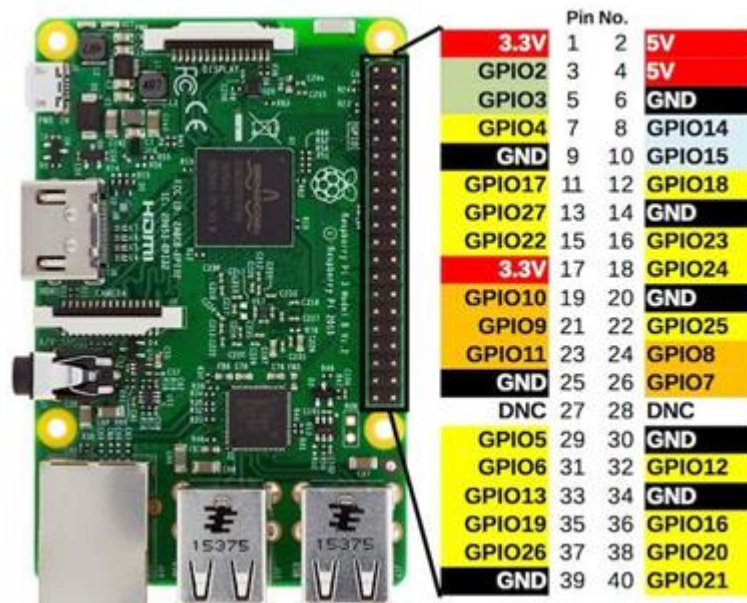


Figura 1.4 Pines GPIO de Raspberry Pi

### 1.2.5 Temporizadores

Los temporizadores en los microcontroladores son circuitos capaces de realizar un conteo de pulsos de reloj a través de un oscilador. Este tipo de tecnología suele ser usada en los microcontroladores para conocer la frecuencia de una señal desconocida, al permitir conocer cuántos pulsos de reloj cambiar la señal de valor. Los temporizadores de un microcontrolador suelen poder editarse a través de la modificación de los registros del micro.

### 1.2.6 Contadores

Los contadores de un microcontrolador suelen ser asociados a los temporizadores para evitar implementar un circuito externo que permita el conteo de los temporizadores. El tamaño de estos acostumbra ser de 8 y 16 bits.

### 1.2.7 Registros

Los registros en un microcontrolador son entidades de memoria que dictan las instrucciones de ejecución de las múltiples prestaciones del microcontrolador. Por ello un microcontrolador con mayor número de registros, aumenta la versatilidad de éste al poder modificar el comportamiento de las múltiples funciones del micro. Debido a la importancia de estas entidades de memoria suelen separarse físicamente de la memoria del programa.

### 1.2.8 Puerto Serie

Los puertos de comunicación de un microcontrolador son usados no sólo para comunicar los microcontroladores con el usuario, sino también para comunicarse con otros microcontroladores y realizar tareas a través de ésta.

La comunicación con puerto serie se basa en la transmisión de bits de datos en forma secuencial, para que un segundo dispositivo pueda reensamblar los bits y completar el mensaje completo. Siendo necesario especificar el inicio y el fin de mensaje. Este tipo de comunicación suele usarse para comunicar dos dispositivos con diferentes controladores, como bien podría ser un Arduino y un módulo Bluetooth HC-06.

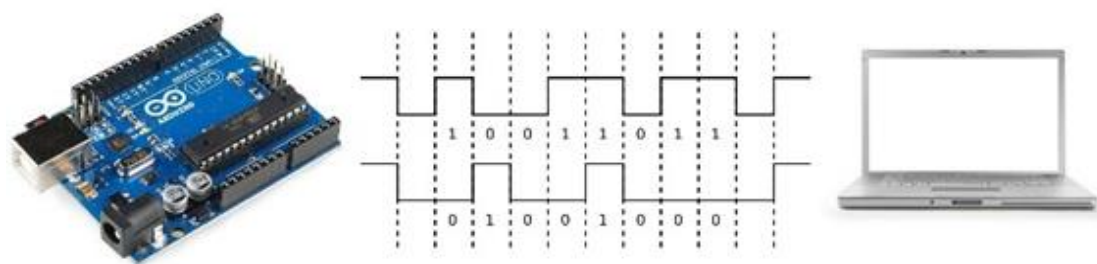


Figura 1.5 Funcionamiento de puerto serie en comunicación con ordenador.

### 1.2.9 ESP8266

Los módulos Wifi ESP8266 son pequeñas tarjetas electrónicas de bajo costo con una pila TCP/IP completa y un microcontrolador clasificadas como SoC(del inglés System on a Chip, sistema en un chip) que incluyen tecnología para conectarse a una WLAN

utilizando el mecanismo de conexión Wifi. Todos los módulos Wifi ESP8266 están basados en el chip ESP8266 producido por el fabricante chino Espressif. A mediados del año 2014 comenzaron salir noticias sobre la primera versión del módulo ESP8266, el ESP-01 que permitía a cualquier microcontrolador o sistema embebido conectarse fácilmente a una red Wifi mediante un conjunto de comandos basado en los comandos Hayes. Desde entonces se han desarrollado múltiples versiones del módulo, cada una con diferentes características.

El conjunto de comandos Hayes, es un estándar abierto de comandos utilizados para configurar y parametrizar dispositivos de red, los caracteres AT preceden a todos los comandos de este estilo.

En general, cualquier módulo ESP8266 cuenta con las siguientes características:

- CPU base: LX106 de la compañía Xetensa, con arquitectura RISC de 32 bit, corriendo a 80 [MHz].
- 64 [KB] de memoria RAM para instrucciones, más 96 [KB] de RAM para datos.
- Memoria flash desde 512 [KB] hasta 16 [MB].
- Tecnología Wifi según la norma IEEE 802.11 b/g/n Wifi.
- Hasta 16 pines de propósito general (GPIO) y convertidor analógico digital.
- Periféricos de comunicación SPI, I<sup>2</sup>C y UART.
- Pines de reinicio y polarización.

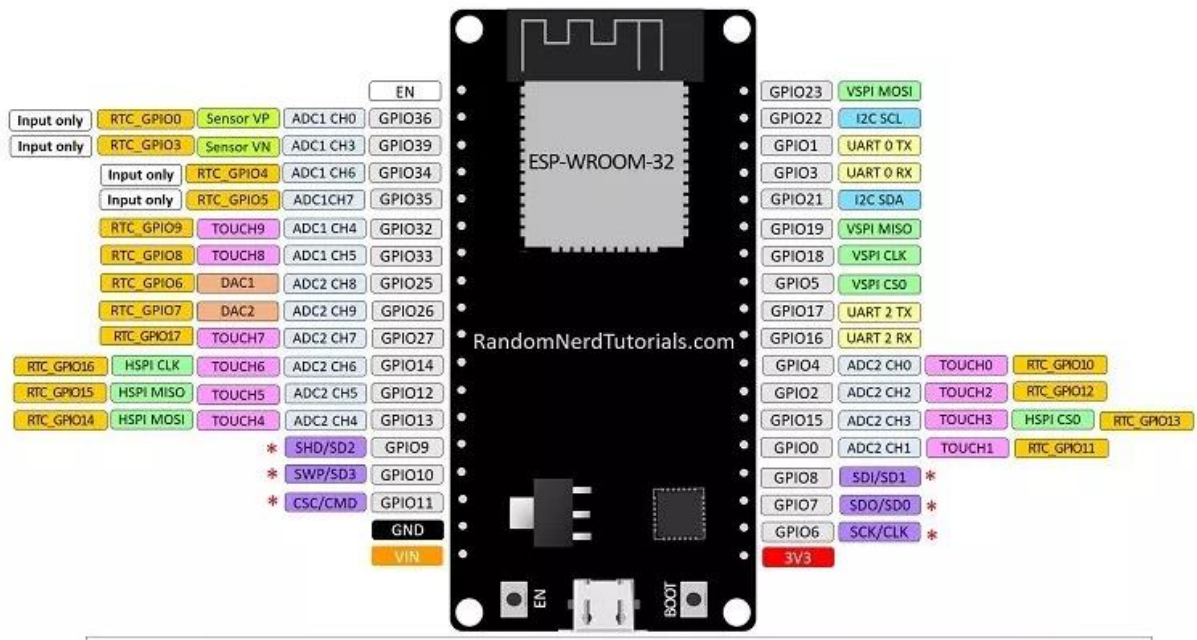


Figura 1.6 ESP32. Nodemcu 8266

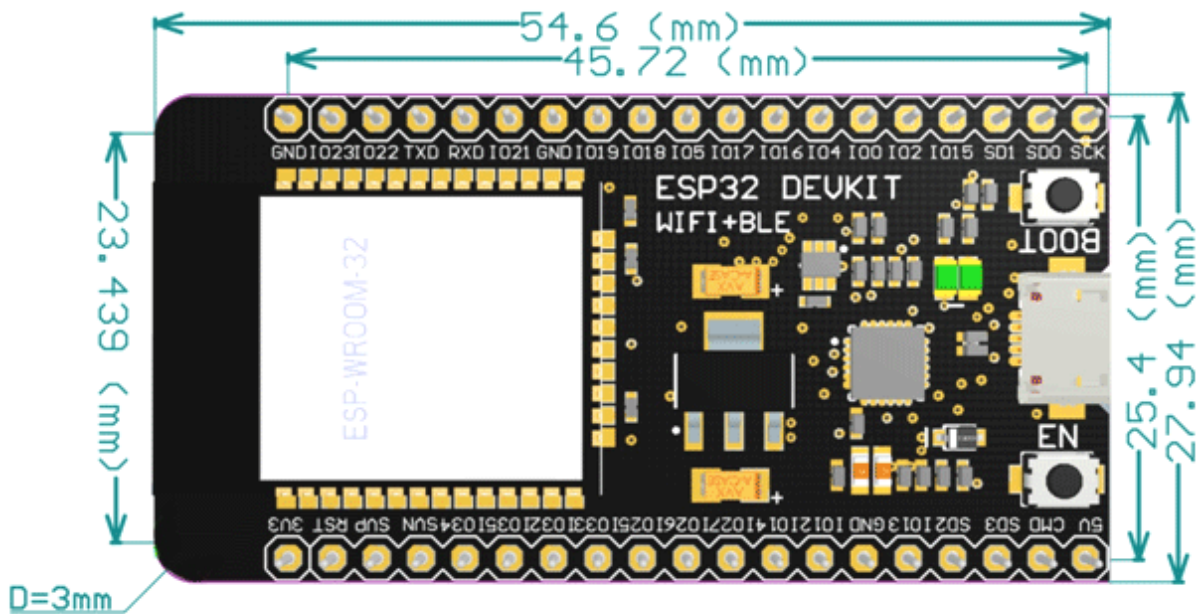


Figura 1.7 ESP32 2D Modelo

### **1.2.9.1 Ventajas y desventajas de los módulos Wifi ESP8266**

Ventajas: Permiten, mediante una interfaz serial asíncrona, a cualquier aplicación en microcontrolador conectarse a Internet. Pueden funcionar como punto de acceso y comportarse como servidores TCP o UDP. Son capaces de funcionar como estaciones, que se conectan a un punto de acceso y conectarse, a un servidor TCP o UDP, como clientes. Son muy baratos, el precio de los módulos varía entre 2 USD y 5 USD. Son muy pequeños, la versión más pequeña tiene un área de apenas un centímetro cuadrado. Existen ejemplos y videos de ciertos módulos ESP8266 estableciendo una conexión con un punto de acceso a más 300 [m] al aire libre usando la antena grabada en el circuito, esta distancia aumenta considerablemente con una antena externa. El fabricante Espressif ha liberado un SDK (paquete de desarrollo de software), con lo que el firmware de los módulos se puede modificar, aunque esto aún está en fase de desarrollo. En función del firmware grabado en la memoria de los módulos, éstos son capaces de establecer un modo de comunicación transparente, donde los datos transmitidos por su receptor serial son encapsulados según el protocolo TCP y enviados a un servidor, del mismo modo, y los datos transmitidos por el servidor se des encapsula en el módulo para luego ser enviados por su transmisor serial. En otras palabras, el módulo se comporta como si de un cable serial se tratara.

Desventajas: La información disponible está muy dispersa y la gran mayoría de esta es una mala traducción del chino al inglés. A pesar del gran éxito que han tenido estos módulos en occidente, aún es muy difícil de conseguirlos en la Ciudad de México, la mejor forma de conseguirlos es ordenarlos por Internet y tardan en llegar unas 5 semanas. Algunos módulos, son enviados con otro firmware diferente al oficial o son versiones piratas vendidas como originales, esto hace que el dispositivo funcione de manera inesperada. Para lograr un máximo alcance los módulos pueden llegar demandar hasta 500[mA] a la fuente de alimentación, esto encarece el diseño de dicha fuente. Algunos módulos tienen errores en la máscara de serigrafía de la tarjeta, o bien dicen haber obtenido la certificación FCC Ce (políticas y procedimientos específicos relacionados con las emisiones de radiofrecuencia), lo cual en algunos casos no es verdad.

## **1.3 Sensores**

### **1.3.1 Tipos de sensores**

El número de sensores disponibles para las distintas magnitudes físicas es tan elevado que, buscar una definición sin antes clasificar los tipos de sensores que se pueden encontrar, sería dar una definición incorrecta. Según el aporte de energía, los sensores se pueden clasificar en moduladores y generadores. Según el tipo de señal de salida, se pueden definir los sensores analógicos y los digitales. Tomando en cuenta su modo de funcionamiento, los sensores también se pueden clasificar por deflexión o comparación. Los sensores también se pueden clasificar por medio del tipo de relación entrada-salida, pueden ser de primer orden, segundo orden o de orden superior.

### **1.3.2 Sensores modulares y generadores**

En los sensores moduladores o activos, la energía de la señal de salida procede, en su mayor parte, de una fuente de energía auxiliar. La entrada solo controla la salida. En los sensores generadores o pasivos, en cambio, la energía de salida es suministrada por la entrada.

Los sensores moduladores requieren en general más hilos que los generadores, ya que la energía de alimentación suele suministrarse mediante hilos distintos a los empleados para la señal. Además, esta presencia de energía auxiliar puede crear un peligro de explosiones en algunos ambientes. Por contra, su sensibilidad se puede modificar a través de la señal de alimentación, lo que no permiten los sensores generadores. La designación de activos y pasivos se emplea en algunos textos con significado opuesto al que se ha dado, por lo que aquí no se hará uso de ella para evitar confusiones.

### **1.3.3 Sensores analógicos y digitales**

En los sensores analógicos la salida varía, a nivel macroscópico, de forma continua. La información está en la amplitud, si bien se suelen incluir en este grupo los sensores



con salida en el dominio temporal. Si es en forma de frecuencia, se denominan, a veces, “casidigitales”, por la facilidad con que se puede convertir en una salida digital.

En los sensores digitales, la salida varía en forma de saltos o pasos discretos. No requieren conversión A/D y la transmisión de su salida es más fácil. Tienen también mayor fidelidad y mayor fiabilidad, y muchas veces mayor exactitud, pero lamentablemente no hay modelos digitales para muchas de las magnitudes físicas de mayor interés.

### 1.3.4 Sensores por deflexión o comparación

En los sensores que funcionan por deflexión, la magnitud medida produce algún efecto físico, que engendra algún efecto similar, pero opuesto, en alguna parte del instrumento, y que está relacionado con alguna variable útil. Un dinamómetro para la medida de fuerzas es un sensor de este tipo en el que la fuerza aplicada deforma un muelle hasta que la fuerza de recuperación de éste, proporcional a su longitud, iguala la fuerza aplicada.

En los sensores que funcionan por comparación, se intenta mantener nula la deflexión mediante la aplicación de un efecto bien conocido, opuesto al generado por la magnitud a medir. Hay un detector del desequilibrio y un medio para restablecerlo. En una balanza manual, por ejemplo, la colocación de una masa en un platillo provoca un desequilibrio, indicado por una aguja sobre una escala. El operario coloca entonces una o varias masas en el otro platillo hasta alcanzar el equilibrio, que se juzga por la posición de la aguja.

<i>Criterio</i>	<i>Clases</i>	<i>Ejemplos</i>
<b>Aporte de energía</b>	<b>Moduladores Generadores</b>	<b>Termistor Termopar</b>
<b>Señal de salida</b>	<b>Analógicos Digitales</b>	<b>Potenciómetro Codificador de posición</b>
<b>Modo de operación</b>	<b>De deflexión De comparación</b>	<b>Acelerómetro de deflexión Servoacelerómetro</b>

Figura 1.8 Clasificación de los sensores

### 1.3.5 Sensores ultrasónicos

Los sensores ultrasónicos son un tipo de sensor de proximidad sin contacto, el documento hecho por Torres, F., y Jara, C. A. (2011) menciona su funcionamiento como:

- Emiten sonido en el rango inaudible a cualquier frecuencia.
- Recibe el eco.

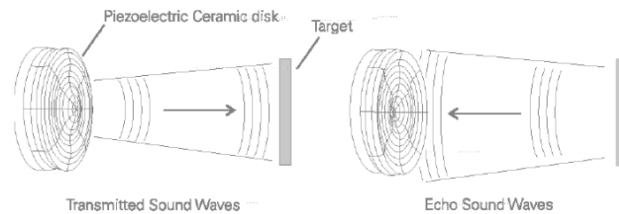


Figura 1.9 Funcionamiento del sensor ultrasónico

La velocidad con la que viajan las ondas emitidas por este sensor se limita por la máxima frecuencia de repetición de pulsos 1Hz a 25Hz

Torres, F., y Jara, C. A enlistan las principales ventajas que tienen estos sensores:

- Rango relativamente alto.
- Detección de objetos independiente del color y del material.
- Detección segura de objetos transparentes.
- Relativamente insensible a la suciedad y el polvo.
- Posibilidad de aplicaciones al aire libre.
- Posibilidad de detección sin contacto con puntos de conmutación de precisión variable.

También mencionan los inconvenientes del uso de estos sensores:

- Objetos con superficies inclinadas, el sonido se desvía.
- Reaccionan con relativa lentitud.

### **1.3.5.1 Sensor Ultrasonico JSN-SR04T**

El sensor JSN-SR04T es un sensor de distancia que utiliza ultrasonido (sonar) para determinar la distancia de un objeto en un rango de 25 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y especialmente por su resistencia al Agua.



Figura 1.10 Sensor JSN-SR04T

El sensor trabaja con ultrasonido y contiene toda la electrónica encargada de hacer la medición. El funcionamiento del sensor es el siguiente: se emite un pulso de sonido (TRIG), se mide la anchura del pulso de retorno (ECHO), se calcula la distancia a partir de las diferencias de tiempos entre el Trig y Echo. El funcionamiento no se ve afectado por la luz solar o material negro (aunque los materiales blandos acústicamente como tela o lana pueden ser difícil de detectar).

Perfecto para aplicaciones donde el sensor estará expuesto a la intemperie, utilizado en automóviles para medir distancia de colisión/parqueo.

#### **Especificaciones técnicas**

- Modelo: JSN-SR04(FZ1655)
- Voltaje de Operación: 5V DC
- Corriente de trabajo: 30mA
- Rango de detección: 25cm-450cm

- Precisión: puede variar entre los 3mm o 0.3cm
- Frecuencia de emisión acústica: 40KHz
- Duración mínima del pulso de disparo (nivel TTL): 10  $\mu$ S.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 mS.
- Ángulo de detección: menor a 50°
- A prueba de agua (parte delantera)
- Diámetro: 22mm
- Longitud: 17mm
- Temperatura de trabajo: -10°C hasta 70°C

Pines de conexión

- VCC (+5V DC)
- TRIG (*Disparo del ultrasonido*)
- ECHO (*Recepción del ultrasonido*)
- GND (0V)

## 1.4 Raspberry Pi

Raspberry Pi es una placa computadora (SBC) de bajo coste, se podría decir que es un ordenador de tamaño reducido, del orden de una tarjeta de crédito, desarrollado en el Reino Unido por la Fundación Raspberry Pi (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012.

### 1.4.1 Historia de la Raspberry Pi

En mayo de 2009, la Fundación Raspberry Pi fue fundada en Caldecote, South Cambridgeshire, Reino Unido como una asociación caritativa que es regulada por la Comisión de Caridad de Inglaterra y Gales.

Los primeros diseños de Raspberry Pi se basaban en el microcontrolador Atmel ATmega644. Sus esquemas y el diseño del circuito impreso están disponibles para su descarga pública.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS y Arch Linux; y promueve principalmente el aprendizaje del lenguaje de programación Python, y otros lenguajes como Tiny BASIC, C y Perl.

### 1.4.2 Hardware

Actualmente existen 2 modelos diferentes de Raspberry Pi. El primero, el modelo A, se diferencia del modelo B, en que tiene un sólo puerto USB, carece de controlador Ethernet, tiene 256MB de RAM por los 512MB del otro modelo y por supuesto cuesta menos que el modelo B, el cual tiene dos puertos USB y controlador Ethernet 10/100. A pesar que el Modelo A no tiene un puerto RJ45, se puede conectar a una red usando un adaptador USB-Ethernet suministrado por el usuario.



Figura 1.11 Model A



Figura 1.12 Model B

### 1.4.3 SoC (System on a Chip, Sistema en un chip) (ARM vs X86)

El procesador en el interior de su Raspberry Pi es un procesador multimedia Broadcom BCM2835 system-on-chip (SoC). Esto quiere decir que la mayor parte de

los componentes del sistema, incluidos la CPU y la GPU junto con el audio y el hardware de comunicaciones, se encuentran integrados dentro de aquel único componente oculto ubicado justo debajo del chip de la memoria de 512 MB en el centro de la placa. No es sólo el diseño del SoC lo que hace al BCM2835 diferente del procesador de un PC o portátil. Lo que lo hace también diferente es que utiliza una arquitectura de conjunto de instrucciones distinta, conocida como ARM.

ARM (Advanced RISC Machine) (Reduced Instruction Set Computer = Ordenador con Conjunto Reducido de Instrucciones).

#### **1.4.4 CPU**

La CPU Contiene un ARM1176JZFS, con unidad de coma flotante, que funciona a 700Mhz y es capaz de soportar overclock a 1GHZ en modo "TURBO" que hace que el SoC de más rendimiento sin reducir el tiempo de vida de la placa y sin perder la garantía. La CPU está basada en la versión 6 de la arquitectura ARM, la cual no es soportada por una gran cantidad de distribuciones Linux, incluyendo Ubuntu.

#### **1.4.5 GPU**

La GPU utilizada es una Dual Core VideoCore IV Multimedia Co-Processor. Es capaz de mover contenidos con calidad Bluray, usando H.264 hasta 40Mbits/s. Dispone un núcleo 3D con soporte para las librerías OpenGL ES2.0 y OpenVG. Es capaz de decodificar 1080p30.

#### **1.4.6 RAM**

La memoria RAM es de 512MB de SDRAM (en su modelo B), en un único módulo, el cual, funciona a 400Mhz en su modo normal y alcanzando los 600Mhz en su versión "TURBO".

### 1.4.7 Almacenamiento

La Raspberry Pi no tiene un disco duro tradicional, para ello dispone de un lector/ranura para memorias SD, un sistema de almacenamiento en estado sólido. El arranque del sistema se hará desde la propia tarjeta SD, con lo que debido a que tiene que albergar todo el sistema operativo, es necesario que la tarjeta sea de al menos 2 GB de capacidad para almacenar todos los archivos requeridos.

Están disponibles Tarjetas SD con el sistema operativo precargado en la tienda oficial de la Raspberry Pi, si no es así para poder arrancar el S.O. será necesario primero instalar (flashear) un sistema operativo en la tarjeta antes de poder trabajar con ella. Tras el arranque inicial de la SD se puede trabajar con almacenamiento de algún dispositivo de disco por USB.

### 1.4.8 Características técnicas

<b>PROCESADOR</b>	Broadcom BCM2837, Cortex-A53 (ARMv8) 64-bit SoC
<b>FRECUENCIA DE RELOJ</b>	1,2 GHz
<b>GPU</b>	VideoCore IV 400 MHz
<b>MEMORIA</b>	1GB LPDDR2 SDRAM
<b>CONECTIVIDAD INALÁMBRICA</b>	2.4GHz IEEE 802.11.b/g/n

	Bluetooth 4.1
<b>CONECTIVIDAD DE RED</b>	Fast Ethernet 10/100 Gbps
<b>PUERTOS</b>	GPIO 40 pines HDMI 4 x USB 2.0 CSI (cámara Raspberry Pi) Toma auriculares / vídeo compuesto Micro SD Micro USB (alimentación)

Figura 1.13 Tabla de especificaciones técnicas de la Raspberry Pi

Utilizamos una Raspberry Pi y no un Logo o un PLC debido a los elevados costos que tiene los últimos, aparte de eso, la Raspberry si puede ser utilizada como servidor web, cosa que Logo y PLC no pueden.

## 1.5 Sistemas informáticos

### 1.5.1 Base de datos

En el libro de Fundamentos de bases de datos (2002): “Se llama base de datos, o también banco de datos, a un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y/o transmisión”. Existen actualmente muchas formas de bases de datos, que van desde una biblioteca hasta los vastos conjuntos de datos de usuarios de una empresa de telecomunicaciones.



Las bases de datos son el producto de la necesidad humana de almacenar la información, es decir, de preservarla contra el tiempo y el deterioro, para poder acudir a ella posteriormente. En ese sentido, la aparición de la electrónica y la computación brindó el elemento digital indispensable para almacenar enormes cantidades de datos en espacios físicos limitados, gracias a su conversión en señales eléctricas o magnéticas.

El manejo de las bases de datos se lleva mediante sistemas de gestión (llamados DBMS por sus siglas en inglés: Database Management Systems o Sistemas de Gestión de Bases de Datos), actualmente digitales y automatizados, que permiten el almacenamiento ordenado y la rápida recuperación de la información. En esta tecnología se halla el principio mismo de la informática.

En la conformación de una base de datos se pueden seguir diferentes modelos y paradigmas, cada uno dotado de características, ventajas y dificultades, haciendo énfasis en su estructura organizacional, su jerarquía, su capacidad de transmisión o de interrelación, etc. Esto se conoce como modelos de base de datos y permite el diseño y la implementación de algoritmos y otros mecanismos lógicos de gestión, según sea el caso específico.

#### **1.5.1.1 Tipos de bases de datos**

Existen diferentes clasificaciones de las bases de datos, atendiendo a características puntuales:

Según su variabilidad. Conforme a los procesos de recuperación y preservación de los datos, podemos hablar de:

- Bases de datos estáticas. Típicas de la inteligencia empresarial y otras áreas de análisis histórico, son bases de datos de sólo lectura, de las cuales se puede extraer información, pero no modificar la ya existente.

- Bases de datos dinámicas. Aparte de las operaciones básicas de consulta, estas bases de datos manejan procesos de actualización, reorganización, añadidura y borrado de información.

Según su contenido. De acuerdo a la naturaleza de la información contenida, pueden ser:

- Bibliográficas. Contienen diverso material de lectura (libros, revistas, etc.) ordenado a partir de información clave como son los datos del autor, del editor, del año de aparición, del área temática o del título del libro, entre otras muchas posibilidades.
- De texto completo. Se manejan con textos históricos o documentales, cuya preservación debe ser a todo nivel y se consideran fuentes primarias.
- Directorios. Listados enormes de datos personalizados o de direcciones de correo electrónico, números telefónicos, etc. Las empresas de servicios manejan enormes directorios clientelares, por ejemplo.
- Especializadas. Bases de datos de información hiperespecializada o técnica, pensadas a partir de las necesidades puntuales de un público determinado que consume dicha información.

### ***1.5.1.2 Ejemplos de bases de datos***

Algunos ejemplos posibles de bases de datos a lo largo de la historia son:

- Guías telefónicas. Aunque en desuso, estos voluminosos libros solían contener miles de números telefónicos asignados a hogares, empresas y particulares, para permitir al usuario dar con el que necesitaba. Eran engorrosos, pesados, pero completos.
- Archivos personales. El conjunto de los escritos de vida de un autor, investigador o intelectual a menudo son preservados en un archivo, que se

organiza en base a la preservación y reproducción de los originales, permitiendo su consulta sin poner en riesgo el documento original.

- Bibliotecas públicas. El perfecto ejemplo de bases de datos, pues contienen miles o cientos de miles de registros pertenecientes a cada título de libro disponible para su préstamo, ya sea en sala o circulante, y del que puede haber más de un mismo ejemplar en el depósito. Los bibliotecólogos se encargan de diseñar estos sistemas y velar por su funcionamiento.
- Registros de transacciones. Las operaciones realizadas con una tarjeta de crédito, así como las llamadas realizadas con un celular, u otro tipo de transacciones comerciales cotidianas, generan todas un conjunto de registros que van a dar a una base de datos de la empresa.
- Historial médico. Cada vez que acudimos al doctor o a un hospital, se actualiza la información respecto a nuestra salud, al tratamiento recibido y demás detalles médicos en un archivo que lleva registro de nuestra historia médica, en caso de que a futuro se requiera conocer datos específicos, como operaciones o tratamientos recibidos.

### **1.5.2 MySQL**

MySQL es una base de datos relacional que utiliza el lenguaje SQL. Se trata de un SGBD (Sistema Gestor de Base de Datos) de código abierto, lanzado en 1995, que más tarde fue adquirido por Sun Microsystems en 2008.

Angel Árias, (2014) dice en su libro que SQL es el nombre de un lenguaje desarrollado para la formulación de búsquedas en bases de datos. Surgió a principios de los años 70, y se basa en el idioma Inglés para nombrar sus comandos, y actualmente está estandarizado por el lenguaje ISO (International Organization for Standardization) y ANSI (Instituto Nacional Estadounidense de Estándares). Actualmente, la mayoría de SGBD relacional y otros formatos tratan de seguir el estándar SQL para formalizar sus consultas, aunque cada empresa que usa el lenguaje lo adapta a sus necesidades.

### **1.5.2.1 SQL, la base de las aplicaciones de MySQL**

En el libro escrito por Helma Spona, (2010) se dice que SQL es la abreviatura de Structured Query Language. A pesar de lo que pueda hacer sospechar su nombre, SQL puede hacer muchas cosas más que realizar consultas en bases de datos.

Con él también se puede:

- Modificar, borrar y agregar tablas en una base de datos.
- Crear bases de datos.
- Crear índices y claves.
- Borrar y agregar datos.
- Calcular valores.
- Agrupar datos.

La norma SQL no se ha impuesto de un modo tan uniforme que se pueda manipular cualquier sistema de gestión de bases de datos soportado por SQL con cualquier aplicación SQL. Existen diferentes dialectos de SQL, que presentan determinadas diferencias. Por Tanto, los comandos no pueden usarse sin más para otros servidores de base de datos o sistemas de gestión de bases de datos. La mayor parte de las instrucciones de SQL son compatibles.

### **1.5.2.2 Servidor de base de datos MySQL**

Es un sistema de gestión de bases de datos relacional, que fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. Las principales características de MySQL son:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.

- Flexible sistema de contraseñas (password) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas (Enríquez, 2011, p. 1-2).

Utilizamos MySQL debido a que es el que más dominamos.

### **1.5.2.3 Phpmyadmin**

Es una herramienta para la administración del servidor de base de datos MySQL, que dispone de una interfaz gráfica y es de libre distribución.

Además, permite realizar todo tipo de operaciones sobre bases de datos como:

- Crear, borrar y modificar tablas.
- Consultar, insertar, modificar y eliminar datos.
- Definir usuarios y asignar permisos.
- Realizar copias de seguridad.
- Puede administrar bases locales y remotas.

Adicionalmente PhpMyAdmin está escrita en PHP y se ejecuta desde cualquier navegador web disponible para acceder a su configuración.

### **1.5.3 Servidores**

Un servidor, es un equipo informático que está al servicio de otras máquinas, ordenadores o personas llamadas clientes y que le suministran a éstos, todo tipo de información. Entre los equipos clientes pueden ser personas u otros dispositivos móviles, impresoras, etc. (Sierra, 2013)

Un servidor en informática será un ordenador u otro tipo de dispositivo que suministra una información requerida por unos clientes.

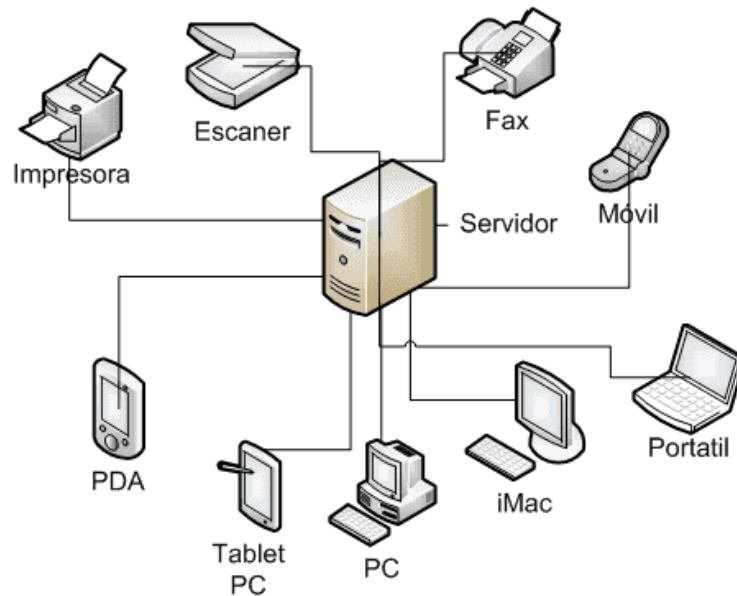


Figura 1.14 Esquema Cliente-Servidor

### 1.5.3.1 Terminos

Términos utilizados habitualmente cuando se habla de servidores, Sierra, (2013):

Proxy: Es un programa u ordenador que hace de intermediario entre dos ordenadores.

DNS: Son las siglas de Domain Name System. Es un sistema por el que se asocia una información con un nombre de dominio. Como cuando introducimos una URL en nuestro navegador, esa información es enviada a un servidor DNS que lo que hace es determinar en qué lugar se encuentra esa página web alojada y nos conecta con ella.

WEB: El término web va asociado a internet, donde los usuarios utilizan sus navegadores web para visitar sitios web, que básicamente se componen de páginas web donde los usuarios pueden acceder a informaciones con texto, videos, imágenes, etc y navegan a través de enlaces o hipervínculos a otras webs.

FTP: Acrónimo de File Transfer Protocol o Protocolo de transferencia de archivos. Es un protocolo utilizado para la transferencia de archivos entre un cliente y un servidor, permitiendo al cliente descargar el archivo desde el servidor o al servidor recibir un archivo enviado desde un cliente. Por defecto FTP no lleva ningún tipo de encriptación

permitiendo la máxima velocidad en la transferencia de los archivos, pero puede presentar problemas de seguridad, por lo que muchas veces se utiliza SFTP que permite un servicio de seguridad encriptada.

**Dedicación:** Normalmente al ser los servidores equipos más potentes y por tanto más caros, se suelen compartir entre varias personas o empresas, permitiéndoles a todos tener un servicio de gran calidad y a un mínimo precio. En este caso se dice que se trata de un servidor compartido. Pero en otros casos puede haber servidores dedicados exclusivamente a una sola persona o empresa si esta puede hacer frente al gasto económico que supone. En este caso se dice que el servidor es “dedicado”.

**POP3 y SMTP:** Hay servidores especializados en correos electrónicos o e-mails. Estos utilizan los protocolos POP3 y SMTP para recibir los correos de nuestro servidor en nuestro cliente, o para enviar desde nuestro cliente un correo al servidor de otro cliente.

**DHCP y TCP/IP:** Cuando un cliente se conecta a un servidor, éste tiene que identificar a cada cliente y lo hace con una dirección IP. Esta dirección ip son 4 pares de números y es única para cada cliente. Así el protocolo TCP/IP permite que cuando nos conectamos a internet se nos asigne una dirección IP que nos identifica. Por otro lado, DHCP es un protocolo de asignación dinámica de host que permite asignar una ip dinámicamente a cada cliente cuando este se conecta con el servidor que le da acceso a internet. En cambio, los servidores al ser máquinas más potentes e importantes suelen tener una IP fija.

### 1.5.3.2 Tipos de Servidores

En esta tabla podemos ver los tipos de servidores más habituales. Sierra, (2013).

DENOMINACIÓN DEL SERVIDOR	DESCRIPCIÓN
<b>Servidor de Correo</b>	Es el servidor que almacena, envía, recibe y realiza todas las operaciones relacionadas con el e-mail de sus clientes.
<b>Servidor Proxy</b>	Es el servidor que actúa de intermediario de forma que el servidor que recibe una petición no conoce quién es el cliente que verdaderamente está detrás de esa petición.
<b>Servidor Web</b>	Almacena principalmente documentos HTML (son documentos a modo de archivos con un formato especial para la visualización de páginas web en los navegadores de los clientes), imágenes, videos, texto, presentaciones, y en general todo tipo de información. Además se encarga de enviar estas informaciones a los clientes.
<b>Servidor de Base de Datos</b>	Da servicios de almacenamiento y gestión de bases de datos a sus clientes. Una base de datos es un sistema que nos permite almacenar grandes cantidades de información. Por ejemplo, todos los datos de los clientes de un banco y sus movimientos en las cuentas.
<b>Servidores Clúster</b>	Son servidores especializados en el almacenamiento de la información teniendo grandes capacidades de almacenamiento y permitiendo evitar la pérdida de la información por problemas en otros servidores.
<b>Servidores Dedicados</b>	Como ya expresamos anteriormente, hay servidores compartidos si hay varias personas o empresas usando un mismo servidor, o dedicados que son exclusivos para una sola persona o empresa.
<b>Servidores de imágenes</b>	Recientemente también se han popularizado servidores especializados en imágenes, permitiendo alojar gran cantidad de imágenes sin consumir recursos de nuestro servidor web en almacenamiento o para almacenar fotografías personales, profesionales, etc. Algunos gratuitos pueden ser: <a href="http://www.imgur.com">www.imgur.com</a> , <a href="http://www.photobucket.com">www.photobucket.com</a> , <a href="http://www.flickr.com">www.flickr.com</a> de Yahoo, o <a href="http://picasaweb.google.com">picasaweb.google.com</a> de Google.

Figura 1.15 Tipos de servidores, Sierra (2013).

La razón por la cual utilizamos un servidor web, es debido a que estamos usando un localhost para almacenar nuestra base de datos, además de ser el tipo de servidor que más dominamos.



### **1.5.3.3 Servidor web**

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente es renderizado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI.

### **1.5.3.4 Arquitectura**

Un servidor web opera mediante el protocolo HTTP, de la capa de aplicación del Modelo OSI. Al protocolo HTTP se le asigna habitualmente el puerto TCP 80. Las peticiones al servidor suelen realizarse mediante HTTP utilizando el método de petición GET, en el que el recurso se solicita a través de la url al servidor Web.

```
GET /index.html HTTP/1.1 HOST: www.host.com
```

En la barra de URL de un navegador cualquiera, la petición anterior sería análoga a la siguiente dirección Web:

```
www.host.com/index.html
```

### **1.5.3.5 Funcionamiento**

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, al teclear `www.google.com` en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y

objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Estas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.
- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

El hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un lenguaje de marcas y HTTP es un "protocolo".

#### **1.5.3.6 Servidor Web Apache**

Es un software de código abierto, seguro y robusto usado por la mayoría de Sistemas Operativos y es implementado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web (www), a través de un servidor HTTP gratuito. Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable (Torres, 2008, p. 20).

## **1.6 Aplicación móvil**

La aplicación móvil es aquel software que utiliza en un dispositivo móvil como herramienta de comunicación, gestión, venta de servicios-productos orientados a proporcionar al usuario las necesidades que demande de forma automática e interactiva.

La app al cubrir una necesidad de forma inmediata, independiente y personalizada, es el soporte más directo para articular estrategias de marketing, usando unos criterios de segmentación más definidos.

### **1.6.1 Usabilidad de una aplicación**

La usabilidad en general tiene que ver con la forma en que se usa algún elemento (herramienta, dispositivo electrónico, etc.), es la facilidad con que se usa y si permite hacer lo que se necesita. Particularmente la usabilidad de una aplicación de software se refiere a la facilidad con que los usuarios pueden utilizar la misma para alcanzar un objetivo concreto. Este nivel de usabilidad no puede medirse o ser evaluado directamente, debido a que depende de diferentes factores.

Formalmente, la definición más utilizada o reconocida de usabilidad es la que se expone en la norma ISO 9241-113, en la cual usabilidad se describe como el grado con el que un producto puede ser usado por usuarios específicos para alcanzar objetivos específicos con efectividad, eficiencia y satisfacción, en un contexto de uso específico. La norma define cómo especificar y medir la usabilidad de productos y aquellos factores que tienen un efecto en la misma; también destaca que la usabilidad en terminales con pantalla de visualización es dependiente del contexto de uso y que el nivel de usabilidad alcanzado dependerá de las circunstancias específicas en las que se utiliza el producto. El contexto de uso lo forman los usuarios, las tareas a realizar, el equipamiento (hardware, software y materiales), así como también los entornos físicos y sociales que pueden influir en la facilidad de uso de un producto.

Para poder especificar o medir la usabilidad, es necesario descomponer los atributos y el contexto de uso en componentes medibles y verificables. Las relaciones que existen entre el usuario, el producto, los atributos, el contexto de uso y los objetivos

que se quieren lograr se pueden observar en el framework de usabilidad propuesto en la norma citada.

### 1.6.2 Efectividad de una aplicación

Está relacionada con la precisión y completitud con la que los usuarios utilizan la aplicación para alcanzar objetivos específicos. La calidad de la solución y la tasa de errores son indicadores de efectividad.

### 1.6.3 Eficiencia de una aplicación

Es la relación entre efectividad y el esfuerzo o los recursos empleados para lograr esta. Indicadores de eficiencia incluyen el tiempo de finalización de tareas y tiempo de aprendizaje. A menor cantidad de esfuerzo o recursos, mayor eficiencia.

### 1.6.4 Satisfacción en una aplicación

Es el grado con que el usuario se siente satisfecho, con actitudes positivas, al utilizar la aplicación para alcanzar objetivos específicos. La satisfacción es un atributo subjetivo, puede ser medido utilizando escalas de calificación de actitud.

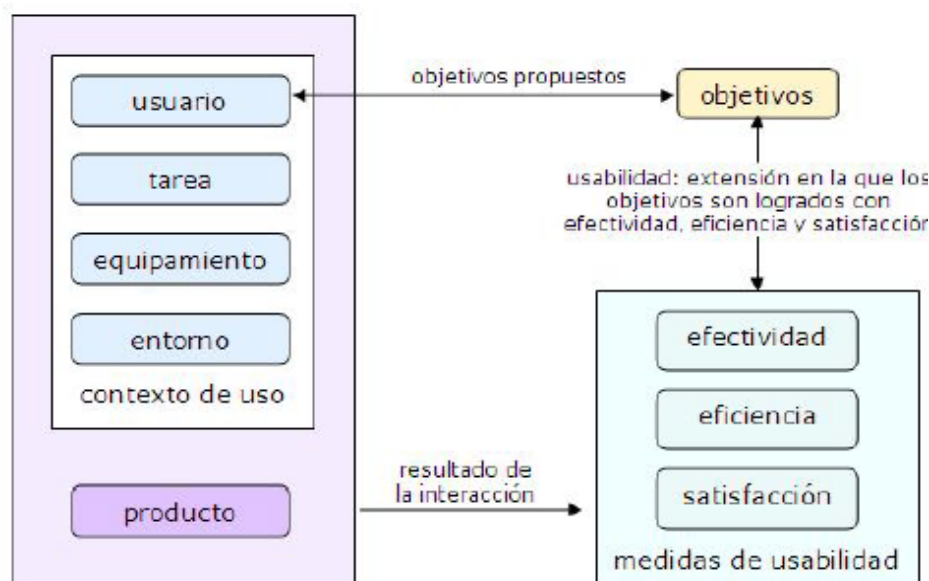


Figura 1.16 Framework de usabilidad

Fuente: Lic. Enríquez Juan Gabriel, Dra. Casas Sandra Isabel, Pág. 4.

### **1.6.5 Adquisición de datos**

La adquisición de dato es un proceso en el cual mide parámetros físicos reales. Dicha información se ingresa a la computadora para su posterior análisis y procesamiento con el fin de obtener una salida de información; la cual puede ser almacenada, desplegada en pantalla o enviada a un sitio remoto utilizando algún método de transmisión inalámbrica.

Para realizar el proceso de adquisición de datos se requiere el uso de un sensor o transductor, el cual proporciona señales eléctricas proporcionales a la magnitud física de la variable a medir. Dicho sensor se puede conectar a la computadora utilizando tarjetas electrónicas para adquirir señales, o bien, puede provenir de sitios remotos si la adquisición se realiza mediante dispositivos de conexión inalámbrica tales como Bluetooth, WiFi o ZigBee. Además, también se pueden tener señales que provengan de puertos o buses de comunicación asociados a la computadora como son el puerto serial, el puerto USB o interfaces PCI-X. Adicionalmente se puede hacer uso de interfaces industriales como GPIB (General Purpose Interfaz Bus) o PXI (Compact PCI Extensions).

### **1.7 Energías renovables**

Las energías renovables son todas las formas de energía alternativas a los combustibles fósiles, que son a saber, capaz de producir electricidad mediante la explotación de fuentes de energía limpia, sostenible y que se renuevan con el tiempo. Las energías renovables son aquellas producidas a partir de fuentes naturales no sujetas a agotamiento, como el sol, el viento, las olas y las mareas, el poder del agua y el calor de la tierra. Desde aquí toman el nombre de energía solar, energía eólica, energía hidroeléctrica y energía geotérmica.

Frente a los efectos contaminantes de combustibles fósiles como el petróleo o el carbón, las energías renovables tienen menos emisiones de carbono, reciclan y son más respetuosas con el medio ambiente.

Los combustibles fósiles crean emisiones de gases efecto invernadero que contribuyen al calentamiento global. Las energías renovables no emiten estos gases y son básicas para frenar el calentamiento global y el cambio climático.

La producción de energías renovables a nivel local, reduce los costes de transportes que tienen las energías fósiles. Potenciar las energías renovables crea puestos de trabajo.

Las energías renovables por su disponibilidad estarán sujetas a menos fluctuaciones de precios, al contrario que el petróleo o el gas.

Potencial ilimitado, frente a los recursos finitos de las energías fósiles, las energías renovables ofrecen un potencial prácticamente ilimitado.

## **1.7.1 Tipos de energías renovables**

### **1.7.1.1 Energía solar**

La energía solar transforma los rayos del sol en electricidad. Lo hace de forma directa usando energía fotovoltaica, o de forma indirecta a través de energía solar concentrada.

Los sistemas de energía solar concentrada usan lentes o paneles solares que acumulan la energía del sol. La energía fotovoltaica usa los paneles solares y materiales semiconductores, de esta forma convierte la luz solar en energía eléctrica mediante el efecto fotoeléctrico.

### **1.7.1.2 Energía solar térmica**

La energía solar térmica, aprovecha la energía del sol para generar calor o energía térmica. La energía se recoge mediante paneles solares o colectores solares se concentra la energía y se usa para calentar el agua a nivel doméstico o industrial.

### **1.7.1.3 Energía eólica**

Es la fuente de energía generada por el viento, por el movimiento continuo de las masas de aire. Se puede transformar directamente en energía mecánica (por ejemplo, molinos de viento) e indirectamente en electricidad (por ejemplo, turbinas eólicas). La energía eólica es una de las fuentes de energía renovables más antiguas. Durante

milenios ha sido la única fuente de energía para la navegación marítima a largas distancias.

La fuerza del viento se transforma en electricidad mediante turbinas de viento. Los parques eólicos pueden tener cientos de turbinas eólicas. El viento da vueltas en las láminas de las turbinas que giran, están conectadas a un generador que produce electricidad.

#### **1.7.1.4 Energía geotérmica**

Es la energía de la Tierra y el calor que proviene del subsuelo. La temperatura aumenta en las profundidades de la tierra. En algunas áreas, la diferencia térmica entre las capas superficial y subterránea es mayor y puede explotarse para calentar y crear una circulación natural de líquidos.

La energía que se obtiene del aprovechamiento del calor generado en el interior de la tierra. Vemos el poder de esta energía en los volcanes o los géiseres. El vapor de agua al pasar por una turbina conectada a un generador produce electricidad.

#### **1.7.1.5 Energía hidráulica**

Aprovecha la energía de la caída del agua desde cierta altura. Este tipo de energía se convierte en energía cinética. El agua a gran velocidad mueve las turbinas y a través de generadores se transforma en electricidad.

Las centrales situadas debajo de las presas o a lo largo de los ríos transforman el movimiento del agua en electricidad y son la principal fuente de energía renovable en el mundo.

#### **1.7.1.6 Biomasa**

Son recursos orgánicos (biológicos) que pueden usarse como combustibles y / o combustibles. Por ejemplo, los desechos del procesamiento agroalimentario se pueden usar como materia prima para producir energía térmica (calor) a través de la combustión. Algunas sustancias orgánicas (por ejemplo, azúcar, cereales, aceites, etc.) se pueden transformar en biocombustibles

Por otro lado, también a través de la fotosíntesis las plantas capturan energía del sol. Esta energía acumulada en maderas, cáscaras de frutos, plantas, y otros residuos orgánicos, al quemarse libera energía acumulada. Esto es la energía de la biomasa.

#### **1.7.1.7 Energía océanos**

La más conocida es la de las mareas, aunque también se trabaja en la energía de las olas y la de los gradientes de temperatura entre el fondo y superficie del océano.

La energía de las mareas aprovecha las diferencias de altura entre la altura media de los mares según la posición relativa de la tierra y la luna, a veces estas diferencias de altura pueden llegar ser de metros. Se usa un alternador para generar energía eléctrica

#### **1.7.1.8 Hidrógeno**

Un elemento muy abundante en el universo, pero no suele encontrarse en estado puro, así que para obtenerlo se necesitan de otras fuentes de energía. El hidrógeno se puede transformar en energía usando una tecnología similar a la fabricación de pilas que transforman la energía química en electricidad.

#### **1.7.1.9 Energía térmica**

La energía que se libera a través del calor, que se puede obtener del sol, la naturaleza, por rozamiento, a través de la combustión etcétera.

#### **1.7.1.10 Energía nuclear**

Se considera energía renovable cuando usa el hidrógeno en lugar del uranio en el proceso de fisión nuclear.



### 1.7.2 Ventajas de las energías renovables

González, J. A. (2009) menciona algunas de las ventajas que tiene utilizar las energías renovables en general, algunas de las mencionadas son:

- Son un recurso inagotable. Las energías renovables, como su propio nombre indica, se pueden reutilizar, es decir, no son fuentes de energía percederas, tal y como son el carbón, el petróleo o los materiales empleados para la energía nuclear. La energía que proviene del viento (eólica), del mar (hidráulica) o del sol (solar) es inagotable o, al menos, no se ven perspectivas de que se vaya a terminar. Por tanto, son fuentes de energía que siempre estarán presentes y que sobrevivirán a la explotación de otro tipo de combustibles.
- Benefician economía y finanzas. Además, producir energía a partir de fuentes renovables también es beneficioso para nuestras economías. Como hemos dicho, gracias a que dependen de fuentes inagotables, a largo plazo pueden garantizar una mayor estabilidad del suministro, reducir la dependencia de proveedores extranjeros e incrementar el desarrollo económico y el empleo.
- Usar energía de fuentes renovables también es beneficioso para nuestras finanzas: si tenemos cuidado con el ahorro de energía y el uso de energía renovable, también podemos ahorrar en nuestras facturas. Gracias al desarrollo de la tecnología, de hecho, las plantas cuestan cada vez menos y tienen un rendimiento cada vez mayor en términos de eficiencia energética.
- No contaminan. Las energías renovables ayudan a cuidar el medioambiente. La mayoría de estas energías no necesitan de una combustión, tal y como sucede en las refinerías o en plantas industriales que usan energías no renovables. Además, ayudan a preservar durante más tiempo los recursos de nuestro planeta. Se estima que si las fuentes renovables se duplicarán para el año 2030, las emisiones nocivas de contaminantes como el amoníaco, las partículas sólidas, el dióxido de azufre se reducirán drásticamente, así como los costos de salud relacionados con la contaminación .
- Frenan en el efecto invernadero. Las energías renovables ayudan a detener también el avance del efecto invernadero. Disminuyen el grado de

contaminación del aire, las emisiones contaminantes y efectos meteorológicos como la lluvia ácida. Muchas de estas energías, como la energía solar, permiten el autoabastecimiento a través de la instalación de sencillas infraestructuras.

De ahí que las eléctricas y petroleras tradicionales luchan a brazo partido para evitar el desarrollo de este tipo de energía que podría permitir a las personas ser autosuficientes y prescindir de los servicios de estas compañías.

Además, cabe destacar que las energías renovables son cada vez más necesarias en un mundo donde el desarrollo sostenido y sostenible es cada vez más importante. Esto redundaría en la existencia de cada vez más puestos de trabajo especializados relacionados con estas formas relativamente nuevas de obtener energía.

### **1.7.3 Sistema fotovoltaico autónomo**

La Energía solar fotovoltaica, un tipo de energía renovable utilizada para generar electricidad. Funciona transformando de forma directa la radiación solar en electricidad gracias a unos Paneles fotovoltaicos, formados de Celdas fotovoltaicas.

La generación de energía solar es uno de los métodos más limpios de producción de energía ideado por el hombre hasta ahora, ya que se basa en la conversión de la captación de la radiación solar y su transformación en electricidad (fotovoltaica) o en calor (térmica), convirtiéndose en un proceso comparable al mecanismo básico de las plantas para generar su energía, conocido como fotosíntesis.

Un sistema fotovoltaico autónomo o aislado (SFA) convierte la energía proveniente del sol en energía eléctrica, almacenándola en una batería para su uso posterior. Es un sistema que no requiere en una conexión a la red eléctrica, trabajando de una forma autónoma para proveer energía a los equipos. Son sistemas que se adaptan bien a lugares remotos sin la conexión a la red, donde hay un bajo consumo de energía y un buen recurso solar. En lugares determinados pueden ser la solución más apropiada para electrificar un edificio o una vivienda.

Entre las ventajas de utilizar los SFA's están:

- La radiación solar es una fuente de energía gratuita, renovable y disponible en muchos sitios.
- No hay gastos recurrentes de combustible.
- La operación y el mantenimiento son relativamente sencillos.
- Se adaptan bien a lugares remotos y pueden ser más económicos que una conexión de red.
- Se pueden ampliar a futuro.
- No emiten gases nocivos y su operación es silenciosa.

En cambio, los inconvenientes son:

- No apto para las instalaciones grandes que tienen una alta demanda de energía.
- Hay que reemplazar las baterías periódicamente.
- El suministro de energía depende de la cantidad de sol disponible.
- El coste inicial es alto.

### **1.7.3.1 SFA de corriente directa**

La electricidad en corriente continua es la que proviene de los módulos fotovoltaicos y que se almacena en las baterías. Un circuito en corriente continua se caracteriza por el flujo de energía, que no cambia de dirección, en donde los conductores se distinguen por tener una polaridad negativa y positiva.

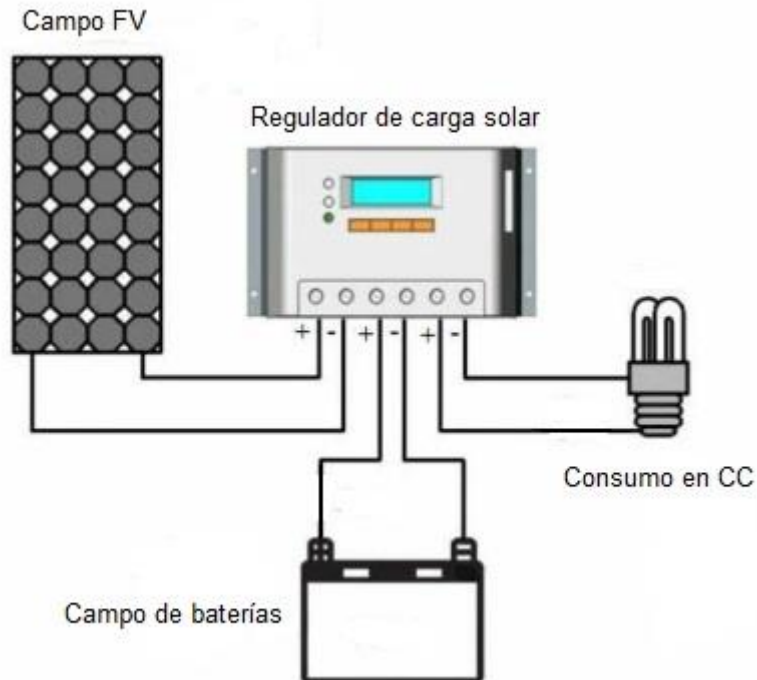


Figura 1.17 SFA en corriente continua.

### 1.7.3.2 Energía Solar

Los rayos solares son una fuente básica de energía inagotable, el 99.98% de la energía proviene del sol como energía radiante y equivale a  $173,000 \times 10^{12}$  Watts - unidad de medida de generación de energía-. De esta energía, el 77% es reflejada o devuelta al exterior. Sólo el 23% es retenido en la tierra, este porcentaje se emplea casi todo en el ciclo hidrológico -evaporación, convección, precipitación y corrientes de agua, entre otras formas-; una pequeña fracción 0.2% da lugar a olas, vientos y fenómenos de convección en la atmósfera y una fracción aún menor 0.02% es capturada y transformada por las plantas en el proceso de fotosíntesis e ingresa de esta forma al sistema trófico que sustenta la vida sobre la tierra.

Esta gran cantidad de energía puede captarse para generar ENERGÍA ELÉCTRICA, una alternativa tecnológica moderna para obtenerla, son los paneles solares formados por celdas fotovoltaicas, que transforman de manera directa la radiación solar en electricidad.

### **1.7.3.3 Panel Solar**

Las celdas fotovoltaicas son dispositivos formados por metales sensibles a la luz que desprenden electrones cuando los rayos de luz inciden sobre ellos, generando energía eléctrica. Están formados por celdas hechas a base de silicio puro con adición de impurezas de ciertos elementos químicos, siendo capaces de generar cada una de 2 a 4 Amperios, a un voltaje de 0.46 a 0.48 Voltios. Estas celdas se colocan en serie sobre paneles o módulos solares para conseguir un voltaje adecuado a las aplicaciones eléctricas; los paneles captan la energía solar transformándola directamente en eléctrica en forma de corriente continua, que se almacena en acumuladores, para que pueda ser utilizada fuera de las horas de luz. Los módulos fotovoltaicos admiten tanto radiación directa como difusa, pudiendo generar energía eléctrica incluso en días nublados.

Una de las ventajas de esta tecnología es porque es modular, lo que permite fabricar desde pequeños paneles, útiles para los techos de las casas o hasta grandes plantas fotovoltaicas que pueden generar gran cantidad de energía. Pero también es una tecnología limpia ya que es renovable e inagotable y no contamina, no emite CO<sub>2</sub> y los gastos de mantenimiento son mínimos.

Los elementos principales de un panel solar son: Generador Solar, un conjunto de paneles fotovoltaicos que captan energía luminosa y la transforman en corriente continua a baja tensión; Acumulador: Almacena la energía producida por el generador y transforma a través de un inversor la corriente continua en corriente alterna; Regulador de carga, su función es evitar sobrecargas o descargas excesivas al acumulador, puesto que los daños podrían ser irreversibles; Inversor (opcional), se encarga de transformar la corriente continua producida por el campo fotovoltaico en corriente alterna, la cual alimentará directamente a los usuarios.

Un sistema fotovoltaico no tiene por qué constar siempre de estos elementos, pudiendo prescindir de uno o más de éstos, teniendo en cuenta el tipo y tamaño de las cargas a alimentar, además de la naturaleza de los recursos energéticos en el lugar de instalación.

### 1.7.3.4 Funcionamiento de una celda solar

“Una celda fotovoltaica es un dispositivo formado por una lámina de material semiconductor, cuyo grosor varía entre los 0.25mm y los 0.35mm, generalmente de forma cuadrada, con una superficie de aproximadamente 100cm<sup>2</sup>” Formación, E. (2007).

Cada célula fotovoltaica se compone de una delgada capa de material tipo N (negativo) y otra de mayor espesor de material tipo P (positivo).

Ambas capas separadas son eléctricamente neutras y al juntarlas se genera un campo eléctrico en la unión “P-N”.

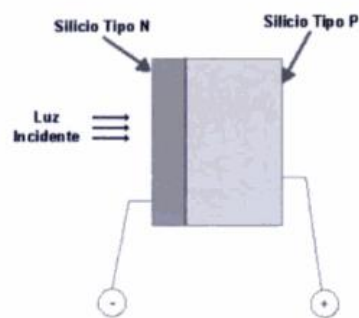


Figura 1.18 Composición de una celda fotovoltaica. Formación, E. (2007)

Cuando la luz incide sobre la celda los fotones rompen el par electrónico hueco. El campo eléctrico de la unión los separa para evitar que se recombinan, llevando los electrones a la zona N y los huecos a la zona P. Mediante un conductor externo, se conecta la capa negativa a la positiva, generando así un flujo de electrones (corriente eléctrica) de la zona P a la zona N.

La superficie de la zona N es la cara que se ilumina.

Mientras la luz siga incidiendo habrá corriente eléctrica, y su intensidad será proporcional a la cantidad de luz que reciba la celda.

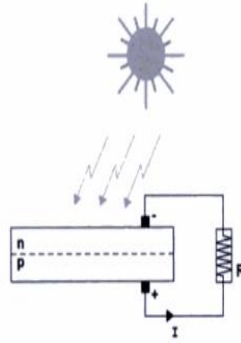


Figura 1.19 Funcionamiento de una celda fotovoltaica. Formación, E. (2007)

### 1.7.3.5 Panel Solar 10W y 12V de potencia

Las características técnicas de los paneles usados en el prototipo son las siguientes:

- Poder máximo: 10W
- Máxima corriente: 0.58A
- Máximo voltaje: 17.1V
- Corriente de corto circuito: 0.65A
- Voltaje de circuito abierto: 21.1V
- Temperatura de operación nominal: 47°C



Figura 1.19 Panel solar de 10W

### **1.7.3.6 Controladores de carga**

Los sistemas fotovoltaicos autónomos (SFA) se componen de un panel solar, una batería, una carga eléctrica y un controlador de carga. En particular, el controlador de carga tiene como objetivo proteger la batería de sub-descargas y sobrecargas manteniendo un control constante en la energía entregada por el panel a la batería, así como también de la batería hacia la carga conectada.

### **1.7.3.7 Controlador de carga solar PWM 20A 12V/24V**

El controlador / regulador de carga solar supervisa el volumen de salida de electricidad, voltios y electricidad en tiempo real de sus paneles solares, funciona con todos los sistemas que tienen un indicador EMI de potencia y se suministran con energía solar.

El controlador / regulador de carga solar ajusta automáticamente los parámetros de carga más eficientes de acuerdo con las temperaturas que lo rodean y la velocidad de potencia es fácilmente ajustable.

El controlador / regulador de carga solar presenta una gestión de período de carga de 4 etapas (EQU, Bulk, ABS y Float) y el circuito de carga mejora la eficiencia de carga y descarga y reduce el consumo de calor.

El controlador / regulador de carga tiene una pantalla LCD que muestra figuras claras por la noche y su cómodo modo de control de carga puede identificar el día y la noche para obtener números precisos.

- Tamaño: 15 cm x 7.8 cm x 3.5 cm
- Peso: 150g
- Temperatura de trabajo: -35 ° C a + 60 ° C
- Color: negro y azul
- Corriente de descarga máxima: 20A
- Voltaje máximo de salida: 12V / 24V
- Batería compatible: célula de plomo
- Carga de batería PWM



- 2 puerto de carga USB: 5V 3A



Figura 1.20 Controlador de carga solar PWM

## 1.8 Baterías de ácido-plomo

El acumulador o batería de plomo, también denominada batería de ácido-plomo es un tipo de batería (batería húmeda) muy común en vehículos convencionales, como batería de arranque, aunque también se utilizan como batería de tracción de vehículos eléctricos. Suele proporcionar una tensión de 6V, 12V u otro múltiplo de 2, ya que la tensión que suministra cada celda es de 2V. Pueden suministrar unas intensidades de corriente relativamente grandes, lo que las hacen ideales para los motores de arranque.



Figura 1.21 Ejemplo de batería ácido-plomo

### 1.8.1 Historia

En 1859, el físico e inventor francés Gaston Planté desarrolló la batería de plomo-ácido, considerada la primera batería eléctrica recargable. El primer modelo consistió en un rollo en espiral de dos láminas de plomo puro separadas por un tejido de lino, sumergido en un recipiente de vidrio que contenía una solución de ácido sulfúrico. Al año siguiente, presentó una batería de plomo de nueve células en la Academia francesa de ciencias, acompañado de un reporte técnico que tituló “Nouvelle pile secondaire d'une grande puissance” (Nueva Pila Secundaria de una Gran Potencia).<sup>8</sup>

En 1881, Camille Alphonse Faure desarrolló un modelo más eficiente y confiable que sería un gran éxito en los primeros automóviles eléctricos. Este tipo de batería sirvió en 1899 para impulsar el automóvil eléctrico, llamado La Jamais Contente, el cual fue el primero en superar la velocidad de 100 km/h. Posteriormente, este dispositivo se convirtió en la primera batería de tipo recargable que fue comercializada. Las baterías en los automóviles actuales hoy todavía trabajan esencialmente bajo el mismo principio.

### **1.8.2 Constitución**

La batería está formada por un depósito de ácido sulfúrico y dentro de él un conjunto de placas de plomo, paralelas entre sí y dispuestas alternadamente en cuanto a su polaridad (positiva (+) y negativa (-). Para evitar la combadura de las placas positivas, se dispone una placa negativa adicional, de forma que siempre haya una placa negativa exterior. Generalmente, en su fabricación, las placas positivas están recubiertas o impregnadas de dióxido de plomo ( $\text{PbO}_2$ ), y las negativas están formadas por plomo esponjoso. Este estado inicial corresponde a la batería cargada, así que el electrolito agregado inicialmente debe corresponder a la batería con carga completa (densidad 1,280 g/ml). Según el número de placas, la corriente (intensidad) suministrada será mayor o menor. Debajo de las placas se deja un espacio para que se depositen eventuales desprendimientos de los materiales que forman las placas. Para que no haya contacto eléctrico directo entre placas positivas y negativas, se disponen separadores aislantes que deben ser resistentes al ácido y permitir la libre circulación del electrolito.

### **1.8.3 Batería CSB GP 1272**

Las características técnicas y las recomendaciones de uso de la batería CSB GP 1272 son adecuadas para un uso continuo, aunque la capacidad sea reducida, si las condiciones del sol son adecuadas puede funcionar durante todo el día sin problema.

Las características son las siguientes:

- Celdas por unidad 6
- Voltaje por unidad 12 V
- Capacidad 7.2 Ah @ velocidad de 20 horas a 1.75V por celda @ 25 ° C (77 ° F)
- Peso (kg) Aprox. 2.40kg. (5.29 libras)
- Corriente máxima de descarga (A) 100A / 130A (5 seg.)
- Resistencia interna aprox. Aprox. 23mΩ
- Rango de temperatura de funcionamiento Descarga: -15 °C ~ 50 °C (5 ° F ~ 122 ° F)

- Carga: -15 °C ~ 40 °C (5 ° F ~ 104 ° F)
- Almacenamiento: -15 °C ~ 40 °C (5 ° F ~ 104 ° F)
- Rango de temperatura de funcionamiento nominal 25 ° C ± 3 ° C (77 ° F ± 5 ° F)
- Voltaje de carga flotante 13.5 a 13.8 VDC / unidad Promedio a 25 ° C (77 ° F)
- Límite máximo de corriente de carga recomendado 2.16A
- Servicio de ecualización y ciclo 14.4 a 15.0 VDC / unidad Promedio a 25 ° C (77 ° F)
- Autodescarga Debe ser más del 75% de la capacidad que antes del almacenamiento después de almacenado durante 6 meses a temperatura ambiente. 25 °C
- Terminal F1 / F2-Faston Tab187 / 250
- El material del contenedor ABS (UL94-HB) y la resistencia a la inflamabilidad de (UL94-V0) pueden estar disponibles a pedido.



Figura 1.22 Batería CSB GP 1272

## **Capítulo 2 Metodología**

## 2.1 Presentación del proyecto

El presente trabajo desarrolla un sistema de control modular para monitorear la disponibilidad de lugares, para el censado de automóviles se diseñó e implementó en PCB un circuito que conecta al sensor con la tarjeta de programación, la información que recolecta el sensor se envía a un servidor fuera de internet instalado en una Raspberry Pi 3, para terminar, los datos almacenados en el servidor son reflejados por una aplicación móvil que los presenta de manera ordenada y gráfica para el consumo del usuario final, los sensores, el servidor y la aplicación móvil están conectadas a un router por fines de demostración no se abrió el servidor a internet.

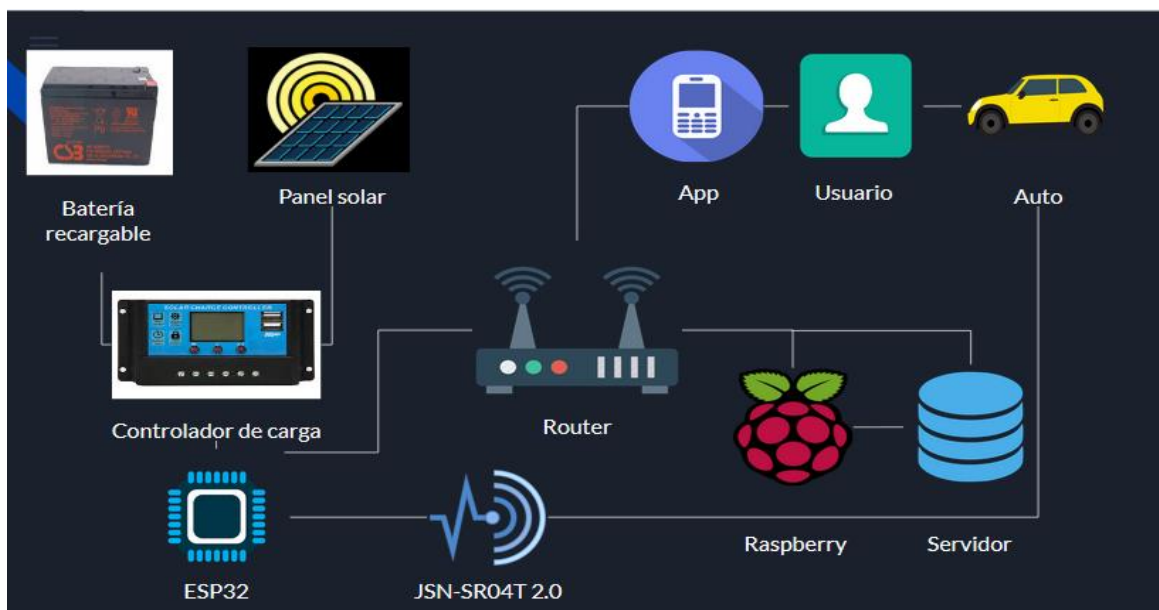


Figura 2.1 Diagrama de conexión

Para las pruebas se inició abriendo un servidor en la Raspberry Pi con sistema operativo Raspbian, después se desarrolló un programa para la conexión de los sensores al router mediante el microcontrolador ESP32, finalizando con la comunicación de la aplicación con el router y la lectura de datos.

## **2.2 Raspberry Pi 3**

### **2.2.1 Instalando la raspberry Pi**

Para iniciar con la instalación y adecuación de la raspberry pi para este proyecto se necesitó:

- Una fuente de alimentación. Una de teléfono con micro USB.
- Una tarjeta micro SD, se utilizó una de 8gb.
- Un teclado y ratón USB.
- Conexión WIFI o Ethernet.
- Un lector grabador de micro SD.
- Un cable HDMI.

#### ***2.2.1.1 Descargar el Sistema Operativo***

Es necesario instalar el sistema operativo llamado Raspbian diseñado y gestionado por la Raspberry Pi Foundation. Visitamos <https://www.raspberrypi.org/downloads/> y bajamos directa la última versión del sistema. (La Raspberry Pi Foundation recomienda descargar un paquete de posibles sistemas llamado NOOBS (New Out Of the Box Software) que incluye un sistema sencillo de instalación, y la posibilidad de arranque múltiple.)

Al querer descargar NOOBS nos ofrece la versión normal y la versión Lite, descargamos la Lite

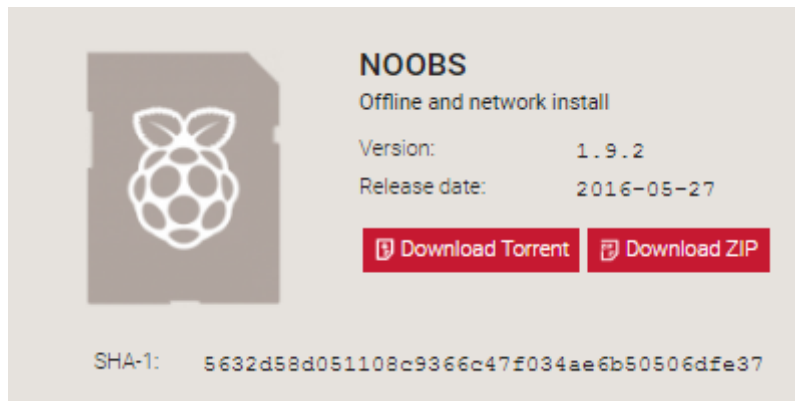


Figura 2.2 Paquete de descarga

La descarga es de 1gb, lo descargamos utilizando Torrent ya que es más rápido. }

### 2.2.1.2 Montando la tarjeta SD

Antes de utilizar la memoria SD hay que formatearla en Fat16 o Fat 32, nosotros lo hicimos con el programa SD Formatter.

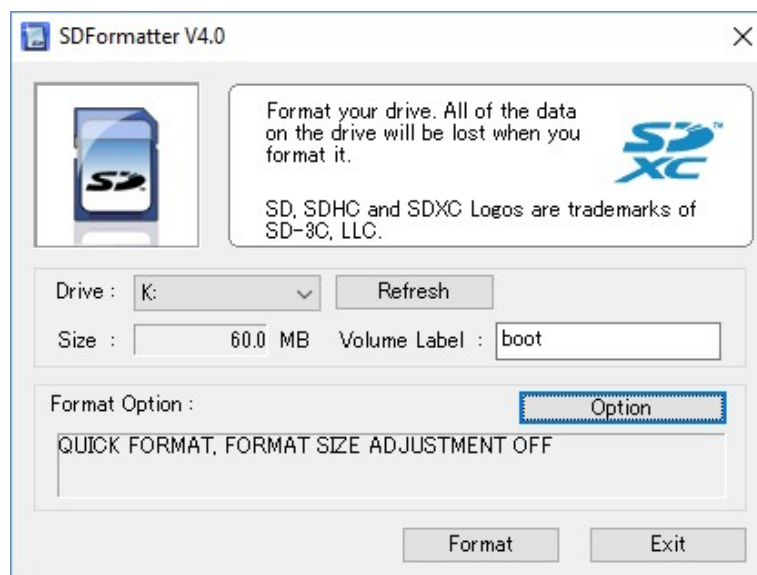


Figura 2.3 Montado de tarjeta SD

NOOBS se descargó como Zip comprimido, hay que extraerlo a una carpeta. Como nosotros utilizamos Windows el archivo se encuentra en: \\equipo\descargas

Después de esto hay que copiar el contenido de la carpeta NOOBS a la Micro SD.



Una vez con los archivos instalados en la Micro SD la retiramos de la computadora y la ponemos en nuestra tarjeta Raspberry Pi.

### 2.2.2 Arranque de Raspberry

Raspbian arranca de manera directa sin necesidad de contraseña y se ve algo así.

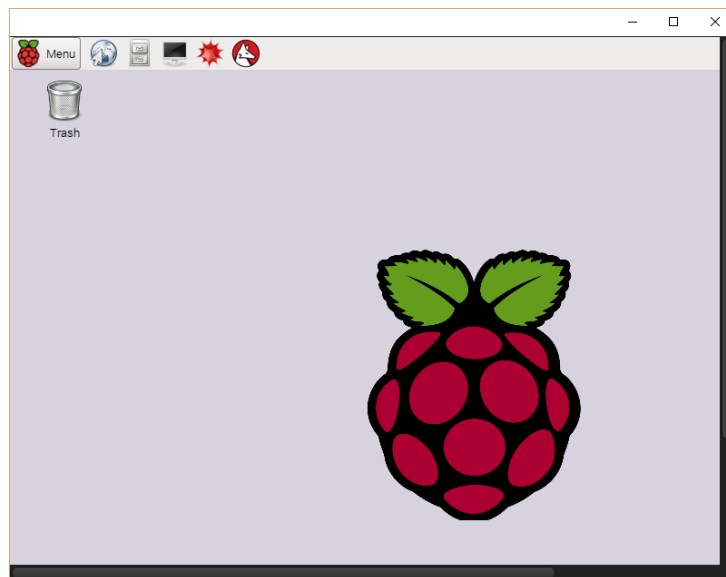


Figura 2.4 Interfaz de usuario

Linux es un sistema operativo con un nivel de seguridad fuerte y es necesario que conozca un usuario con derechos y su contraseña.

El que viene por defecto es:

- Usuario: pi
- Contraseña: raspberry

Linux distingue mayúsculas de minúsculas a diferencia de Windows.

Es necesario configurar algunos elementos del sistema como el idioma, teclado, Wifi, resolución de pantalla, etc.

Raspbian dispone de una utilidad de configuración inicial que corre en el intérprete de comandos.

Para abrir el intérprete de comandos hay un icono en la barra de programas de la parte superior:



Figura 2.5 Intérprete de comandos

Al seleccionarlo se abrirá esta ventana:

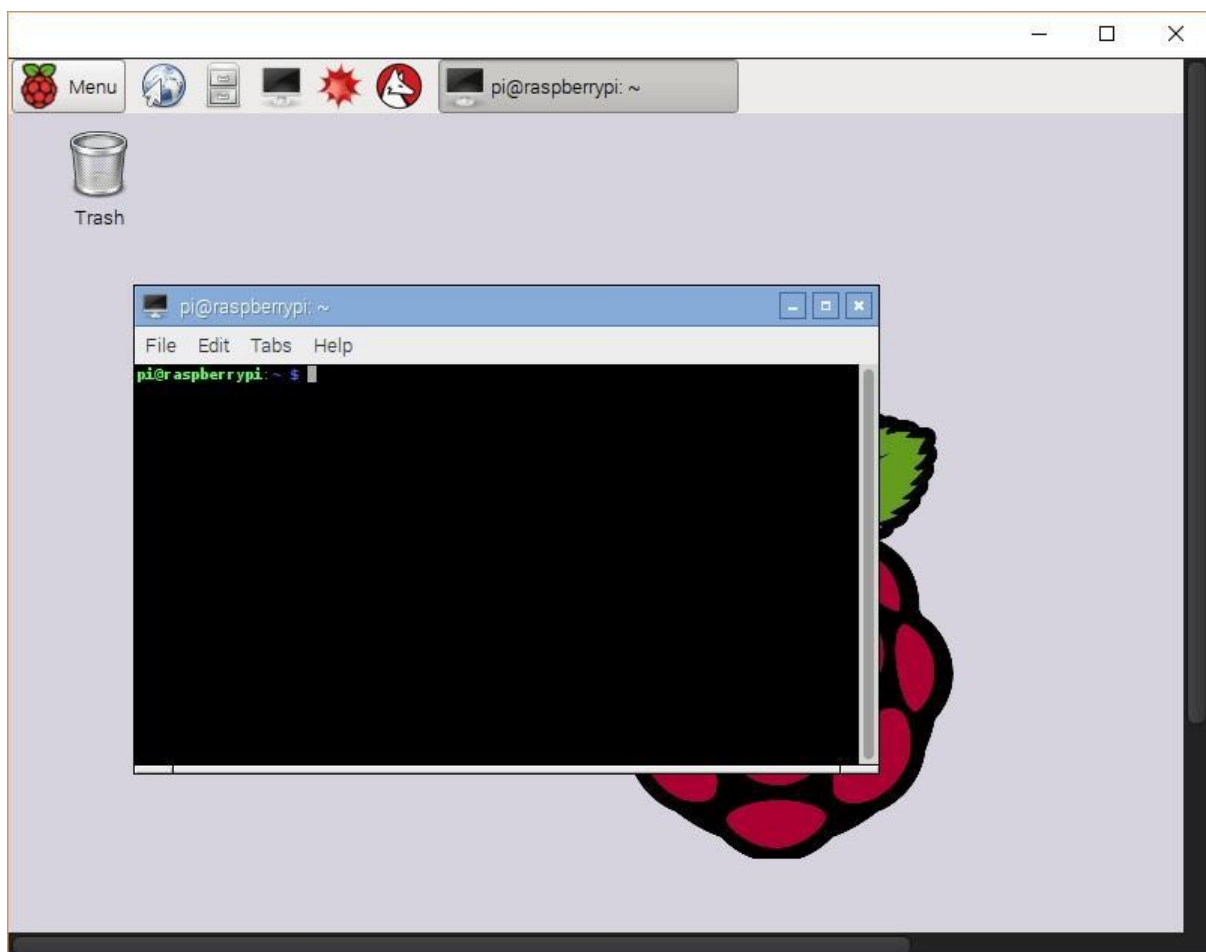


Figura 2.6 Ventana de comandos

En esta ventana ingresamos el comando:

`sudo raspi-config`.

Al ingresar el comando se abre esta ventana, aquí se encuentran todas las opciones de configuración de nuestra raspberry.

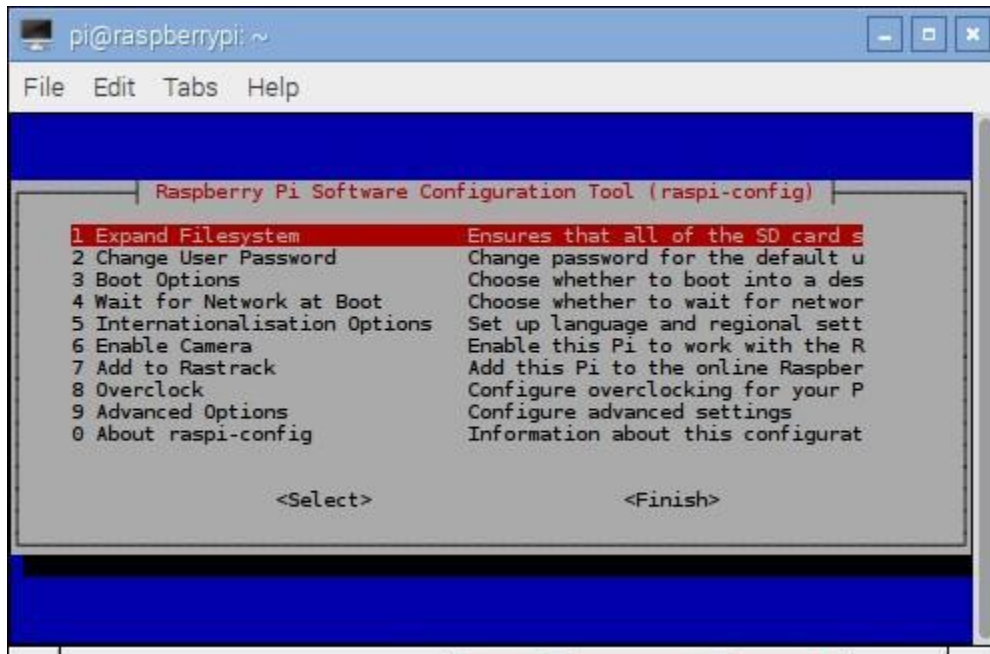


Figura 2.7 Ventana de configuración

### 2.2.2.1 Conectando la Raspberry a Internet

Una vez que ya dejamos lista nuestra configuración inicial conectamos nuestra Raspberry Pi a internet, en este caso lo hicimos por Wifi.

En la parte superior derecha de la pantalla encontramos el símbolo de Red de internet.



Figura 2.8 Barra de estado

Al seleccionarlo nos desplegará la lista de las redes disponibles, primero tuvimos que seleccionar una red abierta a internet para poder instalar y actualizar todos los programas necesarios para el proyecto.

Una vez conectados a una red debemos corroborar su estado desde la ventana de comandos. Si ingresamos en la ventana de comandos:

*ifconfig*

Nos dará la información básica de los interfaces de red y su situación.

### **2.2.2.2 Actualizando el sistema**

Para realizar las actualizaciones necesarias se debe usar una orden desde la línea de comandos.

*apt-get*

Esta requiere que se invoque mediante sudo que es un comando que te permite acceder a la memoria raíz de la Raspberry. Hay dos pasos a seguir para actualizarse y dos órdenes diferentes para ello.

La primera es crear una lista con aquello que debe ser actualizado y la orden es:

*sudo apt-get update*

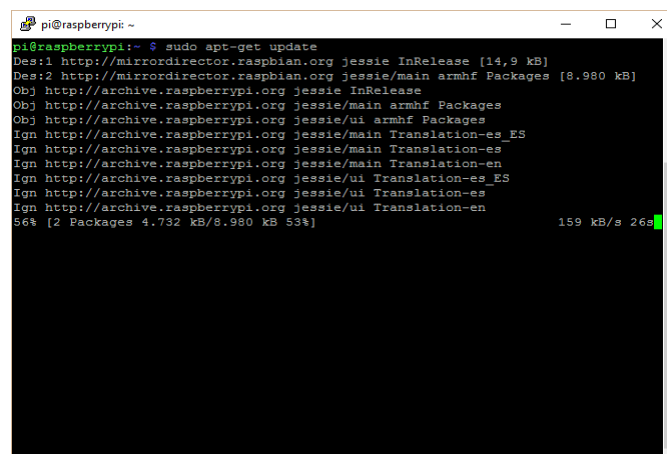
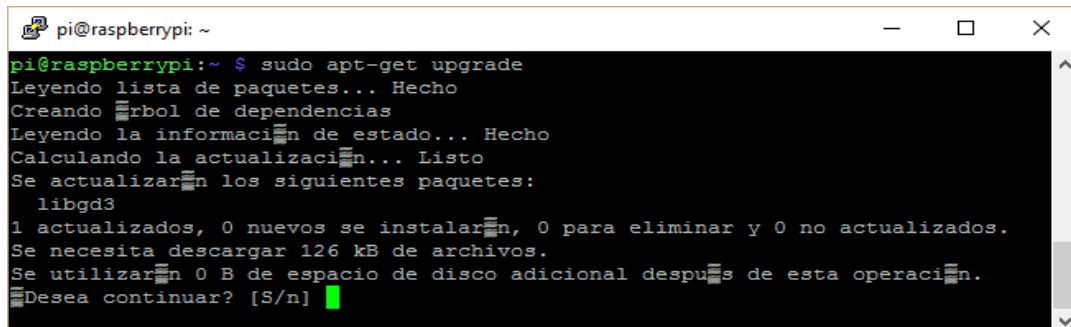


Figura 2.9 Ventana de actualización

Debido a que fue la primera vez que se hizo tardó un buen rato.

Una vez hecho esto la raspberry ya sabe que actualizar, solo es necesario ingresar el comando para que lo haga:

*sudo apt-get upgrade*



```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo apt-get upgrade  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Calculando la actualización... Listo  
Se actualizarán los siguientes paquetes:  
  libgd3  
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
Se necesita descargar 126 kB de archivos.  
Se utilizarán 0 B de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Figura 2.10 Ventana de actualización

Al pulsar intro empezará el proceso de actualizar todos los paquetes\* necesarios.

*\*Un paquete es una colección de programas y datos para ejecutar un programa dado, y que por regla general viene encapsulado en un único fichero comprimido de acuerdo a un formato propio de cada distribución.*

*Al hacer esto nuestro equipo ya está listo para instalar los programas necesarios.*

### **2.2.2.3 Configurando IP estática**

Es necesario cambiar la IP de nuestro Raspberry a estática\* ya que se utilizará de aplicación permanente como un servidor web.

*\*A través de la dirección IP podemos conectarnos con nuestro dispositivo de forma sencilla. Esta IP puede ser:*

- *Dinámica, puede cambiar cada vez que reiniciamos el dispositivo.*
- *Estática, se mantiene fija entre reinicios.*

Para configurar una IP estática en Raspbian debemos editar el fichero `/etc/dhcpd.conf` con el comando:

```
sudo nano /etc/dhcpd.conf
```

Cómo estamos conectados en red inalámbrica ingresamos el comando:

```
interface wlan0
```

Se ingresa el comando `static ip_address=` con la IP deseada y el sufijo `/24` en nuestro caso:

```
static ip_address=192.168.1.99/24
```

Después se especifica la dirección del puerto de enlace y del servidor de nombres de dominio, el router. En el siguiente comando, la dirección IPv4 192.168.1.99 se utiliza como puerto de enlace y servidor DNS.

### **2.2.3 Instalar el servidor web**

Una vez configurada la IP estática se comienza a instalar el servidor Web. El servidor a utilizar es tipo LAMP\*

\*Linux como el sistema operativo, Raspbian en nuestro caso.

Apache como servidor Web.

MySQL para gestionar la base de datos.

PHP como lenguaje de programación.

Primero instalamos Apache ingresando el comando:

```
sudo apt-get install apache2
```

Hecho esto comprobamos escribiendo nuestra IP (192.168.1.99) en el navegador:



Figura 2.11 Ventana de Apache

Luego es necesario instalar PHP utilizando el comando:

```
sudo apt-get install php7.0 libapache2-mod-php7.0
```

Al terminar la descarga es necesario reiniciar la Raspberry con el comando:

```
sudo reboot
```

Al arrancar la Raspberry se comprueba que PHP está funcionando de manera correcta creando el archivo y modificarlo de esta manera:

```
sudo nano /var/www/html/info.php
```

```
<?php
```

```
    phpinfo();
```

```
?>
```

Al abrir el navegador hay que ingresar la IP seguido de /info.php.

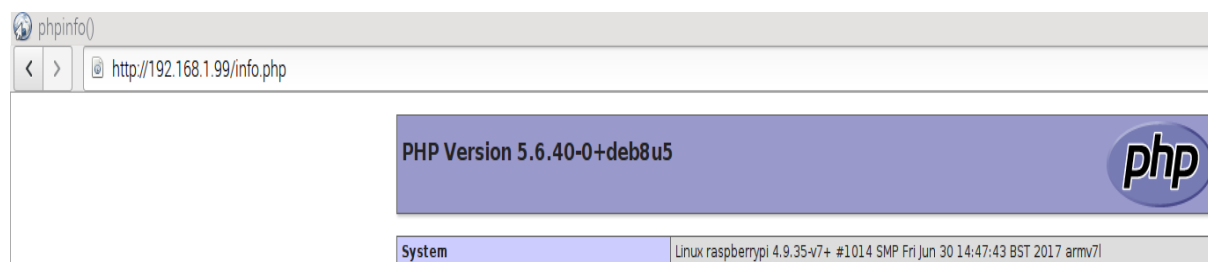


Figura 2.12 Ventana de PHP

### **2.2.3.1 Instalación y configuración de MySQL**

Instalamos MySQL ingresando el comando:

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Durante la instalación nos indicó que ingresemos una contraseña:

Contraseña: raspberry.

Al terminar la instalación reiniciamos el sistema.

Una vez hayamos arrancado de nuevo vamos a instalar PHPMyAdmin para administrar MySQL utilizando el comando:

```
sudo apt-get install php5-mysql phpmyadmin
```

Durante la instalación debemos indicar que estamos usando Apache como tipo de servidor, también indicamos que queremos configurar una base de datos y por último tendremos que ingresar la contraseña que pusimos al instalar MySQL, de igual manera debemos ingresar una contraseña para acceder a PHPMyAdmin:

Contraseña: raspberry.

Al terminar este proceso ingresamos el comando:

```
sudo nano /etc/apache2/apache2.conf
```

Editamos este fichero añadiendo al final la línea:

```
Include /etc/phpmyadmin/apache.conf
```

### **2.2.4 Red local**

Una vez instalados todos los programas necesarios cambiamos a la red WiFi local que creamos para el desarrollo del proyecto.

Red local: EISYI



Contraseña ailerom1

Debido a que nuestra red es local no se puede comprobar entrando a una página de internet así que se tuvo que corroborar utilizando el comando.

*Ifconfig*

### 2.2.4.1 PHPMYAdmin

Una vez hecha la configuración de la nueva red se ingresa en el navegador la IP de la Raspberry seguido de /phpmyadmin y nos abrirá la siguiente ventana en la cual ingresamos el usuario: root y la contraseña: raspberry

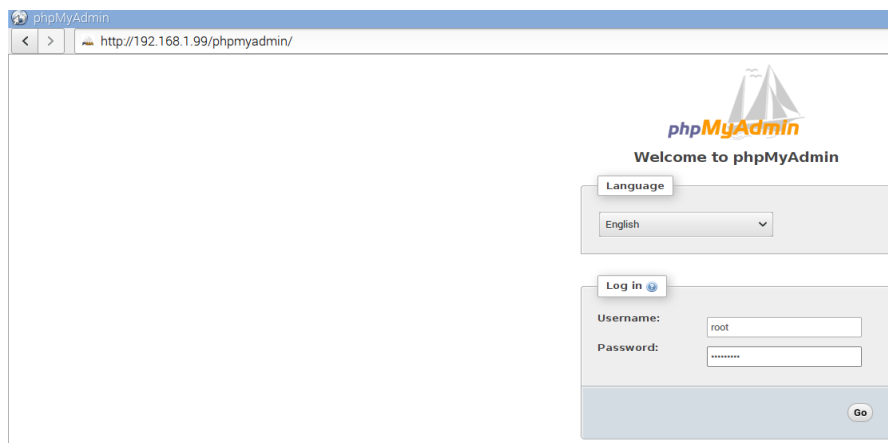


Figura 2.13 Ventana de PHPMYAdmin

Creamos la base de datos seleccionando “Nueva”. El nombre de nuestra base de datos es ESTACIONAMIENTO, en cotejamiento seleccionamos “utf8\_general\_ci” y pulsamos “Crear”.

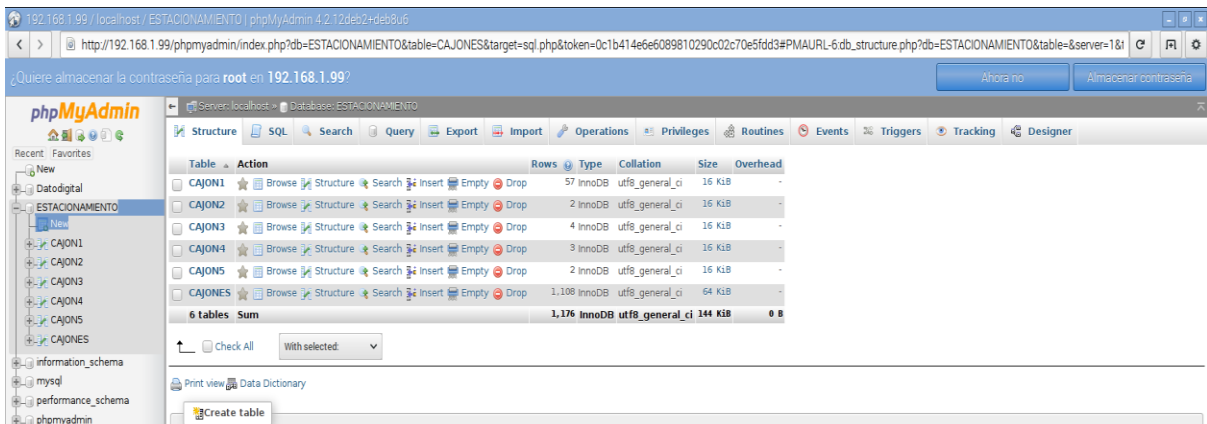


Figura 2.14 Tablas creadas en la base de datos

Creamos 4 tablas para pruebas: CAJON1, CAJON2, CAJON3 y CAJON4.

Cada tabla tiene 2 columnas:

- ID: identificador del registro, la clave primaria y marcaremos la casilla de auto incremento.
- Estado: Indicador de los cajones, disponible y ocupado.

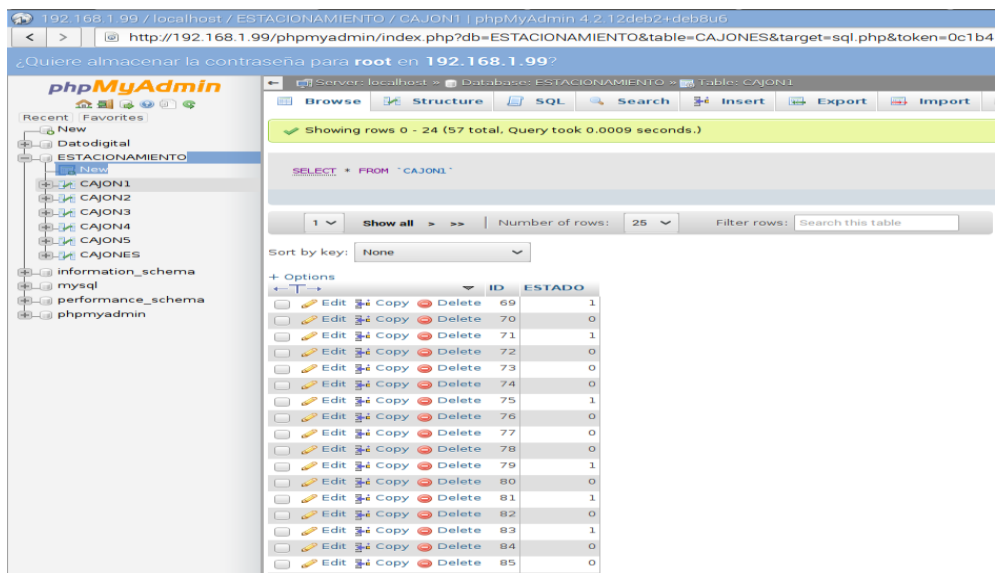


Figura 2.15 Captura de pantalla base de datos cajón 1

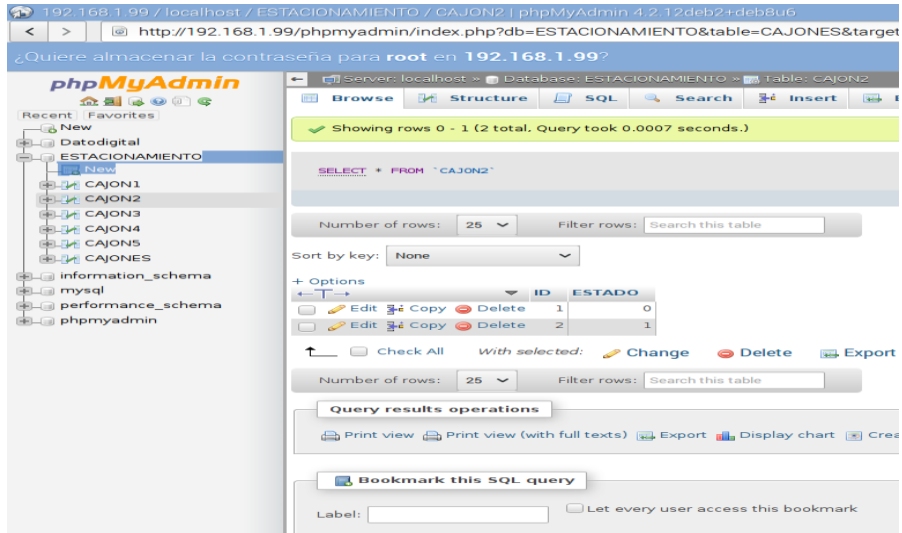


Figura 2.16 Captura de pantalla base de datos cajón 2

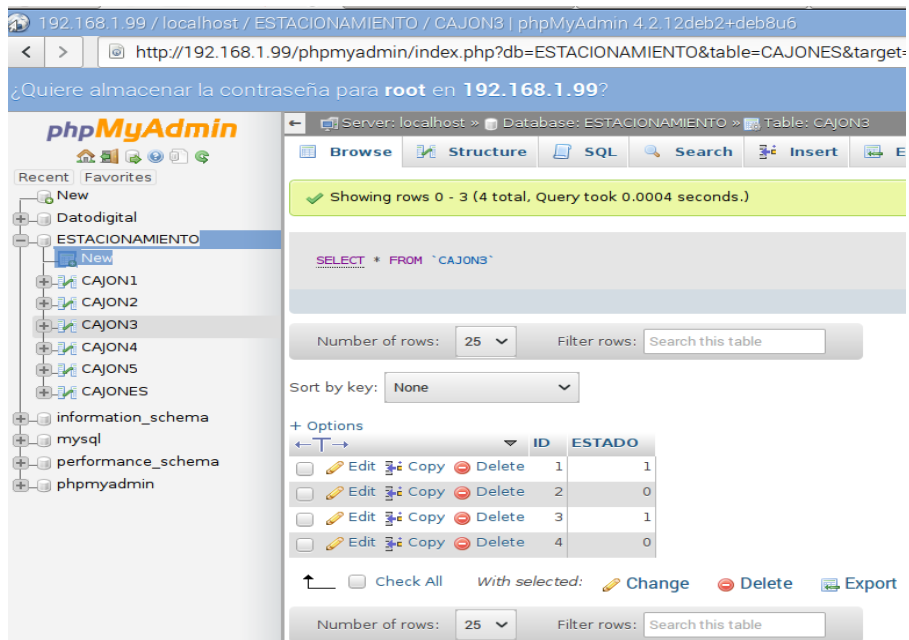


Figura 2.17 Captura de pantalla base de datos cajón 3

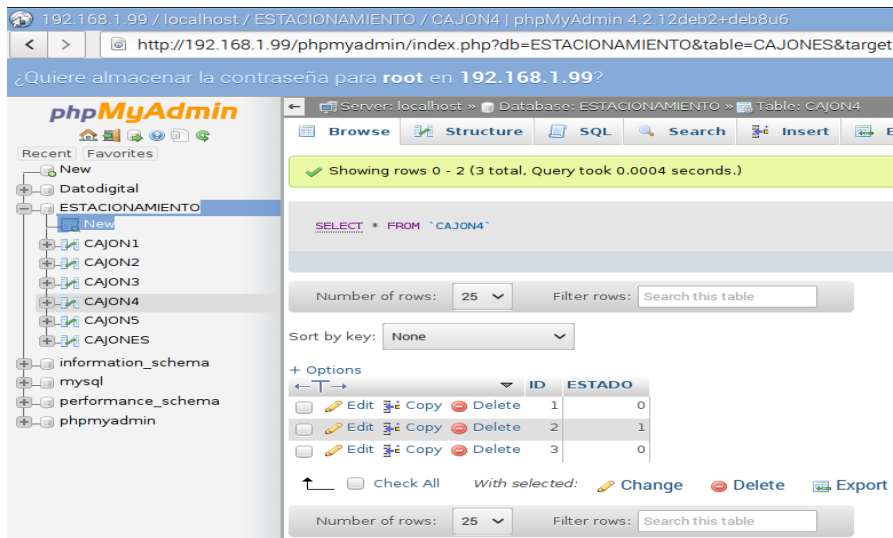


Figura 2.18 Captura de pantalla base de datos cajón 4.

También creamos la tabla principal a la cual llegarán los valores de los 4 sensores instalados, le pusimos de nombre CAJONES

Esta tabla tiene 3 columnas:

- ID: Identificador del registro, la clave primaria y marcaremos la casilla de auto incremento.
- Cajón: Indica cuál fue el cajón (1, 2, 3, 4) que tuvo cambio de estado.
- Estado: Indicador de los cajones, disponible y ocupado.

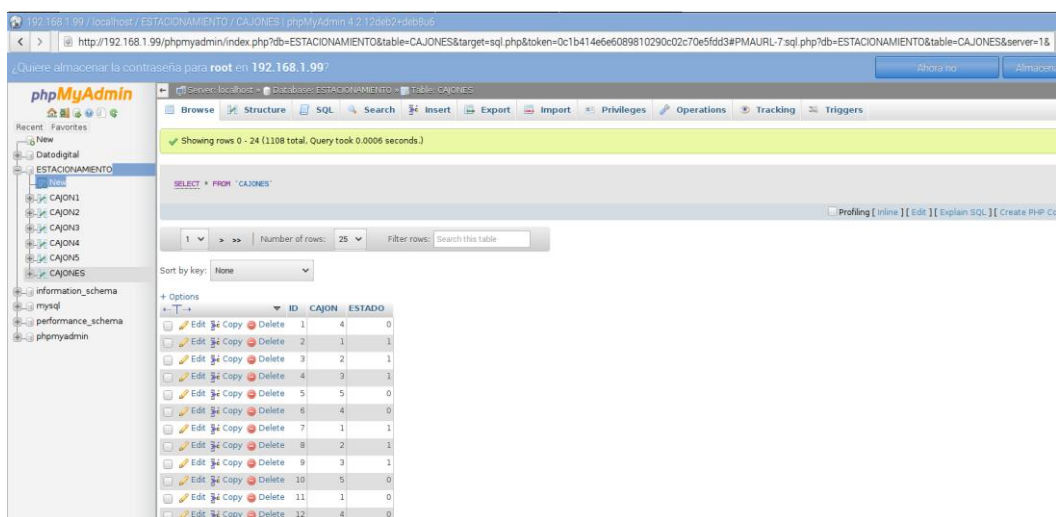


Figura 2.19 Captura de pantalla cajones

### 2.2.4.2 Crear la página para subir valores a la base de datos

Es necesario crear una página con PHP para poder subir datos al servidor. Para ello creamos el archivo config.php el cual tendrá la información para establecer la conexión con la base de datos.

Para crearlo se ingresa el comando:

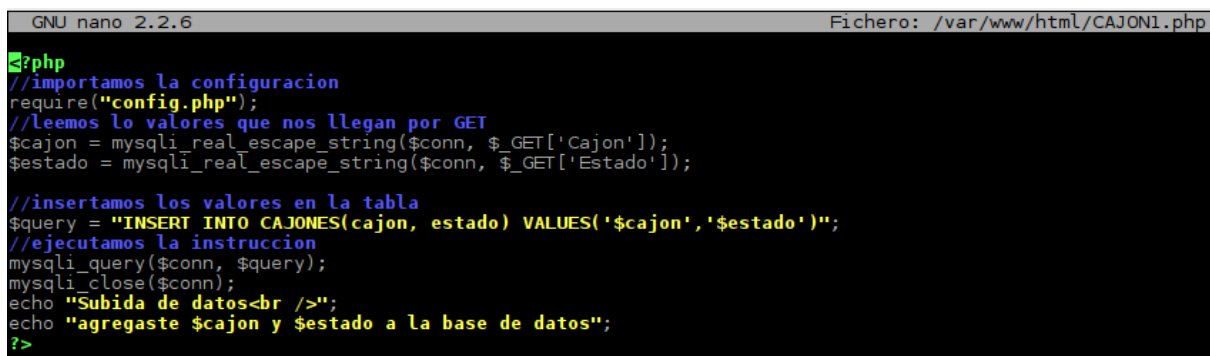
```
sudo nano /var/www/html/config.php
```

Colocamos el host, usuario, contraseña y el nombre de la base de datos:

```
<?php
// config.php
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "Raspberry";
$dbname = "ESTACIONAMIENTO";
//Conexion con la base de datos
$con = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
?>
```

Figura 2.20 Config.php

Creamos los códigos PHP de los cajones (1, 2, 3, 4), primero es necesario llamar el archivo config.php que hicimos para la conexión, la tarea de estos códigos es la de subir los datos.



```
GNU nano 2.2.6 Fichero: /var/www/html/CAJON1.php
<?php
//importamos la configuracion
require("config.php");
//leemos lo valores que nos llegan por GET
$cajon = mysqli_real_escape_string($conn, $_GET['Cajon']);
$estado = mysqli_real_escape_string($conn, $_GET['Estado']);

//insertamos los valores en la tabla
$query = "INSERT INTO CAJONES(cajon, estado) VALUES('$cajon','$estado)";
//ejecutamos la instruccion
mysqli_query($conn, $query);
mysqli_close($conn);
echo "Subida de datos<br />";
echo "agregaste $cajon y $estado a la base de datos";
?>
```

Figura 2.21 Captura de pantalla código PHP CAJON1

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: /var/www/html/CAJON2.php

?php
//importamos la configuracion
require("config.php");
//leemos lo valores que nos llegan por GET
$estado = mysqli_real_escape_string($conn, $_GET['Estado']);

//insertamos los valores en la tabla
$query = "INSERT INTO CAJON2(estado) VALUES('$estado')";
//ejecutamos la instruccion
mysqli_query($conn, $query);
mysqli_close($conn);
echo "Subida de datos<br />";
echo "agregaste $estado a la base de datos";
?>
```

Figura 2.22 Captura de pantalla código PHP CAJON2

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: /var/www/html/CAJON3.php

?php
//importamos la configuracion
require("config.php");
//leemos lo valores que nos llegan por GET
$estado = mysqli_real_escape_string($conn, $_GET['Estado']);

//insertamos los valores en la tabla
$query = "INSERT INTO CAJON3(estado) VALUES('$estado')";
//ejecutamos la instruccion
mysqli_query($conn, $query);
mysqli_close($conn);
echo "Subida de datos<br />";
echo "agregaste $estado a la base de datos";
?>
```

Figura 2.23 Captura de pantalla código PHP CAJON3

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: /var/www/html/CAJON4.php

?php
//importamos la configuracion
require("config.php");
//leemos lo valores que nos llegan por GET
$estado = mysqli_real_escape_string($conn, $_GET['Estado']);

//insertamos los valores en la tabla
$query = "INSERT INTO CAJON4(estado) VALUES('$estado')";
//ejecutamos la instruccion
mysqli_query($conn, $query);
mysqli_close($conn);
echo "Subida de datos<br />";
echo "agregaste $estado a la base de datos";
?>
```

Figura 2.24 Captura de pantalla código PHP CAJON4

Una vez hecho esto probamos los códigos de manera manual desde la URL del navegador, colocando la IP junto al código PHP al que se quiera agregar datos y el dato:

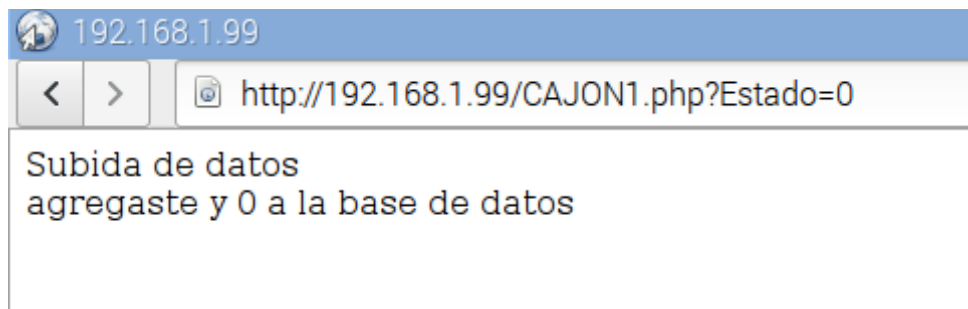


Figura 2.25 Impresión de la URL

La función del código ESTACIONAMIENTO.php (1, 2, 3) es ejecutar config.php para acceder a la base de datos y de ahí ordenar los valores de la tabla desde el último al primer registro, basándose en la variable ID, tomando el último valor subido e imprimiendo el dato.

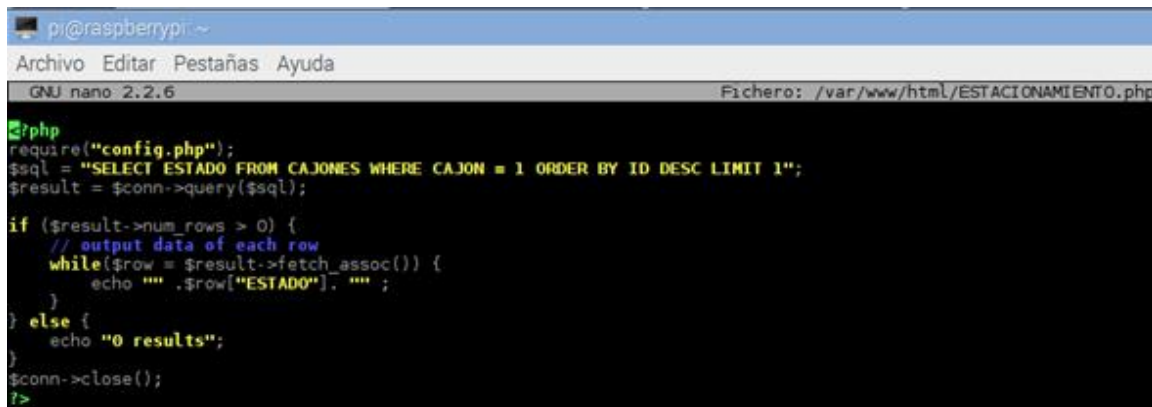
A screenshot of a terminal window showing the PHP code for ESTACIONAMIENTO.php. The code includes a require statement for config.php, a SQL query to select the last record from the CAJONES table where CAJON = 1, and a loop to output the ESTADO of that record. If no results are found, it outputs '0 results'.

Figura 2.26 Captura de pantalla código PHP ESTACIONAMIENTO

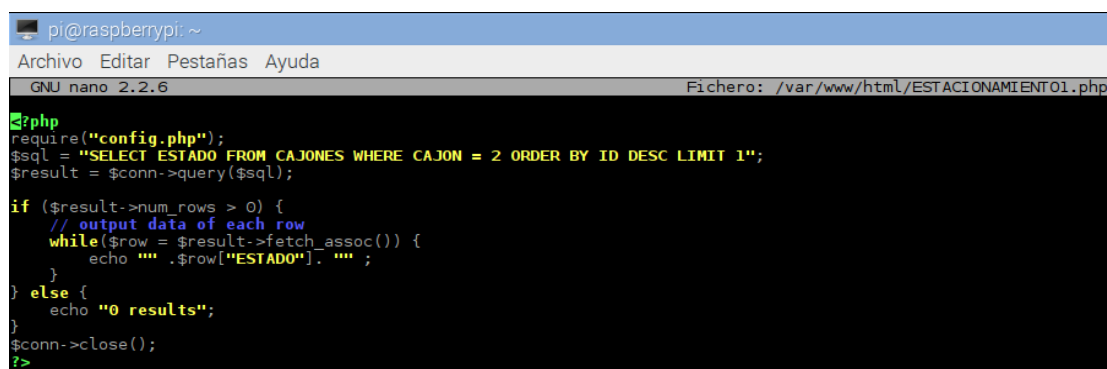
A screenshot of a terminal window showing the PHP code for ESTACIONAMIENTO01.php. The code is similar to the previous one but the SQL query filters for CAJON = 2.

Figura 2.27 Captura de pantalla código PHP ESTACIONAMIENTO01.

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: /var/www/html/ESTACIONAMIENTO2.php
<?php
require("config.php");
$sql = "SELECT ESTADO FROM CAJONES WHERE CAJON = 3 ORDER BY ID DESC LIMIT 1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "" . $row["ESTADO"] . "" ;
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```

Figura 2.28 Captura de pantalla código PHP ESTACIONAMIENTO2

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: /var/www/html/ESTACIONAMIENTO3.php
<?php
require("config.php");
$sql = "SELECT ESTADO FROM CAJONES WHERE CAJON = 4 ORDER BY ID DESC LIMIT 1";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "" . $row["ESTADO"] . "" ;
    }
} else {
    echo "0 results";
}
$conn->close();
?>
```

Figura 2.29 Captura de pantalla código PHP ESTACIONAMIENTO3

## 2.3 Programación del microcontrolador

En esta parte se programó el microcontrolador ESP8266, haciendo un código que le permita la escritura sobre una base de datos de manera inalámbrica. Se trabajó con 4 microcontroladores para evitar el cableado que supondría solo alimentar el sensor desde una base, reduciendo el cableado en un 50%, cada placa trabaja de manera modular para identificar fallas con más facilidad además de que la reparación no supone cortar el servicio de estacionamiento por completo, si no que solo se cerraría un solo cajón de estacionamiento.



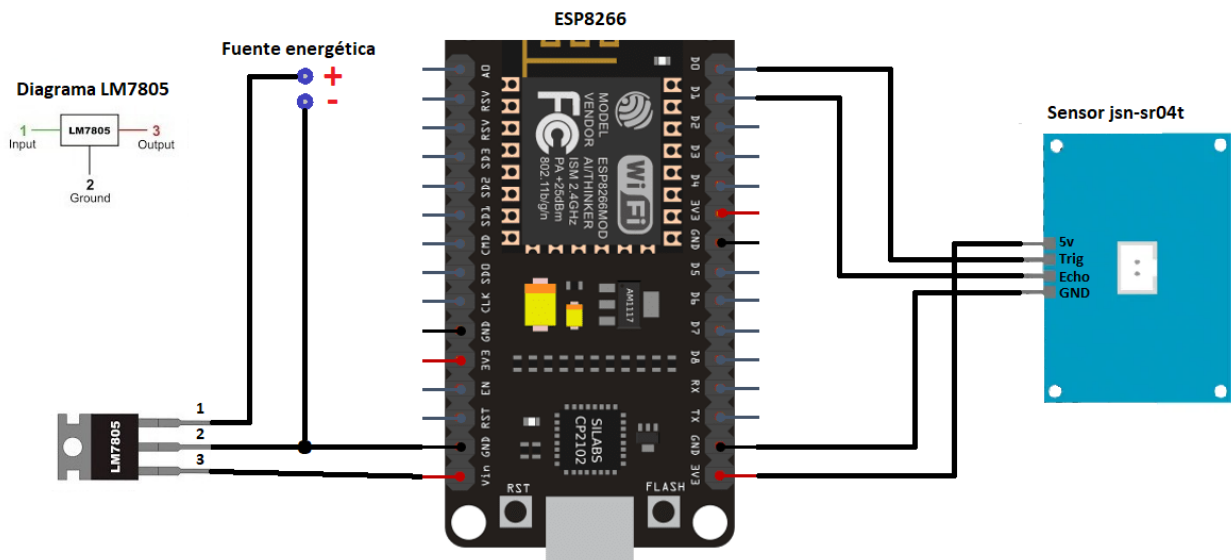


Figura 2.30 Diagrama unifilar de los sensores del tope

La baquelita a utilizar tiene un circuito impreso que permite la alimentación de la placa con un reductor de voltaje previo, la energía que requiere el sensor viene directamente de la salida que proporciona el microcontrolador.

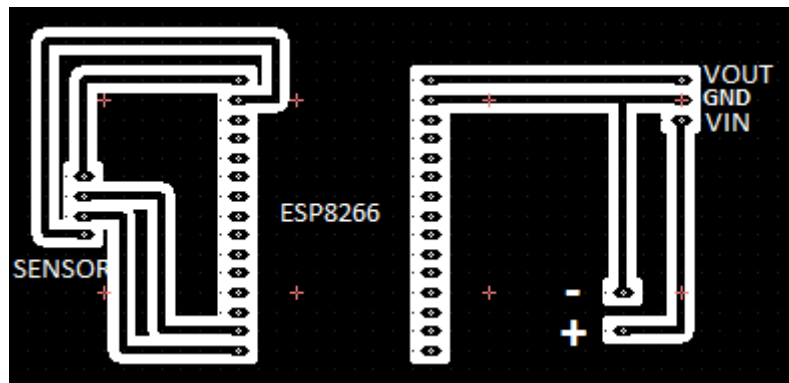


Figura 2.31 Diseño de PCB

### 2.3.1 Drivers necesarios

Para esta parte iniciamos con la instalación de la placa ESP8266 compatible con Arduino, debido a que el microcontrolador es compatible con dicho programador. Inicialmente presionamos inicio y en preferencias encontramos el apartado para agregar placas mediante URL.



Figura 2.32 Inicio Arduino

Agregado el link para la ESP8266 la agregamos a las placas compatibles con el programa con las librerías necesarias para su funcionamiento. Para ello se necesitó entrar a herramientas, en el apartado placa se seleccionó boards manager y se buscó la placa de manera manual.

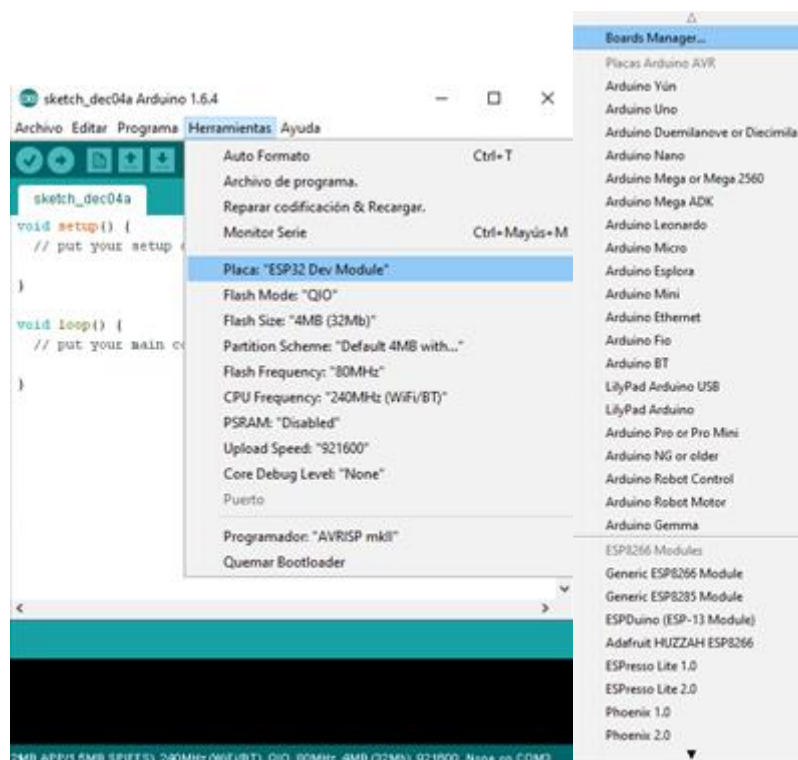


Figura 2.33 Boards Manager



Figura 2.34 ESP8266 Community

## 2.3.2 Código interno

Para el código de que tiene la placa, se necesita que sea capaz de conectarse a una red WiFi, además de conectarse con una IP fija para evitar problemas en el futuro con posibles saturaciones de red o lentitud al momento de conectarse.

Es necesario que detecte de manera correcta los coches que estén estacionados, evitando que suba información errónea por personas que puedan estar pasando y tapen el sensor por muy poco tiempo, así mismo necesitará recibir correctamente la señal de que un coche a dejado el estacionamiento, para no permitir que la aplicación muestre lugares disponibles mientras el sitio esté ocupado o viceversa.

La comunicación WiFi nos permitirá subir información como cliente a una base de datos instalada en una microcomputadora conectada a la misma red que las placas.

Para iniciar se colocó la librería <WiFi.h> que se instala al momento de agregar la placa al programa, esto permitirá habilitar las funciones de conexión inalámbrica que dispone el microcontrolador. Una vez puesta la librería, se ponen los datos de la red y la IP del servidor de donde va a subir la información, se utiliza la variable const char\* para que la variable sea solamente de lectura, y así evitar errores en la programación donde se pueda llegar a cambiar por accidente los datos de conexión.

Las variables a utilizar serán las siguientes:

- estadoanterior: Funciona para guardar cual es el último estado de disponibilidad y cambiará hasta que el estado cambie, así se evita subir información innecesaria a la base de datos.
- datoasubir: Es la variable que subirá el dato final a la base de datos. Disponible: 0 Ocupado: 1.
- contconexion: Es un contador de las veces que se intenta conectar la placa a la red wifi. Sirve para poder mandar un mensaje de error po si no llega a establecer conexión.
- contador: Comprueba si hay diferencia entre esta variable y la variable estadoanterior. Sirve para saber si un coche a llegado o dejado el lugar.
- TRIGPIN: Manda la señal que luego será rebotada si existe un coche.
- ECHOPIN: Recibe la señal enviada y determina la distancia del objeto.
- distance: Guarda en cm la distancia de los objetos censados.
- seg1: Es un contador que evitará que la placa suba información al instante. Evitando falsas lecturas. Cuenta 3s y manda si está ocupado.
- seg0: Es un contador que evitará que la placa suba información al instante. Evitando falsas lecturas. Cuenta 3s y manda si está desocupado.

```

#include <WiFi.h>

//Credenciales de WiFi

const char* ssid    = "EISYI";
const char* password = "aileron1";
const char* host    = "192.168.1.99"; //IP que se otorgo a la raspberry

//Variables

int estadoanterior=10;
int datoasubir=0;
int contconexion = 0;
int contador = 0;
int TRIGPIN=22; //Emisor de onda
int ECHOPIN=23; //Receptor de onda
int distance; //Variable de distancia
int seg1 = 0;
int seg0 = 0;

```

Figura 2.35 Variables

### 2.3.2.1 Void setup

Para el void setup se inicializó el serial para las pruebas que se hicieron antes de tener el código final, los pines digitales donde está conectado el *TRIGPIN* y *ECHOPIN* se programaron para funcionar y luego el *TRIGPIN* se asegura de que no esté encendido al momento de empezar.

```
void setup() {  
  Serial.begin(115200);  
  //Serial.println("");.....  
  
  pinMode(TRIGPIN,OUTPUT);  
  pinMode(ECHOPIN,INPUT_PULLUP);  
  
  digitalWrite(TRIGPIN,LOW);  
}
```

Figura 2.36 Void setup

Después se inicializa la conexión wifi con la sentencia `WiFi.begin` y utilizando las variables antes declaradas. En el `while` de abajo, se intenta conectar 50 veces, de no conseguirlo el serial enviará un error, se utilizó para hacer pruebas. Una vez que se ha conectado se asigna una IP fija, cada cajón de estacionamiento tendrá una IP distinta con terminación .71 a .74.

```
WiFi.begin(ssid, password);  
  
//Cuenta hasta 50 si no se puede conectar lo cancela  
while (WiFi.status() != WL_CONNECTED and contconexion <50) {  
  ++contconexion;  
  delay(500);  
  //Serial.print(".");.....  
}  
  
if (contconexion <50) {  
  //para usar con ip fija  
  IPAddress ip(192,168,1,71);  
  IPAddress gateway(192,168,1,1);  
  IPAddress subnet(255,255,255,0);  
  WiFi.config(ip, gateway, subnet);  
  //Serial.println("");.....  
  //Serial.println("WiFi conectado");.....  
  //Serial.println(WiFi.localIP());.....  
}  
else {  
  //Serial.println("");.....  
  //Serial.println("Error de conexion");.....  
}
```

Figura 2.37 Conexión al router

### 2.3.2.2 Void loop

Para que el sensor tenga mediciones se hizo una intermitencia en la señal de 20 microsegundos de lo contrario las mediciones saldrán incorrectas o simplemente serán 0. Después de recibir el valor en *ECHOPIN* se divide entre 50 para obtener los valores en cm.

```
void loop(){
  digitalWrite(TRIGPIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(20);
  digitalWrite(TRIGPIN, LOW);
  distance = pulseIn(ECHOPIN, HIGH,26000); // Lectura de rebote del pulso
  distance = distance/50; //Valor obtenido se divide entre 50
  Serial.print(distance);
  Serial.println(" cm");
  delay(500); //Se requiere un tiempo de espera para evitar saturacion de onda
}
```

Figura 2.38 Void loop

En esta parte se comprueba que se cense el coche 3 veces antes de subir el dato a la base.

```
//se asegura de que una persona no haya solo pasado por encima del sensor
if(distance<=50&&distance>0){
  delay(1000);
  seg1=seg1+1;
  seg0=0;
}else{
  delay(1000);
  seg0=seg0+1;
  seg1=0;
}

//lo convierte en un dato digital
if(seg1==3){
  delay(1000);
  contador=1;
  seg1=0;
  seg0=0;
}else if(seg0==3){
  delay(1000);
  contador=0;
  seg1=0;
  seg0=0;
}
```

Figura 2.39 Contador

Si el contador es distinto a *datoanterior* entra al if y dependiendo el valor toma el *datoasubir*.

```
//solo entra al if si el estado es diferente
if(contador!=estadoanterior){
    if(contador==1){
        datoasubir=1;
    }else{
        datoasubir=0;
    }
}
```

Figura 2.40 Comparación

Se inicializa el cliente para poder imprimir datos en una URL, la comunicación por defecto en el puerto 80, en los string se anotan los datos que son necesarios para subir información, cada placa cambia el valor del cajón, dependiendo el lugar donde esté ubicado el sensor. En la sentencia `client.print` con GET se obtienen los strings de arriba y se obtiene la variable *datoasubir*, se imprime la URL y con él ejecutamos el archivo `.php` para subir el dato a la base.

```
if(contador!=estadoanterior){
//Serial.print("connecting to ");.....
//Serial.println(host);.....

// Usamos la clase WifiClient para hacer la conexion
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
    //Serial.println("connection failed");.....
    return;
}

// Creamos la URL que se utilizara para la conexion
String url = "/CAJON1.php?";
String dato2 = "Cajon=1a";
String dato1 = "Estado=";

//Serial.print("Requesting URL: ");.....
//Serial.println(url);.....

// Enviara la peticion al servidor
client.print(String("GET ") + url + dato2 + dato1 + datoasubir + "\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
}
```

Figura 2.41 Subida del dato

En el `while` siguiente, si la inactividad supera los 5000 milisegundos, el cliente se detiene para evitar el consumo de recursos innecesarios, cuando se vuelva a subir un dato distinto en la parte de arriba volverá a leer la sentencia para inicializar el cliente,

mientras tanto el último while lee e imprime todo lo que se retorne desde php, se utilizó para saber si se estaban subiendo los datos de manera correcta.

```
unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    //Serial.println(">>> Client Timeout !");.....
    client.stop();
    return;
  }
}

//Lee todas las lineas del servidor
while (client.available()) {
  String line = client.readStringUntil('\r');
  Serial.print(line);
}

//Serial.println();.....
//Serial.println("closing connection");.....

delay(1000);
}
```

Figura 2.42 Ventana de Arduino

Al final del código se iguala el *contador* con *estadoanterior* para que no siga subiendo datos de manera innecesaria.

```
    estadoanterior=contador;
  }
```

Figura 2.43 Reseteo de variable

## 2.4 Desarrollo del prototipo

El prototipo final supone la funcionalidad de 4 cajones de estacionamiento que detectan de manera automática y a tiempo real la disponibilidad de los lugares, dicha información es almacenada en una base de datos la cual consulta una aplicación móvil para mostrarlo al usuario . El sistema de topes está alimentando por una batería que se recarga mediante la energía solar.

El prototipo final se compone por:

- Paneles solares.
- Cableado.



- Caja Legrand con Raspberry Pi y Router.
- Caja Legrand con controlador de carga y batería.
- Topes de hule con Microcontrolador y sensores.

### 2.4.1 Tope de estacionamiento

Cada tope está compuesto por un circuito impreso en el cual se conecta el microcontrolador, el sensor ultrasónico y un divisor de corriente L7805, debido a que la placa y el sensor trabajan a 5v y el voltaje que sale del controlador de carga es de 12v. Cada divisor de voltaje está protegido por un disipador.

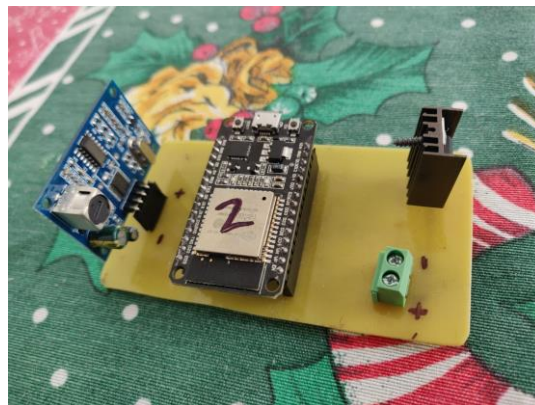


Figura 2.44 Baquelita utilizada para la programación

La razón por la que cada placa tiene un divisor de voltaje es para posibles reparaciones a futuro, donde se tengan muchos más cajones, así solo sería necesario deshabilitar un cajón mientras el sistema sigue funcionando con normalidad.



Figura 2.45 Tope de caucho con sensor ultrasónico

La conexión interna del tope se ve de la siguiente manera.



Figura 2.46 Vista por dentro del tope

Los sensores están colocados atravesando una pared de cada tope, con una inclinación de 20 grados debido a que fue el ángulo donde se percibieron mejores resultados en las pruebas hechas. Para evitar fugas de agua fue reforzado con pegamento y resultó resistente a la lluvia.



Figura 2.47 Sensor visto por dentro

Las baquelitas fueron sellados con una tabla al tope con tornillos, no se recubrieron con ninguna pasta por fines de prototipado, pero para hacerlo completamente resistente a las salpicaduras sería necesario también recubrir la parte de abajo.

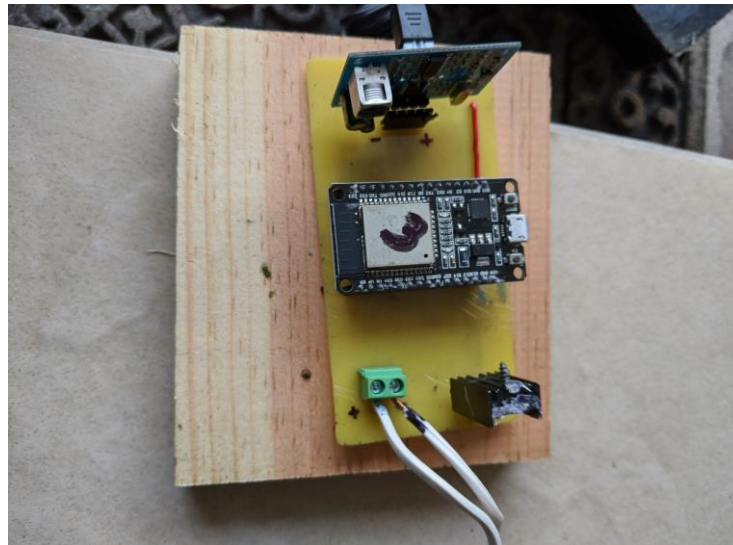


Figura 2.48 Instalado.

### 2.4.2 Caja de alimentación

La caja de control que alimenta al sistema de sensado se compone por una batería de 7000mAh que está conectada a un controlador de carga en los puertos de en medio, que evita que se descargue por completo y asegurando una vida útil más prolongada. Al controlador de carga se conecta la corriente a utilizar para el sistema en los puertos del lado derecho. Del lado izquierdo se conectan los paneles solares que proporcionarán energía a través del controlador. El controlador de carga fue programado solo para funcionar 12h al día.



Figura 2.49 Caja de control

### 2.4.3 Alimentación del sistema

Para la alimentación de los sensores y los microcontroladores se utilizaron dos paneles solares con las siguientes características.

MODULE TYPE:	
Peak Power	(Pmax): 10W
Maximum Power Current	(Imp): 0.58A
Maximum Power Voltage	(Vmp): 17.1V
Short-Circuit Current	(Isc): 0.65A
Open-Circuit Voltage	(Voc): 21.1V
Nominal Operating Cell Temp.	(Noct): 47°C
Application:	DC 12V
Weight:	1.59 KGS
Dimensions:	380x290x28 mm
Maximum System Voltage:	750V
Wind Resistance:	2400Pa
All technical data at standard test condition AM=1.5 E=1000W/m <sup>2</sup> Tc=25°C	
<p><b>WARNING</b> ELECTRICAL HAZARD</p> <p>THIS UNIT PRODUCES DC ELECTRICITY WHEN EXPOSED TO LIGHT. COVER GLASS BEFORE REMOVING TERMINAL JUNCTION BOX LID.</p> 	

Figura 2.50 Características técnicas del panel

En su máximo rendimiento los dos paneles solares nos brindaban una corriente de 250mAh, con un voltaje de 20V, con eso era suficiente alimentar el controlador de

carga que era de 12V, y el consumo de los cuatro topes de estacionamiento era de 88mAh.

Los paneles fueron colocados con dirección al sur a aproximadamente 20° para aprovechar al máximo la luz solar.



Figura 2.51 Inclinación de los paneles

Los dos paneles fueron colocados juntos y sus cables de alimentación fueron conectados en paralelo.

#### 2.4.4 Caja de comunicación

Es la caja donde se encuentra la Raspberry y el router están únicamente conectados a la corriente de luz.



Figura 2.52 Caja de control

## 2.4.5 Instalación

El sistema se instaló en una casa, primero fue necesario delimitar en qué área se colocarán los paneles solares, se colocaron en la parte trasera.



Figura 2.53 Paneles solares

Se utilizó cable calibre 18 de doble filamento y se instaló atravesando la casa desde los paneles solares hasta la caja de alimentación ubicada en la habitación más cercana a la cochera para aprovechar al máximo las redes inalámbricas, para esta tarea se utilizaron 30 metros de cable.

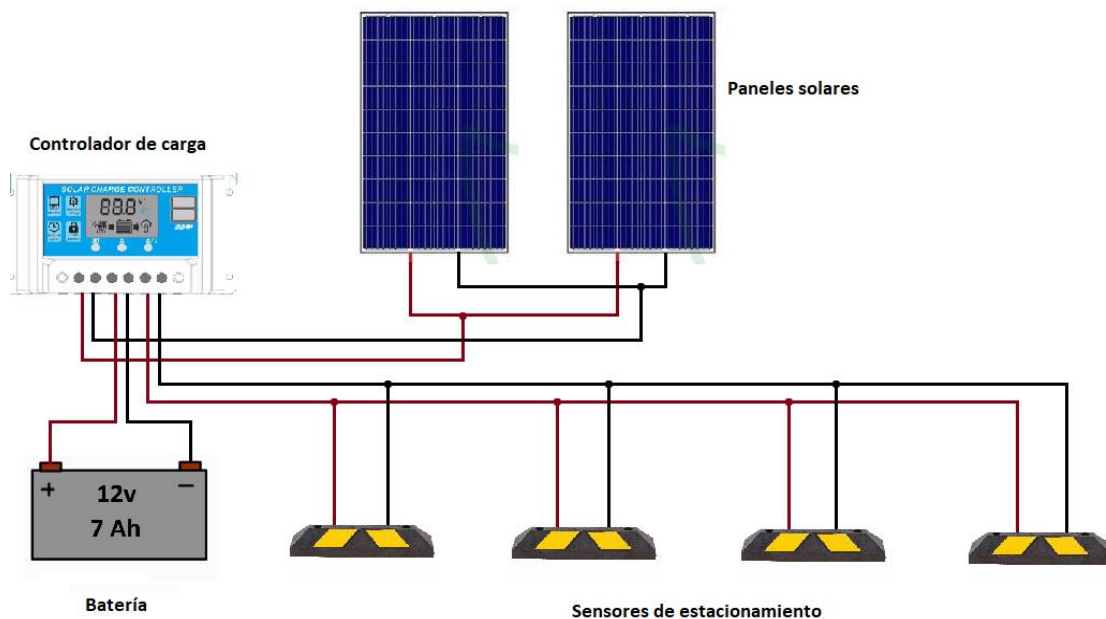


Figura 2.54 Plano unifilar de la conexión externa de los topes

Por una ventana de la habitación sale el cableado que energiza el sistema de topes desde la caja de alimentación.



Figura 2.55 Topes 1, 2 y 3 instalados y funcionando

Se instalaron los topes 1, 2, 3 en una cochera cada uno de estos topes tiene de separación 2.20 mts.

El tope número 4 se instaló en la cochera conjunta para que el sistema tuviera las dimensiones proyectadas.



Figura 2.56 Tope 4 instalado y funcionando

Los topes fueron conectados entre sí en paralelo, para aprovechar el cableado, de cada tope sale la alimentación al siguiente.



Figura 2.57 Vista al ras del suelo de los topes 1, 2, 3.

Se comprobó que el sistema estuviera funcionando de manera correcta utilizando la aplicación móvil para ver si actualizaba su estado.



Figura 2.58 Aplicación móvil en funcionamiento

Una vez comprobamos que todo funcionaba de manera correcta, dejamos el sistema durante un mes en funcionamiento.





Figura 2.59 Sistema de topes en funcionamiento

## 2.5 Aplicación Móvil

El proyecto requiere que la información pueda ser visualizada desde cualquier dispositivo con sistema operativo Android, para ello es necesario crear una aplicación en formato .apk. Debido a las características que se necesitan para la aplicación utilizamos una aplicación web llamada AppInventor, que ofrece las herramientas necesarias para su desarrollo.



Figura 2.60 AppInventor

Las funciones que necesita tener la aplicación es mostrar la disponibilidad de los lugares de estacionamiento, tomando la información desde la base de datos. La aplicación debe ser directa, debido a que el usuario final no puede distraerse mientras es utilizada.

### 2.5.1 Diseño

La aplicación cuenta con imágenes que muestran la disponibilidad de los cajones: Si está ocupado se observa un coche estacionado en el lugar, de lo contrario, el cajón se encontrará vacío.



Figura 2.61 Imagen de la aplicación

### 2.5.2 Programación de la aplicación

La plataforma de AppInventor proporciona herramientas por bloques para la programación.

Al iniciar ingresamos ESTACIONAMIENTO como nombre del proyecto, nos dirigimos a la pantalla de Diseñador en la esquina superior derecha para agregar los elementos necesarios de nuestra aplicación.

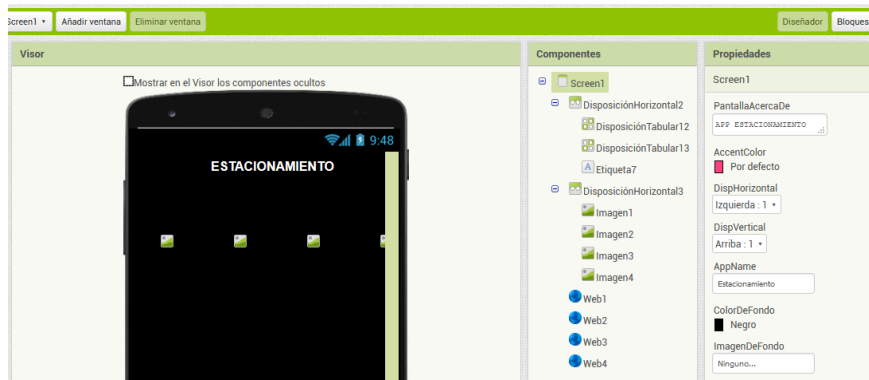


Figura 2.62 Pantalla Diseñador

Del lado izquierdo de la pantalla, en la sección Paleta encontraremos todos los elementos que podemos agregar.

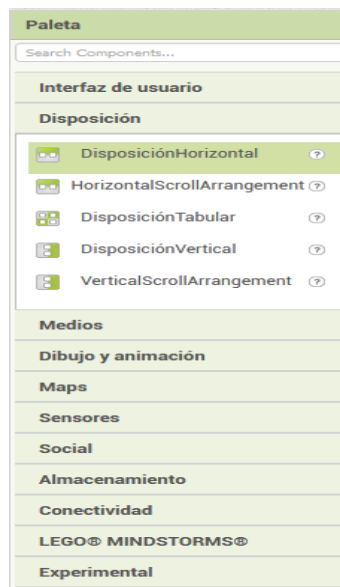


Figura 2.63 Elementos Paleta

Del lado derecho de la pantalla, en la sección Componentes encontraremos todos los elementos agregados a la aplicación.

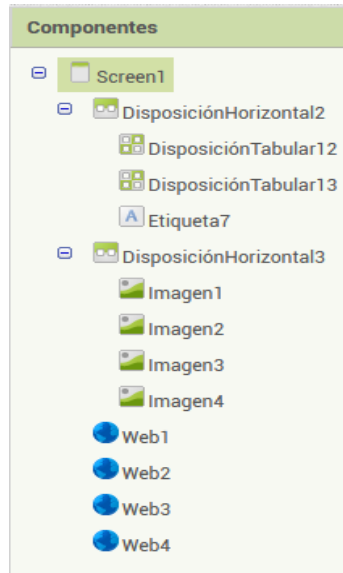


Figura 2.64 Componentes agregados a la APP

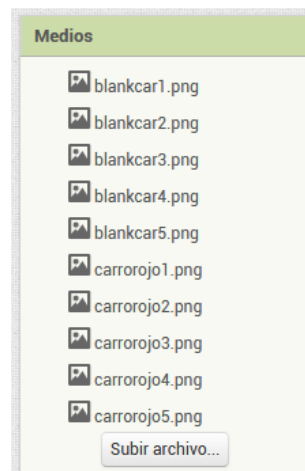


Figura 2.65 Imágenes agregadas a la APP

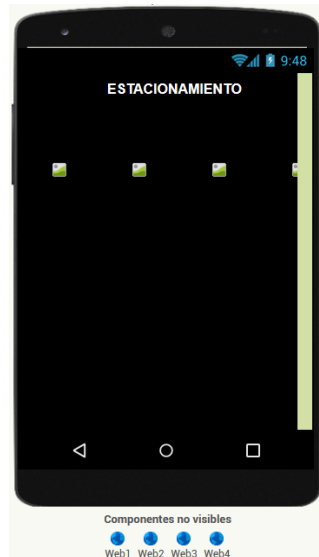


Figura 2.66 Previsualización de APP

Cuando la aplicación es abierta, se ejecutará la instrucción de escribir un URL para comprobar los distintos cajones que tiene agregados a través del código ESTACIONAMIENTO.php, almacenados en la Raspberry PI, como son 4 cajones los que tiene agregados, es necesario que consulte 4 códigos .php. El bloque de la derecha es un contador de 5s, su función es volver a leer los valores que tienen los códigos .php.



Figura 2.67 Bloques de programación

En este bloque, la variable "contenidoDeRespuesta" toma el valor que se imprime después de ejecutar "ESTACIONAMIENTO.php" y lo iguala, si recibe un 1, hace visible en la aplicación, la imagen de "carrerojo.png", si es un 0, la imagen visible es "blankcar.png", que es la imagen del cajón vacío.

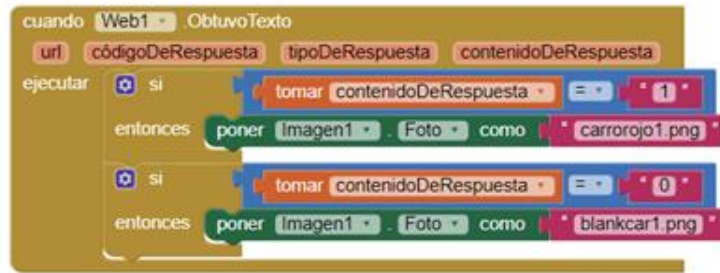


Figura 2.68 En este bloque toma la respuesta de “ESTACIONAMIENTO1.php” y lo compara

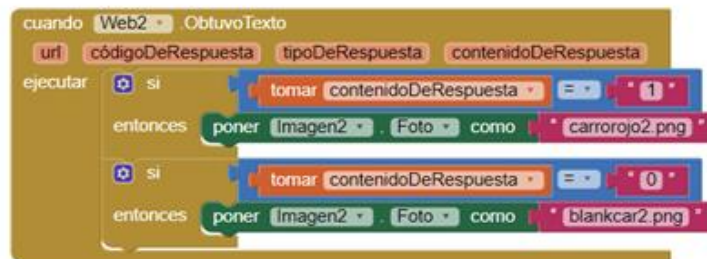


Figura 2.69 En este bloque toma la respuesta de “ESTACIONAMIENTO2.php” y lo compara

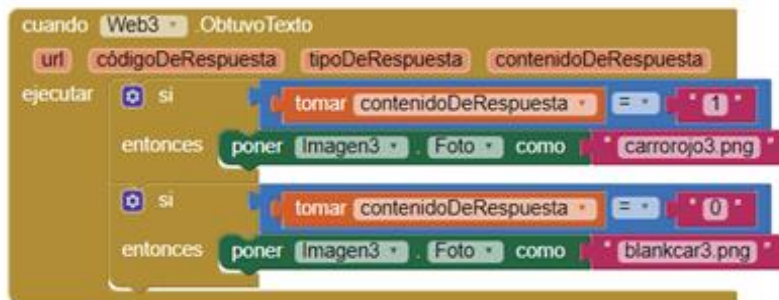


Figura 2.70 En este bloque toma la respuesta de “ESTACIONAMIENTO3.php” y lo compara

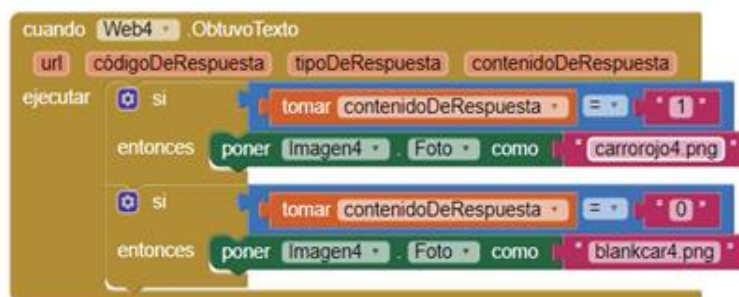


Figura 2.71 En este bloque toma la respuesta de “ESTACIONAMIENTO4.php” y lo compara

## Resultados

Una vez terminada la construcción del prototipo, se realizaron las pruebas correspondientes, analizando su funcionamiento y comprobando que haya cumplido satisfactoriamente las metas planteadas en un inicio.

Después de analizar los resultados obtenidos durante un mes de uso continuo del prototipo, podemos observar que el movimiento promedio existente en 4 cajones instalados es de 13 movimientos al día. El sistema fue utilizado al inicio de 7am a 9pm, pasadas dos semanas se amplió el rango de uso y paso a ser de 5am a 11pm.

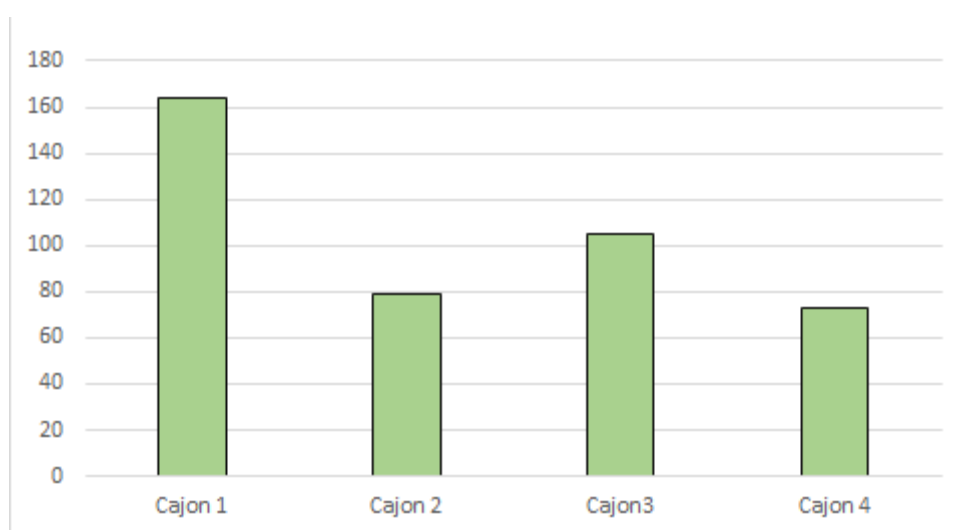


Figura 3 Gráfica del uso de los topes

Consultando la base datos podemos observar el número de veces que cada cajón tuvo un cambio de estado, al consultar los datos se puede saber que los resultados son:

- Cajón 1 = 164
- Cajón 2 = 79
- Cajón 3 = 105
- Cajón 4 = 73

Hubo un total de 421 movimientos y algunos falsos positivos (figura ...) , debido a que cuando el sistema se volvía a encender por las mañanas se reseteaba la variable que controlaba el cambio de estado, haciendo que volviera a subirlo a la base de datos.

Estos falsos positivos no presentan algún problema en el sistema, simplemente reflejan el último dato registrado.

Del tiempo en que estuvo instalado el sistema, existieron 2 días que falló, debido a las condiciones climáticas, lo cual provocó que la batería no se pudiera cargar adecuadamente, causando que el sistema se apagara, este inconveniente representó el 6.66% del tiempo total que estuvo instalado.

La aplicación móvil que desarrollamos funcionó correctamente, reflejó los valores con un tiempo de actualización de 7 segundos, como lo programamos, nunca hubo fallo al momento de obtener los valores y mostrarlos en la aplicación de manera adecuada.

El costo total del sistema con 4 cajones es de \$6,800 aproximadamente, los componentes para cada cajón tienen un costo de \$500 aproximadamente, tomando en cuenta que los costos son basados en la cotización del precio unitario de todos los componentes implicados en el sistema. No incluye costos de fabricación, trabajos de instalación, ni programación de los microcontroladores y bases de datos.

Proyectando el prototipo a un estacionamiento tradicional, donde el lugar más cercano a tu destino final también sea el punto más concurrido y lejano del estacionamiento, tener a disposición esta tecnología evita las vueltas innecesarias a lugares ocupados evitando el tráfico dentro del estacionamiento y agilizando la localización del lugar más adecuado a las necesidades del usuario en ese momento.

Para la realización de este sistema el proyecto cumple con la mayoría de las ramas de la mecatrónica, ya que abarca conocimientos de esta ingeniería que lo definen como tal:

- Computación
- Control
- Electrónica
- Mecánica



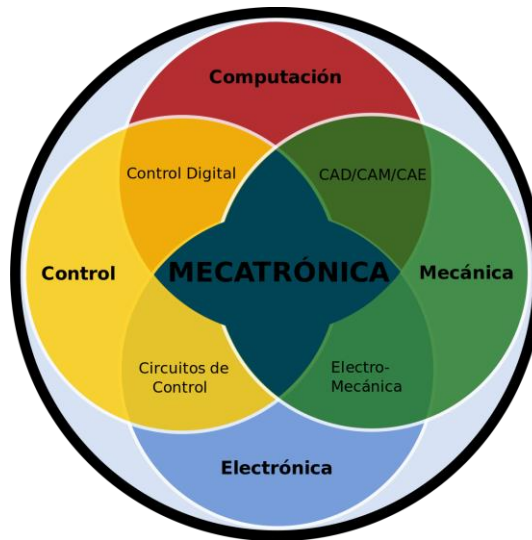


Figura 4 Ramas de la mecatrónica. Ingeniería mecatrónica. (2020)

El sistema fue implementado en una cochera de casa, la evaluación y validación estuvo a cargo del Ing. Daniel Iván Pureco Niño quien quedó enterado de su funcionamiento, también realizó prácticas demostrativas de la operación de dicho proyecto, corroborando el cumplimiento de sus características y especificaciones establecidas.

## Conclusión

El proyecto requirió diversas aplicaciones de conocimientos, obtenidos en base a la investigación y los conocimientos adquiridos durante la carrera.

La implementación del sistema en estacionamientos cumplió de manera satisfactoria con el funcionamiento, censando de manera correcta y con buen tiempo de respuesta a los coches que llegan a estacionarse, mostrando la información en la aplicación. Comprendemos que este proyecto es la primera iteración completamente funcional en estacionamientos y tenemos claro que esta tecnología podría evolucionar en un futuro.

Según la hipótesis planteada al inicio del documento, “La implementación de un prototipo de sistema automático que permita controlar los estacionamientos, tendrá un impacto positivo en lo relacionado a la búsqueda de un espacio libre de parte de un conductor, así como en el aspecto económico, tecnológico y social.”

La hipótesis es válida debido a que el usuario tiene la facilidad de visualizar desde su teléfono móvil la disponibilidad de los lugares de estacionamiento, haciendo más fácil esta tarea, sin necesidad de llegar a comprobarlo en persona. El impacto social apreciado de nuestro proyecto es la disminución en los los tiempos destinados a la tarea de buscar estacionamiento. Del impacto económico podemos concluir que hay una reducción de los insumos del vehículo, debido una reducción en su tiempo de uso, el impacto tecnológico está dado debido a que no se ha implementado una tecnología como esta en los estacionamientos, además de que los avances tecnológicos realizados en esta área han sido muy pocos con el paso del tiempo.

Podemos concluir que debido a que no se utilizó la batería óptima para el sistema, este se apagó un par de ocasiones, al no haber carga suficiente, la batería que se había planteado a utilizar era una de 20000 mAh, al no poder conseguir una de esta capacidad se optó por utilizar una de 7000 mAh.

Se utilizaron sensores ultrasónicos que medían la distancia, con la programación se acoplaron al sistema para que actuaran de manera digital, utilizando los sensores óptimos para este tipo de sistema, en este caso digitales, al ser sensores que se implementan para este tipo de casos el sistema sería más eficaz. Otra mejora que se puede implementar en el sistema de topes, es la utilización de más de un sensor en cada cajón, así generando una programación de compuertas OR asegurando una lectura más eficaz.

Debido a la exigencia energética del circuito, se vio necesario usar más de un panel, debido a que no disponíamos de un panel con las características suficientes.

Dentro de la aplicación móvil podemos mejorar la programación cambiando la plataforma de desarrollo, haciendo de ella más eficiente al momento de registrar los datos, dando aparte nuevas funciones como la hora de llegada, de salida, además de dar un diseño más estético y uniforme al diseño de los celulares actuales. Cambiando la plataforma de desarrollo podemos incluir el ampliar la aplicación a nuevas plataformas móviles, como IOS.

También se puede plantear el cambio de diseño del prototipo del encapsulado de topes, para que se puedan implementar en distintos tipos de estacionamientos o en el lugar que se necesite.

## Bibliografía

- Moreno, E. G. (2001). *Automatización de procesos industriales*. Valencia: Alfaomega.
- Arias, Á. (2014). *Bases de Datos con MySQL: 2ª Edición*. IT Campus Academy.
- Sierra, M. (2013). ¿Qué es un servidor y cuáles son los principales tipos de servidores? (proxy, dns, web, ftp, smtp, etc.), España. Recuperado 24 febrero 2013 de: [http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftpsmtp&catid=57:herramientas-informaticas&Itemid=179](http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftpsmtp&catid=57:herramientas-informaticas&Itemid=179)
- Spona, H. (2010). *Programación de bases de datos con MYSQL y PHP*. Marcombo.
- Ponsa Asensio, P. V., & Arbós, R. V. (2006). *Automatización de procesos mediante la guía GEMMA* (Vol. 102). Universitat Politecnica de Catalunya. Iniciativa Digital Politecnica.
- Ramón Pallás Areny. (2003). *Sensores y acondicionadores de señal*. Barcelona, España: Marcombo.
- Industrial Systems Engineering. (2011). *Esencia y objetivos del proceso de automatización funciones básicas y características de AS*. México, México: UNED
- Navarro, R., Castro, E. O. B., Aguilar, J. A., Tumax, H. J. P. P., Chávez, J. D. N., Ramírez, E. R., ... & Castillo, C. (2004). *Ingeniería de control*. McGraw Hill.
- Valdés, F., & Areny, R. P. (2007). *Microcontroladores fundamentos y aplicaciones con PIC* (Vol. 1149). Marcombo.
- Torres, F., & Jara, C. A. (2011). *Sensores y detectores. Automatización*.

- Silberschatz, A., Korth, H. F., Sudarshan, S., Pérez, F. S., Santiago, A. I., & Sánchez, A. V. (2002). Fundamentos de bases de datos.
- Enríquez, A. (2011). MySQL, México, Recuperado 11 mayo 2013 de: [www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf](http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf)
- Torres, G. (2008). Planificación e implementación de la infraestructura y servicios de red con Windows server 2003 y RedHat Enterprise 5 en la empresa Autorizador S.A. (Tesis de pregrado). Del repositorio del Instituto Politécnico Nacional de México. Recuperado 08 febrero 2013 de: <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/5440/1/PLANIFICACIONEIMPLEMENT.pdf>
- González, J. A. (2009). *Energías renovables*. Sindicato Independiente de Trabajadores de Cádiz.
- Batería de plomo y ácido. (2020, 27 de enero). *Wikipedia, La enciclopedia libre*.
- Formación, E. (2007). *Energía solar fotovoltaica*. FC Editorial.
- Ingeniería mecatrónica. (2020, 4 de marzo). *Wikipedia, La enciclopedia libre*.