

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



DESARROLLO DE UNA APLICACIÓN WEB PARA EL
CONTROL DE UN MOTOR ELÉCTRICO DE 3
VELOCIDADES

Reporte Técnico de Investigación presentado por:

Jose Uriel Gamon Quezada 148789

Requisito para la obtención del título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

M. I. Gabriel Bravo Martínez

Ciudad Juárez, Chihuahua

19 de noviembre de 2019

Ciudad Juárez, Chihuahua, a 7 de Noviembre de 2019

Asunto: Liberación de Asesoría

Mtro. Ismael Canales Valdiviezo
Jefe del Departamento de Ingeniería
Eléctrica y Computación
Presente.-

Por medio de la presente me permito comunicarle que, después de haber realizado las asesorías correspondientes al reporte técnico Desarrollo de una aplicación web para el control de un motor eléctrico de 3 velocidades, del alumno Jose Uriel Gamon Quezada, de la Licenciatura en Ingeniería en Sistemas Computacionales, considero que lo ha concluido satisfactoriamente, por lo que puede continuar con los trámites de titulación intracurricular.

Sin más por el momento, reciba un cordial saludo.

Atentamente

M. I. Gabriel Bravo Martínez
Profesor Investigador IIT

Ccp:
Coordinador del Programa de Sistemas Computacionales
Jose Uriel Gamon Quezada
Archivo



Ciudad Juárez, Chihuahua, a 13 de Noviembre de 2019

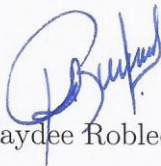
Asunto: Autorización de Publicación

C. Jose Uriel Gamon Quezada

Presente.-

En virtud de que cumple satisfactoriamente los requisitos solicitados, informo a usted que se autoriza la publicación del documento de Desarrollo de una aplicación web para el control de un motor eléctrico de 3 velocidades, para presentar los resultados del proyecto de titulación con el propósito de obtener el título de Licenciado en Ingeniería en Sistemas Computacionales.

Sin otro particular, reciba un cordial saludo.



Mtra. Ivonne Haydee Robledo Portillo

Profesor Titular de Seminario de Titulación II

Declaración de Originalidad

Yo, Jose Uriel Gamon Quezada, declaro que el material contenido en esta publicación fue elaborado con la revisión de los documentos que se mencionan en el capítulo de Bibliografía, y que la solución obtenida es original y no ha sido copiada de ninguna otra fuente, ni ha sido usada para obtener otro título o reconocimiento en otra institución de educación superior.

Uriel G.

Jose Uriel Gamon Quezada

Agradecimientos

Para comenzar quiero agradecer a todas esas personas que me apoyaron para que todo esto fuera posible.

Primeramente quiero agradecer a mis padres, Brisa y José, por haberme apoyado incansablemente desde que comencé mis estudios, que con amor y cariño me impulsaron a seguir adelante, para que con esfuerzo y dedicación yo pudiera ser ingeniero.

También brindo mis agradecimientos a todos mis maestros que con sabiduría me supieron guiar, llevándome por el camino correcto, exigiendo ser mejor cada día para que yo pudiera progresar y obtener mejores conocimientos que me ayudaran en mi carrera académica.

Seguido, agradezco a mis maestros de seminario de titulación I y II, maestro Rene Noriega y la maestra Ivonne Robledo por instruirme este último año con este proyecto de titulación, en como redactar un buen documento de investigación cada día y cada clase con paciencia y sabiduría.

Por último, quiero darle las gracias a mi asesor M. I. Gabriel Bravo Martínez de este proyecto de titulación, que sin la ayuda de él no hubiera sido posible la realización de esta investigación, de este proyecto y estos conocimientos que me deja el haber llevado a cabo dicho proyecto para ser un buen ingeniero.

Dedicatoria

Ese proyecto de titulación se lo dedico a mis padres, ya que es un sueño y un logro para ellos el poder verme titulado, como ingeniero, para que ellos puedan ver que todo el esfuerzo, cariño, tiempo, dinero, paciencia entre otras cosas se ven reflejados en este título y por que sin duda esto no hubiera sido posible sin ellos, ya que ellos son responsables del hombre que soy ahora y que con mucho amor y cariño este proyecto de titulación es por ellos y por toda mi familia.

Índice general

1. Planteamiento del Problema	4
1.1. Antecedentes.....	4
1.2. Definición del problema	7
1.3. Objetivos.....	8
1.3.1. Objetivo General.....	8
1.3.2. Objetivos específicos	9
1.4. Justificación	9
2. Marco teórico	11
2.1. IoT (<i>Internet of Things</i>).....	11
2.2. Generalidades del IoT	13
2.3. Microcontrolador ESP8266.....	14
2.4. Arduino.....	17
2.5. Wifi	19
2.6. Automatización de motores eléctricos.....	20

3. Desarrollo del Proyecto **22**

3.1. Solución propuesta.....	22
3.2. Fases del desarrollo del proyecto	23
3.2.1. Diagrama eléctrico.....	23
3.2.2. Automatización	29
3.2.3. Hardware del sistema	32
3.2.4. Requerimientos	48
3.2.5. Interfaz gráfica.....	54
3.2.6. Aplicación.....	59
3.2.7. Algoritmo de control.....	66
3.2.8. Pruebas	73
3.2.9. Presupuesto	75
3.2.10. Metodología.....	75

4. Resultados y Discusiones **79**

5. Conclusiones **85**

5.1. Con respecto al objetivo de la investigación	85
5.2. Recomendaciones para futuras investigaciones.....	87

Bibliografía **89**

Índice de figuras

3.1. Motor a automatizar	23
3.2. Circuitos y cables del motor	24
3.3. Interruptor de 3 velocidades.....	24
3.4. Extracción del interruptor de velocidades.....	25
3.5. Interruptor de apagado/encendido del foco	25
3.6. Interruptor de velocidades del motor.....	26
3.7. Cables que conducen la energía para darle distintas velocidades al motor .	27
3.8. Revisión de continuidad en interruptor 1.....	28
3.9. Revisión de continuidad en interruptor 2.....	28
3.10. Combinaciones de continuidad del interruptor de velocidad.....	29
3.11. Capacitor del motor.....	30
3.12. Diagrama eléctrico del motor.....	31
3.13. Relevador.....	31
3.14. Relevador.....	32
3.15. Microcontrolador ESP8266	32

3.16. Pulidor	33
3.17. Relevador.....	33
3.18. Relevador.....	34
3.19. Circuito del relevador.....	35
3.20. Análisis físico de la tablilla	36
3.21. Análisis físico de la tablilla	36
3.22. Análisis físico de la tablilla	36
3.23. Cortes a la tablilla de relevador	37
3.24. Cortes a la tablilla de relevador	37
3.25. Cortes a la tablilla de relevador	38
3.26. Pruebas de continuidad.....	38
3.27. Pruebas de continuidad.....	39
3.28. Pruebas de continuidad.....	39
3.29. Pruebas de corriente	40
3.30. Pruebas de corriente	40
3.31. Pruebas de corriente	41
3.32. Pruebas de corriente	42
3.33. Pruebas de corriente	42
3.34. Pruebas de corriente	43
3.35. Conexiones entre el microcontrolador ESP8266 y los relevadores	44
3.36. Conexiones entre el microcontrolador ESP8266 y los relevadores	44

3.37. Pruebas de funcionamiento a relevadores.....	45
3.38. Conexiones entre el microcontrolador ESP8266 y los relevadores	45
3.39. Pruebas de funcionamiento a relevadores.....	46
3.40. Pruebas de funcionamiento a relevadores en aplicación web.....	46
3.41. Posición en la que debe estar encendido el foco.....	47
3.42. Posición en la que esta encendido el foco y la velocidad alta del motor.....	48
3.43. Posición en la que esta encendido el foco y la velocidad media del motor . .	.49
3.44. Posición en la que esta encendido el foco y la velocidad baja del motor . .	.50
3.45. Solado de cables a terminales del relevador	51
3.46. Solado de cables a terminales del relevador	51
3.47. Pegado de cables a terminales del relevador	52
3.48. Pegado de cables a terminales del relevador	52
3.49. Prueba en conjunto.....	53
3.50. Prueba en conjunto.....	53
3.51. Guardado de circuitería	54
3.52. Vista inicial de la aplicación web.....	55
3.53. Color amarillo en botón que indica encendido del foco.....	56
3.54. Encendido del motor en velocidad alta y aparición del botón de apagado total	57
3.55. Cambio de color y valor del botón entre cambio de velocidades.....	58
3.56. Cambio de color y valor del botón entre cambio de velocidades.....	59
3.57. Regreso a estado inicial de los botones	60

3.58. Regreso a estado inicial de los botones	61
3.59. Funcionamiento del ventilador sin el foco	62
3.60. Funcionamiento del ventilador sin el foco	63
3.61. Funcionamiento del ventilador sin el foco	64
3.62. Cabecera del HTML.....	64
3.63. Código de estilos	65
3.64. Botones de Luz principal.....	65
3.65. Botones control del ventilador	66
3.66. Botón de apagado total.....	66
3.67. Librerías.....	67
3.68. Configuraciones de direccionamiento	67
3.69. Constantes a números de pin	68
3.70. Pin de GPIO del ESP8266.....	68
3.71. Configuración de los pines en modo de salida	69
3.72. Parámetros de conexión al Wifi.....	69
3.73. Inicialización del servicio	70
3.74. Servidor HTTP.....	70
3.75. Control de velocidades del motor	71
3.76. Control de velocidades del motor	72
3.77. Código de página HTML.....	73
3.78. Envío de código HTML.....	73

4.1. Vista inicial de la aplicación web.....	80
4.2. Encendido del foco.....	80
4.3. Encendido del motor en velocidad alta	81
4.4. Cambio a velocidad media del motor.....	82
4.5. Cambio a velocidad baja del motor	82
4.6. Apagado del foco en operación con el ventilador	83
4.7. Encendido del foco en operación con el ventilador	83
4.8. Apagado total del sistema.....	84
4.9. Detención del funcionamiento	84

Resumen

En esta investigación se verá cómo se llevó a cabo la automatización y control de un motor eléctrico de 3 velocidades a través de una aplicación web, el ventilador tendrá la posibilidad de conectarse a la red doméstica Wifi del hogar, ejecutar la función de servidor el cual contendrá la aplicación web a la que se va a acceder para poder controlar dicho motor.

Se eliminaron los interruptores físicos del ventilador por una tablilla de relevadores, que tendrán la misma función que los interruptores pero tienen la habilidad de recibir órdenes por un microcontrolador. Como ya se mencionó, se agregó un microcontrolador ESP8266 de arduino, con el cual el ventilador podrá conectarse a la red Wifi del hogar y ejecutar la aplicación web que tendrá las funciones para controlar el funcionamiento del ventilador y se verá cómo se programaron las instrucciones, todo esto en conjunto con un teléfono inteligente funcionando sobre la arquitectura del Internet de las cosas.

Palabras clave: Internet de las cosas, arduino, desarrollo web, sistemas embebidos, automatización.

Abstract

In this investigation, an automation and the control of a 3-Velocity Electric Motor will be carried out by the development of an app. The motor's ventilator will be able to connect through a wireless server linked to the house in which people live in and execute the commands directly through the app.

The ventilator's switches were removed and instead, a relay board was added. This board will have the same function as the original switches, however it'll receive a sign directly from an ESP8266 Arduino microcontroller. This will let the ventilator connect to the house server in order to control it from your smartphone.

Keywords: Internet Things, embedded tech, web development, arduino, automation.

Introducción

El uso de la tecnología como la arquitectura del Internet de las cosas no ha sido usada en toda su magnitud en México. Se sabe que el Internet de las cosas es una arquitectura que permite la comunicación de distintos dispositivos como electrodomésticos, autos, casas, celulares, etc., por medio de la red de redes, el Internet sin la necesidad de la intervención humana, es decir, toda esta comunicación e interacción se da de manera automática.

Como este tipo de tecnología no ha sido explotada al máximo, en México aún hay dispositivos que funcionan a la antigua y esto es así, debido a que los dispositivos comerciales que ya existen son muy caros o difíciles de conseguir, sin embargo, la gente no sabe que en realidad puede realizar este tipo de dispositivos de manera fácil, sencilla y a un bajo costo.

El objetivo del presente documento es que se realice un aporte al desarrollo tecnológico ya existente, con la elaboración de una aplicación para poder controlar un motor eléctrico de tres velocidades a distancia mediante la red Wifi del hogar, así como el promover el concepto "Hazlo tú mismo".

Para comenzar en el primer capítulo se verá qué antecedentes se tienen sobre esta arquitectura del Internet de las cosas basada en la Internet global, la definición de la problemática a la que se enfrenta la humanidad al no contar con tecnología que le facilite la vida y la justificación que sustenta el que se realice este proyecto.

Pasando al segundo capítulo se abordará el tema acerca de los puntos mas importantes que sustentan esta investigación y que permiten llevar a cabo este proyecto. Esto es posible

mediante la definición y el entendimiento del concepto de IoT, cómo funciona, en qué se basa, entre otras. También se hablará sobre las generalidades del IoT y que se esperan en un futuro sobre esta arquitectura. El control del motor es muy importante por lo tanto esta sección contiene información sobre automatización de motores eléctricos y como este motor eléctrico puede conectarse al wifi mediante un microcontrolador ESP8266, así como también al hablar de microcontroladores se explica que es Arduino, el circuito que ayudará a controlar el motor y cómo lo hará. También se habla un poco del wifi y una breve explicación sobre su funcionamiento y que tiene que ver con el IoT.

Continuando con el capítulo 3, se verá como se fue construyendo todo el sistema, desde la aplicación, hasta las pruebas en dicho motor eléctrico de 3 velocidades, el cual pertenece a un ventilador de techo doméstico, se explicará la modificación que se realizó al motor, el hardware que conformará el sistema, el diseño de la interfaz gráfica para la aplicación y del algoritmo de gestión de las velocidades del motor así como la estructura en lenguaje HTML de la aplicación web como el diseño estético de este mismo. Estos son los puntos principales que determinaron la construcción de dicho proyecto.

En el siguiente capítulo se reportará el funcionamiento del sistema y de que manera lo hace, el manejo sencillo e intuitivo de este a través de la aplicación web que puede ser ejecutada tanto en computadoras como en móviles. También se verá un poco de en que lugares funcionó la aplicación dentro del hogar donde se hicieron las pruebas.

Y en el último capítulo se concluirá este proyecto, recapitulando brevemente cada objetivo específico que llevó esta investigación mostrando cómo se llegó a cumplir el objetivo general del proyecto. Además, se describen qué herramientas se utilizaron y cómo se utilizaron, asimismo, se presenta la elección del hardware para la automatización del ventilador y la configuración de estos dispositivos que se eligieron para que pueda ser controlado a través de una aplicación web. También se incluyen algunas consideraciones a tomar en cuenta como proyectos a futuro como mejora a este sistema.

Para finalizar los resultados de este documento de investigación beneficiarán a cualquier persona que utilice este dispositivo a obtener una mejor comodidad en su hogar a la hora de operar su ventilador de techo de 3 velocidades como el tener una mejor visualización del concepto "Hazlo tú mismo".

Capítulo 1

Planteamiento del Problema

Con la evolución de la tecnología se presentan avances importantes y significativos en todos los ámbitos posibles como lo es la ciencia, la medicina, educación, física, etc. En este documento se habla sobre tecnología, un aporte como mejora tecnológica a lo que ya existe y sobre lo que se está desarrollando que es el Internet de las cosas (*Internet of things*, IoT por sus siglas en inglés), presentándose qué antecedentes se tienen sobre esta arquitectura basada en la Internet global, la definición de la problemática a la que se enfrenta la humanidad al no contar con tecnología que le facilite la vida y la justificación que sustenta el que se realice este proyecto.

Con el paso de los años, la forma de realizar las actividades cotidianas han ido cambiando. La vida diaria puede ser más fácil y cómoda con el uso de la tecnología, incluso es capaz de ayudar a mejorar el entorno. Con la conexión a Internet entre diversos dispositivos, estos pueden intercambiar información y realizar actividades como control, automatización y comunicación.

1.1. Antecedentes

Mejorando continuamente el entorno que rodea al ser humano, la tecnología es aquello que algunas personas consideran fuera de su entorno cotidiano, por sus costumbres, por su

forma de pensar o por el hecho de no saber cómo utilizar los aparatos que de ella provienen, sin embargo, la tecnología facilita la vida diaria. *Internet of Things* (IoT por sus siglas en inglés), es una arquitectura emergente basada en la Internet global que facilita el intercambio de bienes y servicios entre redes de la cadena de suministro y que tiene un impacto importante en la seguridad y privacidad de los actores involucrados [1].

Existen muchos tipos de dispositivos que funcionan bajo el paradigma del IoT y el número de dispositivos conectados a Internet se encuentra en aumento. A continuación se verán algunos conocimientos previos que se tienen sobre el tema del IoT.

En el ámbito climático, se sabe que para el pronóstico del clima de un lugar, se basa en la recopilación de datos cuantitativos sobre las condiciones meteorológicas, a través de una estación que es la encargada de procesar dichos datos. Dentro del IoT, existen estaciones conectadas al Internet, en la que cualquier usuario puede acceder a una plataforma IoT mediante un sitio web para saber el clima de un área sin tener que ir a ésta [2]. También hay otras áreas en las que el IoT es una herramienta eficiente como lo es la gestión en el consumo de energía eléctrica.

En lo que respecta a la gestión de la energía, Siriwat Wasoontarajaroen, Khwanchai Pawasan y Vithaya Chamnanphrai de la facultad de Ingeniería, Universidad de Tecnología de Rajamangala, desarrollaron un dispositivo para monitorizar el consumo de energía eléctrica con IoT [3]. Este dispositivo mostró ser muy efectivo, proporcionando datos confiables al usuario sobre el consumo de energía eléctrica, de este modo se puede llevar a cabo una gestión eficiente en el consumo de energía eléctrica, lo cual se puede monitorizar desde cualquier lugar donde haya Internet.

También el IoT se aplica a la agricultura; el monitoreo mediante IoT es utilizado para delinear una técnica manejable y sensores fáciles de instalar, para detectar y especificar el nivel de humedad del suelo [4], incluso predecir la precipitación estacional para planificar una cosecha futura. Este dispositivo de IoT es una excelente herramienta para los agricultores, ya

que el uso computacional es el más aceptado por el bajo costo, su alta persistencia y menor demanda de trabajo.

En otros ámbitos, se tiene el posicionamiento de personas u objetos en interiores que presentan un novedoso sistema de posicionamiento interior (IPS) basado en la fusión. Este fusiona los datos sensoriales de tres tecnologías populares de sensores para teléfonos inteligentes: Wifi, Bluetooth Low Energy (BLE) y *Pedestrian Dead Reckoning* (PDR) para lograr una precisión superior de localización en interiores [5]. El sistema propuesto se puede usar para una amplia base de aplicaciones relacionadas con Internet de las Cosas (IoT) que involucran movilidad y localización. En este enfoque, primero se comparó la precisión de posicionamiento en interiores de los esquemas de toma de huellas dactilares utilizando Wifi y BLE a través de sus valores de intensidad de señal recibida (RSS). De este modo, todos los elementos en el entorno del usuario tendrán un posicionamiento en el interior de alguna edificación con la certeza de ser ubicado.

En las actividades cotidianas del hogar o actividades básicas, el sistema domótico del hogar es un proceso de automatización o adaptación a las actividades básicas del hogar [6], como el control de los aparatos de iluminación, calefacción, ventilación y aire acondicionado, bajo el comando del usuario. Se utiliza un dispositivo de mano dedicado el cual es ideal para proporcionar una interfaz de usuario en un sistema de automatización del hogar, debido a su portabilidad y su amplia gama de capacidades. Este tipo de tecnología podría ayudar a automatizar las actividades cotidianas o facilitar su realización.

En el monitoreo de energía sobre dispositivos electrónicos en el hogar, con el advenimiento de Internet y la era computacional, no solo se tiene la oportunidad de enviar y recibir datos entre humanos, sino también entre los dispositivos ayudaría a ahorrar energía sin control humano sobre estos dispositivos y se puede aplicar para resolver el creciente problema de la gestión de energía [7]. Una solución propuesta es un sistema de monitoreo de energía barato y fácil de implementar y administrar para el uso diario de energía eléctrica con el fin de superar

los errores humanos, el trabajo manual y la reducción de costos en el consumo de energía con más eficiencia para el sistema de administración de energía.

En estudios enfocados especialmente al Internet de las cosas se tiene que, *Internet of Things* se refiere a una combinación de dispositivos físicos que tienen sensores y capacidades de conexión que les permiten interactuar entre sí (máquina a máquina) y se pueden controlar de forma remota a través del motor de nube. El éxito de un dispositivo IoT depende de la capacidad de los sistemas y dispositivos para muestrear, recopilar y analizar datos de manera segura, y luego transmitirlos a través de enlaces, protocolos o selecciones de medios basados en los requisitos establecidos, todo sin la intervención humana [8].

Afirmando lo anterior, se tiene que la visión de la conectividad de la red y la capacidad de computación se revelan en futuras mejoras en el campo de esta tecnología, que se extiende a sensores, actuadores, RFID, software, electrónica que permiten que estos objetos recopilen e intercambien información de forma remota [9]. Para finalizar con los antecedentes del IoT, también se puede aplicar al uso doméstico [10], como lo es el control automatizado de las luces en un hogar, lo que brinda mayor comodidad a aquellos usuarios que tienen una agenda muy ocupada. Este tipo de personas se verían beneficiadas ya que les facilitaría el manejo de iluminación de su casa desde cualquier lugar donde tengan acceso a Internet.

1.2. Definición del problema

El IoT plantea que todos los objetos alrededor del entorno del usuario como refrigeradores, tostadoras, aires acondicionados, televisiones o en general aparatos electrónicos/mecánicos estén interconectados entre sí a través de Internet, para llevar a cabo procesos automatizados o gestionar de una mejor manera dichos aparatos. En el último año esta tendencia ha estado dando lugar a un crecimiento exponencial, sin embargo, aún está en aumento; no hay una gran variedad de dispositivos tecnológicos de distintos tipos conectados al IoT, sobre todo en

países del tercer mundo, así como también algunos de estos dispositivos aún se manejan de manera manual.

El control a distancia de las cosas se ha vuelto muy popular y en estos días a través de la Internet, lo cual brinda más comodidad y facilidad de manejar el dispositivo en cuestión. El rápido desarrollo del IoT hace que el hogar inteligente se haga una realidad como la gente espera [10].

El sistema de hogar inteligente crea un entorno más cómodo, seguro, confortable y como se hace saber, inteligente [10]. El IoT puede resolver los problemas a los que se enfrentan las personas que tienen horarios ocupados y que cuentan con muy poco tiempo para descansar en casa, por lo que el usuario al llegar a casa tiene que hacer todo de una manera manual, por ejemplo, encender las luces de su hogar, abrir las cortinas o bien encender su ventilador de techo y manejar sus velocidades, dándole menos tiempo de disfrutar su estadía en casa, puesto que tiene que llegar haciendo estas cosas.

El usuario puede depender de las máquinas y dispositivos automatizados, como los teléfonos inteligentes. Tener el control de un dispositivo en cuestión desde la palma de su mano mejora notablemente la tecnología existente, sin tener que levantarse hasta el lugar del dispositivo, todo desde donde está, haciendo más cómoda la vida del usuario.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar una aplicación wifi web para controlar un motor de 3 velocidades (IoT), para darle más comodidad al usuario a la hora de manejar su ventilador de techo desde cualquier parte de su hogar, con tan solo tener a la mano la aplicación web y estar conectado a la red local de su casa (Wifi).

1.3.2. Objetivos específicos

- Determinar el diagrama eléctrico del motor a automatizar
- Realizar la automatización del motor de 3 velocidades
- Escoger hardware, microcontrolador, plataforma, conexiones eléctricas/electrónicos
- Analizar requisitos para la aplicación web
- Diseñar la interfaz gráfica para la aplicación
- Desarrollar la aplicación de control
- Diseñar algoritmo de gestión de control de las velocidades del motor
- Realizar pruebas

1.4. Justificación

En la actualidad, es que muchos dispositivos electrónicos, mecánicos, etc., no pueden ser controlados de manera remota ni automatizada, por lo que la justificación de este proyecto será introducir una mejora tecnológica con esta aplicación web y darle más comodidad al usuario al poder controlar un ventilador de techo, a distancia y de manera automática a través de la señal Wifi. El usuario en su hogar no tiene el tiempo, dinero, facilidad y la comodidad de poder controlar su ventilador por IoT. La solución para esto es el realizar una aplicación web para que el usuario pueda llevar a cabo este control. Otros beneficios que se obtienen es el ahorro de energía contribuyendo al medio ambiente.

La ventaja de controlar el motor eléctrico de tres velocidades de un ventilador de techo mediante la red Wifi (IoT) es que brinda más comodidad y de manera remota, tal y como una persona no se levanta de su sillón para cambiar de canal a su televisor, pero en este caso

no tendrá que levantarse para prender, apagar o cambiar de velocidad a su ventilador de techo, ya que la mayoría de las veces el ser humano no apaga ciertos aparatos por pereza, en cambio siendo automatizado a través del celular se podrá apagarlo o encenderlo fácilmente sin necesidad de pararse o acercarse al ventilador.

El usuario podrá controlar su ventilador desde su smartphone conectándose a la red Wifi y desde cualquier lugar que se encuentre dentro de su hogar en rango a la señal Wifi y manejar el encendido y apagado de su ventilador, así como también, la velocidad de este, mediante una aplicación web la cual podrá realizar el manejo del ventilador.

Capítulo 2

Marco teórico

En este capítulo se abordará el tema acerca de los puntos mas importantes que sustentan esta investigación y que permiten llevar a cabo este proyecto. Esto es posible mediante la definición y el entendimiento del concepto de IoT, cómo funciona, en qué se basa, entre otras. También se hablará sobre las generalidades del IoT y que se esperan en un futuro sobre esta arquitectura.

El control del motor es muy importante por lo tanto esta sección contiene información sobre automatización de motores eléctricos y como este motor eléctrico puede conectarse al Wifi mediante un microcontrolador ESP8266, así como también al hablar de microcontroladores se explica que es Arduino, el circuito que ayudará a controlar el motor y cómo lo hará.

Se habla un poco del Wifi y una breve explicación sobre su funcionamiento y que tiene que ver con el IoT.

2.1. IoT (*Internet of Things*)

Internet de las cosas o IoT (*Internet of Things*), se refiere a la conexión de objetos electrónicos a Internet, este concepto proviene del avance de la tecnología y de la necesidad de

compartir y gestionar las cosas que rodean al ser humano [11].

Existen bastantes áreas de implementación para el IoT, pero los campos donde se presenta fuertemente son: la adopción generalizada de redes basadas en el protocolo IP, la economía en la capacidad de cómputo, los avances en el análisis de datos y el surgimiento de la computación en la nube, salud, herramientas de aprendizaje, seguridad, optimización de procesos, agricultura, entre otros. En los últimos años, el concepto de IoT ha tenido varias definiciones, por lo que a continuación se citan algunas:

Coordination and Support Action for Global RFID-related Activities and Standardisation (CASAGRAS) define al Internet de las cosas como: “Una infraestructura de red global, que enlaza los objetos físicos y virtuales a través de la explotación de la captura de datos y capacidades de comunicación. Esta infraestructura incluye desarrollos existentes y en evolución del Internet y las redes. Ofrecerá identificación específica de objetos, sensores y la capacidad de conexión como base para el desarrollo de servicios cooperativos y aplicaciones independientes. Se caracterizarán por un alto grado de captura de datos de manera autónoma, transferencias de eventos, conectividad de red e interoperabilidad”[12].

Cluster of European Research projects on the Internet of Things (CERP-IoT); define el Internet de las cosas como “Una infraestructura de red global y dinámica con capacidades de auto configuración basada en estándares y protocolos de comunicación interoperables donde las cosas físicas y virtuales tienen identidades, atributos físicos y personalidades virtuales y utilizan interfaces inteligentes, y se integran a la perfección en la red de información”[12].

European Research Cluster for the Internet of Things (IERC) junto con el *ITU Telecommunication Standardization Sector (UIT-T)*, definen el IoT como: “Infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras”[12].

2.2. Generalidades del IoT

Para el *Cluster of European Research Projects*: “En la IoT se espera que “las cosas” conviertan a los participantes activos en procesos sociales, de información y negocios en donde “estas cosas” sean capaces de interactuar y comunicarse entre ellos mismos y con el ambiente a través del intercambio de datos e información detectada a su alrededor, mientras reaccionan automáticamente a los eventos del “mundo físico/real” y son influenciados a través de procesos en ejecución que disparan acciones y crean servicios con o sin la intervención directa del ser humano” [13].

En general, se puede afirmar que todos los objetos que integran o participan en Internet, puedan comunicarse desde cualquier lugar del mundo y en cualquier instante a través del conjunto de tecnologías de información y comunicación con la finalidad de brindar o utilizar servicios de la red que permitan el control y monitorización de dichos participantes en tiempo real o no de manera automática.

Las tecnologías que están dando paso al IoT han avanzado rápidamente, en los próximos años se espera ver más servicios que las exploten. Las tecnologías necesarias tendrán que facilitar la gestión de una gran cantidad de datos, captura, encaminamiento y análisis en función del tipo de aplicación ya sea en tiempo síncrono o asíncrono. La gestión de la gran cantidad de datos y la alta potencia de procesamiento se relaciona con las áreas de *Big Data* y *Cloud Computing*. El desafío está en la administración a gran escala de los recursos de detección, procesamiento de cómputo y almacenamiento. En el caso del volumen de datos que se espera genere el IoT se perfila con un crecimiento exponencial [13].

Las tecnologías manejadas por la IoT exigen un gran consumo de energía e incluso se están desarrollando propuestas de soluciones para garantizar la preservación del medio ambiente específicamente tales entornos.

En un mundo con sistemas IoT, los objetos tendrán conexiones inteligentes y ubicuas,

se espera que se tenga un mayor uso de la red Wifi y el acceso al Internet inalámbrico 4G-LTE (Long Term Evolution). Los sistemas serán complejos con elevado número de objetos interactuando alrededor de una persona, procesando en línea y en paralelo todos los eventos generados por cada objeto.

El IoT conectará distintos tipos de dispositivos como lo son el teléfono inteligente y electrodomésticos, los cuales se encontrarán en entornos inteligentes con capacidad de ejecutar funciones de manera automática. Entre las áreas con potencial aplicación se tienen: *Smart Cities* (y regiones), *Smart Car and mobility*, *Smart Home* y Asistencia Personal, *Smart Industries*, Seguridad Pública, Protección Ambiental y Energética, Agricultura y Turismo todas formando parte del IoT.

El dispositivo por excelencia a incentivar el uso de aplicaciones IoT será el teléfono inteligente. Actualmente, los teléfonos inteligentes están ya equipados con una serie de sensores y actuadores: cámara, el micrófono, acelerómetros, indicador de temperatura, altavoces, pantallas, etc. Pero se espera que existan otra serie de productos innovadores de detección portátiles.

En definitiva, las aplicaciones de los conceptos de perspectivas da un potencial crecimiento debido a los avances tecnológicos, pero al mismo tiempo habrá que confrontar varios retos como son la gestión de gran cantidad de datos, la falta de capacidad de procesamiento en línea, la gestión del consumo de energía, la interoperabilidad entre sensores y una falta de integración de sus componentes. Todo esto es fundamental para la explotación de los datos a través nuevos productos y servicios basados en la ubicuidad y en la innovación [13].

2.3. Microcontrolador ESP8266

El ESP8266 es un microcontrolador de bajo coste y muy utilizado en aplicaciones IoT y plataformas móviles. De forma general se podría decir que ESP8266 combina un procesador de

propósito general con un transceptor Wifi (Transmisor y receptor). Por ello se puede utilizar tanto como host de una aplicación (el procesador es el encargado de ejecutar la aplicación) que como un adaptador Wifi para cualquier otro procesador [14].

Cuando se utiliza como host el microcontrolador arranca desde una memoria flash externa. Por esta razón los módulos ESP-01 y ESP-12 incluyen una memoria flash de 8 Mbits, la BG25Q80A, donde residen los programas.

A continuación se listan las características más importantes que conforman el microcontrolador ESP8266:

- Arquitectura RISC de 32 bits.
- Memoria de programa de 64 KB.
- Memoria de datos de 96 KB.
- IEEE 802.11 b/g/n WiFi:
 - TR switch, RF balun, LNA, amplificador de potencia de RF y red de adaptación de impedancias.
 - Soporte de autenticación WEP y WPA/WPA2.
 - Pila TCP/IP.
- 16 pines GPIO.
- SPI, I2C.
- UART.
- Conversor ADC de 10-bit.

La programación y ejecución de los programas en el ESP8266 implica dos modos de funcionamiento diferentes, el modo UART y el Flash, respectivamente. Estos modos se configuran asignando diferentes combinaciones lógicas en dos de los terminales del dispositivo (GPIO0 y GPIO2).

En el modo UART para cargar un programa en el ESP8266, es necesario encenderlo o reiniciarlo (reset) teniendo el pin GPIO0 a nivel bajo ($LOW = 0\text{ V}$) y el GPIO2 a nivel alto ($HIGH = 3.3\text{ V}$). El pin GPIO2 está por defecto a *HIGH*, ya que tiene una resistencia de *pull-up* interna, por tanto, se puede dejar desconectado.

En el modo Flash una vez que el programa está cargado, es necesario ejecutarlo. Para ello hay que poner GPIO0 y GPIO2 a nivel alto ($HIGH = 3.3\text{ V}$). Tanto el GPIO0 como el GPIO2 están por defecto a *HIGH*, ya que tienen una resistencia de *pull-up* interna, por lo que pueden quedar desconectados.

En el caso del módulo ESP-01 el cambio de un modo de funcionamiento a otro, programación a ejecución y viceversa, se realiza de forma manual lo cual es una limitación importante. Sin embargo, en el ESP-12, el módulo proporciona la circuitería interna necesaria para poder hacer esta acción mediante un puerto USB, sin necesidad de modificar conexiones durante la programación y/o ejecución de un programa.

Este microcontrolador tiene un firmware que permite la programación mediante comandos AT. El funcionamiento de este microcontrolador programándolo con comandos AT es totalmente funcional, pudiendo realizar cualquier operación.

Todos los módulos de la familia ESP integran el microcontrolador ESP8266, pero en cada módulo varían los terminales del microcontrolador disponibles en el conector del módulo[14]. En el caso del ESP-01 los terminales disponibles son ocho:

- 1.GND.

- 2.GPIO2, entrada/salida de propósito general.

- 3.GPIO0, entrada/salida de propósito general.
- 4.RXD, donde se recibirán los datos de entrada del puerto serie. Trabaja a 3.3 V.
- 5.TXD, donde se transmitirán los datos hacia el puerto serie. Trabaja a 3.3 V.
- 6.CH_PD, pin para el apagado, activo a nivel bajo.
- 7.RESET", pin para el reset, activo a nivel bajo.
- 8.VCC, pin de alimentación. El rango de alimentación es de 3.3 a 3.6 voltios. La corriente suministrada en el momento de arranque debe ser superior a 200 mA.

2.4. Arduino

Arduino se conforma de tres cosas:

Una **placa hardware** libre que incorpora un microcontrolador programable y una serie de pines-hembra (los cuales están unidos internamente a las patillas de E/S del microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores y actuadores [15].

Cuando se habla de “placa hardware”, se esta refiriendo en concreto a una PCB (del inglés *printed circuit board*, o sea, placa de circuito impreso). Las PCBs son superficies fabricadas de un material no conductor (normalmente resinas de fibra de vidrio reforzada, cerámica o plástico) sobre las cuales aparecen laminadas ("pegadas") pistas de material conductor (normalmente cobre). Las PCBs se utilizan para conectar eléctricamente, a través de los caminos conductores, diferentes componentes electrónicos soldados a ella. Una PCB es la forma más compacta y estable de construir un circuito electrónico (en contraposición a una *breadboard*, *perfboard* o similar) pero, al contrario que estas, una vez fabricada, su diseño

es bastante difícil de modificar. Así pues, la placa Arduino no es más que una PCB que implementa un determinado diseño de circuitería interna.

Cuando se habla de “placa Arduino”, se debe especificar el modelo concreto, ya que existen varios modelos oficiales, cada una con diferentes características (como el tamaño físico, el número de pines-hembra ofrecidos, el modelo de microcontrolador incorporado y como consecuencia, entre otras cosas, la cantidad de memoria utilizable, etc.). Conviene conocer estas características para identificar qué placa Arduino es la que convendrá más en cada proyecto.

De todas formas, aunque puedan ser modelos específicos diferentes, los microcontroladores incorporados en las diferentes placas Arduino pertenecen todos a la misma “familia tecnológica”, por lo que su funcionamiento en realidad es bastante parecido entre sí. Todos los microcontroladores son de tipo AVR, una arquitectura de microcontroladores desarrollada y fabricada por la marca Atmel.

Un **software** (entorno de desarrollo) gratis, libre y multiplataforma (ya que funciona en Linux, MacOS y Windows) que se debe instalar en un ordenador y que permite escribir, verificar y guardar (“cargar”) en la memoria del microcontrolador de la placa Arduino el conjunto de instrucciones que se desea que este empiece a ejecutar. Es decir: permite programarlo. La manera estándar de conectar el computador con la placa Arduino para poder enviarle y grabarle dichas instrucciones es mediante un simple cable USB, gracias a que la mayoría de placas Arduino incorporan un conector de este tipo [15].

Los proyectos Arduino pueden ser autónomos o no. En el primer caso, una vez programado el microcontrolador, la placa no necesita estar conectada a ningún computador y puede funcionar autónomamente si dispone de alguna fuente de alimentación. En el segundo caso, la placa debe estar conectada de alguna forma permanente (por cable USB, por cable de red Ethernet, etc.) a un computador ejecutando algún software específico que permita la comunicación entre este y la placa y el intercambio de datos entre ambos dispositivos. Este

software específico se debe programar mediante algún lenguaje de programación estándar como Python, C, Java, Php, etc., y será independiente completamente del entorno de desarrollo Arduino, el cual no se necesitará más, una vez que la placa ya haya sido programada y esté en funcionamiento.

Un **lenguaje de programación** libre. Por “lenguaje de programación”, se entiende cualquier idioma artificial diseñado para expresar instrucciones (siguiendo unas determinadas reglas sintácticas) que pueden ser llevadas a cabo por máquinas. Concretamente dentro del lenguaje Arduino, se encuentran elementos parecidos a muchos otros lenguajes de programación existentes (como los bloques condicionales, los bloques repetitivos, las variables, etc.), así como también diferentes comandos asimismo llamados “órdenes” ó “funciones”, que permite especificar de una forma coherente y sin errores las instrucciones exactas que se quiere programar en el microcontrolador de la placa. Estos comandos se escriben mediante el entorno de desarrollo Arduino [15].

Con Arduino se pueden realizar multitud de proyectos de rango muy variado: desde robótica hasta domótica, pasando por monitorización de sensores ambientales, sistemas de navegación, telemática, etc. Realmente, las posibilidades de esta plataforma para el desarrollo de productos electrónicos son prácticamente infinitas y tan solo están limitadas por la imaginación del ser humano.

2.5. Wifi

Cuando se habla de tecnología Wifi, realmente se está haciendo referencia a la WI-FI Alliance. Se trata de una organización sin ánimo de lucro, que engloba a un amplio grupo de fabricantes, con el objetivo de promocionar el uso de la tecnología inalámbrica en redes de área local, y asegurando la compatibilidad entre fabricantes con base a los estándares IEEE 802.11 [16].

Las tecnologías Wifi pueden aplicarse en la red de acceso, dando lugar a los denominados *hotspot* o islas digitales [17].

Un *hotspot* es una área en la que existe conectividad inalámbrica basada en puntos de acceso Wifi y en la que se concentran cierto número de usuarios inalámbricos (aeropuertos, hoteles, etc.) y en la que se proporciona un servicio de acceso que, generalmente, es la conexión a Internet, aunque pueden existir otros.

La red de interconexión se encarga de conectar entre sí las diferentes islas digitales extendiendo así la cobertura de la red Wifi y, además, de la conexión a Internet. La comunicación entre estas islas digitales se puede hacer mediante varias tecnologías: VSAT, LMDS, Cable, ADSL, VDSL, PLC, etc. [17].

2.6. Automatización de motores eléctricos

Entre los controles remotos convencionales, las figuras del dispositivo, las configuraciones y los métodos de control son diferentes entre sí. Actualmente, los teléfonos inteligentes han sido muy utilizados por las personas. Controlar los aparatos eléctricos por teléfono inteligente tiene su racionalidad y necesidad de conveniencia. Los equipos domésticos, como computadoras, impresoras, monitores, proyectores y televisores, ya han brindado capacidades de red con conexión inalámbrica (Wifi) o cableada (Ethernet). En comparación con otras tecnologías inalámbricas locales y personales (por ejemplo, Bluetooth y ZigBee), los equipos Wifi (por ejemplo, teléfonos inteligentes y Wifi AP) se equipan comúnmente en la mayoría de las casas [18].

El motor tradicional de frecuencia fija requiere una gran corriente para iniciar el funcionamiento del motor. Cuando la temperatura cumple con la configuración del aire acondicionado (o refrigerador), el motor deja de funcionar, de lo contrario, reinicia la operación. Por lo tanto, la iteración de arranque y apagado del motor causa una utilización ineficiente de la energía

y un ruido de operación desagradable.

La principal ventaja del control al utilizar una señal digital es que las señales del maestro de control (es decir, el procesador) para controlar el esclavo (es decir, el motor) son todas digitales. Eso significa que no se requiere más conversión de señal entre los formatos de digital y analógico. Además, la capacidad anti-interferencia de la señal digital es mucho mejor que la de la analógica. La razón es que una condición de error de la señal digital incurre solo cuando la intensidad del ruido de interferencia es lo suficientemente fuerte como para tener un impacto sustancial que provoca que el nivel de voltaje de las señales cambie de alto a bajo (o de bajo a alto).

El sistema que controla el motor incluye una interfaz de control de usuario en el teléfono inteligente. A través de una conexión al punto de acceso Wifi, los mensajes de control del usuario pueden pasarse a la placa de desarrollo. El sistema incorporado procesará los mensajes recibidos por un co-diseño de hardware y software y luego generará señales a un módulo de potencia para impulsar el funcionamiento del motor.

Se integra el sistema de control remoto diseñado en un ventilador eléctrico comercial para implementar un ventilador eléctrico de tipo IoT controlado por teléfono inteligente. El usuario puede conectarse al servidor web basado en IP en el ventilador eléctrico para acceder a la página de control. Las configuraciones de control incluyen control de potencia, ajuste de velocidad y un modo de sabiduría que se implementa por hardware para hacer la velocidad de la espina de baja a alta, luego de alta a baja, iterativamente [18].

Capítulo 3

Desarrollo del Proyecto

En esta sección, se verá cómo se fue construyendo todo el sistema, desde la aplicación, hasta las pruebas en dicho motor eléctrico de 3 velocidades, el cual pertenece a un ventilador de techo doméstico, se explicará la modificación que se realizó al motor, el hardware que conformará el sistema, el diseño de la interfaz gráfica para la aplicación y del algoritmo de gestión de las velocidades del motor así como la estructura en lenguaje HTML de la aplicación web como el diseño estético de este mismo. Estos son los puntos principales que determinaron la construcción de dicho proyecto, que se verá a continuación.

3.1. Solución propuesta

La solución propuesta se definió como la introducción de una mejora tecnológica a lo que ya existe, con el apoyo del control de un motor de 3 velocidades de un ventilador de techo a través de una aplicación web, basada en la arquitectura del Internet de las cosas, con el ventilador y el teléfono conectados a la red Wifi del hogar.

3.2. Fases del desarrollo del proyecto

3.2.1. Diagrama eléctrico

En esta parte se explicará cómo se elaboró el diagrama eléctrico del motor, ya que es esencial para el funcionamiento con la aplicación.

Primeramente se obtuvo el motor a automatizar (Ver Figura 3.1), este motor corresponde a un ventilador de techo de 3 velocidades, el tradicional de uso doméstico en la mayoría de los hogares de hoy en día.



Figura 3.1: Motor a automatizar
Fuente: Elaboración propia.

Se procederá a abrir el motor, dejando al descubierto todos sus circuitos como se puede ver en la Figura 3.2, de esta manera se podrá observar que partes lo conforman como cables, circuitos, botones, interruptores, entre otros.

Para poder seguir con el objetivo de determinar el diagrama eléctrico del motor, se debe localizar cada una de las partes que conforman el motor, comenzando por el interruptor de 3 velocidades y encendido/apagado de este, como se puede apreciar en las Figuras 3.3 y 3.4.



Figura 3.2: Circuitos y cables del motor
Fuente: Elaboración propia.



Figura 3.3: Interruptor de 3 velocidades
Fuente: Elaboración propia.

Una vez localizado el interruptor de velocidades (Figura 3.5), se procederá a localizar el interruptor de encendido y apagado del foco, ya que la mayoría de los ventiladores de techo comúnmente poseen una luz, como funcionalidad de iluminación en la habitación que se haya instalado.



Figura 3.4: Extracción del interruptor de velocidades
Fuente: Elaboración propia.



Figura 3.5: Interruptor de apagado/encendido del foco
Fuente: Elaboración propia.

Ya que se identificaron las partes necesarias en las que se va a enfocar para el funcionamiento y control del motor, se procederá a determinar el diagrama eléctrico de este de la siguiente forma.

Se tomará el interruptor de velocidades como se muestra en la Figura 3.6, junto con los

cables que se encuentran soldados a él. Estos cables son los de color negro, morado, gris y café. El interruptor ya no será necesario para el funcionamiento del motor, ya que el propósito de éste proyecto es eliminar el control manual a través de la aplicación web.



Figura 3.6: Interruptor de velocidades del motor

Fuente: Elaboración propia.

Seguido de esto, se comenzará a remover los cables (Figura 3.7) que estan conectados al interruptor con cuidado de no dañarlos (negro, morado, cafe, gris). Teniendo bien identificados estos cables, el que estén por color nos va a facilitar el realizar el diagrama del motor.

Al retirar un cable es necesario etiquetarlo con el color del cable que estaba conectado en el interruptor (en cada orificio donde se desconectó el cable de color), esto para saber con mayor certeza donde estaba conectado cada cable. Con los cables desconectados del interruptor, se tomará el interruptor de velocidades, el cuál ayudará a saber de que manera se interconectan estos cables.

Se procederá a verificar donde el interruptor se encuentra abierto (es decir, la posición en el que el ventilador debe estar apagado), una vez hecho esto se necesitará de un multímetro para poder ver que pasa cuando se cambian las posiciones del interruptor, para este caso, se pondrá el multímetro en modo de continuidad y se procede a tirar del cordón para cambiar de



Figura 3.7: Cables que conducen la energía para darle distintas velocidades al motor

Fuente: Elaboración propia.

posición el interruptor y después se verifica la continuidad, en total deben de ser 4 posiciones: Una para la posición de apagado, la segunda para la posición de alta velocidad del ventilador, la tercera para la velocidad media y la última para la velocidad baja de este.

Una vez cambiado a la primera posición se comienza a probar las diferentes formas en las que puede estar el interruptor donde cada color hace continuidad como se puede apreciar en las siguientes Figuras (Figura 3.8 y Figura 3.9).

Se registraron todas las coincidencias en una tabla como en la siguiente (Figura 3.10). Se encontró que el cable negro es el común con los demás colores, por lo tanto, este es el puente entre cada uno de los cables a la hora de dar la velocidad al motor. Se determinó que este cable es el común ya que aparece en todas las combinaciones posibles que se llevaron a cabo en las pruebas de continuidad.

Otro punto importante que cabe resaltar es el detectar el capacitor principal del motor, el cual es el responsable de darle la potencia al motor según se encuentre el interruptor de velocidades, ya que contiene el diagrama de como se interconectan a la hora de manipular el interruptor de velocidades, es decir, será la base para poder encontrar el diagrama del motor



Figura 3.8: Revisión de continuidad en interruptor 1.
Fuente: Elaboración propia.



Figura 3.9: Revisión de continuidad en interruptor 2.
Fuente: Elaboración propia.

como se muestra en la Figura 3.11

Basándose en el diagrama que muestra el capacitor del motor se puede comenzar a determinar el diagrama eléctrico del motor, uniéndolo con las combinaciones que se obtuvieron a la hora de probar la continuidad del interruptor (Figura 3.12).

	V0	V1	V2	V3
Negro		Gris	Café/Morado	Morado
Gris		Negro		
Café			Negro/Morado	
Morado			Negro/Café	Negro

V1: Alta

V2: Media

V3: Baja

Común: Negro	V0	V1	V2	V3
Gris		X		
Café			X	
Morado			X	X

Figura 3.10: Combinaciones de continuidad del interruptor de velocidad.

Fuente: Elaboración propia.

3.2.2. Automatización

La manera tradicional en que un aparato se utiliza es yendo hasta él, e interactuando físicamente con este, presionando botones e interruptores que lo conforman para poder realizar las cosas o actividades para las que esta diseñado, sin embargo es mucho mejor poder operarlo a distancia a través del Internet de las cosas. En esta sección se verá la modificación que se hizo en el ventilador de techo para que este pueda ser controlado por una aplicación web.

Para comenzar, se debe analizar cómo entran las órdenes al ventilador para poder encender y apagar el foco y el motor con sus cambios de velocidad. El ventilador utiliza dos interruptores, los cuales basta con tirar de ellos para ponerlos en otra posición, por ejemplo, cuando se tira del interruptor de la luz, esta enciende, si se vuelve a tirar de ella la luz se apaga, para el motor funciona de la misma forma, se tira del interruptor una vez y este comienza en la velocidad mas baja, se vuelve a tirar por una segunda vez o tercera este cambiará de velocidad de media a alta respectivamente, tirando por una última vez el motor se detendrá



Figura 3.11: Capacitor del motor
Fuente: Elaboración propia.

y se pondrá en reposo.

Este es el único medio de entrada por el cual el ventilador recibe las órdenes para comenzar a funcionar, es decir, se tienen que manipular directamente y en el lugar donde se encuentra el ventilador para que funcione. Por lo tanto, es necesario reemplazar estos interruptores por el hardware que realizará la misma tarea pero de manera digital. Dicho hardware es un relevador arduino de 4 módulos como el que se puede observar en las Figuras 3.13 y 3.14.

Este relevador suplirá las funciones de los interruptores físicos, dicho funcionamiento será explicado más a fondo en la siguiente sección 3.2.3 "Hardware del sistema". Ahora para que estos relevadores puedan realizar la función de los interruptores se agregará un microcontrolador de arduino ESP8266, el cual dará la posibilidad de que el ventilador se

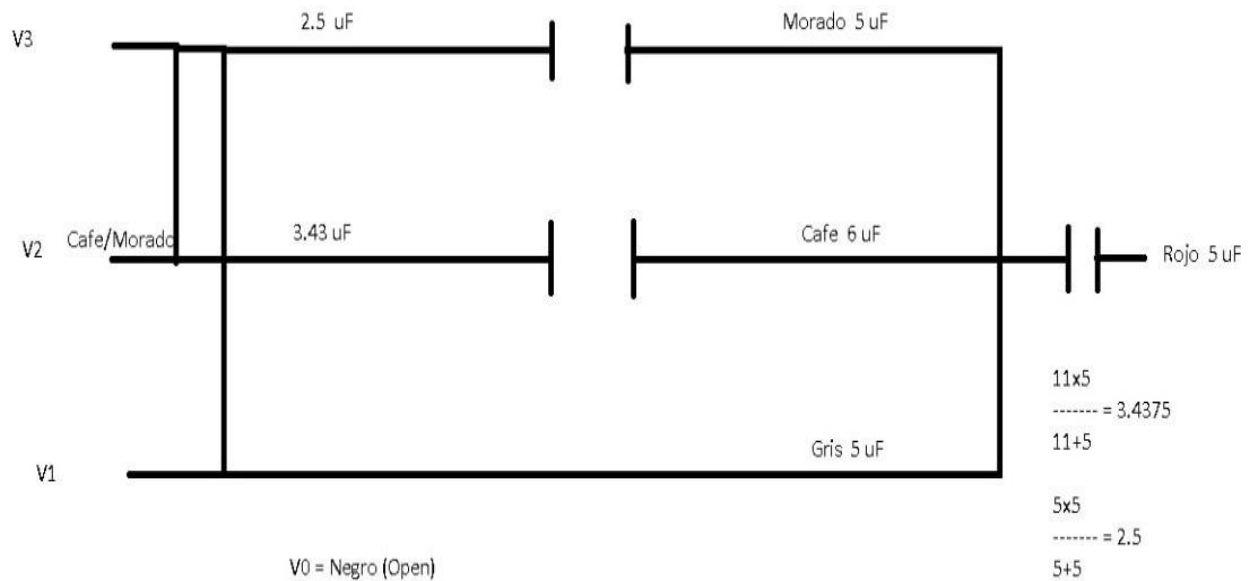


Figura 3.12: Diagrama eléctrico del motor
Fuente: Elaboración propia.



Figura 3.13: Relevador
Fuente: Elaboración propia.

conecte a red doméstica Wifi, funcionará como servidor donde corre la aplicación web que dará todas las órdenes al motor y el intercambio de estados entre cada módulo del relevador (Ver Figura 3.15).

Estas partes son las que se modificaron, se removieron los interruptores que venían de fábrica con el ventilador y se conectaron con las terminales de cada módulo del relevador,



Figura 3.14: Relevador
Fuente: Elaboración propia.



Figura 3.15: Microcontrolador ESP8266
Fuente: <https://www.steren.com.mx/catalogsearch/result/?q=esp8266>

este fue conectado al microcontrolador ESP8266 para poder ser controlado por este mismo y que el motor pueda conectarse al Wifi dando la función de servidor para poder controlarlo a través de la aplicación web.

3.2.3. Hardware del sistema

A continuación se mostrará el hardware que conforma el sistema, como dispositivos electrónicos y eléctricos, conforme se va ensamblando cada dispositivo.

Al principio se observó como optimizar el espacio donde se iban a introducir los dispositivos electrónicos y para ello se colocó cada dispositivo donde se encuentran los cables que conducen electricidad al motor, se observó como quedaría cada uno por lo cual la tablilla de los relevadores ocupaba mucho espacio y se procedió a rebajar el plástico que conforma la tablilla para optimizar el espacio (solamente donde no había pistas electrónicas), cuidadosamente sin dañar las partes funcionales de este. Se utilizó un pequeño pulidor para realizar los cortes como se muestra en la Figura 3.16.



Figura 3.16: Pulidor
Fuente: Elaboración propia.

La tablilla de relevadores a la cuál fue utilizada se puede observar en las Figuras (3.17 y 3.18).



Figura 3.17: Relevador
Fuente: Elaboración propia.



Figura 3.18: Relevador
Fuente: Elaboración propia.

Este relevador funciona de la siguiente manera: Ayudará a sustituir la función que realizan los interruptores que encienden y apagan el foco así como los que controlan el encendido/apagado del motor y sus velocidades, se eligió un relevador de 4 módulos ya que el primero llevará a cabo la conexión entre el encendido y el apagado del foco, y los otros tres realizarán la conexión de las velocidades del motor.

En este relevador se va a centrar en los 3 pines principales que se ven en la foto (conexiones que tienen los tornillos), los dos de las orillas son los que serán conectados entre sí, uno es llamado normalmente abierto, cuando no esta pasando corriente a través del circuito y el normalmente cerrado, donde está conectado con el brazo del relevador. Al momento de energizar el relevador mandándole una señal (para probar el relevador únicamente), este activa la bobina provocando un magnetismo que atraerá el brazo del normalmente abierto cerrando el circuito dejando pasar la corriente ahora por esta salida, para mejor entendimiento se puede observar la Figura 3.19.

Para evitar dañar cualquier pista de este circuito se observó de manera cuidadosa físicamente la tablilla (Figuras 3.20, 3.21 y 3.22) para ver por donde circulan pistas y donde no, y así poder cortar la tablilla como se comentó en párrafos anteriores.

Cuando se observó que partes del plástico estaban libres se procedió a realizar los cortes

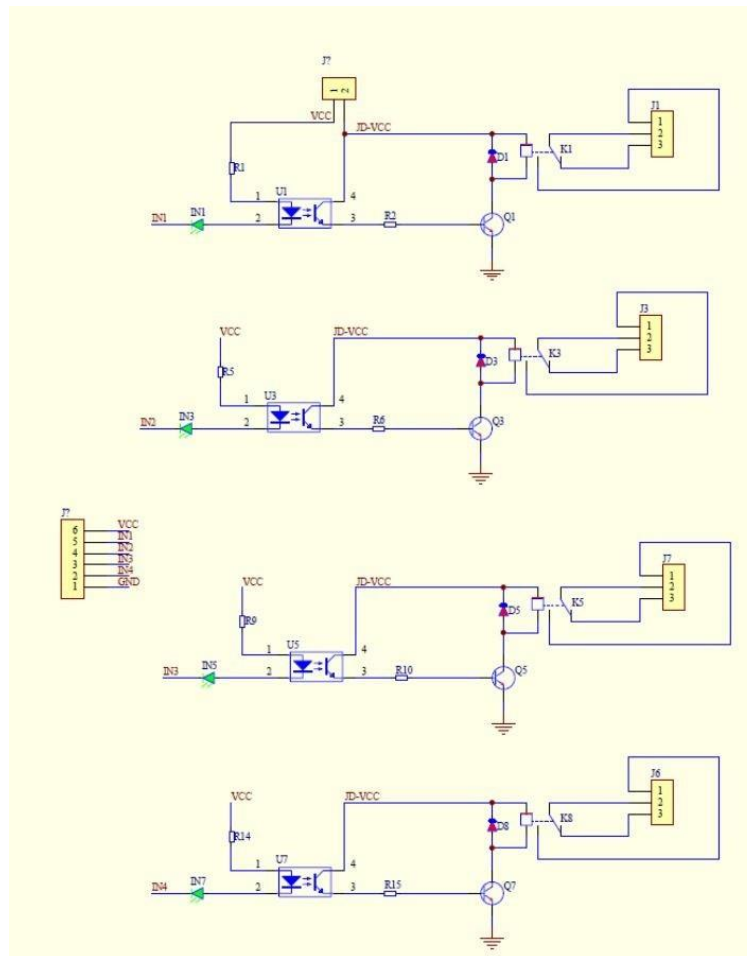


Figura 3.19: Circuito del relevador

Fuente: <http://arduinolearning.com/code/5-volt-4-channel-arduino-relay-module-example.php>

para optimizar el espacio a usar en el lugar donde se posicionará. Los cortes se observan en las Figuras 3.23, 3.24 y 3.25.

Prosiguiendo con el proceso se puso a prueba el funcionamiento del relevador, simulando en un protoboard las conexiones que tendrá el sistema y la programación del circuito ESP8266 que se explicará en la sección 3.2.6 de este documento.

Se energizó el circuito una vez todo conectado y con el multímetro en modo de medición de continuidad se verificó según el diagrama del circuito del relevador que haya continuidad,



Figura 3.20: Análisis físico de la tablilla
Fuente:Elaboración propia.



Figura 3.21: Análisis físico de la tablilla
Fuente:Elaboración propia.

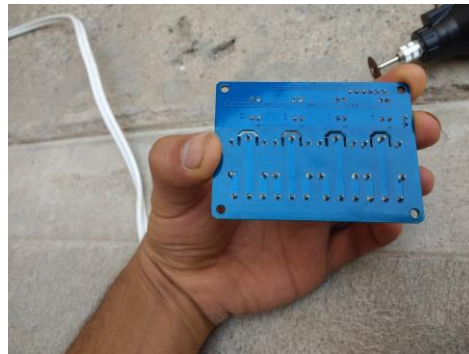


Figura 3.22: Análisis físico de la tablilla
Fuente:Elaboración propia.

esto se sabe en el momento en el que se escucha un clic dentro del relevador y el multímetro emite un ruido de "bip"(Ver Figuras 3.26, 3.27 y 3.28).

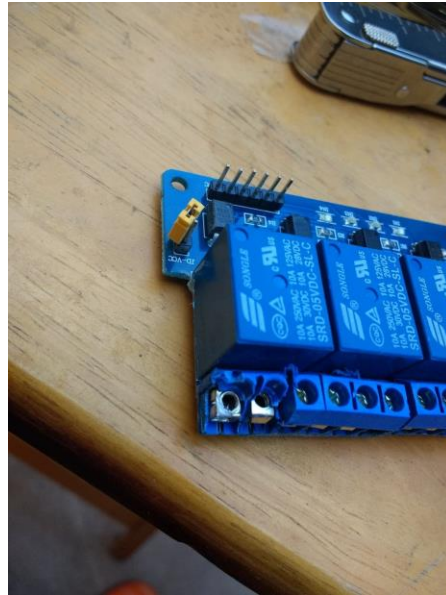


Figura 3.23: Cortes a la tablilla de relevador
Fuente:Elaboración propia.

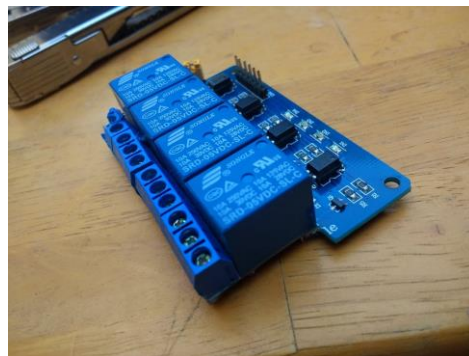


Figura 3.24: Cortes a la tablilla de relevador
Fuente:Elaboración propia.

En los distintos clics que se escucharon y en el proceso de prueba se comenzó con las lecturas de corriente, por lo que a cada clic (circuito cerrado), luego circuito abierto (segundo clic) y nuevamente circuito cerrado se obtuvieron estas lecturas respectivamente (cuando el circuito se encontraba abierto marcaba un uno. Ver Figuras 3.29, 3.30 y 3.31):

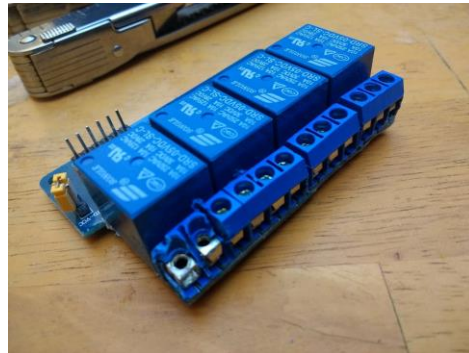


Figura 3.25: Cortes a la tablilla de relevador
Fuente:Elaboración propia.

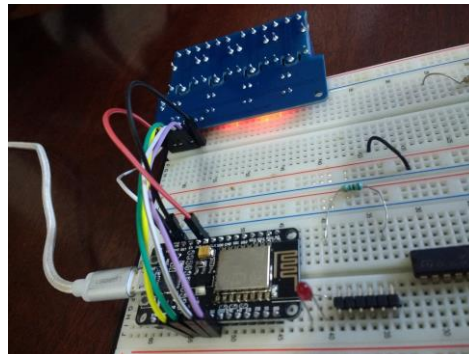


Figura 3.26: Pruebas de continuidad
Fuente:Elaboración propia.

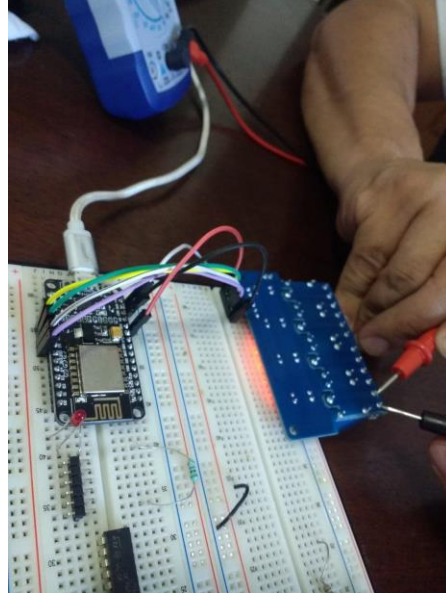


Figura 3.27: Pruebas de continuidad
Fuente:Elaboración propia.

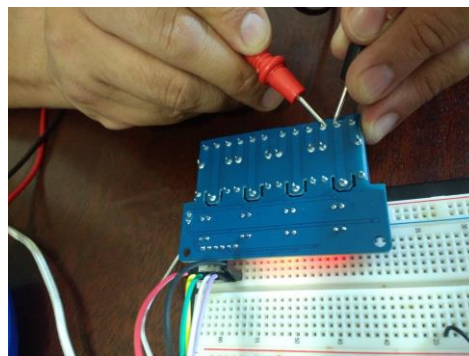


Figura 3.28: Pruebas de continuidad
Fuente:Elaboración propia.



Figura 3.29: Pruebas de corriente
Fuente:Elaboración propia.



Figura 3.30: Pruebas de corriente
Fuente:Elaboración propia.



Figura 3.31: Pruebas de corriente
Fuente:Elaboración propia.

Se hizo la prueba del otro lado del circuito y se obtuvo el mismo resultado como se puede observar en las Figuras siguientes (3.32, 3.33 y 3.34).

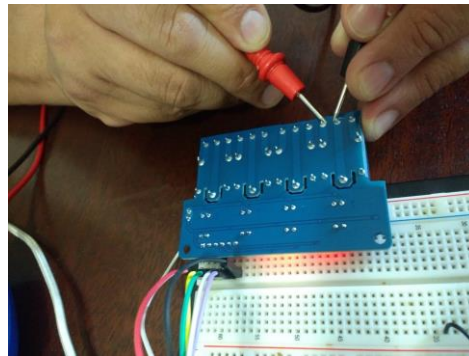


Figura 3.32: Pruebas de corriente
Fuente:Elaboración propia.

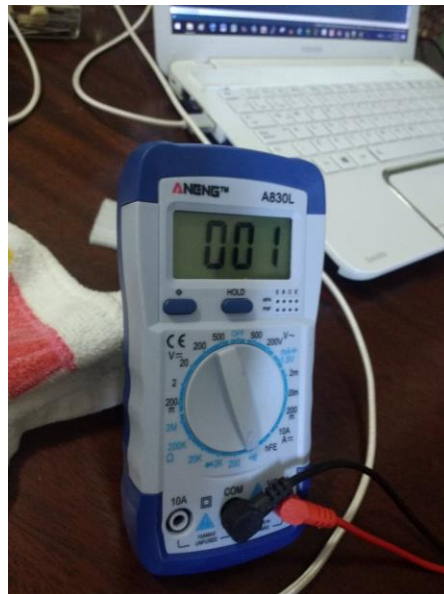


Figura 3.33: Pruebas de corriente
Fuente:Elaboración propia.



Figura 3.34: Pruebas de corriente
Fuente:Elaboración propia.

Después de probar el relevador y las conexiones, se procede a ver de que manera trabaja realizando la conexión directa al circuito de relevadores por cables, como se puede ver en la Figura 3.35 (diagrama de circuito) se muestran las conexiones del relevador al circuito ESP8266 de arduino (dispositivo el cual contiene las funciones principales de la aplicación, servidor y conexión al Wifi).

Utilizando este diagrama como base, se realizaron las conexiones necesarias en el circuito que ya se tenía en el protoboard, ver Figura 3.36. Como se puede apreciar en esta figura el circuito comenzó a ser probado nuevamente y se observa en el led de la tablilla, y en la segunda Figura otro número de leds encendidos (3.37).

En las siguientes dos figuras se puede observar otro patrón de leds encendidos ya que cada uno indica que relevador esta encendido (Figuras: 3.38 y 3.39).

Una vez hechas estas pruebas se comenzó a examinar la aplicación en el teléfono, como este smartphone ejecuta la aplicación web que esta corriendo en el circuito ESP8266, y así

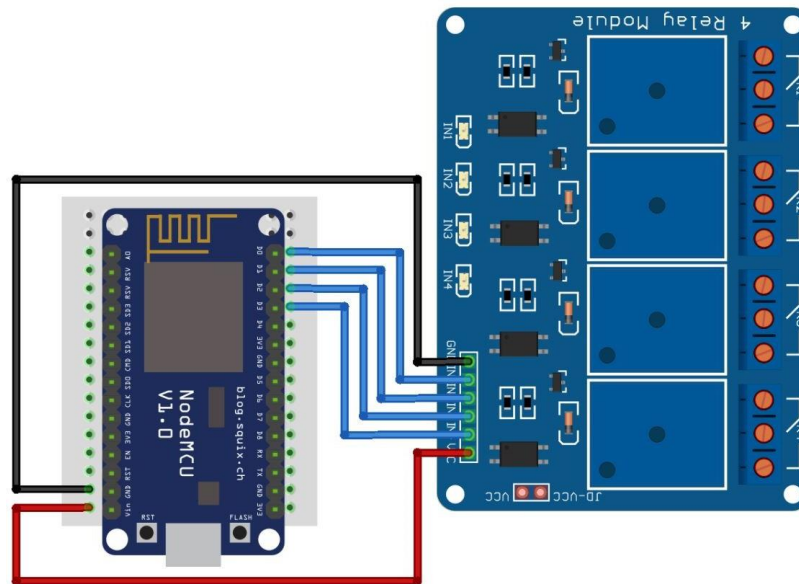


Figura 3.35: Conexiones entre el microcontrolador ESP8266 y los relevadores
Fuente:<http://invent.module143.com/home-automation-using-nodemcu-blynk/>

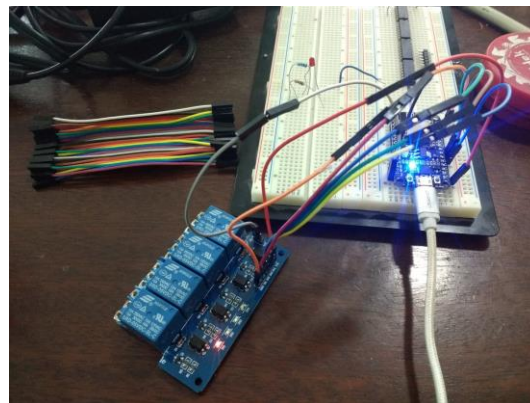


Figura 3.36: Conexiones entre el microcontrolador ESP8266 y los relevadores
Fuente:Elaboración propia.

observar cómo se comunican estos dispositivos, como se intercambia información entre estos.

Para ello se programó la aplicación web en HTML, como se puede ver en las siguientes Figuras (3.41, 3.42, 3.43 y 3.44) cada botón hace algo distinto, uno enciende el foco, otro controla las velocidades del motor hasta apagarlo y el último cumple la función de apagar

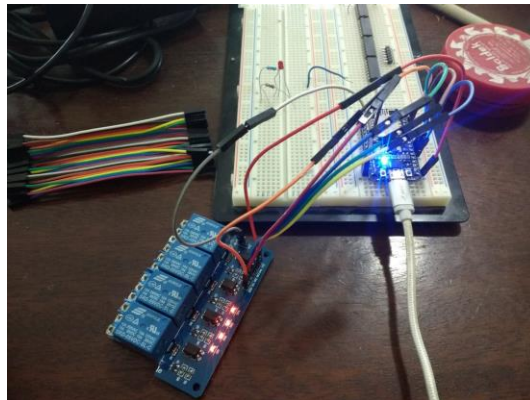


Figura 3.37: Pruebas de funcionamiento a relevadores
Fuente:Elaboración propia.

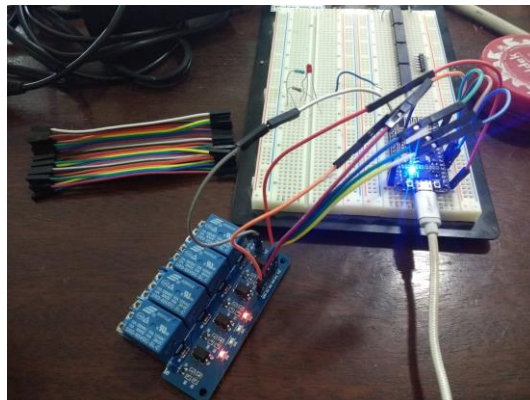


Figura 3.38: Conexiones entre el microcontrolador ESP8266 y los relevadores
Fuente:Elaboración propia.

todo el sistema.

De este modo, se puede probar nuevamente el funcionamiento de los relevadores pero ahora de forma manual, observando como responde a cada una de las órdenes que se le dan desde la aplicación web móvil. En la siguiente Figura (3.40) la aplicación se encuentra en su estado inicial, con el ventilador y todas las funciones apagadas.

Para continuar, se comenzaron a mandar señales para observar el comportamiento de los relevadores a través del smartphone y las funciones programadas para esta aplicación web

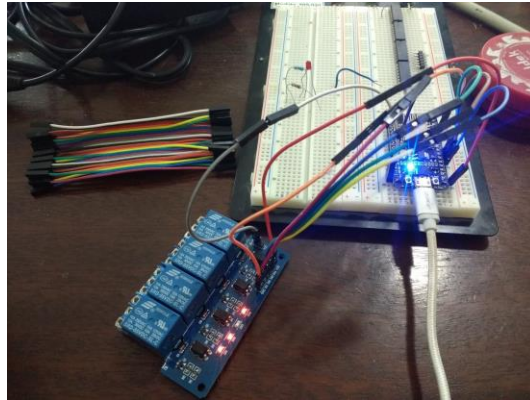


Figura 3.39: Pruebas de funcionamiento a relevadores
Fuente:Elaboración propia.

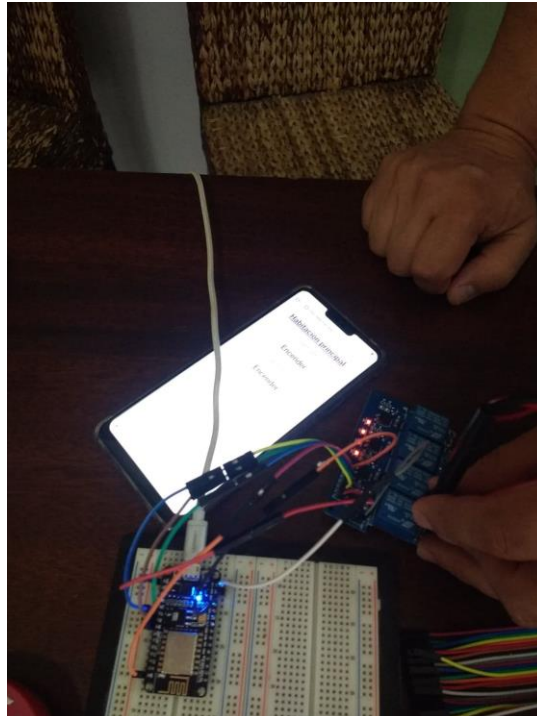


Figura 3.40: Pruebas de funcionamiento a relevadores en aplicación web
Fuente:Elaboración propia.

móvil.

Con estas pruebas de funcionamiento al relevador junto con la aplicación se tienen iden-

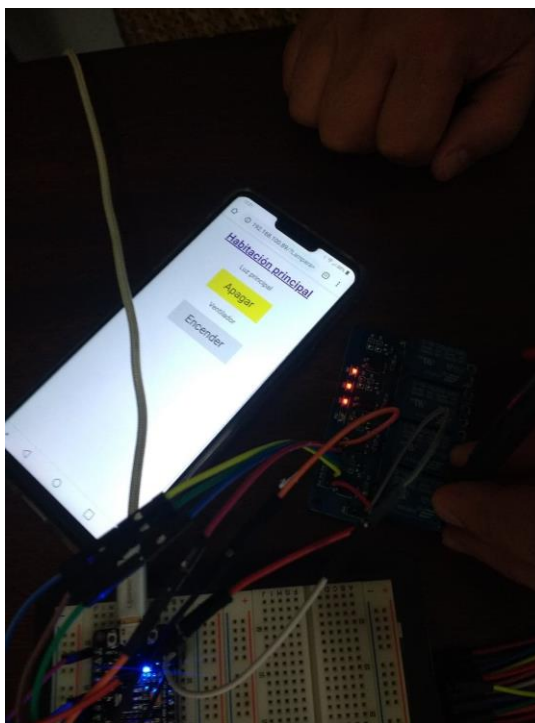


Figura 3.41: Posición en la que debe estar encendido el foco
Fuente:Elaboración propia.

tificados los lugares donde deben ir conectados los cables, por lo cual se comenzó a soldar estos cables en las terminales de los relevadores como se aprecia en las figuras (3.45 y 3.46).

Las terminales que donde no fue posible fijar con soldadura se utilizó pegamento de silicon caliente, como se puede apreciar en la Figura 3.47, para no dañar el material (plástico). También se fijó la terminal que proporcionará energía eléctrica al motor (Ver Figura 3.48).

Para finalizar con la fase de armado del hardware de todo el sistema, se probó todo en conjunto, el microcontrolador ESP8266, los relevadores, el teléfono con la aplicación, todo conectado al Wifi, y se logró encender exitosamente el foco y el encendido/apagado con las velocidades del motor (Ver Figuras: 3.49 y 3.50).

Por último, se juntan todos los componentes y se posicionan cuidadosamente en el espacio dentro de la carcasa del motor (donde van todos los circuitos) y cerrar la tapa con sus tornillos

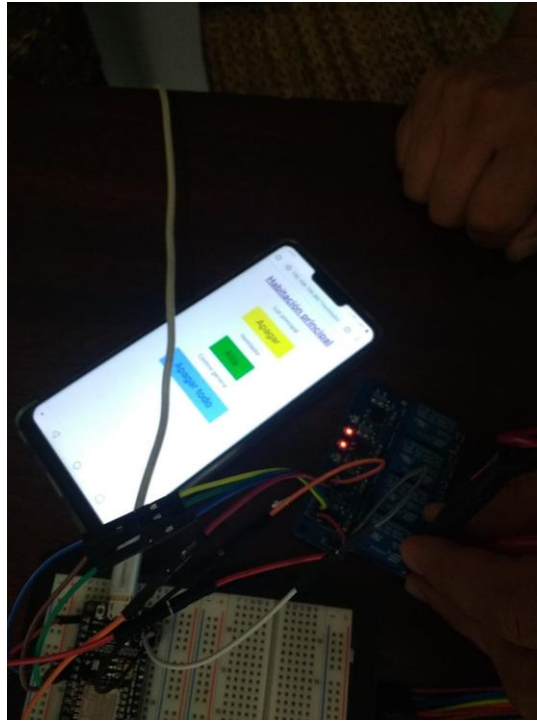


Figura 3.42: Posición en la que esta encendido el foco y la velocidad alta del motor
Fuente:Elaboración propia.

correspondientes para poder proteger todo el hardware del sistema (Figura: 3.51).

3.2.4. Requerimientos

Prosiguiendo con la realización de este proyecto, es necesario llevar a cabo un análisis sobre los aspectos necesarios que debe cubrir el funcionamiento del sistema, en este caso específico la aplicación móvil. En este apartado, se verá el análisis de los requisitos necesarios para la aplicación, tales como funcionalidad, comunicación, interfaz gráfica de usuario, entre otras.

Todo esto con el fin de cumplir con el propósito del presente proyecto, con base a funcionalidad y la detección y corrección de posibles errores que se puedan presentar a la hora de intentar operar el ventilador.

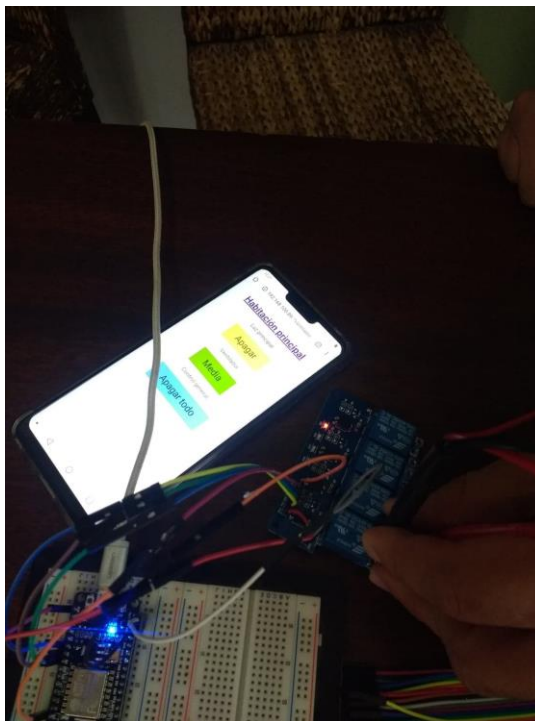


Figura 3.43: Posición en la que esta encendido el foco y la velocidad media del motor
Fuente:Elaboración propia.

Comenzando, el propósito de la aplicación es el poder controlar el ventilador, por ende significa que va a sustituir los controles manuales incluidos por defecto en el ventilador (control de encendido/apagado del foco y el control de velocidades del motor).

Primeramente es necesario incluir botones que den la posibilidad de que al presionarlos puedan efectuar las funciones que realizan los interruptores físicos del ventilador. Se integrará un botón el cual pueda encender y apagar el foco, realizándolo de la forma que indique cuando este se encuentre encendido, cambiando el color del botón y las letras de este, por ejemplo dicho botón se encuentra en apagado, es decir, en color gris y con la leyenda "encender", al momento de presionar cambiará de color gris a amarillo y la palabra ahora dirá "apagar". Para los botones que controlan las velocidades la idea será esencialmente la misma pero intercambiará entre tres opciones, la primera es como el primer botón (encendido/apagado del motor), estará en la opción de "encender" con el botón en color gris, al presionar este

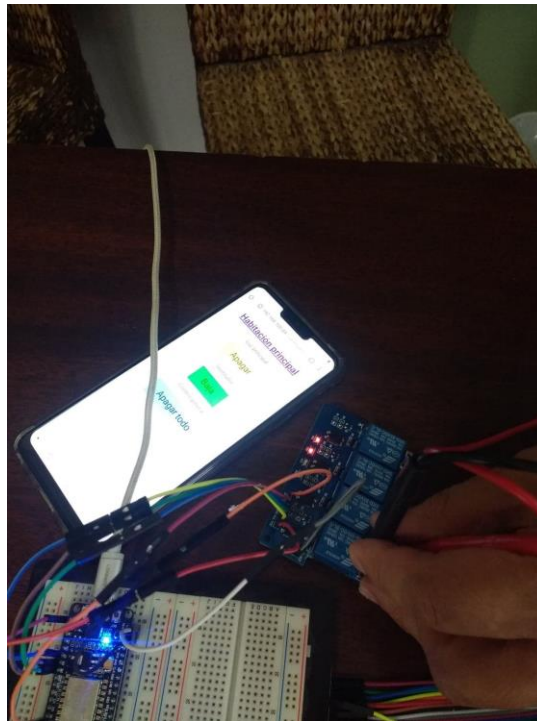


Figura 3.44: Posición en la que esta encendido el foco y la velocidad baja del motor
Fuente:Elaboración propia.

deberá cambiar al color verde, la intensidad de tal color indicará la velocidad en la que se encuentra junto con el cambio de la leyenda, el cual dirá alta, media o baja según se haya presionado el botón, entre cada color cambiará de verde, a verde limón y por último un verde con un poco más de saturación pero un poco mas claro, respectivamente según la velocidad en la que se encuentre funcionando el motor.

Este análisis inicial sobre el funcionamiento general del ventilador ayudarán a controlar el motor, sin embargo, se incluirá un botón extra, el cual ayudará a apagar por completo el ventilador, tanto foco, como el motor exactamente con la misma lógica que tienen los botones para el foco y las velocidades.

Un punto importante a tomar en cuenta es especificar de que manera la aplicación le dará dichas órdenes al ventilador, por lo cual, como ya se sabe el sistema funciona sobre el Internet



Figura 3.45: Solado de cables a terminales del relevador
Fuente:Elaboración propia.



Figura 3.46: Solado de cables a terminales del relevador
Fuente:Elaboración propia.

de las cosas. La aplicación usará la conexión a la red Wifi del hogar (Red LAN) mediante el teléfono móvil y el ventilador se conectará también a dicha red a través del microcontrolador ESP8266 el cual esta programado para controlar las funciones de este.

Además dicha aplicación será web, ya que es requerida la facilidad de acceso al control del ventilador por cualquier persona que se encuentre en el hogar y conectado a la red Wifi,



Figura 3.47: Pegado de cables a terminales del relevador
Fuente:Elaboración propia.



Figura 3.48: Pegado de cables a terminales del relevador
Fuente:Elaboración propia.

sin necesidad de perder el tiempo en instalación y configuración del teléfono. Por lo tanto, el microcontrolador ESP8266 funcionará como servidor, ya que este contiene el algoritmo de control y la aplicación web móvil para poder manejar el ventilador.

Para finalizar, la manera de acceder a dicha aplicación se necesita que sea a través de la red Wifi del hogar (Red LAN) y como el microcontrolador funcionará como servidor se deberá



Figura 3.49: Prueba en conjunto
Fuente:Elaboración propia.



Figura 3.50: Prueba en conjunto
Fuente:Elaboración propia.

acceder a esta mediante una dirección IP local (por ejemplo: 192.168.100.89), primeramente abriendo algún navegador instalado en el móvil y escribiendo dicha dirección en el buscador de este y abrirá la aplicación que controlará el ventilador.



Figura 3.51: Guardado de circuitería
Fuente:Elaboración propia.

3.2.5. Interfaz gráfica

Al momento de utilizar una aplicación web móvil, que es similar al uso de una aplicación móvil para cualquier sistema operativo, se vuelve importante el aspecto de la facilidad de uso, el entendimiento y que tan intuitivo es un sistema para su uso. En esta sección se verá el análisis para determinar el diseño de la interfaz gráfica que tendrá la aplicación.

El diseño estará centrado en el uso de botones para el control de las funciones de este ventilador, es necesario que cada botón realice la función que tienen los interruptores físicos, por lo que se añadirá 3 botones en la interfaz de la aplicación.

Se comenzará con el botón del foco, luego el de las velocidades y al final un botón extra que pueda apagar todo en el sistema, todo en dicho orden, para una mejor visualización serán posicionados en el centro superior de la pantalla.

El tamaño de los botones será de tamaño grande, es necesario de esta manera ya que se dirige al mayor tipo de usuarios. Los botones tendrán un poco de movimiento de un lado hacia otro al intercambiar entre valor (apagado/encendido o entre las velocidades) junto con el color. Inicialmente estarán en color gris, para indicar que el ventilador se encuentra apagado totalmente (Ver Figura 3.52).

Se requiere de una interfaz gráfica únicamente con los elementos necesarios para realizar

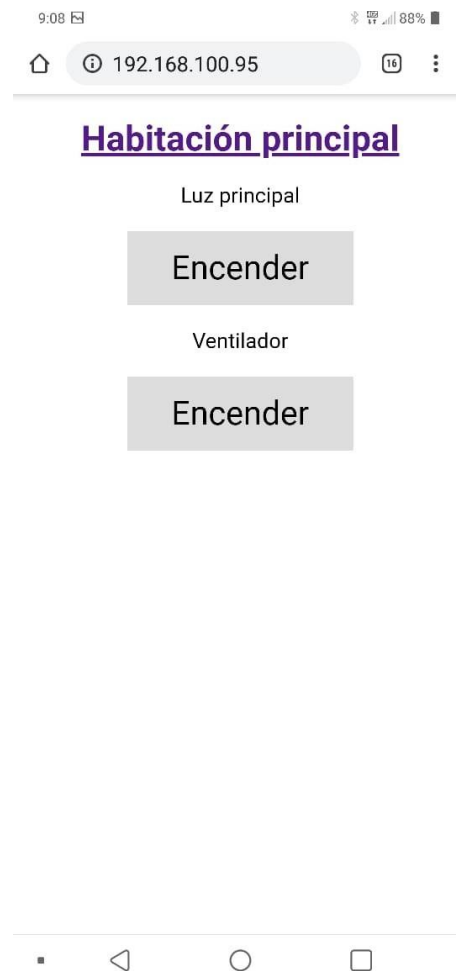


Figura 3.52: Vista inicial de la aplicación web
Fuente:Elaboración propia.

las funciones principales de este sistema como lo son el encendido/apagado del ventilador y el foco así como el apagado total de todo el sistema.

El siguiente paso en el diseño de la interfaz gráfica es el color en el botón de encendido y apagado del foco con el cambio de color de este de gris a amarillo y si se apaga pasará de amarillo a gris, esto con el fin de que el usuario pueda ver cuando el foco indica que esta encendido y cuando no (Ver Figura 3.53).

Si es requerido el encendido ahora del motor, debe presionarse el botón de encendido del

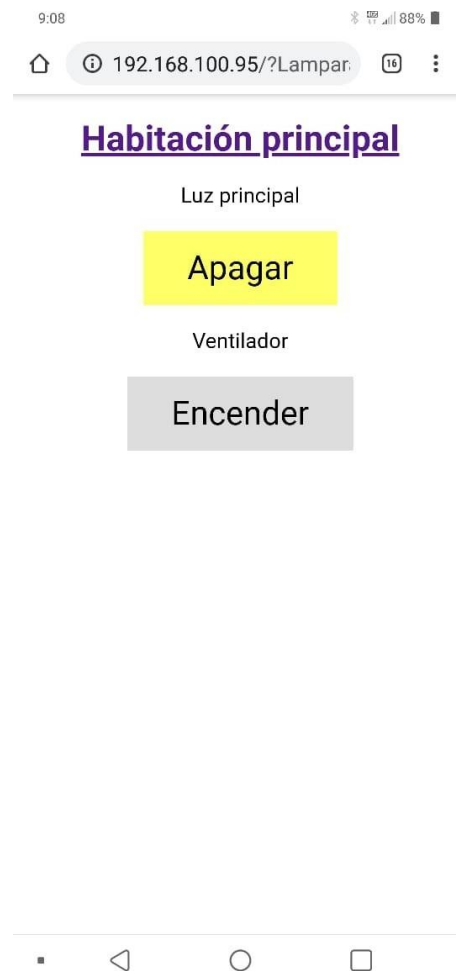


Figura 3.53: Color amarillo en botón que indica encendido del foco
Fuente:Elaboración propia.

ventilador, este funcionará con la misma lógica del botón anterior, cambiará de un color a otro al encender, comenzando con la velocidad mas alta del motor con un color que indique en que velocidad se encuentra (verde bandera). Además de que tendrá un movimiento al intercambiar de valor y color (verde limón y verde fosforescente), se agregará el botón extra que ya se mencionó, que es el de apagado total de todo el sistema, este no cerrará ni saldrá de la aplicación, únicamente apagará todo el ventilador incluyendo el foco (Ver Figura 3.54).

Para intercambiar entre cada velocidad funciona de la misma forma que si fuera el in-



Figura 3.54: Encendido del motor en velocidad alta y aparición del botón de apagado total
Fuente:Elaboración propia.

terruptor físico que traen por defecto, por lo tanto, presionando el botón una segunda vez cambiará a la velocidad media y si se vuelve a presionar por tercera vez cambiará a la velocidad baja (botón de ventilador), de la misma forma que la primera vez que se presiona tendrá un ligero movimiento y cambio de color (Ver Figuras 3.55).

Tanto el botón de encendido del foco como el de las velocidades del motor, si se presionan una vez más (segunda vez en el botón del foco y cuarta vez en ventilador) vuelven a su estado inicial en color gris y con el valor de encender (Ver Figuras 3.57 y 3.58).



Figura 3.55: Cambio de color y valor del botón entre cambio de velocidades
Fuente:Elaboración propia.

Para el funcionamiento del motor no es necesario que esté encendido el foco, esto para cuando se le de un uso en el día, de este modo funciona exactamente igual que todo encendido en conjunto pero únicamente con el ventilador encendido (ver Figuras 3.59, 3.60 y 3.61).

Para finalizar con las pruebas del diseño de esta interfaz gráfica, se puede ver en la siguiente figura (ver Figura 3.58) que al presionar el botón de apagar este debe de desaparecer de la pantalla quedando únicamente los botones de encendido del foco y ventilador.



Figura 3.56: Cambio de color y valor del botón entre cambio de velocidades
Fuente:Elaboración propia.

3.2.6. Aplicación

En este sistema se controlará un motor eléctrico de 3 velocidades, el cual requiere de una aplicación que lo controle, una interfaz con la que el usuario pueda interactuar y así poder controlar dicho motor. Por lo tanto se desarrollará una aplicación web en HTML que en conjunto con el código en C de arduino (código que contiene el algoritmo de control) producirá las tareas que se requieren para controlar el motor del ventilador.

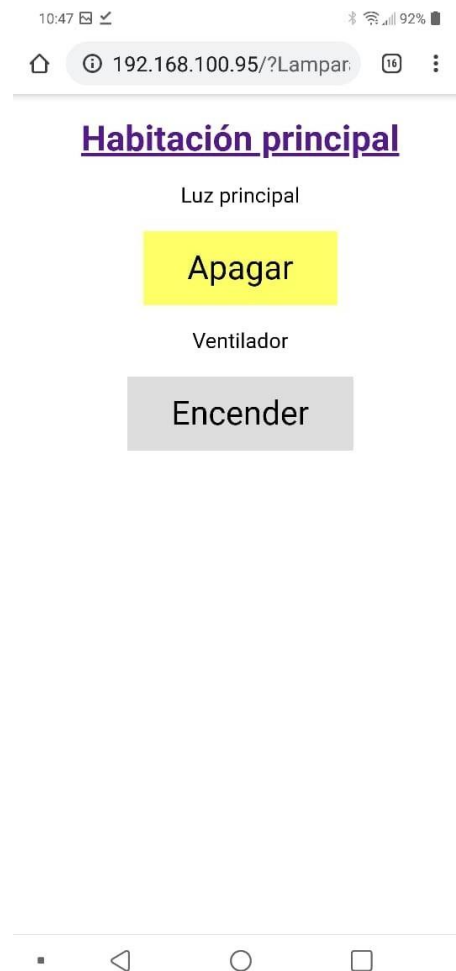


Figura 3.57: Regreso a estado inicial de los botones
Fuente:Elaboración propia.

Primeramente como en todo código HTML se tienen las cabeceras las cuales definen el tipo de documento HTML, versión, entre otras cosas. En la primera línea de esta cabecera se especifica el tipo de contenido del documento HTML y en la siguiente sentencia se comienza una escala de medida para la ventana en la cual el contenido no se deforme para el mayor número de tamaños para cada pantalla (ver Figura 3.62).

En el siguiente bloque de código se encuentra la definición del estilo CSS para la vista y el diseño que se le dará a la estructura HTML, la cual cuenta con la especificación de una

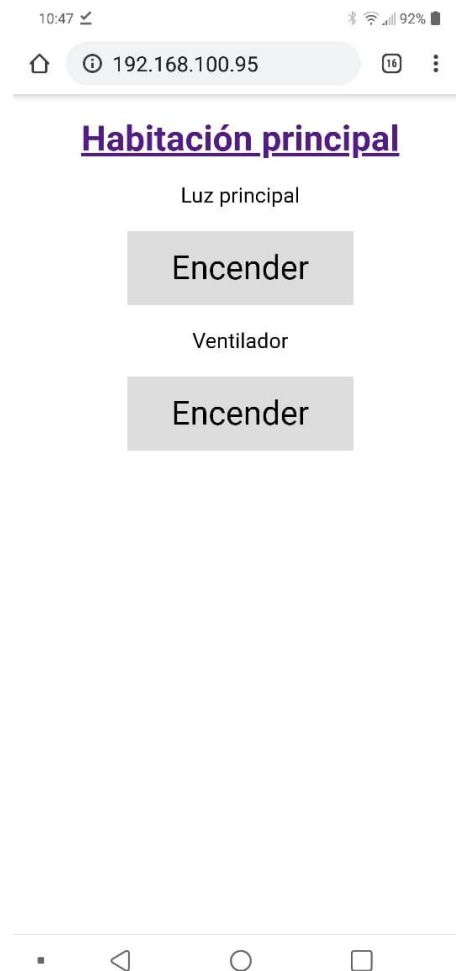


Figura 3.58: Regreso a estado inicial de los botones
Fuente:Elaboración propia.

animación para los botones que dará un movimiento entre cambio de valor del botón que irá disminuyendo su margen izquierdo, provocando el movimiento deseado.

Al bloque de código CSS para el cuerpo HTML, tiene una serie de definiciones las cuales especifican el tipo de letra, eliminación de márgenes así como la alineación del texto y de como aparezcan los elementos en la pantalla. Esto con el fin de que todo aparezca en el centro de la pantalla al cargar la aplicación y que no se deforme en caso de cambiar las dimensiones de la pantalla.

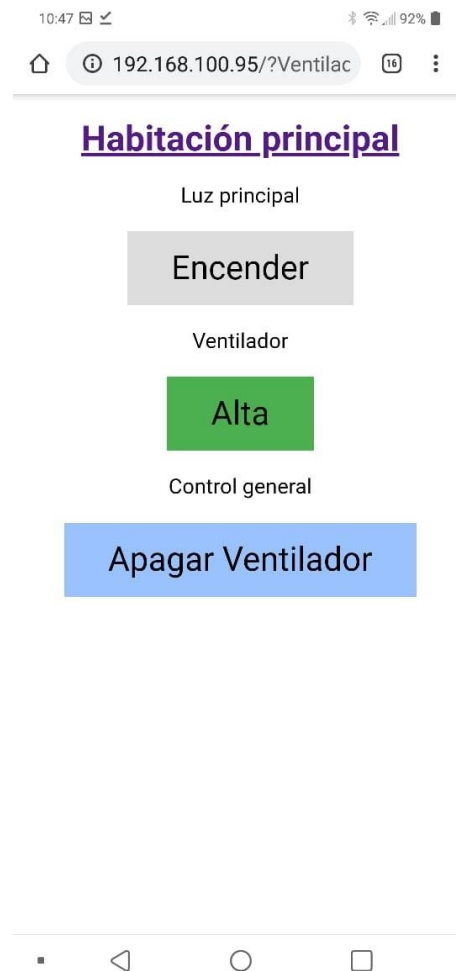


Figura 3.59: Funcionamiento del ventilador sin el foco
Fuente:Elaboración propia.

Y para finalizar con los estilos CSS, en el último bloque de código de este, las sentencias le dan estilos a los botones, se les da color, tamaño, bordes, márgenes, una decoración al texto y tamaño de fuente tanto a los botones, como a cada párrafo que se vaya a poner en el documento HTML (ver Figura 3.63).

Como siguiente diseño de la aplicación se comienza a programar la estructura del HTML, primeramente se programa un h1 (encabezado de tamaño 1, el mas grande) para poner como título "Habitación principal", esto se puede modificar según la ubicación donde se encuentre



Figura 3.60: Funcionamiento del ventilador sin el foco
Fuente:Elaboración propia.

el ventilador y seguido de otra leyenda la cual indica el objeto que se va a prender en este caso el título "Luz Principal". Cuando inicia la aplicación, la variable lámpara (se explicará en la siguiente sección 3.2.7) se encuentra en un estado inicial 0, por lo que si es así, en la aplicación web pondrá el botón en color gris con su leyenda "Encender", si se encuentra en cualquier otro estado, se cambiará el color y leyenda del botón a amarillo y "Apagar". Al final de la figura, se observa en la última línea el título escrito que es para el siguiente botón que es "Ventilador"(ver Figura 3.64).



Figura 3.61: Funcionamiento del ventilador sin el foco

Fuente:Elaboración propia.

```
header = "<!DOCTYPE html><html>\n<head>\n";
header += "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\">\n";
header += "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\n";
header += "<link rel=\"icon\" href=\"data:,\">>\n";
```

Figura 3.62: Cabecera del HTML

Fuente:Elaboración propia.

La lógica para el botón que controla las velocidades del ventilador es exactamente la misma que la de la lámpara, sin embargo en este caso se usa entre tres valores diferentes

```
header += "<style>\n";
header += ".saludos{animation-duration: 2s;animation-name: slidein;}
@keyframes slidein {from {margin-left: 100%;width: 300%}to{margin-left: 0%;width: 100%;}}\n";
header += "html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";
header += ".button { background-color: #4CAF50; border: none; color: black; padding: 16px 40px;\n";
header += "text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}\n";
header += ".button2 {background-color: #FFFF66;}.button3 {background-color: #99c2ff;}.button4 {background-color: #dddddd;}.button5 {background-color: #00ff00;}.button6 {background-color: #99cc00;}p{font-size: 1.2em;}\n";
header += "</style>\n";
header += "<title>Habitación principal</title>\n";
```

Figura 3.63: Código de estilos

Fuente:Elaboración propia.

```
header += "</head>\n";
header += "<body><h1><a href='\"http://\"';header +=Wifi.localIP().toString();header += \"\">Habitación principal</a></h1>\n";
header += "<p>Luz principal</p>\n";
if(Lampara == 0)
    header += "<p><a href='\"/?Lampara=on\"'><button class='\"button button4\"'>Encender</button></a></p>\n";
else
    header += "<p><a href='\"/?Lampara=off\"'><button class='\"button button2\"'>Apagar</button></a></p>\n";
header += "<p>Ventilador</p>\n";
```

Figura 3.64: Botones de Luz principal

Fuente:Elaboración propia.

y tres colores diferentes para indicar en que velocidad se encuentra. Como en el botón de encendido/apagado de la luz principal, se encuentra en estado 0, por lo tanto, este estará en color gris y con el valor de "Encender", una vez sea presionado, cambiará a valor 3, color verde y leyenda "Alta", y a cada presión al botón cambiará de valor, si se presiona otra vez, cambiará a valor 2 y por tercera vez a valor 3, lo cual respectivamente cambiará a color verde limón con leyenda "Media" y por último a color verde fosforescente con leyenda "Baja"(Ver figura 3.65).

Para finalizar con el desarrollo de la aplicación web, se agregó un botón extra a los que ya desarrollan la función principal que hacían los interruptores físicos, este botón es un interruptor de apagado general, el cual vuelve a los estados iniciales las variables de ventilador y lámpara a 0 y que todo en el sistema se apague, este aparecerá únicamente cuando el ventilador se encuentre encendido. Este tendrá dos valores que serán "Apagar todo" y "Apagar ventilador", este último para el caso en que se encuentre encendido solo el ventilador. Al apagar todo este desaparecerá y se quedaran únicamente los botones de

```

switch(Ventilador)
{
    case 3:
        if(Vent3)
        {
            header += "<p><span class='saludos'><a href='\"/?Ventilador=2\"><button class='button '\">Alta</button></a></span></p>\n";
            Vent3 = 0;
        }
        else
            header += "<p><a href='\"/?Ventilador=2\"><button class='button '\">Alta</button></a></p>\n";
        break;
    case 2:
        header += "<p><a href='\"/?Ventilador=1\"><button class='button button6\">Media</button></a></p>\n";
        break;
    case 1:
        header += "<p><a href='\"/?Ventilador=0\"><button class='button button5\">Baja</button></a></p>\n";
        break;
    case 0:
        header += "<p><a href='\"/?Ventilador=3\"><button class='button button4\">Encender</button></a></p>\n";
        break;
}

```

Figura 3.65: Botones control del ventilador

Fuente:Elaboración propia.

encendido/apagado del ventilador y la luz principal en su estado inicial (ver Figura 3.66).

```

if(Ventilador)
{
    header += "<p>Control general</p>\n";
    if(Lampara)
        header += "<p><a href='\"/?Ventilador=0&Lampara=off\"><button class='button button3\">Apagar todo</button></a></p>\n";
    else
        if(Ventilador != 1)
            header += "<p><a href='\"/?Ventilador=0\"><button class='button button3\">Apagar Ventilador</button></a></p>\n";
}
header += "</body>\n";
header += "</html>\n";

```

Figura 3.66: Botón de apagado total

Fuente:Elaboración propia.

3.2.7. Algoritmo de control

La correcta y eficaz ejecución de un programa se encuentra en el análisis y diseño del algoritmo (pasos a seguir) para poder resolver un problema de la mejor manera. En esta sección se explicará como funciona el algoritmo programado en arduino para el microcontrolador ESP8266.

Como primer paso y en todo aspecto en el que sea requerido realizar algún proyecto, se

deben analizar las herramientas necesarias para poder realizar dicho trabajo, en programación lo que se necesitará como herramientas son las librerías estándar que contienen las funciones que se necesitarán a la hora de programar las acciones que este programa debe hacer (ver Figura 3.67).

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
ESP8266WebServer server(80);
String header;
```

Figura 3.67: Librerías
Fuente:Elaboración propia.

En el anterior bloque de código se incluyen las librerías que permitirán que el microcontrolador ESP8266 realice la función de servidor web y pueda conectarse al Wifi. También se incluyó una variable de tipo string la cual al final de este código, guardará el código HTML que será desplegado al ejecutar la aplicación. Una vez incluidas estas librerías se comenzó a realizar las configuraciones correspondientes en cuanto a la red, asignando una dirección IP (dirección por la cual se accederá a la aplicación web mediante un navegador de Internet), especificación de la puerta de enlace y máscara de subred (ver Figura 3.68).

```
IPAddress ip(192, 168, 100, 89);
IPAddress gateway(192, 168, 100, 1);
IPAddress subnet(255, 255, 255, 0);
```

Figura 3.68: Configuraciones de direccionamiento
Fuente:Elaboración propia.

En el siguiente segmento de código se crean variables constantes (ver figura 3.69) las cuales indican el color del cable que maneja las velocidades según los encontrados en la sección de

”Hardware del sistema” y se les asigna un número entero el cual indica el número de pin al que se encuentra conectado cada cable de color (ver Figura 3.70).

```
// Assign output variables to GPIO pins
const int Vent_Gris = 16;
const int Vent_Cafe = 14;
const int Vent_Morado = 12;
const int Lampara_Control = 13;
```

Figura 3.69: Constantes a números de pin

Fuente:Elaboración propia.

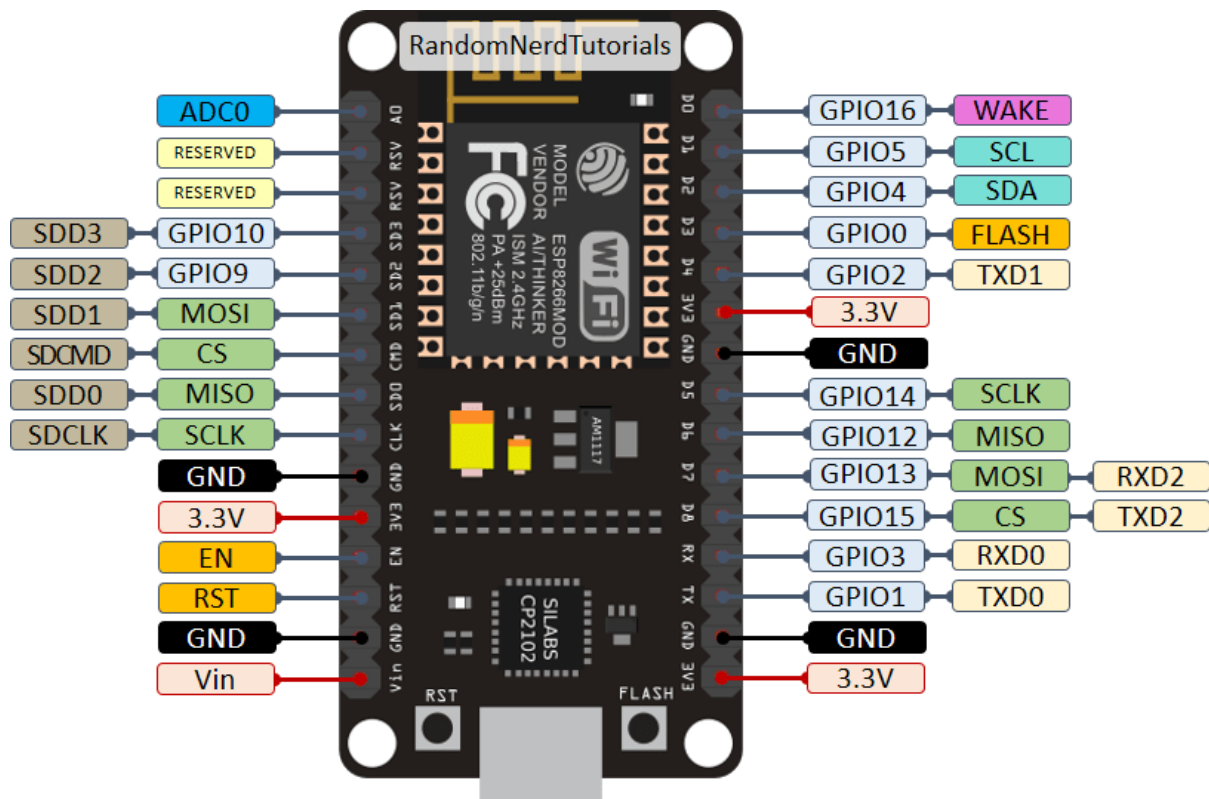


Figura 3.70: Pin de GPIO del ESP8266

Fuente: <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>

Un programa en arduino debe tener las funciones ”setup” y ”loop”, sin estas funciones no es posible ejecutar un programa en arduino, por lo cual para comenzar se define la primera

función respectivamente, estableciendo el modo en el que funcionarán los pines del ESP8266 explicados anteriormente, se configurarán en modo salida, identificando cada uno con el color que se le asignó (ver Figura 3.71).

```
void setup() {  
    delay(1000);  
    Serial.begin(115200);  
    pinMode(Vent_Gris, OUTPUT);  
    pinMode(Vent_Cafe, OUTPUT);  
    pinMode(Vent_Morado, OUTPUT);  
    pinMode(Lampara_Control, OUTPUT);  
    Serial.print("Espera");delay(1000);  
}
```

Figura 3.71: Configuración de los pines en modo de salida

Fuente: Elaboración propia.

En el siguiente bloque de código se configuró los parámetros de red para el microcontrolador ESP8266, comenzando por la dirección IP, su puerta de enlace y la mascara de subred, se le asigna un nombre al host, en este caso el microcontrolador y se comienza la conexión Wifi con el nombre de la red y su contraseña (conexión a la red Wifi, ver Figura 3.72).

```
Serial.print("Setting AP (Access Point)...");  
  
WiFi.config(ip, gateway, subnet);  
  
WiFi.mode(WIFI_STA);  
WiFi.hostname("uriel");  
//delay(1000);  
WiFi.begin("Totalplay-A767", "A767C5A759FFJ4RY");
```

Figura 3.72: Parámetros de conexión al Wifi

Fuente: Elaboración propia.

Después de que se configuró la conexión a la red doméstica del hogar, el programa espera a que el enlace entre el microcontrolador y el Wifi sea establecido, una vez realizado con

éxito el programa procede a inicializar el servidor (ver Figura 3.73). Una vez que ha sido inicializado el servicio se procede a emplear dicho servidor como HTTP para que así pueda ser desplegada la aplicación web (Ver Figura 3.74).

```
while (WiFi.status() != WL_CONNECTED) {  
  
    delay(500);  
    Serial.println("Esperando para conectarse...");  
  
}  
Serial.println(WiFi.localIP());  
server.on("/", manejaRaiz);  
server.begin();
```

Figura 3.73: Inicialización del servicio

Fuente: Elaboración propia.

```
void loop() {  
  
    server.handleClient();  
  
}
```

Figura 3.74: Servidor HTTP

Fuente: Elaboración propia.

Acercándose al final de este programa, en el siguiente segmento de código comienza el control de las velocidades del motor y el foco, por lo que se comienza con el foco, el código comienza con la lógica determinando si el foco se encuentra encendido, es decir, si desde la aplicación web se mando una orden de encender el foco, por lo que si es así la salida de este pin se enciende para que el foco pueda emanar luz, analíticamente sucede lo mismo si se manda la orden de apagar el foco, siendo así únicamente la salida se apaga por lo tanto el foco apaga.

Seguido de esto y con la misma lógica se controlan las salidas hacia las velocidades del motor, por lo que si se envía la primer orden de encender la velocidad mas alta del motor como primer paso, este cierra los otros dos canales (velocidades media y baja) y abre la salida para la velocidad mas alta, eso basándose en el análisis que se hizo en la sección de hardware del sistema para identificar que combinación de colores coincidentes con el cable común (negro) son las que provocan cada velocidad en el motor. En las siguientes bloques condicionales se hace lo mismo con la velocidad media y baja, encendiendo el canal correspondiente para este así como para la baja y cerrando los canales de las velocidades que no se están utilizando (ver Figura 3.75 y 3.76).

```
void manejaRaiz() {  
  WiFiClient client = server.client();  
  if (server.arg("Lampara")== "on")  
  {  
    Serial.println("Lampara on");  
    Lampara = 1;  
    digitalWrite(Lampara_Control, HIGH);  
  }else if (server.arg("Lampara")== "off")  
  {  
    Serial.println("Lampara off");  
    Lampara = 0;  
    digitalWrite(Lampara_Control, LOW);  
  }  
  if (server.arg("Ventilador")== "0")  
  {  
    Serial.println("Ventilador 0");  
    Ventilador = 0;  
    digitalWrite(Vent_Gris, LOW);  
    digitalWrite(Vent_Cafe, LOW);  
    digitalWrite(Vent_Morado, LOW);  
  }else if (server.arg("Ventilador")== "1")
```

Figura 3.75: Control de velocidades del motor

Fuente: Elaboración propia.

Por último, ya definidas estas funciones, se comienza a escribir el código de la aplicación web HTML en una variable de tipo string como se puede ver en la Figura 3.77, definido el código HTML por último se utiliza una función para enviarle al servidor dicho código y que pueda ser ejecutado en este (ver Figura 3.78).

```
}else if (server.arg("Ventilador")== "1")
{
    Serial.println("Ventilador 1");
    Ventilador = 1;
    digitalWrite(Vent_Gris, LOW);
    digitalWrite(Vent_Cafe, LOW);
    digitalWrite(Vent_Morado, HIGH);
}else if (server.arg("Ventilador")== "2")
{
    Serial.println("Ventilador 2");
    Ventilador = 2;
    digitalWrite(Vent_Gris, LOW);
    digitalWrite(Vent_Cafe, HIGH);
    digitalWrite(Vent_Morado, HIGH);
}else if (server.arg("Ventilador")== "3")
{
    Serial.println("Ventilador 3");
    Ventilador = 3;
    Vent3 = 1;
    digitalWrite(Vent_Gris, HIGH);
    digitalWrite(Vent_Cafe, LOW);
    digitalWrite(Vent_Morado, LOW);
}
```

Figura 3.76: Control de velocidades del motor

Fuente: Elaboración propia.

```

header = "<!DOCTYPE html><html>\n<head>\n";
header += "<meta http-equiv='Content-Type' content='text/html; charset=utf-8'\n";
header += "<meta name='viewport' content='width=device-width, initial-scale=1'\n";
header += "<link rel='icon' href='data:,\n";

header += "<style>\n";
header += ".saludos{animation-duration: 2s;animation-name: slidein;}@keyframes slidein {from {margin-left: 100%;width: 300%;}to{margin-left: 0%;width: 100%;}}\n";
header += "html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";
header += ".button { background-color: #4CAF50; border: none; color: black; padding: 16px 40px;\n";
header += "text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}\n";
header += ".button2 {background-color: #FFFF66;}.button3 {background-color: #99c2ff;}.button4 {background-color: #dddddd;}.button5 {background-color: #00ff00;\n";
header += "</style>\n";
header += "<title>Habitación principal</title>\n";

header += "</head>\n";
header += "<body><h1><a href='http://';header +=WiFi.localIP().toString() ;header += '\n'>Habitación principal</a></h1>\n";
header += "<p>Luz principal</p>\n";
if (Lampara == 0)
header += "<p><a href='\"/?Lampara=on\"><button class='\"button button4\">Encender</button></a></p>\n";
else
header += "<p><a href='\"/?Lampara=off\"><button class='\"button button2\">Apagar</button></a></p>\n";
header += "<p><Ventilador</p>\n";
switch (Ventilador)
{

```

Figura 3.77: Código de página HTML

Fuente: Elaboración propia.

```
server.send(200, "text/html", header);
```

Figura 3.78: Envío de código HTML

Fuente: Elaboración propia.

3.2.8. Pruebas

En la fase de pruebas, se llevó a cabo el funcionamiento del motor eléctrico de 3 velocidades para descartar cualquier problema que tenga que ver con defectos de fábrica, como la buena conducción de electricidad de los cables, funciones de capacitor y el foco.

Como se vio en la sección 3.2.3 Hardware del sistema se hicieron algunas pruebas de funcionamiento a los relevadores, únicamente para evitar problemas a la hora de armar la circuitería que conforma el motor con el fin de si se tenía algún fallo imprevisto a la hora de probar el funcionamiento del motor tener claro desde donde se falló y dónde está el error, de la forma en que se instalaba una cosa y se probaba inmediatamente.

Una vez que todos los componentes del sistema estén en su lugar se comenzaron a realizar las pruebas del funcionamiento empezando por el foco, se conectó el ventilador a la energía,

por lo cual la programación del microcontrolador lleva a cabo las configuraciones necesarias de la aplicación y las conexiones al Wifi del hogar, todo listo para ser manejado mediante la aplicación.

Cuando se conecta a la energía este no debe de hacer ningún tipo de movimiento en la rotación del motor o en la luz del foco, ya que si lo hace hay algún error en las conexiones entre los relevadores o se han dañado. Prosiguiendo con el proceso, se conecta el teléfono que se vaya a utilizar para llevar a cabo el control del ventilador, se debe abrir algún navegador web que se desee accediendo a la dirección IP ya preestablecida (192.168.100.89) y se podrá ver la aplicación. Al acceder a la aplicación todo se debe encontrar apagado, hasta esta parte todo se encuentra en orden y funcionando cómo se programó.

Una vez en la interfaz de la aplicación se dirigió al botón de encendido o apagado del foco, se presiona y se puede observar como el microcontrolador acciona los relevadores para realizar el puente de la energía y pueda encender el foco (ver Figuras 3.49 y 3.50). Por lo cual en la función de este botón se cumple con éxito.

Seguido de esto se apagó el foco, se procedió a verificar el funcionamiento principal de este proyecto que es el control de un motor de 3 velocidades, primeramente se presiono el botón de encendido de las velocidades, por lo cuál según la leyenda del botón se encuentra en la velocidad mas alta, y efectivamente así era, se volvió a presionar para cambiar de velocidad hasta llegar a la velocidad mas baja y apagar el motor. Todo esto se lleva a cabo por las órdenes que le da el microcontrolador ESP8266 a los relevadores que cumplen la función que cumplían los interruptores físicos que tenía el ventilador.

Como se observó este funcionamiento, se procedió a encender todo, y todo funcionaba muy bien correctamente para así al final probar el botón de "apagar todo", el cual se presionó y apagó todo el sistema, solo el ventilador y el foco ya que el microcontrolador y los relevadores deben estar encendidos ya que se encuentran conectados directamente a la energía eléctrica a la espera de cualquier orden de quien se conecte a la aplicación, cumpliendo así con el

objetivo principal de este proyecto.

3.2.9. Presupuesto

En cuanto al presupuesto que es necesario para llevar a cabo el proyecto no es muy elevado. Primeramente se fija en los precios de ventiladores estándar como los tradicionales que se usan mas comúnmente en los hogares, por lo que se acudió a un proveedor de calidad en sus productos con el cual se eligió el ventilador que tuviera el espacio necesario entre los cables para poder meter toda la circuitería y que incluyera un foco. El costo de dicho ventilador fue de \$49.97 dolares.

En cuanto al microcontrolador ESP8266 se vieron distintos proveedores del cual se eligió el que tenía el precio mas económico tratándose del mismo modelo de tarjeta que ya se menciona con un costo de \$180.00 pesos. Existen diferentes modelos para esta tarjeta, sin embargo la de mejor calidad y que se ajusta a las características de dicho proyecto es el ESP8266 de arduino.

Y para finalizar con los costos más significativos de este proyecto se tiene el precio de la tarjeta de relevadores de 4 módulos con un valor de \$111.00 pesos, tanto los relevadores como la tarjeta del microcontrolador ESP8266 se compraron con distintos proveedores, así teniendo un costo total del proyecto en pesos de \$1,218.00 pesos aproximadamente ahorrándose entre \$1,000.00 a \$4,000.00 pesos comparándose con el precio de un ventilador comercial.

3.2.10. Metodología

Analizando las cosas necesarias para dicho proyecto, el uso de hardware, programación de software que se comunica con el hardware a través del Internet, se reconoció que este es un sistema embebido, por lo que se investigó alguna metodología que este orientada a sistemas embebidos.

Con base en lo anteriormente investigado, se procedió a elegir una metodología orientada a sistemas embebidos, ya que este proyecto es de tal tipo. Un sistema embebido es un sistema de procesamiento de información de uso específico integrado en otro sistema de mayor tamaño y conformado por componentes de hardware y software [19].

A menudo los diseñadores de sistemas cometen el error de comenzar a diseñar e implementar soluciones que no han sido completamente especificadas y que corresponden a problemas a los que les falta delimitación, lo cual conduce a la construcción de sistemas que no satisfacen las necesidades de los clientes y que incurren en el aumento de los costos y en el incumplimiento de los plazos establecidos [19].

Ciertas metodologías orientadas a sistemas embebidos únicamente permiten expresar los requisitos que están en el nivel de sistema, es decir, los que tienen que ver con funciones, pero olvidan los requerimientos que están en niveles externos, en donde se encuentran las relaciones con el sistema principal y otros sistemas embebidos. A continuación se muestran algunas metodologías que ofrecen mayor adaptación en el desarrollo de sistemas embebidos y que siguen los conceptos de meta o escenario.

Comenzando con las metodologías orientadas por escenarios se tiene primero a ScenIC que se basa en objetivos, tareas, obstáculos y acciones llevadas a cabo por actores [19]. La siguiente metodología orientada por escenarios es llamada SRAM que propone una combinación de escenarios y prototipos tempranos [19].

Con las metodologías orientadas a metas se tiene a KAOS que se basa en técnicas de inteligencia artificial para definir metas, acciones, estados de forma consistente y rigurosa [19]. Por último se tiene a ABC-Besoins que se orienta a metas y agentes para lograr un mayor entendimiento de la problemática a resolver [19].

En las metodologías orientadas por escenarios que se mencionaron son mas fáciles de entender para el usuario, pero traen confusión e imprecisión y no poseen una forma de determinar cuando se han seleccionado suficientes escenarios para abarcar toda la funcionalidad

del sistema a desarrollar.

Por otro lado, las metodologías orientadas por metas tiene una dificultad mayor, pero es posible tener resultados más precisos y exactos e incluir requisitos de niveles externos como por ejemplo el sistema principal si se tratara de un sistema embebido con subsistemas que lo conforman, es decir, no sólo expresan funcionalidades. KAOS tiene la desventaja de hacer el tratamiento de requisitos de manera aislada, mientras que ABC-Besoins, con el concepto de “Intervención de agentes” logra integrar y relacionar los requisitos de diferentes contextos [19].

Por estas razones ya mencionadas se optó por otra metodología tradicional que es el proceso estándar del desarrollo de software que lleva como nombre metodología en cascada con los siguientes pasos a desarrollar:

- **Requerimientos:** Análisis de las necesidades que debe cubrir la aplicación para cumplir el objetivo.
- **Diseño del sistema:** Identificación de elementos que van a conformar el sistema, tanto de hardware como de software y elementos de red.
- **Diseño del programa:** Diseño del algoritmo que tendrá el programa para el control del motor.
- **Codificación:** La implementación en código del algoritmo establecido para el funcionamiento de la aplicación.
- **Pruebas:** Se ensamblan cada uno de los elementos para comprobar que funcionen correctamente.

La fase de requisitos consiste en que se especifican las necesidades del sistema a desarrollar. La especificación de requisitos sirve como base para la negociación entre los desarrolladores y clientes del sistema y también para planear y controlar el proceso de desarrollo [20].

En la fase de diseño, se busca comprender los requisitos del sistema con el propósito de estructurar la arquitectura de este. Se transforma la arquitectura obtenida en una arquitectura especializada [20].

En la parte de codificación, se expresa la arquitectura del sistema en una forma aceptable para la computadora (es decir, el código programado). Se combinan los componentes creados de manera independiente para formar el sistema completo [20].

Se verifica y valida el sistema a nivel de componentes individuales y su integración. Éste es uno de los aspectos más críticos del desarrollo y debe desarrollarse repetidas veces con las demás actividades. Se busca descubrir cualquier defecto en los requerimientos, diseño, codificación e integración. Las pruebas se hacen en segmentos de programa sencillos hasta el sistema completo [20].

Capítulo 4

Resultados y Discusiones

En este capítulo se reportará el funcionamiento del sistema y de que manera lo hace, el manejo sencillo e intuitivo de este a través de la aplicación web que puede ser ejecutada tanto en computadoras como en móviles. También se verá un poco de en qué lugares funcionó la aplicación dentro del hogar donde se hicieron las pruebas.

Para comenzar el sistema consiste en manejar un motor eléctrico de 3 velocidades a través de la red Wifi mediante una aplicación web, introduciendo un ventilador de techo convencional a la arquitectura del Internet de las cosas, con dicha aplicación web corriendo sobre un teléfono inteligente, tableta o computadora.

El sistema funciona de la siguiente manera, primeramente con el ventilador ya conectado a la energía eléctrica se accede a un navegador web y se escribe la dirección IP ya configurada en el microcontrolador ESP8266 para acceder a la aplicación web. Una vez hecho esto, se observará en la pantalla donde se haya accedido a la página web una interfaz como en la Figura 4.1.

El uso de este sistema tiene dos opciones: Foco y ventilador con 3 velocidades. En esta parte se comienza a encender el foco, se debe presionar el botón que dice encender en el apartado “Luz principal”, en la aplicación se sabrá que esta encendido por que el botón cambia de color a amarillo y la leyenda dice “apagar”, por lo tanto este debe encender como se observa en la siguiente Figura (4.2).



Figura 4.1: Vista inicial de la aplicación web

Fuente: Elaboración propia.

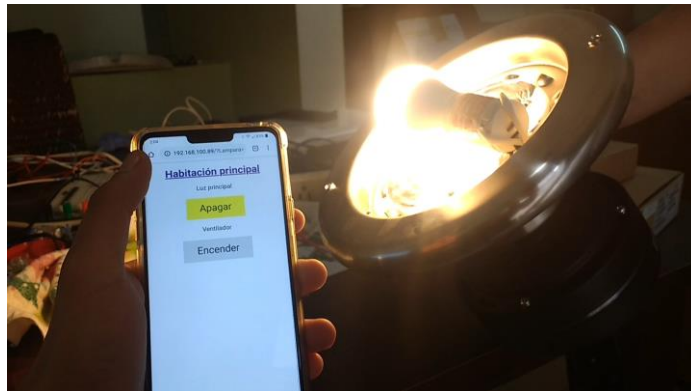


Figura 4.2: Encendido del foco

Fuente: Elaboración propia.

Este también puede trabajar junto con las velocidades del motor, por lo que se puede encender el ventilador (encendido del motor) presionando el botón de “Encender” del apartado “Ventilador”, este inmediatamente comienza a girar en la velocidad mas alta de las 3 que conforman sus velocidades (alta, media y baja), en la Figura 4.3 se puede observar un poco

como este comienza a girar.



Figura 4.3: Encendido del motor en velocidad alta

Fuente: Elaboración propia

Si el usuario desea que el ventilador opere en una velocidad mas baja, este puede volver a presionar el mismo botón de ventilador por lo que la velocidad cambiará a “Media”, el color del botón cambiará a verde limón y la leyenda de este botón dirá “Media” de este modo el usuario sabrá que se cambió de velocidad y al verificarlo en el ventilador, este contendrá una velocidad mas baja (ver Figura 4.4). Funciona de la misma forma para bajar aún mas la velocidad, presionando por una tercera vez dicho botón este estará a su mínima potencia, si es presionado por cuarta vez este se detendrá por completo, también cambiará de color y leyenda (Figura 4.5).

También el ventilador puede trabajar sin la necesidad de que el foco este encendido, por lo que se puede apagar este sin ningún problema, también independientemente de la velocidad en la que se encuentre operando el ventilador así como también se puede volver a encender si el usuario lo desea (ver Figuras 4.6 y 4.7).

En dicha aplicación se le agregó una funcionalidad extra que no simula el funcionamiento que tenían los controles físicos del ventilador. Es un botón el cual apaga todo el sistema, muy útil en los casos que el usuario tenga que salir del hogar y únicamente pueda apagar todo con un solo clic. Este botón es de color azul claro y tiene como leyenda “Apagar todo”, si este es

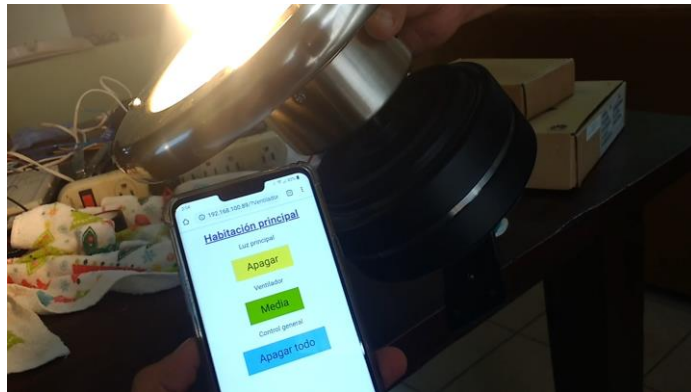


Figura 4.4: Cambio a velocidad media del motor
Fuente: Elaboración propia.

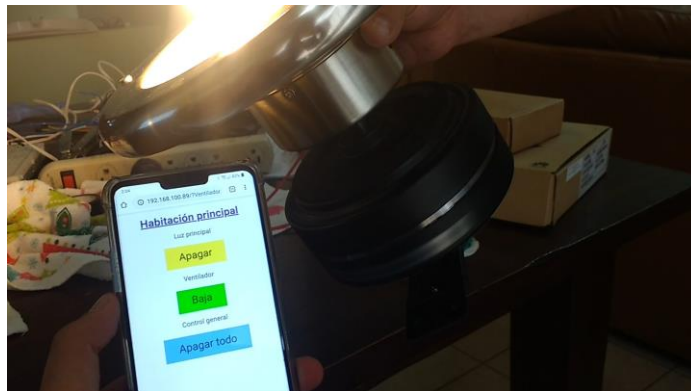


Figura 4.5: Cambio a velocidad baja del motor
Fuente: Elaboración propia.

presionado, apagará todo el sistema como se puede ver en la Figura 4.8 por consecuencia en la siguiente F.igura (4.9) se puede apreciar como se va deteniendo la rotación del motor.

El ventilador puede estar posicionado en cualquier lugar donde haya alcance con la red Wifi del hogar, como mínimo los módems tradicionales tienen un alcance de 10 metros a la redonda, por lo que si este se encuentra en este rango podrá conectarse a la red y funcionar, incluso en un hogar de dos pisos. Para esta muestra de funcionamiento se posicionó en la sala principal (No instalado), el cual se encuentra aproximadamente de 5 a 6 metros del punto de acceso (módem).



Figura 4.6: Apagado del foco en operación con el ventilador
Fuente: Elaboración propia.

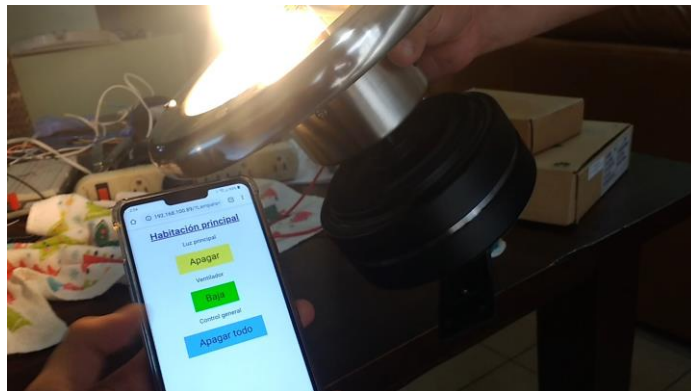


Figura 4.7: Encendido del foco en operación con el ventilador
Fuente: Elaboración propia.

La aplicación web fue ejecutada en un teléfono inteligente, en la cual se probó el funcionamiento en distintos puntos del hogar donde se encontraba el equipo, una casa de un solo piso (180 metros cuadrados), el ventilador funcionó correctamente, además se enviaron órdenes desde el patio y fuera de la casa recibiendo dichas órdenes aún así.

Se detectó que el teléfono inteligente seguía conectado a la red Wifi, en la calle frente a dicho hogar. Se probó conexión en 4 habitaciones diferentes de donde se encontraba el ventilador. Cabe resaltar que el alcance depende mucho del módem y teléfono o dispositivo que se utilice para operar el ventilador, ya que algunos puntos de acceso pueden irradiar una

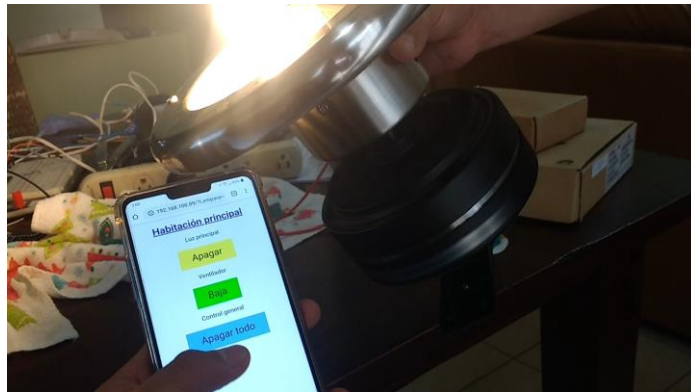


Figura 4.8: Apagado total del sistema
Fuente: Elaboración propia.



Figura 4.9: Detención del funcionamiento
Fuente: Elaboración propia.

mejor frecuencia con mas alcance, así como cualquier otro teléfono inteligente o dispositivo móvil con un sensor de mayor capacidad de alcance a la detección de redes Wifi.

No es necesario que el punto de acceso provea Internet, únicamente con que haya señal Wifi es suficiente, ya que el ventilador junto con la aplicación web funcionan sobre la red doméstica del hogar. Siendo así este sistema se une a la arquitectura del Internet de las cosas.

Capítulo 5

Conclusiones

La arquitectura del Internet de las cosas, esta avanzando cada vez más, el uso de la red de redes se ha vuelto una necesidad para las comunicaciones hoy en día. En esta sección se concluirá este proyecto, recapitulando brevemente cada objetivo específico que llevó esta investigación mostrando cómo se llegó a cumplir el objetivo general del proyecto. Además, se describen qué herramientas se utilizaron y cómo se utilizaron, asimismo, se presenta la elección del hardware para la automatización del ventilador y la configuración de estos dispositivos que se eligieron para que pueda ser controlado a través de una aplicación web. También se incluyen algunas consideraciones a tomar en cuenta como proyectos a futuro como mejora a este sistema.

5.1. Con respecto al objetivo de la investigación

En el presente proyecto se trataron diversos puntos, de los cuales permitieron llegar a cumplir el objetivo principal de este proyecto, como por ejemplo, determinar el diagrama eléctrico del motor. Esto es de suma importancia, debido a que se debe analizar el funcionamiento normal del motor eléctrico, donde proporcionó la posibilidad de que obteniendo el diagrama se determinara de qué modo funcionaban los interruptores y circuitos eléctricos del motor, qué salidas iban a estos interruptores y qué cables le pertenecían para que se pudiera

automatizar el motor, saber de qué forma modificar dicho motor para que se realizara el control a distancia.

En la automatización del motor se llevó a cabo la modificación de este y se identificaron los elementos que debían modificarse para que fuera capaz de conectarse al Wifi, desplegar y ejecutar una aplicación web, así como controlar los cambios de velocidad en el motor, por lo que se optó por un microcontrolador, es pequeño, versátil y de buena calidad para el uso que se necesitaba. Cumple con las funciones para las que se le necesita que son el ejecutar y desplegar una aplicación web, ya que este puede llevar a cabo la función de servidor, también es capaz de conectarse a la red Wifi, sin embargo, no es posible que este pueda manipular directamente los cambios de velocidad pero si controlar otro circuito de arduino, por consiguiente, se modificaron los interruptores que ya tenía dicho ventilador por una tablilla de relevadores de 4 módulos, el cual 3 serán orientados a las 3 velocidades del motor y uno al del foco ya que puede realizar la función de un interruptor pero puede ser controlado por un microcontrolador, que le dará las órdenes para que efectué la función de interruptor que se desea.

Con respecto al hardware del sistema, tras el análisis hecho se eligió qué tipo de dispositivos serían útiles, ya que en la parte de automatización se requería reemplazar el trabajo que llevaban a cabo los interruptores físicos y el cómo controlarlos a través de la red Wifi se procedió a elegir una tablilla de relevadores, las cuales ayudarían a suplir los interruptores físicos del motor pero con la cualidad de poder ser controlados por un microcontrolador. Para el control, fue necesaria una aplicación web y una interfaz para el usuario, para ejecutar esta aplicación web se eligió un microcontrolador ESP8266 ya que este brindó la posibilidad de que el motor pudiera conectarse a la red Wifi, se le pueda programar un algoritmo de control en arduino para la gestión de las velocidades y funcione como servidor para que el usuario se conecte y pueda visualizar la aplicación web a través de su teléfono móvil.

Al momento de llegar a reunir los requerimientos de la aplicación, se definió prácticamente

la interfaz gráfica del usuario, se obtuvo la idea de qué botones incluir y cómo suplir lo que hacia cada interruptor físico del motor. Con base en este análisis y esta idea se procedió a definir el algoritmo de control en arduino y de qué manera trabajaría en conjunto con la aplicación web que se desarrolló. Definidos los requerimientos necesarios se programó la estructura HTML y código CSS para darle vista a dicha estructura. En el algoritmo de control se definió cómo conectarse al microcontrolador ESP8266 con las configuraciones pertinentes.

Al final se llevaron a cabo pruebas de lo que se desarrolló y la parte que se modificó en el motor, la aplicación funcionó según lo previsto, cada botón enviaba la orden al microcontrolador y este la ejecutaba, cambió de velocidad según se le indicó, además, se probó el funcionamiento y el encendido del foco. En cuanto el ventilador es conectado a la red eléctrica este inicializa la aplicación y el algoritmo de control junto con la aplicación web a la espera de un cliente, se conecta automáticamente a la red Wifi con las configuraciones ya programadas, se conectó el teléfono móvil inteligente a la aplicación web, se dieron órdenes desde distintas habitaciones del hogar y distancias con respecto al punto de acceso (Wifi) funcionando correctamente, todo este sistema funcionando sobre la red Wifi, por lo tanto este sistema funciona en la arquitectura del Internet de las cosas. Por lo tanto se alcanzó exitosamente el objetivo general del presente proyecto.

5.2. Recomendaciones para futuras investigaciones

Para finalizar, en el proceso de desarrollo del sistema y ensamble se hicieron algunas observaciones importantes de forma tal que pueden ser tomados como futuras investigaciones con el objeto de mejorar el sistema desarrollado en la presente investigación:

- Integrar una base de datos: Con el fin de poder guardar datos como horarios de funcionamiento

- Desarrollar una aplicación móvil: Para Android o IOS.
- Sistema de automatización para el hogar: Un proyecto a futuro podría darse al momento de que este sistema de control del motor de 3 velocidades forme parte de otro mas grande, un sistema de automatización para el hogar que controle sobre la arquitectura del Internet de las cosas las luces del hogar, el aire acondicionado, la calefacción del hogar, gestión de energía eléctrica, entre otras cosas.

Conforme la arquitectura del Internet de las cosas vaya tomando un uso mas frecuente, se pueden encontrar mejores usos en el hogar para ayudar al ser humano con las actividades cotidianas de una mejor manera, mas útil y eficaz.

Bibliografía

- [1]J. Salazar and S. Silvestre, “Internet de las cosas”, *Techpedia. České vysoké učení technické v Praze Fakulta elektrotechnická*, 2016.
- [2]M. Kusriyanto and A.A. Putra, “Weather Station Design Using IoT Platform Based On Arduino Mega”, in *2018 International Symposium on Electronics and Smart Devices (ISESD)*. oct 2018, pp. 1–4, IEEE.
- [3]S. Wasoontarajaroen, K. Pawasan, and v. Chamnanphrai, “Development of an IoT device for monitoring electrical energy consumption”, in *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)*. oct 2017, pp. 1–4, IEEE.
- [4]S. Athani, C H Tejeshwar, M.M Patil, P. Patil, and R. Kulkarni, “Soil moisture monitoring using IoT enabled arduino sensors with neural networks for improving soil management for farmers and predict seasonal rainfall for planning future harvest in North Karnataka — India”, in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. feb 2017, pp. 43–48, IEEE.
- [5]R. Ijaz, M.A. Pasha, N.U. Hassan, and C. Yuen, “A novel fusion methodology for indoor positioning in IoT-based mobile applications”, in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. feb 2018, pp. 742–747, IEEE.
- [6]G.V. Vivek and M.P. Sunil, “Enabling IOT services using WIFI - ZigBee gateway for a home automation system”, in *2015 IEEE International Conference on Research in*

- Computational Intelligence and Communication Networks (ICRCICN)*. nov 2015, pp. 77–80, IEEE.
- [7]W. Hlaing, S. Thepphaeng, V. Nontaboot, N. Tangsunantham, T. Sangsuwan, and C. Pira, “Implementation of WiFi-based single phase smart meter for Internet of Things (IoT)”, in *2017 International Electrical Engineering Congress (iEECON)*. mar 2017, pp. 1–4, IEEE.
- [8]S. Saloni and A. Hegde, “WiFi-aware as a connectivity solution for IoT pairing IoT with WiFi aware technology: Enabling new proximity based services”, in *2016 International Conference on Internet of Things and Applications (IOTA)*. jan 2016, pp. 137–142, IEEE.
- [9]N. V. Rajeesh Kumar, B. S. B. Praveen, A. V. Sarveswara Reddy, and B. Baron Sam, “Study on IOT with reference of M2M and WiFi”, in *2017 International Conference on Information Communication and Embedded Systems (ICICES)*. feb 2017, pp. 1–6, IEEE.
- [10]N. K. Walia, P. Kalra, and D. Mehrotra, “An IOT by information retrieval approach: Smart lights controlled using WiFi”, in *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*. jan 2016, pp. 708–712, IEEE.
- [11]D. R. Bautista, J.A Parra-Valencia, and C. D. Guerrero, “IoT: Una aproximación desde ciudad inteligente a universidad inteligente”, *Revista Ingenio UFPSO*, vol. 13, no. 1, pp. 9–20, jun 2017.
- [12]J. L. Peña Merizalde and G. Elizabeth, “Estudio del modelo de referencia del internet de las cosas (iot), con la implementación de un prototipo domótico”, B.S. thesis, Quito: EPN, Ecuador, 2016.

- [13]L Jurado Pérez, A Velásquez Vargas Washington, and N Vinueza Escobar, “Estado del arte de las arquitecturas de internet de las cosas (iot)”, B.S. thesis, Departamento de Ingeniería de Sistemas Telemáticos - DIT, UPM, Madrid, España, 2014.
- [14]S. Canedo de Prado, *Sistema de control domótico basado en WiFi*, PhD thesis, Escuela de Ingeniería de Telecomunicación, U . de Vigo, Pontevedra, España, 2017.
- [15]O. T. Artero, *ARDUINO. Curso práctico de formación*, RC Libros, 2013.
- [16]R. Castro, “Avanzando en la seguridad de las redes wifi”, *Boletín de RedIRIS*, vol. 73, pp. 23–33, 2005.
- [17]D. Roldán Martínez and J. Alba, “Aplicaciones de wifi”, *Bit*, , no. 145, pp. 68–70, 2004.
- [18]W.C. Tsai, Y.J. Shih, and T.M. Tsai, “IoT-type Electric Fan: Remote-Controlled by Smart-Phone”, in *2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)*. may 2016, pp. 12–15, IEEE.
- [19]L. González and G. Urrego, “Modelo de requisitos para sistemas embebidos: Model of requirements for embedded systems”, *Revista Ingenierías Universidad de Medellín*, vol. 7, no. 13, pp. 111–127, 2008.
- [20]A. Weitzenfeld and S. Guardati, “Capítulo 12: Ingeniería de software: el proceso para el desarrollo de software”, *Libro: Introducción a la Computación. CENGAGE Learning. México, DF*, 2007.