



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

RED MALLADA CON DISPOSITIVOS DE BAJO CONSUMO PARA IOT

ORELLANA PLACENCIA MIGUEL ANGEL
INGENIERO DE SISTEMAS

MACHALA
2022



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

RED MALLADA CON DISPOSITIVOS DE BAJO CONSUMO PARA
IOT

ORELLANA PLACENCIA MIGUEL ANGEL
INGENIERO DE SISTEMAS

MACHALA
2022



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

RED MALLADA CON DISPOSITIVOS DE BAJO CONSUMO PARA IOT

ORELLANA PLACENCIA MIGUEL ANGEL
INGENIERO DE SISTEMAS

HERNANDEZ ROJAS DIXYS LEONARDO

MACHALA, 24 DE FEBRERO DE 2022

MACHALA
2022

Tesis de Angel Orellana

INFORME DE ORIGINALIDAD

3%

INDICE DE SIMILITUD

5%

FUENTES DE INTERNET

0%

PUBLICACIONES

2%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1

medium.com

Fuente de Internet

2%

2

veryxs.com

Fuente de Internet

2%

Excluir citas

Activo

Excluir coincidencias < 2%

Excluir bibliografía

Activo

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, ORELLANA PLACENCIA MIGUEL ANGEL, en calidad de autor del siguiente trabajo escrito titulado RED MALLADA CON DISPOSITIVOS DE BAJO CONSUMO PARA IOT, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 24 de febrero de 2022



ORELLANA PLACENCIA MIGUEL ANGEL
0706025608

DEDICATORIA

El presente trabajo es dedicado a toda mi familia por darme la fortaleza suficiente para superarme, pero con todo mi corazón a mi madre por brindarme todo el apoyo emocional y ser un ejemplo de superación, humildad y sacrificio.

A todos mis amigos que estuvieron presentes a lo largo de mi vida universitaria y sobre todo a mis docentes por forjarme con los conocimientos adquiridos.

Sr. Miguel Ángel Orellana Placencia

AGRADECIMIENTO

Agradezco a Dios por darme la vida y ser mi guía en cada uno de mis pasos, también por permitirme tener y disfrutar a toda mi familia, por los consejos en cada decisión o proyecto logrado y esa confianza de creer en mí para lograr esta meta.

A mis docentes de la Escuela de Informática por sus enseñanzas para poder desarrollarme profesionalmente y haberme dedicado todo su tiempo en promover sus conocimientos y sobre todo a mi tutor, el Ing. Dixys Leonardo Hernández Rojas por ser el mentor para la elaboración de este proyecto.

Sr. Miguel Ángel Orellana Placencia

RESUMEN

Con el constante avance en la tecnología y los conocimientos conjuntos del ser humano para asumir el fin de satisfacer los problemas de la sociedad en los diferentes ámbitos sociales o culturales, es inevitable el desarrollo de nuevos sistemas de comunicación para intercambiar la información necesaria. El Internet de las cosas (IoT) es la descripción de una red de todos los objetos físicos para conectarse a través de internet y es importante para las nuevas tecnologías inalámbricas que se posicionan mundialmente. En la actualidad los hogares tienen implementados una red inalámbrica Wi-Fi por el simple hecho de no adaptarse al control de nuevos sistemas para la conectividad de los dispositivos, por ello la tecnología BLE prevalece la relación de dispositivos y personas para que el sistema no sea totalmente autónomo ni pierda la esencia humana, también posee una red práctica y que pueda funcionar a la par con otra tecnología sin interrumpir ningún proceso.

Para la utilización del protocolo Bluetooth Low Energy (BLE) es necesario la ampliación de dispositivos con microcontroladores que son necesarios para requerir procesos y que ofrezcan el servicio Bluetooth (conectar con un dispositivo móvil), por lo cual el módulo ESP 32 lo convierte en favorito para integrarse en una red y el valor que tiene en el mercado electrónico lo hace accesible para incursionar fácilmente en redes. Además, que su estructura interna lleva ventaja para adaptarse con algunos marcos de desarrollo como: Arduino y Espressif que fue seleccionado por que tiene los elementos necesarios para construir a base del protocolo BLE, en otras palabras, es necesario el manejo del módulo ESP 32 para que sea el encargado de presentar ejemplos comunes de la vida real adaptándose con otros periféricos como la toma de temperatura, la presentación de información para que sea implementado en una área domótica y se tiene como finalidad comprender un sistema de iluminación RGB (red, green, blue) en dos tipos de modelos ESP 32 para alcanzar el control que permite conseguir con la aplicación nRF Mesh. El diseño que ofrece es adaptable para incorporar nuevos nodos o dispositivos provisionados para estructuras de largas distancias que lo convierte en una red funcional controlando el sistema de iluminación RGB.

Se culmina el proyecto con las pruebas obtenidas que fueron de ayuda para ilustrar la capacidad que tiene una red BLE en una estructura compleja, de manera que la conexión que posee el módulo ESP WROOM 32 a menos de 20 metros con el móvil es importante para conocer el rango de alcance y así a futuro diseñar la red de dispositivos que permitan la conectividad entre nodos. En lo particular la creación de grupos que se puede

acceder por medio de la App nRF Mesh es esencial para la gestión de los elementos según sus características y que tan ordenada puede ser la App nRF Mesh cuando se agregan grandes cantidades de nodos aprovisionados, por otra parte, se comprobó que la aplicación no es perjudicial para el usuario y que solo solicita los permisos básicos para su correcto funcionamiento.

Palabras claves: Tecnologías inalámbricas, IoT, ESP32, Aprovisionamiento, BLE Mesh.

ABSTRACT

With the constant advancement in technology and joint knowledge of human being to assume the purpose of meeting the problems of society in different social or cultural fields, the development of new communication systems to exchange the necessary information is inevitable. The Internet of Things (IoT) is the description of a network of all physical objects to connect through the internet and is important for new wireless technologies that are positioned globally. Currently homes have implemented a Wi-Fi wireless network for the simple fact of not adapting to the control of new systems for the connectivity of devices, so the BLE technology prevails the relationship of devices and people so that the system is not totally autonomous or lose the human essence, also has a practical network and that can work on par with other technology without interrupting any process.

For the use of the Bluetooth Low Energy (BLE) protocol it is necessary to expand devices with microcontrollers that are necessary to require processes and offer the Bluetooth service (connect to a mobile device), so the ESP 32 module makes it a favorite for integration into a network and the value it has in the electronic market makes it accessible to easily venture into networks. In addition, that its internal structure takes advantage to adapt with some development frameworks such as: Arduino and Espressif which was selected because it has the necessary elements to build based on the BLE protocol, in other words, it is necessary the management of the ESP 32 module to be in charge of presenting common examples of real life adapting with other peripherals such as taking temperature, the presentation of information to be implemented in a home automation area and it is intended to understand an RGB lighting system (red, green, blue) in two types of ESP 32 models to achieve the control that allows to achieve with the nRF Mesh application. The design offered is adaptable to incorporate new nodes or devices provisioned for long distance structures that turns it into a functional network controlling the RGB lighting system.

The project ends with the tests obtained that were helpful to illustrate the capacity of a BLE network in a complex structure, so that the connection that has the ESP WROOM 32 module at less than 20 meters with the mobile is important to know the range of reach and thus in the future design the network of devices that allow connectivity between nodes. In particular the creation of groups that can be accessed through the App nRF Mesh is essential for the management of the elements according to their characteristics and how tidy can be the App nRF Mesh when large numbers of provisioned nodes are

added, on the other hand, it was found that the application is not harmful to the user and only requests the basic permissions for proper operation.

Keywords: Wireless technologies, IoT, ESP32, Provisioning, BLE Mesh.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
1 CAPÍTULO I: DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS.....	3
1.1 Ámbito de Aplicación: descripción del contexto y hechos de interés.....	3
1.2 Establecimiento de requerimientos.....	3
1.3 Justificación del requerimiento a satisfacer	4
2 CAPÍTULO II: DESARROLLO DEL PROYECTO.....	5
2.1 Definición del prototipo tecnológico	5
2.2 Fundamentación teórica del prototipo	6
2.2.1 Internet de las cosas (IoT)	6
2.2.1.1 Aplicaciones IoT	6
2.2.2 Tecnologías inalámbricas	7
2.2.2.1 ZigBee y Thread	7
2.2.2.2 Z-Wave.....	8
2.2.2.3 LoRaWan	9
2.2.2.4 6LowPAN	9
2.2.2.5 Wifi	9
2.2.2.6 Bluetooth	10
2.2.2.7 Comparativa	11
2.2.3 Microcontroladores	11
2.2.4 Bluetooth Mesh	12
2.2.4.1 Pila del protocolo	13
2.2.4.2 Composición de nodos.....	14
2.2.4.3 Seguridad.....	15
2.3 Objetivos del prototipo.....	15
2.3.1 Objetivo General	15
2.3.2 Objetivos Específicos.....	15
2.4 Diseño del prototipo.....	15

2.4.1	Herramientas de desarrollo.....	15
2.4.1.1	Hardware.....	15
2.4.1.2	Software	16
2.4.2	Diseño del Hardware	17
2.4.2.1	Sensor de Temperatura DS18B20	17
2.4.2.2	Display LCD 16x2.....	18
2.4.2.3	LEDC Started	20
2.4.3	Diseño BLE Mesh	20
2.4.3.1	Conexión para los módulos ESP32.....	20
2.4.3.2	Conexión en grupos en los módulos con control de luz RGB.....	21
2.5	Ejecución y/o ensamblaje del prototipo	23
2.5.1	Instalación del Sensor Temperatura DS18B20	24
2.5.2	Instalación del Display LCD 16x2	28
2.5.3	Instalación del LEDC Started.....	31
2.5.4	Instalación de la Red BLE Mesh.....	35
2.5.4.1	Aprovisionamiento con la App nRF Mesh.....	39
2.5.4.2	Asignación de grupos a los nodos.....	45
3	CAPÍTULO III: EVALUACIÓN DEL PROTOTIPO	49
3.1	Plan de evaluación	49
3.2	Resultados de evaluación	49
3.2.1	Pruebas de distanciamiento de los nodos	49
3.2.2	Pruebas en los Grupos de la malla.....	54
3.2.3	Seguridad de la App nRF Mesh	56
3.3	Conclusiones.....	59
3.4	Recomendaciones.....	60
	REFERENCIAS BIBLIOGRÁFICAS	61

ÍNDICE DE FIGURAS

Figura 1: Estructura propuesta con BLE Mesh	5
Figura 2: Topologías de ZigBee	8
Figura 3: Capas de protocolo	8
Figura 4: Pila de protocolo.....	9
Figura 5: Descripción de la interconexión IoT.....	10
Figura 6: Arquitectura de Harvard	12
Figura 7: Historia de Bluetooth	13
Figura 8: Pila de protocolo BLE	13
Figura 9: Nodo con conexiones.....	14
Figura 10: Tipos de nodos BLE Mesh	14
Figura 11: Sensor de temperatura LM35.....	18
Figura 12: Modelado del circuito Temperatura DS18B20.....	18
Figura 13: Modelado del circuito Display LCD 16x2.....	19
Figura 14: Modelado del circuito LEDC started.....	20
Figura 15: Red BLE Mesh	21
Figura 16: Conexión LED para el módulo ESP-WROOM-32.....	22
Figura 17: Conexión LED para el módulo ESP-WROOM-32D	22
Figura 18: Red BLE Mesh para sistema de iluminación	23
Figura 19: Estructura para el desarrollo de aplicaciones del ESP32	23
Figura 20: Circuito del sensor de temperatura.....	27
Figura 21: Datos recibidos del sensor de temperatura	27
Figura 22: Conexión del Display LCD 16x2.....	30
Figura 23: Datos recibidos del display LCD 16x2.....	30
Figura 24: Mensaje presentado por pantalla	31
Figura 25: Conexión del circuito LEDC started.....	34
Figura 26: Datos recibidos de la aplicación	35
Figura 27: Ejecución del circuito.....	35
Figura 28: Datos recibidos del nodo	39
Figura 29: Interfaz de inicio App nRF Mesh.....	39
Figura 30: Escaneo del módulo para uno o más módulos.....	40
Figura 31: Identify and provision.....	41
Figura 32: Aprovisionamiento del nodo	41
Figura 33: Nodo aprovisionado.....	42

Figura 34: Elementos del nodo.....	43
Figura 35: Clave de la aplicación.....	43
Figura 36: Control de la aplicación	44
Figura 37: Información recibida de la App nRF Mesh.....	44
Figura 38: Interfaz de grupo nRF Mesh.....	45
Figura 39: Crear grupos Mesh.....	46
Figura 40: Nodo ESP32.....	46
Figura 41: Asignar el elemento al grupo.....	47
Figura 42: Elemento suscrito al grupo	48
Figura 43: Grupo 1 con el elemento asignado.....	48
Figura 44: Circuito ESP-WROOM-32	49
Figura 45: Circuito ESP-WROOM-32D.....	50
Figura 46: nRF Mes conectado a un nodo.....	50
Figura 47: ESP32 disponible	51
Figura 48: Control de iluminación RGB - rojo	51
Figura 49: Control de iluminación RGB - azul.....	52
Figura 50: Limitación del rango Bluetooth	52
Figura 51: Conexión de 2 nodos.....	53
Figura 52: Aprovisionamiento del nodo nuevo.....	53
Figura 53: Grupos en la App.....	54
Figura 54: Suscribiendo elementos a los grupos.....	55
Figura 55: Grupos con sus elementos.....	55
Figura 56: Led agrupados.....	56
Figura 57: Virus Total	56
Figura 58: Escanear el dispositivo.....	57
Figura 59: Escaneo de nRF Mesh	57
Figura 60: Información y reportes.....	58
Figura 61: Permisos de la aplicación.....	58

ÍNDICE DE TABLAS

Tabla 1: Comparación de tecnologías inalámbricas	11
Tabla 2: Características ESP-WROOM-32	12
Tabla 3: Requisitos del Hardware	16
Tabla 4: Requisitos computacionales	16
Tabla 5: Requisitos del Software	17
Tabla 6: Conexión de pines entre ESP32 y LCD 16x2	19

INTRODUCCIÓN

Durante la era tecnológica, el desarrollo de dispositivos inteligentes ha sido fundamental para la automatización de procesos. Según [1], los circuitos electrónicos son cada vez más útiles dentro de nuestro diario vivir, desde servirnos para poner la hora en una alarma de radio, preparar un café exactamente a nuestro gusto, hasta llegar a la navegación automática de aviones, automóviles y barcos. El Internet de las cosas (IoT) tiene el objetivo de conectarse a través de internet entre dispositivos que se logran integrar el software, sensores, aplicaciones, entre otras tecnologías para implantar una conexión primordial.

Las tecnologías inalámbricas son adaptables a los diferentes proyectos que emplean dispositivos inteligentes para ser aplicables a varios campos como domótica que convierte una vivienda en un hogar inteligente, industrias y las redes de sensores inalámbricas (WSN) para monitorizar los parámetros de temperatura, presión, entre otros. La tecnología del Bluetooth clásico o nombrado como Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR), se establece en la topología de punto a punto que ayuda a emplearse en dispositivos para transferencias de datos o la transmisión inalámbrica del audio convirtiéndose en un protocolo estándar para sistemas como los auriculares o altavoces inalámbricos.

Bluetooth Low Energy (BLE) está diseñada con las determinadas normas del Bluetooth Clásico y posee características favorables para su implementación desde la topología en malla (incorpora también Point to Point - Broadcast) hasta su funcionamiento en baja potencia que se dirige para la transmisión de datos a más de 40 canales en una banda de frecuencia ISM 2,4 GHz. Considerando que la tecnología Bluetooth Low Energy (BLE), se ha convertido en la elección definitiva de muchos desarrolladores de IoT porque se ha integrado masivamente en los teléfonos inteligentes [2], también conocida como Bluetooth Smart [3].

La presente investigación, se centra en la nula automatización que tienen los hogares y el control ineficaz sobre los sistemas de iluminación, la implementación de una red BLE Mesh ofrece un sistema de control simple, ordenada y escalable para ir agregando nuevos dispositivos a la red para una estructura de largas distancias y que permite coexistir con la red Wi-Fi.

Este documento presenta una estructura de varios capítulos detallados de la siguiente manera:

Capítulo I: se detalla los motivos por el cual se desarrollará el proyecto, una breve descripción de los antecedentes y la necesidad de impulsar nuevas propuestas tecnológicas en el ámbito del Internet de las Cosas aplicando requerimientos primordiales de las tareas en la vida cotidiana.

Capítulo II: se estructura con la definición del prototipo tecnológico, los objetivos que serán alcanzados y el proceso de ensamblaje o ejecución del mismo.

Capítulo III: es la parte fundamental del proyecto ya que se realiza un plan de evaluación utilizando diferentes métodos para enfocar el uso de esta red BLE Mesh y que culmina con las conclusiones y recomendaciones para apoyar las bases de implementación en proyectos futuros.

1 CAPÍTULO I: DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1 Ámbito de Aplicación: descripción del contexto y hechos de interés

Por el desarrollo de la tecnología es inevitable la innovación, el Internet de las cosas (IoT) está alcanzando un camino importante en la red de informática global en donde ubica que todos y todo estará conectado a internet [4]. El IoT es considerado como una evaluación automática de aprendizaje de máquina a máquina (M2M), refiriéndose a la interconexión de cosas (televisores, sensores, actuadores, etc.) incluso para una red que funciona sin la intervención de las personas [5].

Según [6], en la actualidad todos los dominios de aplicación IoT muestran una creciente preocupación de las amenazas de seguridad, ataques y vulnerabilidades. La idea general de IoT es que sea autosuficiente para que pueda autorizar la conexión y el intercambio de datos seguros entre aplicaciones y dispositivos, el aumento de los dispositivos conectados a internet es irremediable no usar algún tipo de dispositivo inteligente [7]. Por lo tanto, la tecnología ayuda como herramienta de colaboración y toma de decisiones en un mundo en el que converge lo físico con lo digital [8].

Con la gran variedad de tecnología en IoT que participan activamente para el desarrollo de la comunicación, implementando un gran número de protocolos y tecnologías a base de los requisitos del dispositivo como el NFC, Bluetooth, Wi-Fi directo, entre otros [9] [10]. Con respecto al proyecto propuesto al implementar estos dispositivos con Bluetooth de baja energía (BLE) en malla es una gran solución para que estos se puedan comunicar entre sí, aunque un dispositivo falle es la clave para que la transmisión de datos llegue al receptor sin problemas. Para el desarrollo también se puede relacionar con la domótica que es una tecnología aplicada a la automatización inteligente del hogar que proporciona seguridad y comunicación entre el sistema y usuario que permitirá reducir el tiempo para las tareas cotidianas.

1.2 Establecimiento de requerimientos

Con el paso del tiempo las nuevas tecnologías de la información proveen sistemas de comunicación inalámbricas que pueden emplearse en los diferentes dispositivos por medio del bluetooth, los usuarios dependen de ello para transferir archivos, intercambiar datos, etc, sin poseer conocimientos avanzados en estructuras de red. IoT prevalece un gran impacto en la sociedad y los negocios con los millones de usuarios que hacen uso del Internet, para su vida laboral y social todo es posible por las diferentes tecnologías inalámbricas [11].

Bluetooth Low Energy proporciona estas funciones, pero con topologías de comunicación de Point to Point, Broadcast y Mesh, siendo la topología Mesh una red adaptable y escalable para solucionar cualquier tipo de problema estructural sin interrumpir la señal Wi-Fi. Promoviendo el bajo consumo de los dispositivos. Según [12], es primordial tener un control y administración de dicha energía con la finalidad de regular su uso y no generar pérdida innecesaria en periodos extensos de tiempo.

La propuesta tecnológica va dirigida a que apoye este nuevo protocolo BLE para la implementación de comunicación en malla que será el encargado de que el mensaje no se interrumpa. El usuario podrá conectarse mediante una aplicación para Android o IOS y adquirir aprovisionarse con los diferentes ESP32 (microcontrolador principal) para que obtengan una expansión de conexión de dispositivos y que se logrará simular un sistema de control de iluminación RGB siendo capaz de funcionar a largas distancias alcanzado en el proceso ejemplos reales de asistencias para el usuario.

1.3 Justificación del requerimiento a satisfacer

El Internet de las cosas aborda diferentes campos de aplicación como la agronomía, industria, salud, domótica. También un conjunto de impresoras con red multifuncionales requiere el monitoreo, control y la administración de estos equipos [13], es decir todo son procesos que facilitan el trabajo, supervisan o minimizan los riesgos humanos por ello es necesario automatizarse hacia un futuro que todo esté relacionado con IoT. Los múltiples avances que se han desarrollado en la tecnología les permite implementar sistemas inteligentes que sean capaces de solucionar problemas reales o apoyarse en el proceso.

Las tecnologías inalámbricas son las que ofrecen diversos protocolos de comunicación para que sea seguro, veloz y de bajo presupuesto. Estos mismos protocolos cuando son implementados tienen como propósito la relación entre personas y máquinas capaces de coexistir cuando se habla de una conectividad inalámbrica.

El proyecto tecnológico ofrece el enfoque necesario para lograr implementar una tecnología domótica idóneo para que el hogar se convierta en un sistema inteligente funcional que ampliamente apoyado con Bluetooth Low Energy tiene características importantes de ser una tecnología que favorece la conexión de dispositivos inteligentes y que prevalece el control humano, siendo BLE una tecnología inalámbrica de bajo presupuesto.

2 CAPÍTULO II: DESARROLLO DEL PROYECTO

2.1 Definición del prototipo tecnológico

En las últimas décadas el desarrollo de la tecnología ha sido fundamental para el confort de la vida humana, con la implementación de dispositivos electrónicos en los campos de la industria, agricultura, salud, automotriz, entre otros. Específicamente la domótica es la automatización y control de dispositivos (on/off) como climatización e iluminación que a un conjunto aplicada de tecnología se convierte en un hogar o vivienda inteligente [14].

La tecnología inalámbrica es la encargada de que el usuario obtenga la información adecuada y que pueda realizar las tareas dejando atrás el recorrido de grandes distancias, en la Figura 1 se presenta una estructura de red BLE Mesh que logra ser escalable además que conectada a una Aplicación nRF Mesh sea capaz de comunicarse con los módulos ESP32 sin tener la necesidad de interrumpir la señal Wi-Fi o conectarse a un router. Como un proceso de automatización al microcontrolador del ESP32 será el encargado de ofrecer ejemplos reales en la utilización de dispositivos, conceptualizando la idea de integrar en una red BLE Mesh en los diferentes campos de aplicación ahorrando energía, tiempo y respuesta en la comunicación.

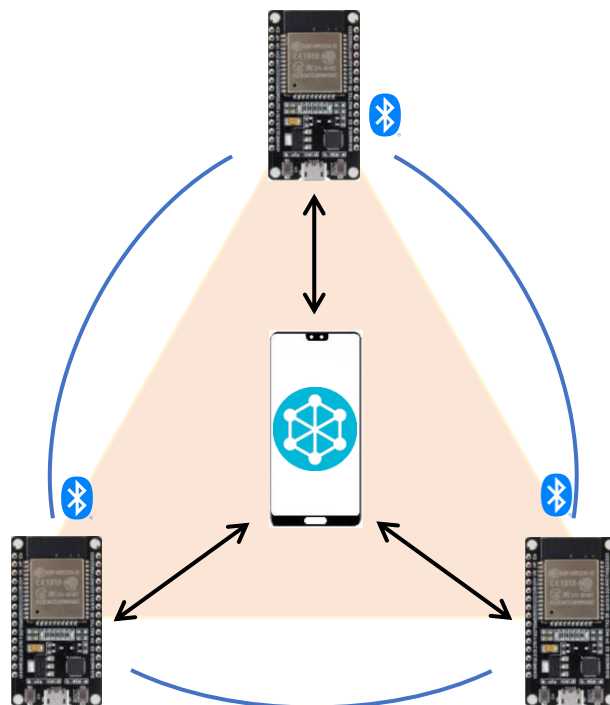


Figura 1: Estructura propuesta con BLE Mesh

Fuente: Elaboración del Autor

2.2 Fundamentación teórica del prototipo

2.2.1 Internet de las cosas (IoT)

Internet es un modelo para el desarrollo que va alentando al crecimiento tecnológico en la comunidad mundial, por el mismo motivo del uso desproporcionado del internet llega IoT como expresión de vincular el mundo físico [15]. Es un sueño futurista estar en comodidad con todos los dispositivos o lugares modernizados por el cual, según [16] permite a los profesionales estar a la altura para crear múltiples soluciones para que se puedan comunicar a través de internet para procesos automatizados y poder brindar servicios.

Al inicio de toda evolución tecnológica existen diferencias de opiniones, teorías o pruebas, por eso [17] puntualiza que los dispositivos IoT presentan problemas de privacidad y seguridad, por ello los fabricantes de dispositivos dan prioridad a estos problemas. El proceso de las tecnologías dan paso a la recopilación de datos para crear arquitecturas, protocolos, dominios de aplicación para poder solventar los problemas en el desarrollo [18] [19].

El IoT no se puede comparar solo con la tecnología de máquina a máquina (M2M), a la vez que [20] lo relaciona al internet de las cosas con los contenidos virtuales ya que no se limita solo a los dispositivos físicos y se introduce en un entorno que rodea a las personas.

2.2.1.1 Aplicaciones IoT

La razón más influyente para aplicar IoT es la relación del ser humano con la tecnología que brinda una automatización de los recursos, el comportamiento inteligente impulsado por un dispositivo es exponerse a una atmósfera totalmente diferente pero la disminución de errores lo hace en una tecnología sostenible [21] [22]. Se describen algunos ejemplos de aplicaciones IoT:

- El Sistema de Información Ambiental no puede obtener mediciones ambientales en tiempo real por el cual se implementa M2M para aplicar Internet de las cosas como una tecnología que se apoya en la comunicación [23].
- Según [24], la agricultura tradicional se ha envuelto transformado por la gran importancia de IoT, convirtiéndose en un plan para reducir el consumo de agua con la implementación de sensores. Además, se puede utilizar en sensor polarográfico de clorofila y sensor espectrofotométrico de clorofila [25]. La

agricultura de precisión es otro campo que al combinarse con la información diversa y las Tecnologías de la comunicación administrando los procesos y datos llega a aumentar la productividad y eficiencia [26] [27].

- La geolocalización en donde se puede estimar la posición geográfica de un objeto o persona específicamente en adultos mayores para agendar un registro de sus actividades, también, los diferentes sensores proporcionan capacidades de plug and play para la obtención de detectores de sensores activos de forma precisa y rápida [28].
- Según [29], en la salud se aplica para el monitoreo de la presión arterial requiriendo fases de diseño se puede construir un prototipo funcional a base de dispositivos IoT. En la atención médica actual todas las ciudades enfrentan desafíos para mejorar la salud en general por ello IoT tiene un valor fundamental [30].

2.2.2 Tecnologías inalámbricas

Existen diversas tecnologías para la aplicación del Internet de las cosas, cada una de ellas tendrán diferentes atributos como el modelo de comunicación, relación de costos, consumo de energía y el rendimiento. Se analizarán las siguientes tecnologías:

2.2.2.1 ZigBee y Thread

La tecnología ZigBee fue creado por ZigBee Alliance, que es basada en el estándar IEEE 802.15.4, se usa por lo general en aplicaciones-- con baja velocidad para transmitir cantidades pequeñas de 250 kbps con una arquitectura de red que tiene como atributos a maestro y esclavo logrando así una comunicación bidireccional [31].

ZigBee se presenta como un protocolo de red inalámbrica que su topología más empleada es en malla, pero también reconoce la topología de estrella como se muestra en la Figura 2, siendo su mayor ventaja la escalabilidad dirigida a aplicaciones de domótica y control remoto.

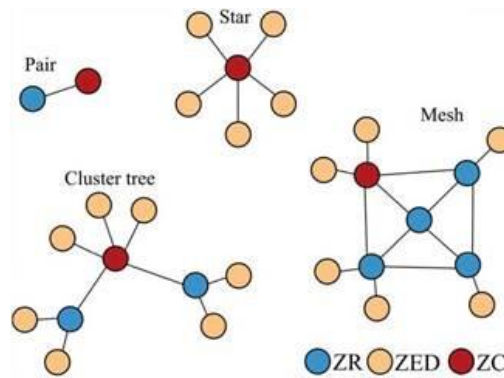


Figura 2: Topologías de ZigBee

Fuente: Tomado de [32]

La tecnología Thread es desarrollada y dirigida por Thread Group, con una solución de red escalable basado en el protocolo 6LowPAN con arquitecturas de controlador para flexibles directorios [33]. Diseñado para que sea compatible con redes mesh y con IEEE 802.15.4.

2.2.2.2 Z-Wave

En sus inicios Z-Wave se diseñó para iluminación en un control de sistemas, pero se desarrolló en un protocolo de automatización gestionado por Z-Wave Alliance. Z-Wave fue originalmente ZenSys como protocolo de IoT patentado. Los mensajes se intercambian de manera asíncrona a través del módulo RF como tramas para las unidades de datos del protocolo MPDU [34].

Según [35] Z-Wave es una red con topología de malla que implementa ondas de radio con baja energía para la comunicación de los nodos que dentro del alcance interactúan entre sí. En la Figura 3 se describen las capas de protocolo.

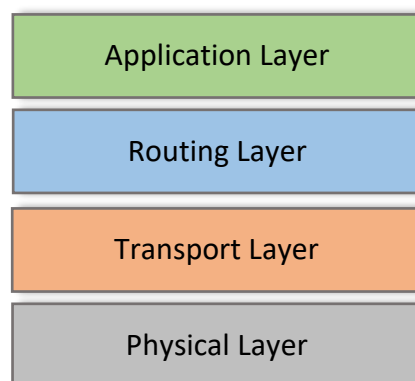


Figura 3: Capas de protocolo

Fuente: Elaboración del Autor

2.2.2.3 LoRaWan

El protocolo LoRaWan es desarrollado por LoRa Alliance que se especializa en redes de área amplia o WAN. Posee una tecnología de capa física que modula las señales con una comunicación bidireccional además que es difícil de detectar o atascar, los datos están entre los 300 bps y 50 kbps según sea el ancho de banda del canal que a la vez puede ser recibido simultáneamente [36] [37].

2.2.2.4 6LowPAN

IPv6 Low-Power Wireless Personal Area Network (6LowPAN) brinda soporte a la comunicación IPv6 que mejoran su funcionalidad desarrollando un estándar de comunicación que puede ser implementado para redes de sensores inalámbricos [38].

Desde su diseño hasta su aplicación permite su compatibilidad con internet utilizando un nodo Gateway basado en Arduino o Raspberry Pi, sin embargo su operación son por periodos de larga duración y no logra ser muy óptima [39]. Además, en la Figura 4 se presenta la estructura de la pila de protocolo del 6LowPAN.

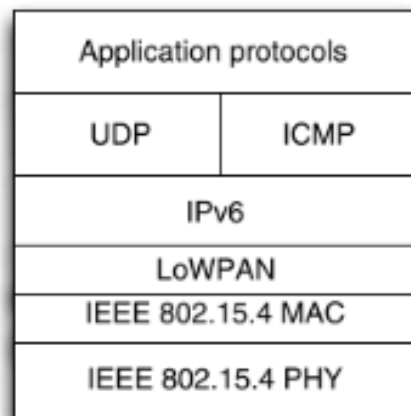


Figura 4: Pila de protocolo

Fuente: Tomado de [40]

2.2.2.5 Wifi

El protocolo Wifi sigue el estándar IEEE 802.11 para cualquier red comercial para el área local inalámbrica. Según [41] en los últimos años se ha realizado una gran cantidad de trabajos para monitorizar de forma remota las variables en un entorno de centros de datos para sensores de redes inalámbricas, humedad, temperatura y sistemas de enfriamiento.

Wifi básicamente se emplea en una banda de frecuencia de 2,4 y 5 GHz utilizada por diversas tecnologías como identificación RF [42]. Es considerable en su velocidad de transferencia que alcanza los 600 Mbps, por el cual en una red Wifi Mesh fácilmente podría escalar a más de 1000 nodos sin necesitar una gran infraestructura tecnológica específica.

2.2.2.6 Bluetooth

La tecnología bluetooth es una gran solución inalámbrica y proporciona flexibilidad en topologías de ancho de banda y rango. Pero existe dos tipos de Bluetooth: Bluetooth Classic se usa especialmente para aplicaciones de transmisión de audio para comunicación de punto a punto y Bluetooth Low Energy (BLE) para un funcionamiento de baja potencia además que admite diferentes tipologías desde punto a punto hasta que se extiende a broadcast hasta una topología en malla [43]. En la Figura 5 es un detallado de IoT en las diferentes conexiones y aplicaciones además del bluetooth.

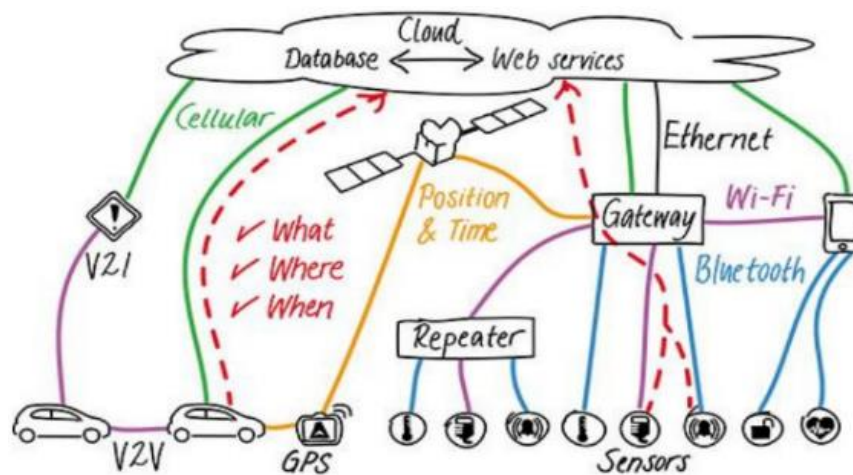


Figura 5: Descripción de la interconexión IoT

Fuente: Tomado de [44]

2.2.2.7 Comparativa

En la Tabla 1 se presenta una comparativa de las diferentes tecnologías, con relación a los atributos que poseen cada una de ellas y de manera adecuada sus características para que sea fácilmente relacionado con el tipo de proyecto a implementar.

Tabla 1: Comparación de tecnologías inalámbricas

	IEEE 802.15.4 (ZigBee -Thread)	Z-Wave	LoRaWAN	Wi-Fi	Bluetooth C	BLE
Velocidad	250 kbps	100 kbps	50 kbps	54 Mbps – 1.3 Gbps	2 Mbps	2 Mbps
Frecuencia	2.4 GHz	868 MHz	Varias	2.4 – 5.8 GHz	2.4 GHz	2.4 GHz
Cobertura	30 m - 100 m	30 m - 50 m	2 km - 20 km	15 m - 100m	10 m - 100m	10 m – 1.5 km
Topologías	Punto a punto Estrella Mesh	Mesh	Estrella	Estrella Mesh	Punto a punto	Punto a punto Broadcast Mesh
Seguridad	AES 128	AES 128	AES 128	AES CCM 128	AES 128	AES 128

Fuente: Tomado de [45] [46]

2.2.3 Microcontroladores

El microcontrolador es la parte fundamental para cualquier dispositivo programable permitiendo así ejecutar las tareas grabadas en su memoria además de ser el tipo de procesador más implementado en el mundo ya que es posible utilizarlo por sus puertos de entrada y salida [47]. En la actualidad existen empresas encargadas en su desarrollo como: Espressif, Arduino, Xbee, entre otros, que permiten que se ensamblan fácilmente en cualquier proyecto relacionado con IoT.

El funcionamiento interno de un microcontrolador consta de su memoria no volátil diseñados para almacenar la programación, memoria de escritura/lectura, líneas de entrada/salida y el procesador para que pueda obtener mejores rendimientos se emplea la arquitectura Harvard como se puede observar en la Figura 6, con dos tipos de memorias de manera independiente y posee una característica dual que cuenta con su propio sistema.

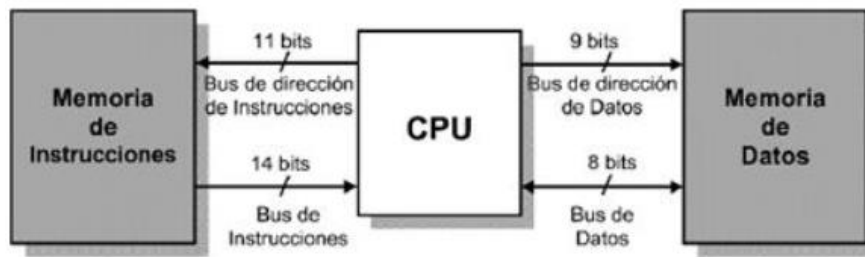


Figura 6: Arquitectura de Harvard

Fuente: Tomado de [48]

La familia Espressif posee diferentes microcontroladores que apoyan el proceso para IoT, desde su arquitectura con tecnologías inalámbricas hasta su compatibilidad con dispositivos inteligentes. Se detalla en la Tabla 2 las principales características del ESP-WROOM-32 como la conectividad del Wi-Fi y Bluetooth.

Tabla 2: Características ESP-WROOM-32

Chip (Module)	ESP32 (ESP-WROOM-32)
CPU	Tensilica Xtensa LX6 32 bit Dual-Core at 160/240 MHz
SRAM	520 KB
Flash	2 MB (max. 64MB)
Voltage	2.2V to 3.6V
Programmable	Free (C, C++, Lua, etc.)
Wi-Fi	802.11 b/g/n
Bluetooth	4.2 BR/EDR + BLE

Fuente: Tomado de [49]

El módulo ESP32 puede ser fácilmente localizado en el mercado electrónico por su bajo precio además admite plataformas como Arduino y ESP-IDF que cuentan con bastante documentación [50]. Su utilidad aumenta cuando puede soportar una red BLE Mesh para un proyecto a gran escala que lo convierte en un buen producto para un sistema inteligente.

2.2.4 Bluetooth Mesh

La nueva tecnología se ha convertido con el paso del tiempo en una importante herramienta para los servicios de iluminación y encendido/apagado de dispositivos. En la Figura 7 se presenta la poca historia del bluetooth en la era tecnológica.

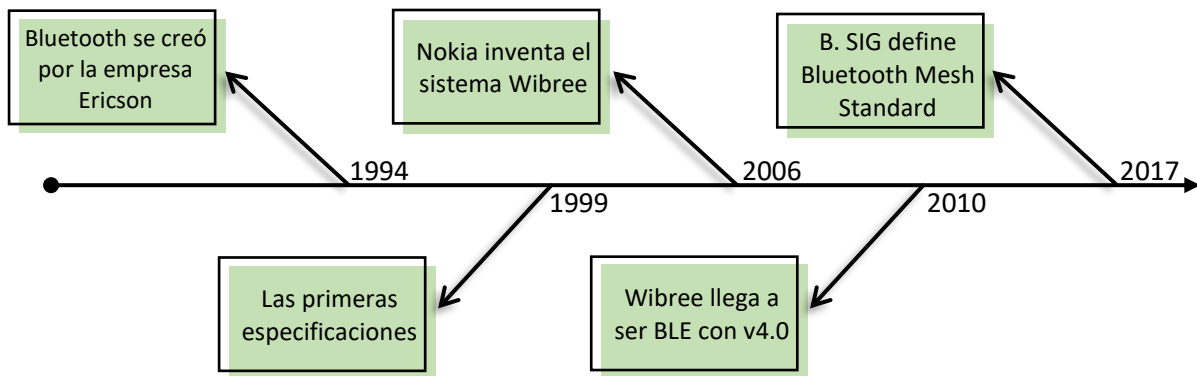


Figura 7: Historia de Bluetooth

Fuente: Elaboración del Autor

Bluetooth SIG en el año 2017 lanzando el protocolo Bluetooth Mesh como su novedosa tecnología desarrollada, siendo BLE una transmisión inalámbrica de corta alcance por lo cual su baja complejidad y el menor costo. Está incorporado desde la versión 4.0 de Bluetooth Core [51] [52]. Optimizando la vida útil en los dispositivos con una batería pequeña aligerando así el ciclo de trabajo.

2.2.4.1 Pila del protocolo

Bluetooth Low Energy está conformado por una pila de protocolos ya que es el código que lo implementa. En la Figura 8 se puede observar que es una comunicación interna de BLE como el controlador encargado de especificar la frecuencia inalámbrica, la interfaz de controlador del host y el host que facilita una conexión y la transmisión sean seguras.

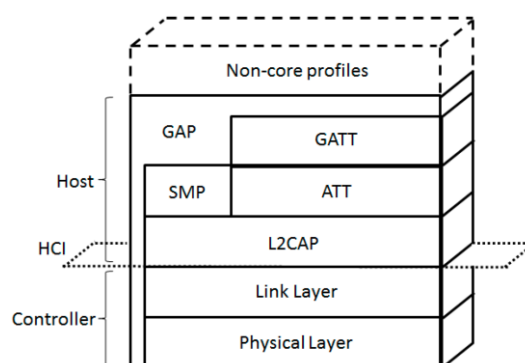


Figura 8: Pila de protocolo BLE

Fuente: Tomado de [53] [54]

2.2.4.2 Composición de nodos

Los nodos son puntos de intersección, unión o la conexión para la comunicación que en un conjunto pueden formar algún tipo de topología, en la Figura 9 se puede observar el estado del nodo que emite el ID del nodo o el ID del cluster esperando otro nodo para relacionarse.

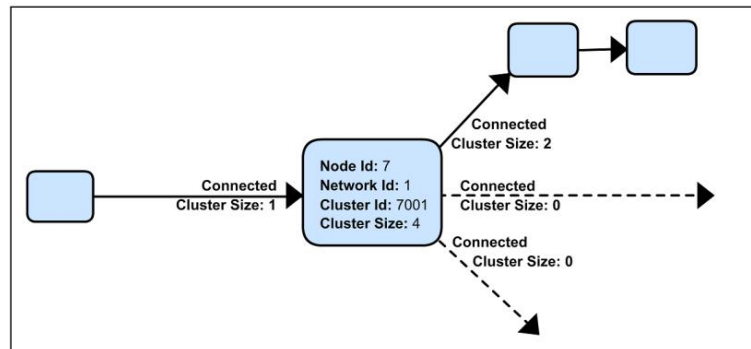


Figura 9: Nodo con conexiones

Fuente: Tomado de [55]

En toda estructura de comunicación existen diferentes tipos de nodos, en la Figura 10 los nodos más importantes son el relay node que es exclusivo para la transmisión de datos, proxy node funciona para dispositivos BLE por GATT y aprovisioner node ayuda al registro de nodos en la BLE Mesh y configura o distribuye las diferentes claves de seguridad.

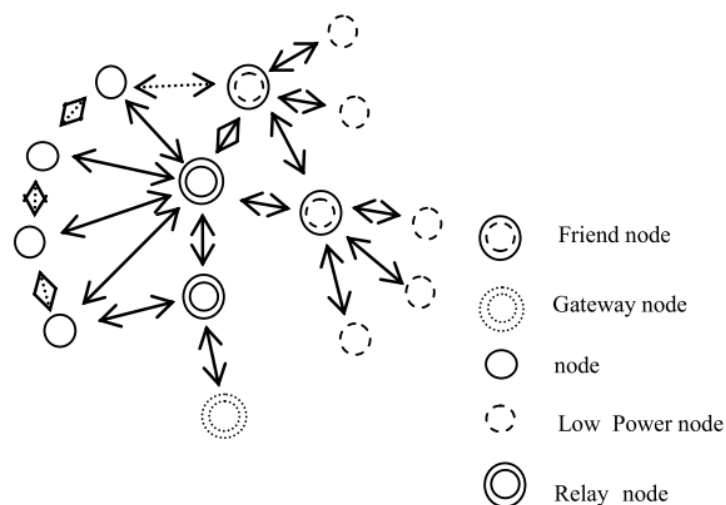


Figura 10: Tipos de nodos BLE Mesh

Fuente: Tomado de [56]

2.2.4.3 Seguridad

La red inalámbrica es el administrador para designar la conexión en los nodos sin la necesidad de pasar por cables la transmisión y recepción. Existen algunas vulnerabilidades que pueden afectar la estructura inalámbrica: Denial of Service, Eavesdropping, Man in the Middle, Relay attack y Replay Attack [57], por el mismo caso la seguridad de la red es importante y por ello BLE implementa el Advanced Encryption Standard (AES 128) que es una estructura de cifrado por bloques.

2.3 Objetivos del prototipo

2.3.1 Objetivo General

Implementar un sistema de dispositivos ESP32 mediante una red de Bluetooth Low Energy Mesh para la comunicación y control de un sistema de iluminación RGB en el teléfono móvil.

2.3.2 Objetivos Específicos

- Analizar las diferentes tecnologías inalámbricas y comparar sus atributos más importantes.
- Configurar los periféricos internos del ESP32 como ejemplos prácticos para el uso en IoT.
- Diseñar una red BLE entre el módulo ESP32 y el nRF Mesh móvil para una comunicación de punto a punto.
- Implementar una red BLE Mesh en los módulos ESP-WROOM-32 y ESP32-WROOM-32D.
- Establecer grupos en los módulos ESP32 con la aplicación nRF Mesh para un sistema de iluminación.

2.4 Diseño del prototipo

2.4.1 Herramientas de desarrollo

2.4.1.1 Hardware

Para la implementación de un sistema BLE Mesh y los ejemplos prácticos es necesario el uso de algunos dispositivos. En la Tabla 3 se detallan los componentes necesarios para el desarrollo de este proyecto.

Tabla 3: Requisitos del Hardware

Hardware	Definición
Módulo ESP32	Los módulos ESP-WROOM-32 y ESP32-WROOM-32D se implementan en la red.
Sensor de temperatura DS18B20	Capaz de calcular la temperatura del ambiente.
Display LCD 16x2	Presenta información por medio de una pantalla azul.
Breadboard	Brinda el soporte necesario para los componentes y ranuras de conexión.
LED	Simula un sistema de iluminación.
Resistencia	Fundamental para evitar la sobrecarga de energía.
Cable USB A / micro-USB B	Establece la conexión entre el módulo ESP32 y el computador.
Cables Dupont	Establece una conexión entre los dispositivos

Fuente: Elaboración del autor

Todos los componentes antes mencionados serán implementados según sea la función del circuito por el cual el uso de computadora o portátil (Windows, Linux o macOS) es requerido para la instalación de programas para la programación o estructurar el sistema de circuitos que serán empleados los componentes. En la tabla 4 se especifican con detalle los requisitos mínimos.

Tabla 4: Requisitos computacionales

Hardware	Requisitos
Portátil	8 GB RAM (mínimo) Procesador Intel Core i5 o i7 (preferido) Tarjeta de video NVIDIA GeForce MX130
Teléfono móvil	Android 8.1 (mínimo)

Fuente: Elaboración del autor

2.4.1.2 Software

Para realizar determinadas tareas es necesario implementar programas basado en el desarrollo de componentes lógicos, en la Tabla 5 se detallan los diferentes softwares que se implementan para el propósito de este proyecto.

Tabla 5: Requisitos del Software

Software	Definición
Python	Es un lenguaje de programación de multiparadigma y parcialmente soporta la orientación a objetos, además que el módulo ESP32 requiere de Python para su desarrollo.
Git	Funcionalmente es un software de control de versiones distribuidas más implementado.
Visual Studio Code	Proporciona las herramientas necesarias para desarrollar diversos tipos de programas como un editor de código fuente.
PlatformIO	Es una extensión IDE de Visual Studio Code permitiendo desarrollar en diferentes plataformas con el objetivo de implementar dispositivos de IoT.
CMake Tools	Es un conjunto de herramientas multiplataforma con el fin de crear y empaquetar software en código abierto.
nRF Mesh	Es una aplicación móvil para el aprovisionamiento, control y las configuraciones en redes de malla Bluetooth.

Fuente: Elaboración del autor

2.4.2 Diseño del Hardware

En las herramientas de desarrollo se detallaron los requisitos principales para el desarrollo del proyecto, por el cual en este apartado se presentará el diseño del prototipo con el hardware requerido. Basado en IoT con ejemplos de la vida real que son requeridos en un hogar, negocio, entre otros. El diseño va desde la estructura de los pines de los componentes hasta su conexión con el módulo ESP32 en la placa protoboard.

2.4.2.1 Sensor de Temperatura DS18B20

El dispositivo electrónico DS18B20 pertenece a la familia de sensores de temperatura que el mismo nombre lo indica permite calcular o estimar la temperatura por medio de una señal eléctrica pueden alcanzar un rango de estimación entre -55°C y 125°C . El sensor está compuesto por 3 pines o patas:

- VDD es la tensión positiva de alimentación es el principal para el funcionamiento del sensor.
- DQ conocido como el pin de datos que reciben todos los datos del protocolo 1-Wire.

Tabla 6: Conexión de pines entre ESP32 y LCD 16x2

ESP32	LCD
GND	VSS
+ 5V	VDD
GND	V0
33	RS
5	RW
19	E
12	D0
13	D1
14	D2
18	D3
25	D4
26	D5
27	D6
32	D7
+ 5V	A
GND	K

Fuente: Elaboración del autor

En la Figura 13 se puede observar el circuito modelado con base al registro de la Tabla 6 el cable Dupont hembra a hembra ofrece una gran ventaja si la conexión requiera conectar directamente al módulo ESP32 y no por medio del protoboard.

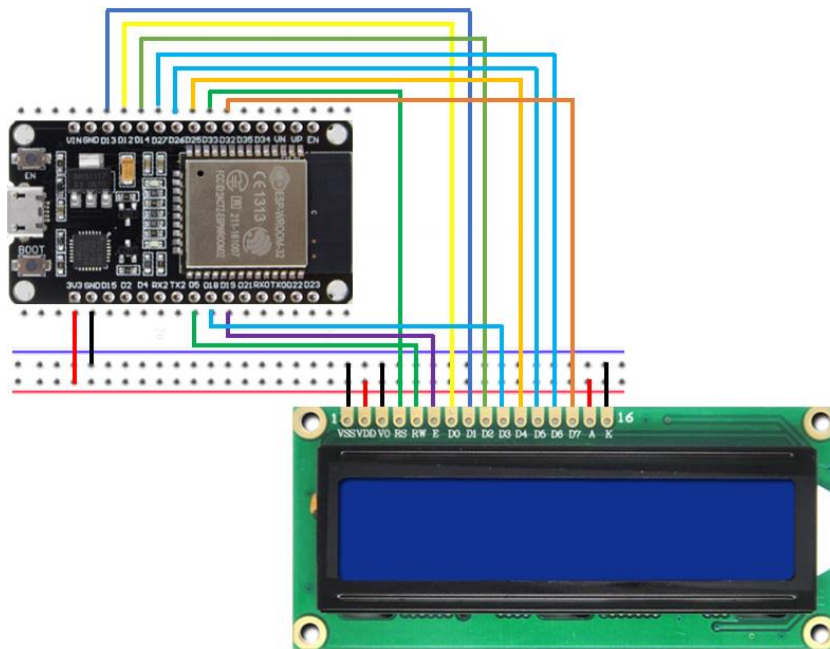


Figura 13: Modelado del circuito Display LCD 16x2

Fuente: Elaboración del autor

2.4.2.3 LEDC Started

La Figura 14 es una representación gráfica de la interacción de Leds para manipular los cambios de tiempo entre ellos.

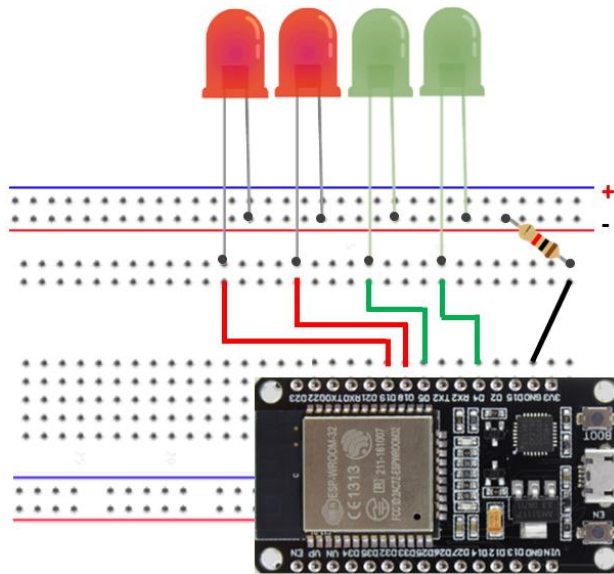


Figura 14: Modelado del circuito LEDC started

Fuente: Elaboración del autor

2.4.3 Diseño BLE Mesh

2.4.3.1 Conexión para los módulos ESP32

La red BLE Mesh es un conjunto de módulos ESP32 que interactúan entre sí, además de poseer un sistema escalable. En la Figura 15 se presenta una estructura de red BLE en donde se relaciona la App nRF Mesh con el nodo 1 y se va aprovisionando con los demás nodos para obtener un sistema escalable para que los nodos esperen a ser aprovisionados y que ejecute las ordenes cuando el cliente requiera.

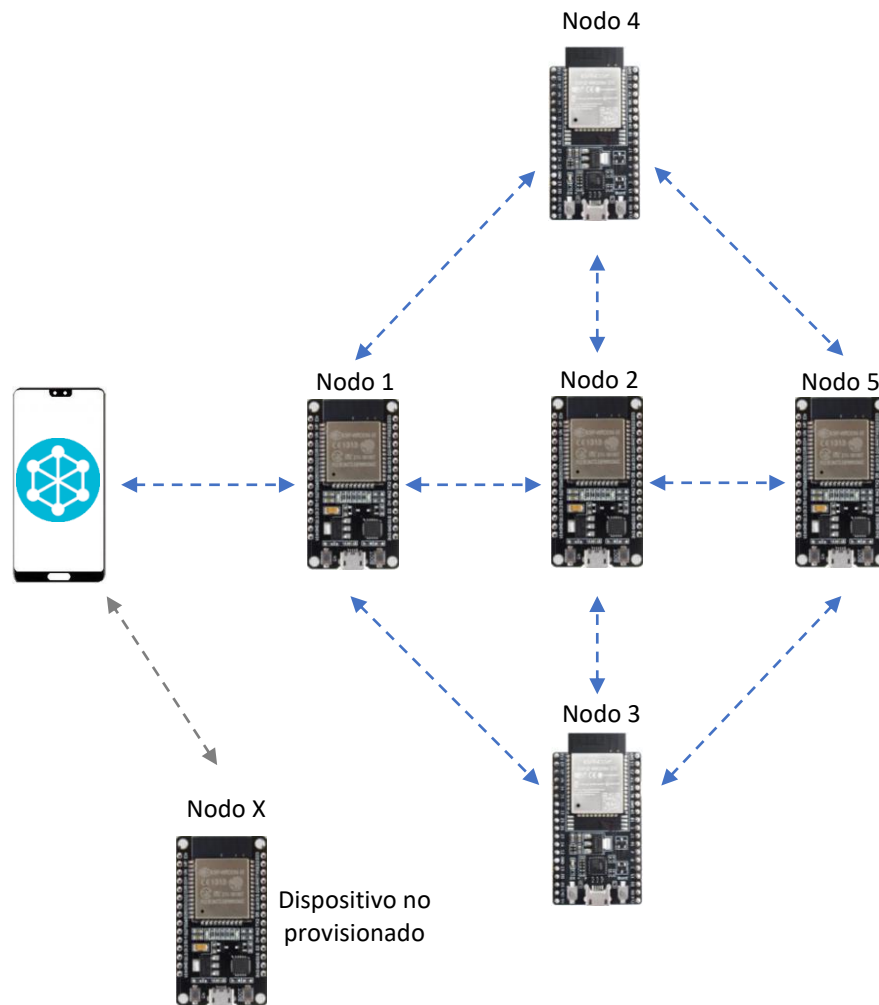


Figura 15: Red BLE Mesh

Fuente: Elaboración del autor

2.4.3.2 Conexión en grupos en los módulos con control de luz RGB

Los módulos ESP32 tienen varias funciones y una de ellas es aplicado al sistema de control de iluminación RGB (red – green - blue) que conectado a cada módulo su modelado de circuito es diferente, como se puede observar en la Figura 16 el diseño es para la conexión del ESP-WROOM-32 con sus respectivos pines de configuración.

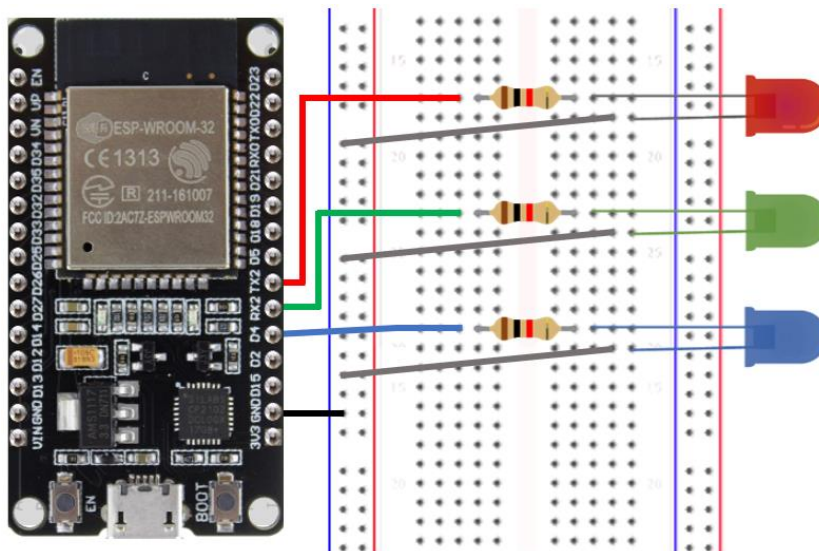


Figura 16: Conexión LED para el módulo ESP-WROOM-32

Fuente: Elaboración del Autor

El ESP-WROOM-32D presenta una similitud de pines con respecto al ESP-WROOM-32 por ello la Figura 17 es la estructura correcta de la utilización del GPIO, cada conexión tiene una entrada de datos o energía y salida a GND regulados por una resistencia.

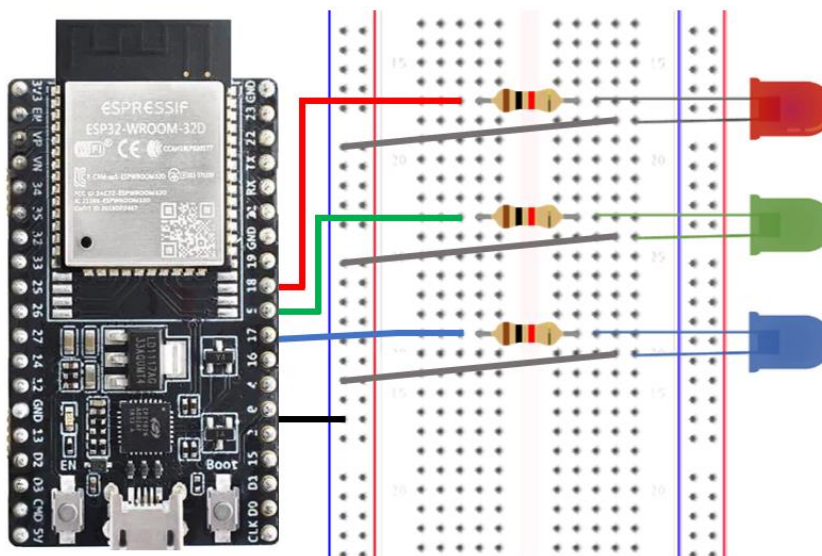


Figura 17: Conexión LED para el módulo ESP-WROOM-32D

Fuente: Elaboración del Autor

El sistema escalable de la red permite conectar los nodos en grupos para tener una estructura más ordenada y su configuración permite nombrar los grupos de acuerdo a las funciones que ocupan. En la Figura 18

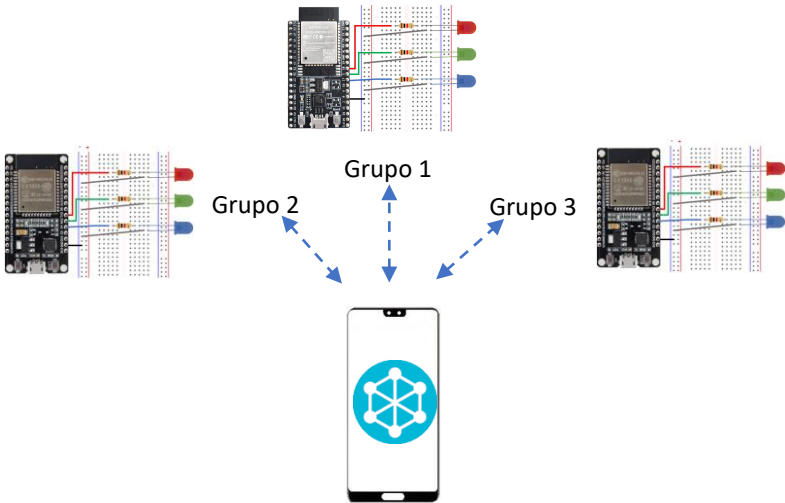


Figura 18: Red BLE Mesh para sistema de iluminación
Fuente: Elaboración del autor

2.5 Ejecución y/o ensamblaje del prototipo

En esta sección se describe el desarrollo de la estructura lógica de programación para el módulo ESP32 y la unión física de los componentes electrónicos a base del diseño antes propuesto. En la Figura 19 se presenta la estructura interna del proyecto y la interacción del ESP32 con el computador.

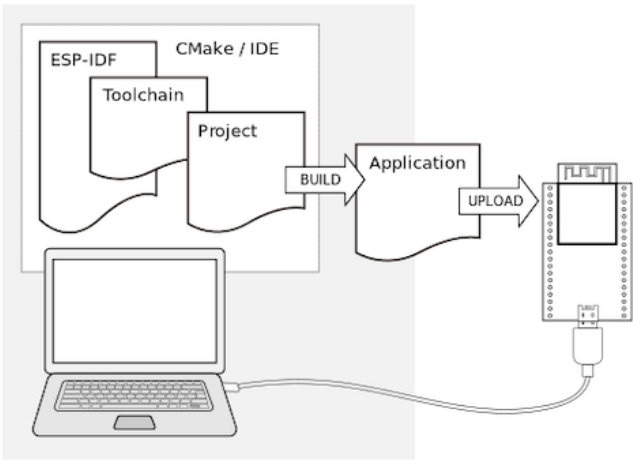


Figura 19: Estructura para el desarrollo de aplicaciones del ESP32
Fuente: Tomado de [59]

Como se establece la estructura en la Figura 19 el proyecto se crea en PlatformIO y Visual Studio Code permite desarrollar la aplicación a continuación, se detallará el esquema que se implementara para el desarrollo de los proyectos bajo la plataforma de ESP-IDF:

```
project_dir
├── include
├── src
│   ├── CMakeLists.txt
│   └── main.c
├── CMakeLists.txt
└── platformio.ini
```

2.5.1 Instalación del Sensor Temperatura DS18B20

Para la implementación de este tipo de sensor es necesario el uso de algunas librerías para la recepción de la temperatura y el número de modelo del sensor para ser exactos en su aplicación.

```
#include <ds18x20.h>
#include <esp_log.h>
#include <esp_err.h>
```

Se especifica el tipo de módulo con el cual se va a trabajar definiéndolo únicamente un tipo de tarjeta.

```
#define CONFIG_IDF_TARGET_ESP82
```

En la siguiente condición se procede a tomar en cuenta el modelo del ESP que se definió antes caso contrario elegirá automáticamente otro número de pin.

```
#if defined(CONFIG_IDF_TARGET_ESP82)
static const gpio_num_t SENSOR_GPIO = 4;
#else
static const gpio_num_t SENSOR_GPIO = 17;
#endif
```

En la toma de la temperatura se estableció un bucle de 500 milisegundos, es decir, que por pantalla presentará de forma repetida, además que permite un máximo de ocho

sensores a la vez cuando se requiera trabajar en datos de temperatura para varios ambientes.

```
static const uint32_t LOOP_DELAY_MS = 500;
static const int MAX_SENSORS = 8;
static const int RESCAN_INTERVAL = 8;
```

Es el nombre inicial con el cual se presentan los resultados.

```
static const char *TAG = "ds18x20_test";
```

En la función void se detallará lo más importante de la aplicación.

Realiza un conteo de los sensores conectados.

```
ds18x20_addr_t addrs[MAX_SENSORS];
float temps[MAX_SENSORS];
size_t sensor_count = 0;
```

Establece el modo de extracción.

```
gpio_set_pull_mode(SENSOR_GPIO, GPIO_PULLUP_ONLY);
```

Permite realizar un escaneo del sensor conectado y presenta el modelo caso contrario no encontrará ningún sensor o tiene errores el dispositivo.

```
res = ds18x20_scan_devices(SENSOR_GPIO, addrs, MAX_SENSORS, &sensor_count);
if (res != ESP_OK)
{
    ESP_LOGE(TAG, "Sensors scan error %d (%s)", res, esp_err_to_name(res));
    continue;
}

if (!sensor_count)
{
    ESP_LOGW(TAG, "No sensors detected!");
    continue;
}

ESP_LOGI(TAG, "%d sensors detected", sensor_count);
```

Ayuda que solo presente el número máximo de sensores.

```
if (sensor_count > MAX_SENSORS)
    sensor_count = MAX_SENSORS;
```

En este apartado el código muestra el modelo de sensor y los datos de la temperatura en Celsius y Fahrenheit.

```
for (int i = 0; i < RESCAN_INTERVAL; i++)
{
    ESP_LOGI(TAG, "Measuring...");

    res = ds18x20_measure_and_read_multi(SENSOR_GPIO, addrs, sensor_count,
    temps);
    if (res != ESP_OK)
    {
        ESP_LOGE(TAG, "Sensors read error %d (%s)", res, esp_err_to_name(res));
        continue;
    }

    for (int j = 0; j < sensor_count; j++)
    {
        float temp_c = temps[j];
        float temp_f = (temp_c * 1.8) + 32;
        ESP_LOGI(TAG, "Sensor %08x%08x (%s) reports %.3f°C (%.3f°F)",
        (uint32_t)(addrs[j] >> 32), (uint32_t)addrs[j],
        (addrs[j] & 0xff) == DS18B20_FAMILY_ID ? "DS18B20" : "DS18S20",
        temp_c, temp_f);
    }

    vTaskDelay(pdMS_TO_TICKS(LOOP_DELAY_MS));
}
```

Circuito del sensor de temperatura

El protoboard será la base para implementar los componentes, por el cual vamos a recibir los datos de la temperatura ambiental. En la Figura 20 mostrará el módulo ESP32 conectado a un cable USB como entrada de energía y los cables Dupont realizando la conexión con el sensor de temperatura.

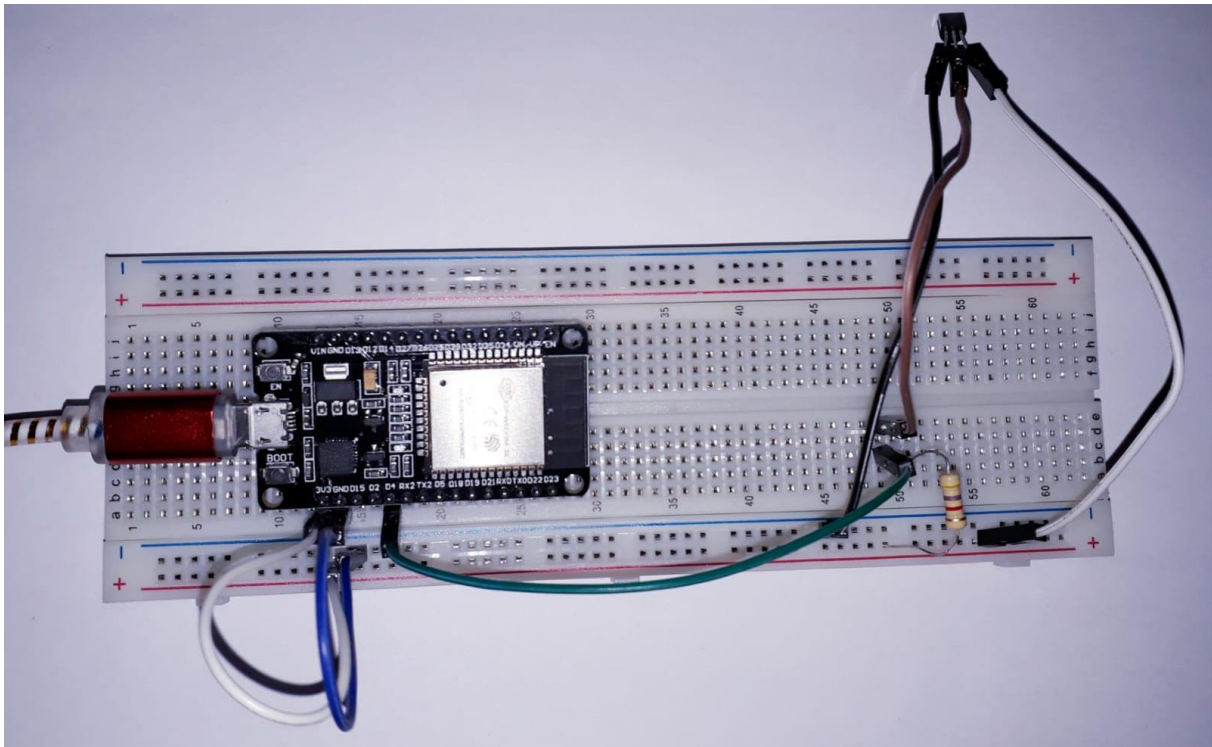


Figura 20: Circuito del sensor de temperatura

Fuente: Elaboración del autor

En la Figura 21 se puede observar que primero detecta al sensor para luego testarlo y presentar un reporte de los datos de la temperatura este proceso lo realiza cada 500 milisegundos para obtener mejores resultados de la temperatura ambiental.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
-----
es[0;32mI (0) cpu_start: Starting scheduler on APP CPU.es[0m
es[0;32mI (10) ds18x20_test: 1 sensors detected.es[0m
es[0;32mI (20) ds18x20_test: Measuring...es[0m
es[0;32mI (1861) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 27.562°C (81.613°F).es[0m
es[0;32mI (2361) ds18x20_test: Measuring...es[0m
es[0;32mI (3121) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 27.562°C (81.613°F).es[0m
es[0;32mI (3621) ds18x20_test: Measuring...es[0m
es[0;32mI (4381) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 27.625°C (81.725°F).es[0m
es[0;32mI (4881) ds18x20_test: Measuring...es[0m
es[0;32mI (5641) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 27.562°C (81.613°F).es[0m
es[0;32mI (6141) ds18x20_test: Measuring...es[0m
es[0;32mI (6901) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 27.562°C (81.613°F).es[0m
es[0;32mI (7401) ds18x20_test: Measuring...es[0m
es[0;32mI (8161) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 28.188°C (82.738°F).es[0m
es[0;32mI (8661) ds18x20_test: Measuring...es[0m
es[0;32mI (9421) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 29.375°C (84.875°F).es[0m
es[0;32mI (9921) ds18x20_test: Measuring...es[0m
es[0;32mI (10681) ds18x20_test: Sensor 4d00000da36dc028 (DS18B20) reports 30.062°C (86.113°F).es[0m
es[0;32mI (11191) ds18x20_test: 1 sensors detected.es[0m
es[0;32mI (11191) ds18x20_test: Measuring...es[0m

```

Figura 21: Datos recibidos del sensor de temperatura

Fuente: Elaboración del autor

2.5.2 Instalación del Display LCD 16x2

La utilización de librerías es esencial para la implementación del LCD 16x2 para la relación del código fuente con el dispositivo electrónico.

```
#include "driver/gpio.h"
#include "sdkconfig.h"
#include <math.h>
#include <string.h>
```

Inicializar los diferentes tipos de datos para estructurar los datos.

```
void lcd_init();
void lcd_cmd(unsigned char);
void lcd_data(unsigned char);
void lcd_decode(unsigned char);
void lcd_string(unsigned char*);
```

Se establecen los números de pines del ESP32 a utilizar con el LCD 16x2.

```
int lcd_pins[11] = {12,13,14,18,25,26,27,32,33,5,19};
```

Se presenta por pantalla la información requerida.

```
unsigned char msg[] = "Welcome";
```

Se realiza una estructura de control para estar pendiente de la ejecución de la aplicación.

```
void app_main(void)
{
    printf("Codigo esta empezando\n");
    while (1)
    {
        printf("printing msg on LCD\n");
        lcd_init();
    }
}
```

La función `lcd_init` se inicia los pines y llama las demás funciones.

```
void lcd_init()
{
    for(int i=0;i<11;i++)
    {
        gpio_pad_select_gpio(lcd_pins[i]);
        gpio_set_direction(lcd_pins[i],GPIO_MODE_OUTPUT);
    }
    lcd_cmd(0x38);
    lcd_cmd(0x01);
    lcd_cmd(0x0E);
    lcd_cmd(0x80);
    lcd_string(msg);
}

void lcd_decode(unsigned char info)
{
    unsigned char temp;
    for(int i=0;i<8;i++)
    {
        temp=pow(2,i);
        gpio_set_level(lcd_pins[i],(info&temp));
    }
}

void lcd_cmd(unsigned char cmd)
{
    lcd_decode(cmd);
    gpio_set_level(lcd_pins[8],0);
    gpio_set_level(lcd_pins[9],0);
    gpio_set_level(lcd_pins[10],1);
    vTaskDelay(10/portTICK_PERIOD_MS);
    gpio_set_level(lcd_pins[10],0);
    vTaskDelay(10/portTICK_PERIOD_MS);
}

void lcd_data(unsigned char data)
{
    lcd_decode(data);
    gpio_set_level(lcd_pins[8],1);
    gpio_set_level(lcd_pins[9],0);
    gpio_set_level(lcd_pins[10],1);
    vTaskDelay(10/portTICK_PERIOD_MS);
    gpio_set_level(lcd_pins[10],0);
    vTaskDelay(10/portTICK_PERIOD_MS);
}
```

Circuito del Display LCD 16x2

Para este tipo de conexiones se debe observar la Tabla 6 para conectar correctamente los pines y no tener un sobrecargada de energía que puede ocasionar el fallo del circuito interno del ESP32, en la Figura 22 es la conexión del ESP32 y el display LCD 16x2 realizando la relación el cable Dupont por medio del protoboard.

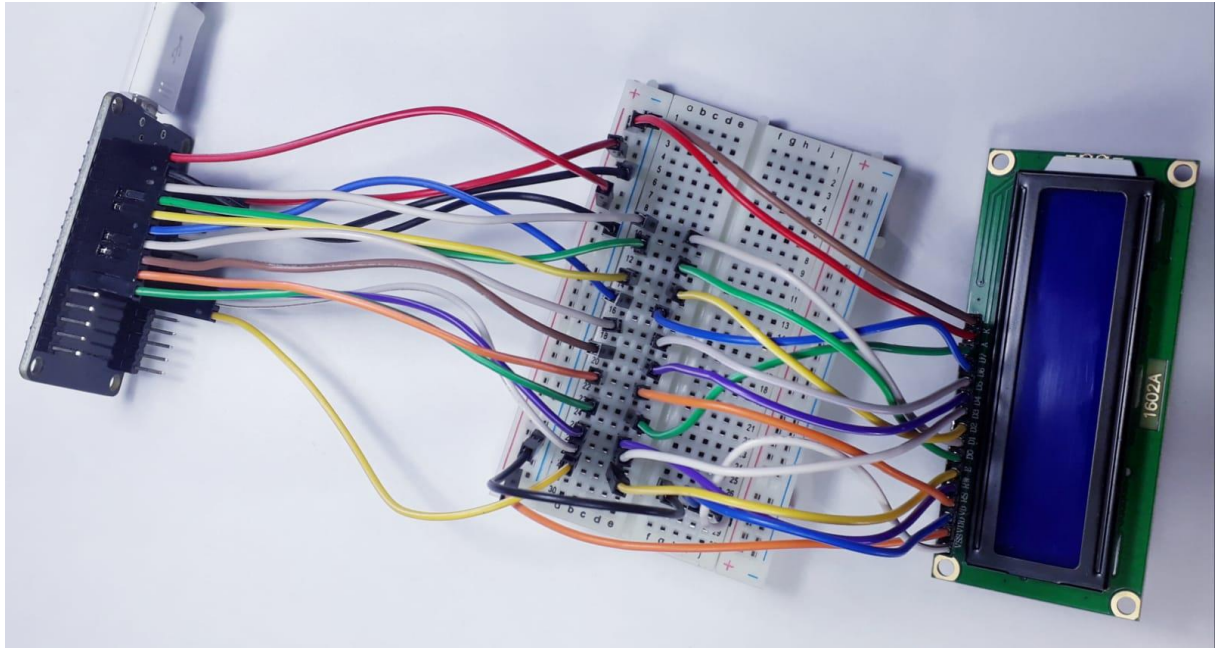


Figura 22: Conexión del Display LCD 16x2

Fuente: Elaboración del autor

En la Figura 23 es la información que se recibe después de ejecutar la aplicación que detalla que el código está comenzado y el mensaje está presentando por pantalla.

```
es[0;32mI (284) spi_flash: detected chip: generices[0m
es[0;32mI (287) spi_flash: flash io: dioes[0m
es[0;32mI (292) cpu_start: Starting scheduler on PRO CPU.es[0m
es[0;32mI (0) cpu_start: Starting scheduler on APP CPU.es[0m
Codigo esta empezando
printing msg on LCD
printing msg on LCD
printing msg on LCD
```

Figura 23: Datos recibidos del display LCD 16x2

Fuente: Elaboración del autor

El mensaje presentado en la Figura 24 es de forma continua, es decir, no para en presentar el mensaje ingresado para generar que el mensaje no sea interrumpido.

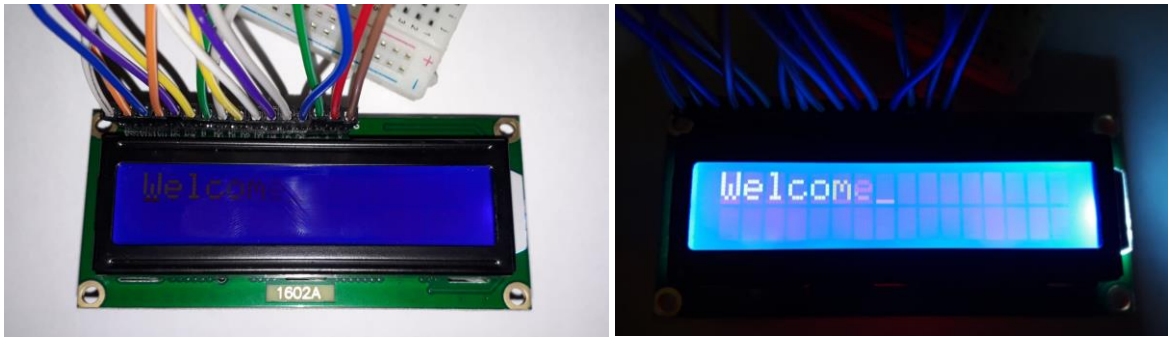


Figura 24: Mensaje presentado por pantalla

Fuente: Elaboración del autor

2.5.3 Instalación del LEDC Started

Es una serie de LED que se relacionan entre sí, se comunican de acuerdo al rango de encendido por lo mismo se implementan librerías que hacen uso del LEDC y el GPIO.

```
#include "driver/ledc.h"  
#include "esp_err.h"  
#include "driver/gpio.h"  
#include "sdkconfig.h"
```

Se implementa la inicialización del módulo ESP32, configurando el temporizador del LEDC (0-1) determinando la frecuencia y se especifica que el GPIO que se desea utilizar con sus respectivos canales de comunicación.

```
#ifdef CONFIG_IDF_TARGET_ESP32  
#define LEDC_HS_TIMER          LEDC_TIMER_0  
#define LEDC_HS_MODE          LEDC_HIGH_SPEED_MODE  
#define LEDC_HS_CH0_GPIO      (18)  
#define LEDC_HS_CH0_CHANNEL   LEDC_CHANNEL_0  
#define LEDC_HS_CH1_GPIO      (19)  
#define LEDC_HS_CH1_CHANNEL   LEDC_CHANNEL_1  
#endif  
#define LEDC_LS_TIMER          LEDC_TIMER_1  
#define LEDC_LS_MODE          LEDC_LOW_SPEED_MODE  
#ifdef CONFIG_IDF_TARGET_ESP32S2  
#define LEDC_LS_CH0_GPIO      (18)  
#define LEDC_LS_CH0_CHANNEL   LEDC_CHANNEL_0  
#define LEDC_LS_CH1_GPIO      (19)  
#define LEDC_LS_CH1_CHANNEL   LEDC_CHANNEL_1  
#endif  
#define LEDC_LS_CH2_GPIO      (4)  
#define LEDC_LS_CH2_CHANNEL   LEDC_CHANNEL_2  
#define LEDC_LS_CH3_GPIO      (5)
```

```

#define LEDC_LS_CH3_CHANNEL    LEDC_CHANNEL_3

#define LEDC_TEST_CH_NUM      (4)
#define LEDC_TEST_DUTY        (4000)
#define LEDC_TEST_FADE_TIME    (3000)

```

En la función del void se prepara la estructura del controlador del LED.

```

int ch;

ledc_timer_config_t ledc_timer = {
    .duty_resolution = LEDC_TIMER_13_BIT, // resolución PWM
    .freq_hz = 5000, // frecuencia de señal PWM
    .speed_mode = LEDC_LS_MODE, // modo del temporizador
    .timer_num = LEDC_LS_TIMER, // índice del temporizador
    .clk_cfg = LEDC_AUTO_CLK, // selección automática del reloj
};

ledc_timer_config(&ledc_timer);

```

Se establece la configuración del temporizador para los canales de una baja velocidad.

```

#ifdef CONFIG_IDF_TARGET_ESP32
    ledc_timer.speed_mode = LEDC_HS_MODE;
    ledc_timer.timer_num = LEDC_HS_TIMER;
    ledc_timer_config(&ledc_timer);
#endif

ledc_channel_config_t ledc_channel[LEDC_TEST_CH_NUM] = {
#ifdef CONFIG_IDF_TARGET_ESP32
    {
        .channel = LEDC_HS_CH0_CHANNEL,
        .duty = 0,
        .gpio_num = LEDC_HS_CH0_GPIO,
        .speed_mode = LEDC_HS_MODE,
        .hpoint = 0,
        .timer_sel = LEDC_HS_TIMER
    },
    {
        .channel = LEDC_HS_CH1_CHANNEL,
        .duty = 0,
        .gpio_num = LEDC_HS_CH1_GPIO,
        .speed_mode = LEDC_HS_MODE,
        .hpoint = 0,
        .timer_sel = LEDC_HS_TIMER
    },
#endif
#ifdef CONFIG_IDF_TARGET_ESP32S2
    {
        .channel = LEDC_HS_CH2_CHANNEL,
        .duty = 0,
        .gpio_num = LEDC_HS_CH2_GPIO,
        .speed_mode = LEDC_HS_MODE,
        .hpoint = 0,
        .timer_sel = LEDC_HS_TIMER
    },
#endif
};

```



```

    {
        .channel    = LEDC_LS_CH2_CHANNEL,
        .duty       = 0,
        .gpio_num   = LEDC_LS_CH2_GPIO,
        .speed_mode = LEDC_LS_MODE,
        .hpoint     = 0,
        .timer_sel  = LEDC_LS_TIMER
    },
    {
        .channel    = LEDC_LS_CH3_CHANNEL,
        .duty       = 0,
        .gpio_num   = LEDC_LS_CH3_GPIO,
        .speed_mode = LEDC_LS_MODE,
        .hpoint     = 0,
        .timer_sel  = LEDC_LS_TIMER
    },
};

```

El controlador LED.

```

for (ch = 0; ch < LEDC_TEST_CH_NUM; ch++) {
    ledc_channel_config(&ledc_channel[ch]);
}

```

Se inicializa el servicio.

```

ledc_fade_func_install(0);

while (1) {
    printf("1. LEDC fade up to duty = %d\n", LEDC_TEST_DUTY);
    for (ch = 0; ch < LEDC_TEST_CH_NUM; ch++) {
        ledc_set_fade_with_time(ledc_channel[ch].speed_mode,
                                ledc_channel[ch].channel, LEDC_TEST_DUTY,
LEDC_TEST_FADE_TIME);
        ledc_fade_start(ledc_channel[ch].speed_mode,
                        ledc_channel[ch].channel, LEDC_FADE_NO_WAIT);
    }
    vTaskDelay(LEDC_TEST_FADE_TIME / portTICK_PERIOD_MS);

    printf("2. LEDC fade down to duty = 0\n");
    for (ch = 0; ch < LEDC_TEST_CH_NUM; ch++) {
        ledc_set_fade_with_time(ledc_channel[ch].speed_mode,
                                ledc_channel[ch].channel, 0, LEDC_TEST_FADE_TIME);
        ledc_fade_start(ledc_channel[ch].speed_mode,
                        ledc_channel[ch].channel, LEDC_FADE_NO_WAIT);
    }
    vTaskDelay(LEDC_TEST_FADE_TIME / portTICK_PERIOD_MS);
}

```

```

printf("3. LEDC set duty = %d without fade\n", LEDC_TEST_DUTY);
for (ch = 0; ch < LEDC_TEST_CH_NUM; ch++) {
    ledc_set_duty(ledc_channel[ch].speed_mode, ledc_channel[ch].channel,
LEDC_TEST_DUTY);
    ledc_update_duty(ledc_channel[ch].speed_mode,
ledc_channel[ch].channel);
}
vTaskDelay(1000 / portTICK_PERIOD_MS);

printf("4. LEDC set duty = 0 without fade\n");
for (ch = 0; ch < LEDC_TEST_CH_NUM; ch++) {
    ledc_set_duty(ledc_channel[ch].speed_mode, ledc_channel[ch].channel,
0);
    ledc_update_duty(ledc_channel[ch].speed_mode,
ledc_channel[ch].channel);
}
vTaskDelay(1000 / portTICK_PERIOD_MS);
}

```

Circuito del LEDC started

El ensamblaje del circuito se realiza sobre la base del protoboard, en la Figura 25 se observa que la implementación es simple para ejemplificar que el sistema no tiene un control adecuado del led ni puede ser manipulado por el usuario.

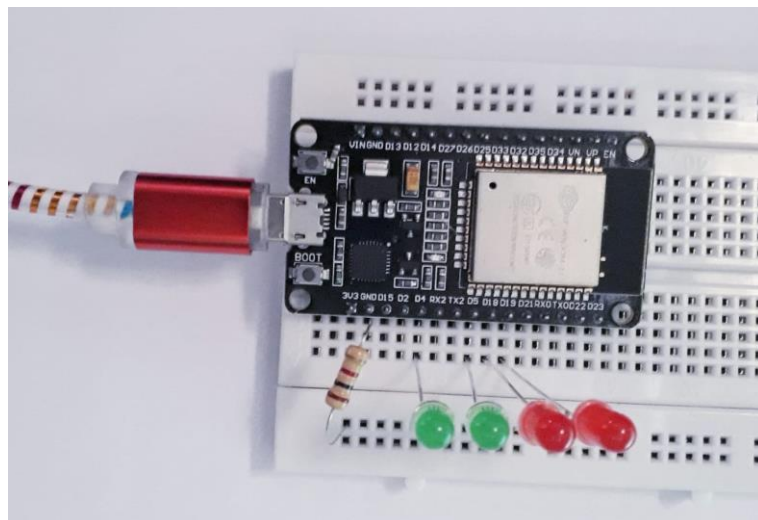


Figura 25: Conexión del circuito LEDC started

Fuente: Elaboración del autor

```
es[0;32mI (288) spi_flash: detected chip: generic
es[0;32mI (291) spi_flash: flash io: dio
es[0;32mI (296) cpu_start: Starting scheduler on PRO CPU.
es[0;32mI (0) cpu_start: Starting scheduler on APP CPU.
1. LEDC fade up to duty = 4000
2. LEDC fade down to duty = 0
3. LEDC set duty = 4000 without fade
4. LEDC set duty = 0 without fade
1. LEDC fade up to duty = 4000
2. LEDC fade down to duty = 0
3. LEDC set duty = 4000 without fade
4. LEDC set duty = 0 without fade
```

Figura 26: Datos recibidos de la aplicación

Fuente: Elaboración del autor

Se reciben datos (Figura 26) para el encendido con o sin desvanecimiento de la luz estableciendo un ciclo automático, en la Figura 27 presenta el circuito con sus respectivos controles de luz.

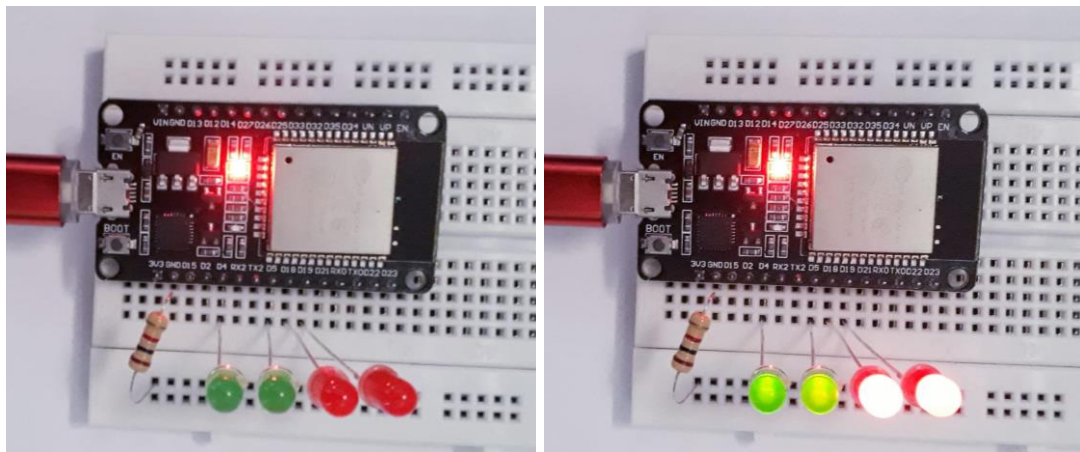


Figura 27: Ejecución del circuito

Fuente: Elaboración del autor

2.5.4 Instalación de la Red BLE Mesh

Al desarrollar una red BLE Mesh se debe tener en cuenta para cuantos dispositivos requiere la instalación total para no quedar corto en el rango de distancia de los nodos. El archivo principal main.c tiene como fin la inicialización del nodo como la pila de protocolo Bluetooth Low Energy como se puede observar en la función app_main.

```
void app_main(void)
{
    esp_err_t err;

    ESP_LOGI(TAG, "Initializing...");
```

```

board_init();

err = nvs_flash_init();
if (err == ESP_ERR_NVS_NO_FREE_PAGES) {
    ESP_ERROR_CHECK(nvs_flash_erase());
    err = nvs_flash_init();
}
ESP_ERROR_CHECK(err);

err = bluetooth_init();
if (err) {
    ESP_LOGE(TAG, "esp32_bluetooth_init failed (err %d)", err);
    return;
}

ble_mesh_get_dev_uuid(dev_uuid);

/* Initialize the Bluetooth Mesh Subsystem */
err = ble_mesh_init();
if (err) {
    ESP_LOGE(TAG, "Bluetooth mesh init failed (err %d)", err);
}
}

```

Esta función `ble_mesh_init()` es la razón por la cual habilita el Bluetooth Mesh que registra la devolución de la llamada del aprovisionamiento además que activa las funciones de escaneo y publicidad para que sea visible.

```

static esp_err_t ble_mesh_init(void)
{
    esp_err_t err = ESP_OK;

    esp_ble_mesh_register_prov_callback(example_ble_mesh_provisioning_cb);
    esp_ble_mesh_register_config_server_callback(example_ble_mesh_config_server_c
b);
    esp_ble_mesh_register_generic_server_callback(example_ble_mesh_generic_server
_cb);

    err = esp_ble_mesh_init(&provision, &composition);
    if (err != ESP_OK) {
        ESP_LOGE(TAG, "Failed to initialize mesh stack (err %d)", err);
        return err;
    }

    err = esp_ble_mesh_node_prov_enable(ESP_BLE_MESH_PROV_ADV |
ESP_BLE_MESH_PROV_GATT);

```

```

if (err != ESP_OK) {
    ESP_LOGE(TAG, "Failed to enable mesh node (err %d)", err);
    return err;
}

ESP_LOGI(TAG, "BLE Mesh Node initialized");

board_led_operation(LED_G, LED_ON);

return err;
-}

```

Se declaran los elementos necesarios para el funcionamiento del sistema de control, depende mucho del tipo de control que se desea realizar.

```

static esp_ble_mesh_elem_t elements[] = {
    ESP_BLE_MESH_ELEMENT(0, root_models, ESP_BLE_MESH_MODEL_NONE),
    ESP_BLE_MESH_ELEMENT(0, extend_model_0, ESP_BLE_MESH_MODEL_NONE),
    ESP_BLE_MESH_ELEMENT(0, extend_model_1, ESP_BLE_MESH_MODEL_NONE),
};

```

Las matrices de modelos se utilizan para indicar el número de las estructuras de los modelos existentes.

```

static esp_ble_mesh_model_t root_models[] = {
    ESP_BLE_MESH_MODEL_CFG_SRV(&config_server),
    ESP_BLE_MESH_MODEL_GEN_ONOFF_SRV(&onoff_pub_0, &onoff_server_0),
};

static esp_ble_mesh_model_t extend_model_0[] = {
    ESP_BLE_MESH_MODEL_GEN_ONOFF_SRV(&onoff_pub_1, &onoff_server_1),
};

static esp_ble_mesh_model_t extend_model_1[] = {
    ESP_BLE_MESH_MODEL_GEN_ONOFF_SRV(&onoff_pub_2, &onoff_server_2),
};

```

En el archivo board.h se define el tipo de placa a implementar con los pines que se desea utilizar.

```

// se define el tipo de placa
#define CONFIG_BLE_MESH_ESP_WROOM_32
//#define CONFIG_BLE_MESH_ESP_WROOM_32D

```

```

#if defined(CONFIG_BLE_MESH_ESP_WROOM_32)
#define LED_R GPIO_NUM_15
#define LED_G GPIO_NUM_4
#define LED_B GPIO_NUM_17

#elif defined(CONFIG_BLE_MESH_ESP_WROOM_32D)
#define LED_R GPIO_NUM_15
#define LED_G GPIO_NUM_2
#define LED_B GPIO_NUM_4
#endif

#define LED_ON 1
#define LED_OFF 0

```

El archivo board.c se inicia con el proceso para el control del sistema de iluminación.

```

static void board_led_init(void)
{
    for (int i = 0; i < 3; i++) {
        gpio_pad_select_gpio(led_state[i].pin);
        gpio_set_direction(led_state[i].pin, GPIO_MODE_OUTPUT);
        gpio_set_level(led_state[i].pin, LED_OFF);
        led_state[i].previous = LED_OFF;
    }
}

void board_led_operation(uint8_t pin, uint8_t onoff)
{
    for (int i = 0; i < 3; i++) {

        if (led_state[i].pin != pin) {
            continue;
        }
        if (onoff == led_state[i].previous) {
            ESP_LOGW(TAG, "led %s is already %s",
                    led_state[i].name, (onoff ? "on" : "off"));
            return;
        }
        gpio_set_level(pin, onoff);
        led_state[i].previous = onoff;
        return;
    }

    ESP_LOGE(TAG, "LED is not found!");
}

```

Cuando se ejecuta el código y se carga al módulo ESP32 se presenta los datos de inicio como se puede observar en la Figura 28 que índice que el nodo está inicializado.

```

[0;32mI (467) spi_flash: detected chip: generic[0m
[0;32mI (471) spi_flash: flash io: dio[0m
[0;32mI (476) cpu_start: Starting scheduler on PRO CPU.[0m
[0;32mI (0) cpu_start: Starting scheduler on APP CPU.[0m
[0;32mI (485) EXAMPLE: Initializing...[0m
[0;32mI (505) BTDM_INIT: BT controller compile version [bfbbe1e][0m
[0;32mI (505) system_api: Base MAC address is not set[0m
[0;32mI (505) system_api: read default base MAC address from EFUSE[0m
[0;32mI (515) phy_init: phy_version 4670,719f9f6, Feb 18 2021,17:07:07:[0m
[0;32mI (855) EXAMPLE_INIT: BLE Host Task Started[0m
[0;32mI (1465) EXAMPLE: ESP_BLE_MESH_PROV_REGISTER_COMP_EVT, err_code 0[0m
GAP procedure initiated: discovery; own_addr_type=0 filter_policy=0 passive=1 limited=0 filter_duplicates=0 duration=forever
[0;32mI (1465) EXAMPLE: BLE Mesh Node initialized[0m
GAP procedure initiated: advertise; disc_mode=0 adv_channel_map=0 own_addr_type=0 adv_filter_policy=0 adv_itvl_min=32 adv_itvl_max=32
[0;32mI (1485) EXAMPLE: ESP_BLE_MESH_NODE_PROV_ENABLE_COMP_EVT, err_code 0[0m
GAP procedure initiated: stop advertising.
GAP procedure initiated: advertise; disc_mode=2 adv_channel_map=0 own_addr_type=0 adv_filter_policy=0 adv_itvl_min=32 adv_itvl_max=32

```

Figura 28: Datos recibidos del nodo

Fuente: Elaboración del autor

2.5.4.1 Aprovisionamiento con la App nRF Mesh

El aprovisionador es el encargado de realizar la conexión con los nodos y se configura para que pertenezca a la misma red BLE Mesh. La aplicación nRF Mesh cumple esa función de presentar de manera visual cómo se aprovisionan los módulos ESP32. En la Figura 29 se muestra la interfaz de inicio de la App nRF Mesh que es intuitiva para su fácil utilización.

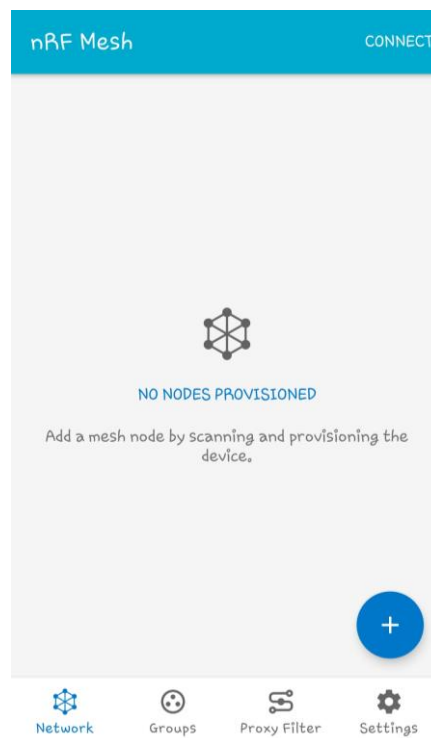


Figura 29: Interfaz de inicio App nRF Mesh

Fuente: Elaboración del autor

Cuando el módulo ESP32 está cargado con el programa de la red BLE Mesh se conecta a una fuente de energía y se logra realizar una conexión con el nRF Mesh como se presenta en la Figura 30 el ESP32 disponible. Como es una red escalable cada vez que un nodo esté visible se realiza esta misma configuración.

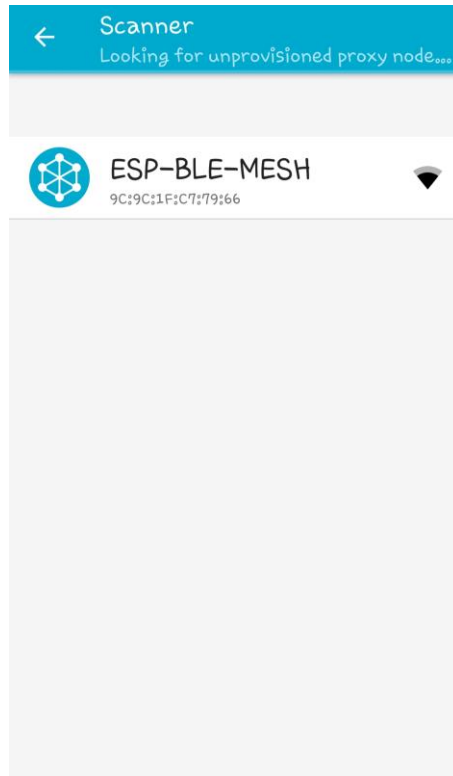


Figura 30: Escaneo del módulo para uno o más módulos

Fuente: Elaboración del autor

En la Figura 31 el nodo ESP se logra identificar y luego en “provisión” para empezar el proceso de aprovisionamiento.

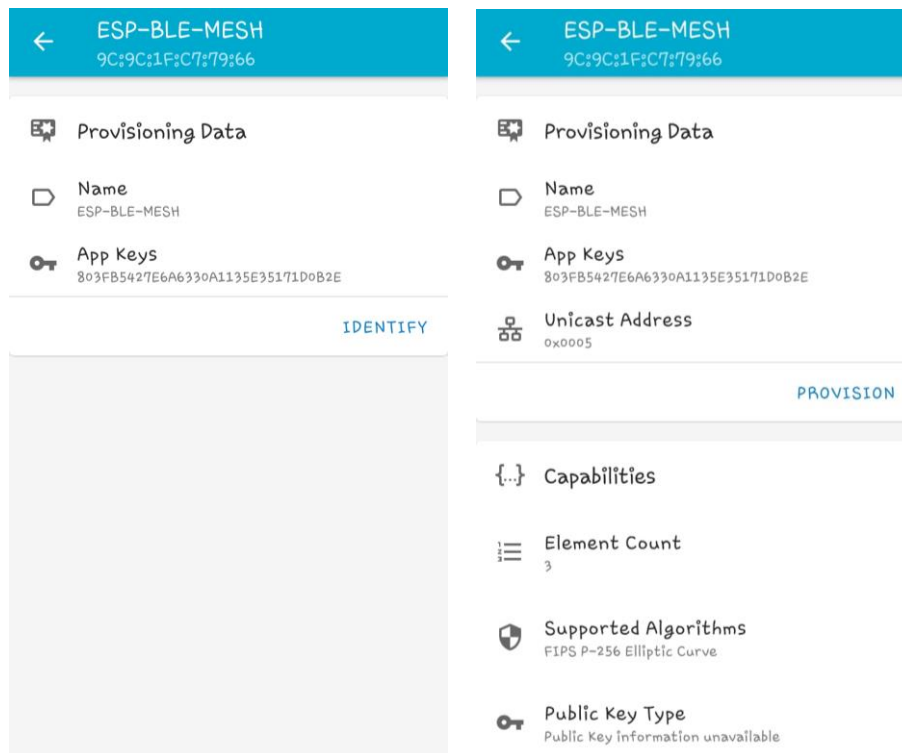


Figura 31: Identify and provision

Fuente: Elaboración del autor

El proceso de aprovisionamiento es la configuración de la red del nodo como se presenta en la Figura 32 realizando todos los procesos y completando el aprovisionamiento.

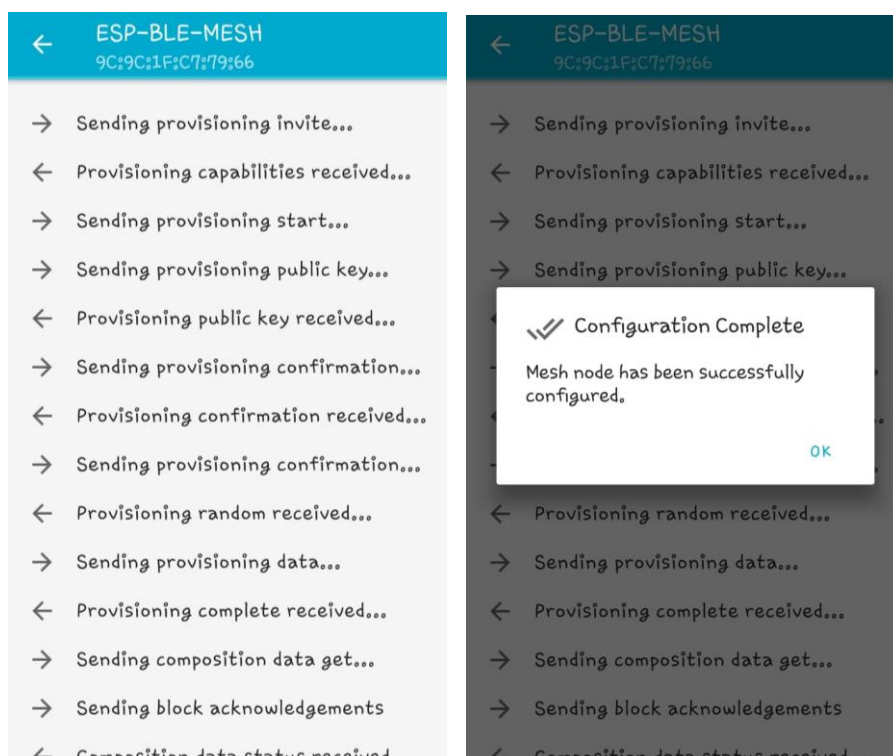


Figura 32: Aprovisionamiento del nodo

Fuente: Elaboración del autor

Cuando el proceso terminó en la Figura 33 se presenta un nodo aprovisionado en tal caso será una lista de todos los nodos disponibles.

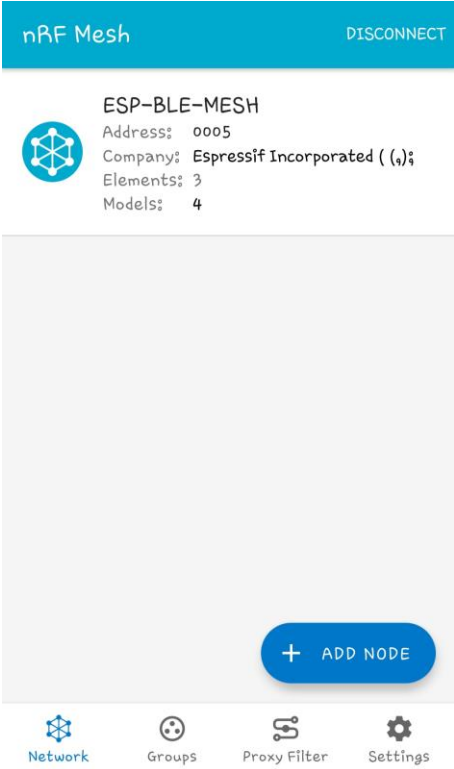


Figura 33: Nodo aprovisionado

Fuente: Elaboración del autor

La configuración del nodo es la estructura de los elementos establecidos en el código, cada elemento disponible como se observa en la Figura 34 se realiza la configuración individual.

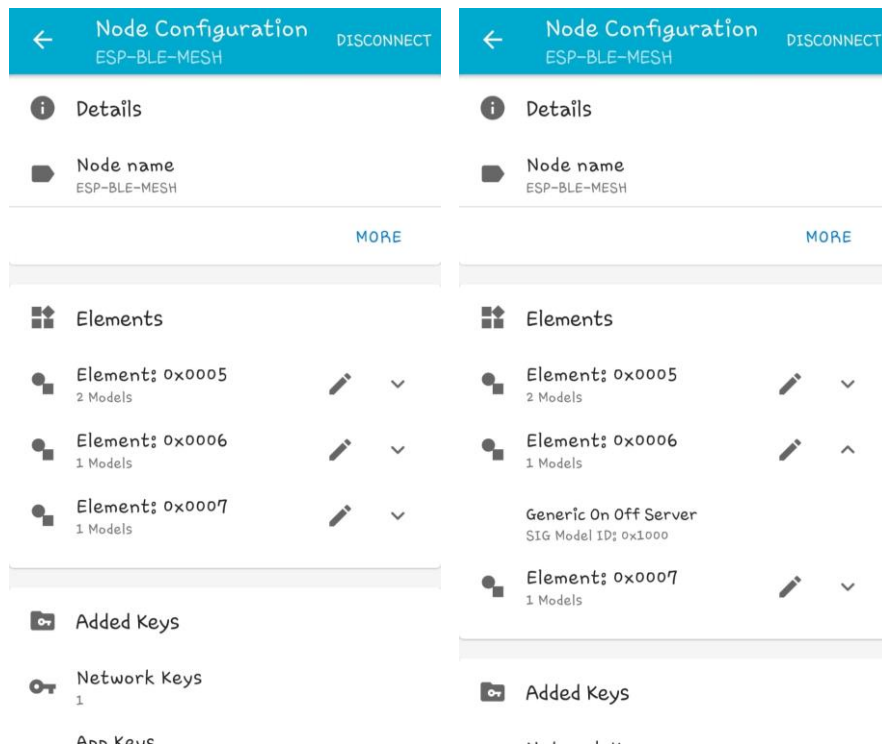


Figura 34: Elementos del nodo

Fuente: Elaboración del autor

El elemento del nodo se establece una clave de enlace (BIND KEY) que es “Application Key” 1 el mismo realiza una operación interna para generar el control del sistema en la Figura 35 presenta el proceso del elemento.

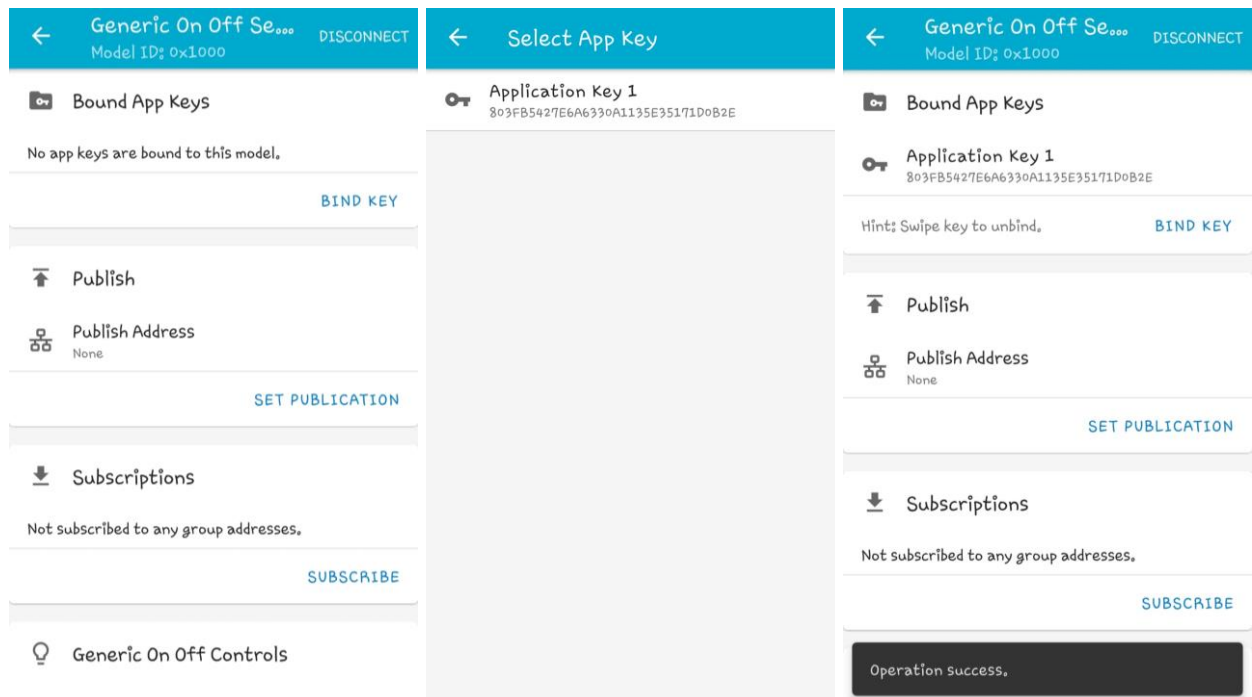


Figura 35: Clave de la aplicación.

Fuente: Elaboración del autor

El proceso de aprovisionar el módulo ESP32 finaliza con el control del sistema genérico. La Figura 36 muestra el estado listo de la función encendido o apagado.

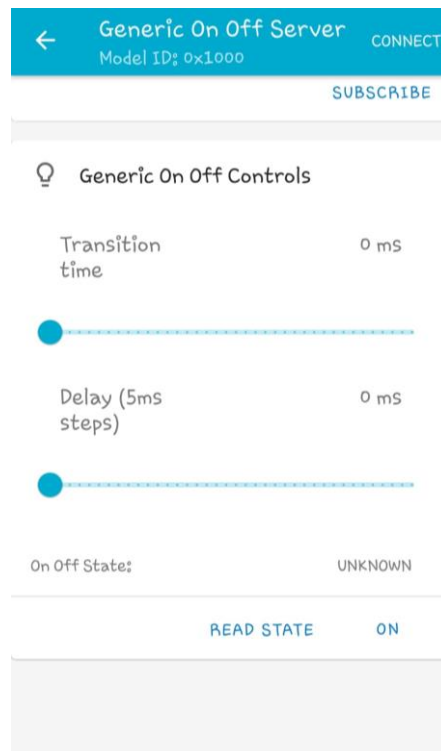


Figura 36: Control de la aplicación

Fuente: Elaboración del autor

En la pantalla del VSC se presentan los detalles de los procesos de la aplicación nRF Mesh (Figura 37). La función *ESP_BLE_MESH_NODE_PROV_LINK_OPEN_EVENT* establece que el módulo ESP32 y la App están aprovisionados por el cual cuando se configura el elemento indica la dirección unicast 0x0002 y la función *ESP_BLE_MESH_MODEL_OP_APP_KEY_ADD* se monitoriza la clave de la aplicación.

```
es[0;32mI (13516) NimBLE: mtu update event; conn_handle=0 cid=4 mtu=23
es[0m
es[0;32mI (13616) NimBLE: subscribe event; conn_handle=0 attr_handle=20 reason=1 prevn
es[0m
es[0;32mI (13616) EXAMPLE: ESP_BLE_MESH_NODE_PROV_LINK_OPEN_EVT, bearer PB-GATTes[0m
es[0;32mI (15416) NimBLE: GATT procedure initiated: notify; es[0m
es[0;32mI (15416) NimBLE: att_handle=20
--T.0m
es[0;32mI (17876) EXAMPLE: ESP_BLE_MESH_NODE_PROV_LINK_CLOSE_EVT, bearer PB-GATT
es[0;32mI (17886) EXAMPLE: ESP_BLE_MESH_NODE_PROV_COMPLETE_EVTes[0m
es[0;32mI (17896) EXAMPLE: net_idx: 0x0000, addr: 0x0002es[0m
es[0;32mI (17896) EXAMPLE: flags: 0x00, iv_index: 0x00000000es[0m
es[0;32mI (29246) EXAMPLE: ESP_BLE_MESH_MODEL_OP_APP_KEY_ADDes[0m
es[0;32mI (29256) EXAMPLE: net_idx 0x0000, app_idx 0x0000es[0m
es[0;32mI (29256) AppKey: 80 3f b5 42 7e 6a 63 30 a1 13 5e 35 17 1d 0b 2e es[0m
```

Figura 37: Información recibida de la App nRF Mesh

Fuente: Elaboración del autor

2.5.4.2 Asignación de grupos a los nodos

La aplicación nRF Mesh tiene más funcionalidades para la red BLE Mesh como: Network, Groups, Proxy Filter que permiten al usuario estructurar la red con las necesidades que posea. La creación de grupos permite agrupar a los nodos o los elementos RGB por sus colores por lo mismo se presenta la interfaz de inicio de grupos en la App nRF Mesh (Figura 38)

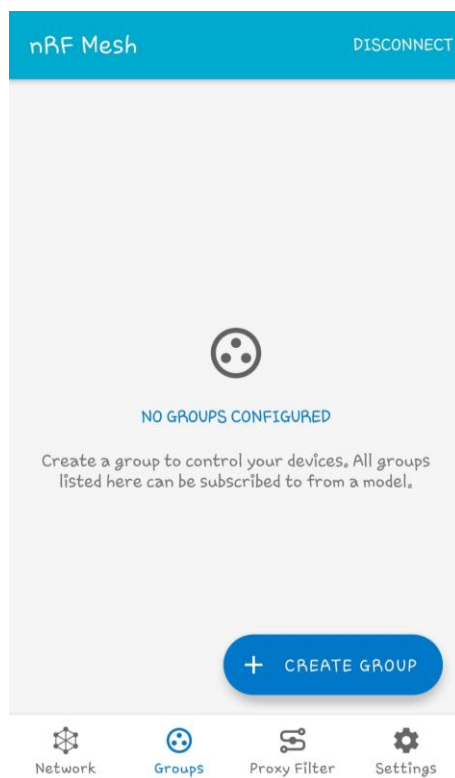


Figura 38: Interfaz de grupo nRF Mesh

Fuente: Elaboración del autor

Se establece un nombre del grupo distintivo con su funcionalidad y su ubicación como se describe en la Figura 39.

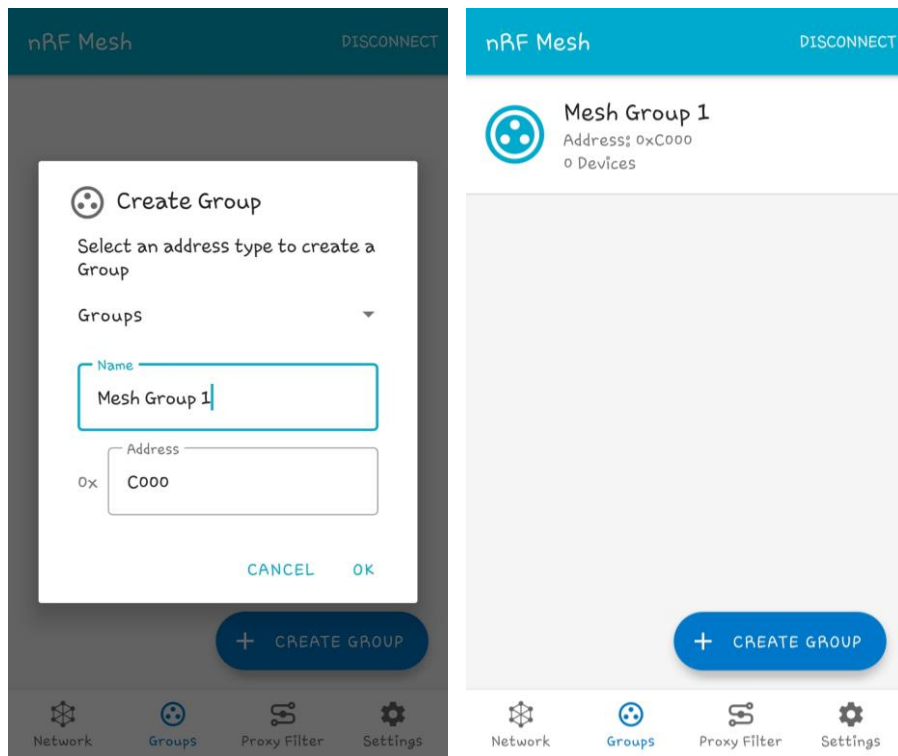


Figura 39: Crear grupos Mesh

Fuente: Elaboración del autor

En la red (network) de la App nRF Mesh se establece el vínculo con el grupo creado (Mesh group 1) como se presenta en la Figura 40.

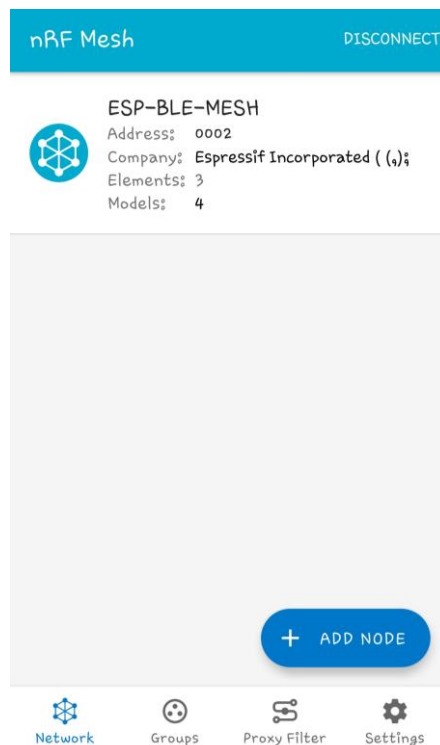


Figura 40: Nodo ESP32

Fuente: Elaboración del autor

En la Figura 41 para la asignación se debe tener en cuenta que elementos vamos agregar al grupo, cada elemento puede pertenecer a un grupo distinto y la agrupación depende de las funciones que se requiera ejecutar. Para el elemento 0x0003 se le suscribe al Mesh group 1 como se presenta en la Figura 42.

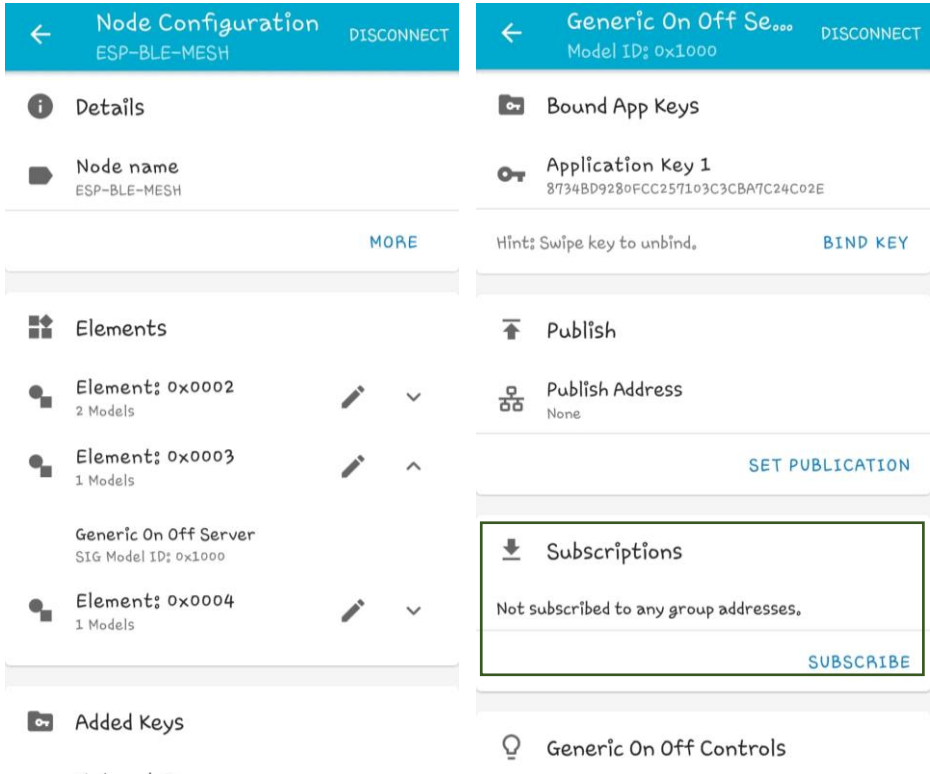


Figura 41: Asignar el elemento al grupo

Fuente: Elaboración del autor

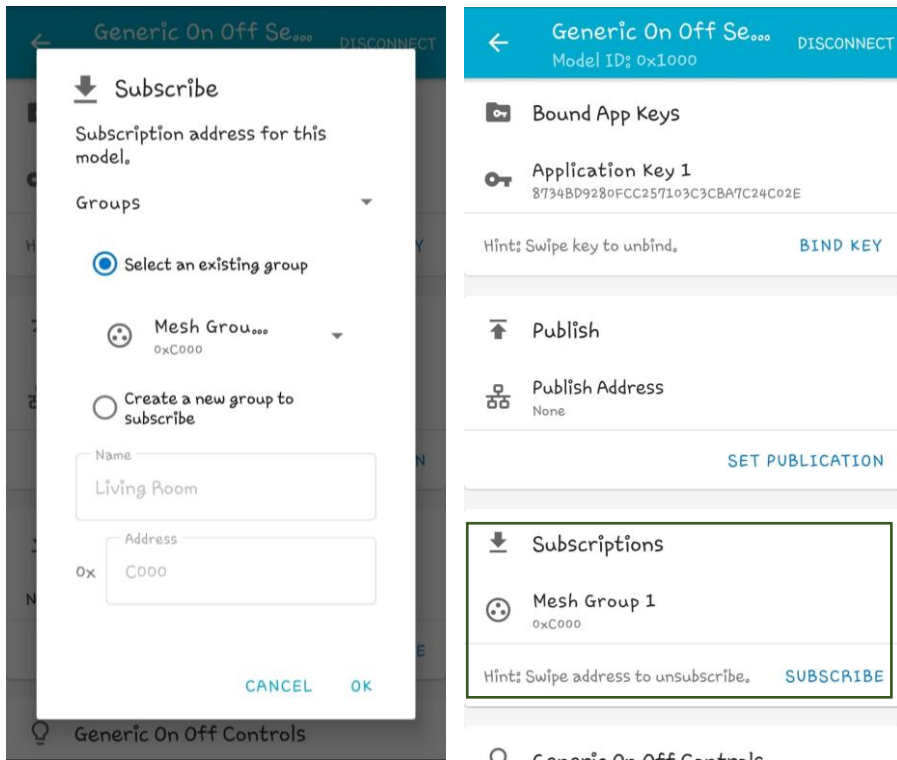


Figura 42: Elemento suscrito al grupo

Fuente: Elaboración del autor

El Mesh Group 1 tiene un elemento asignado (Figura 43) asumiendo el rol de la función que tiene el elemento 0x003.

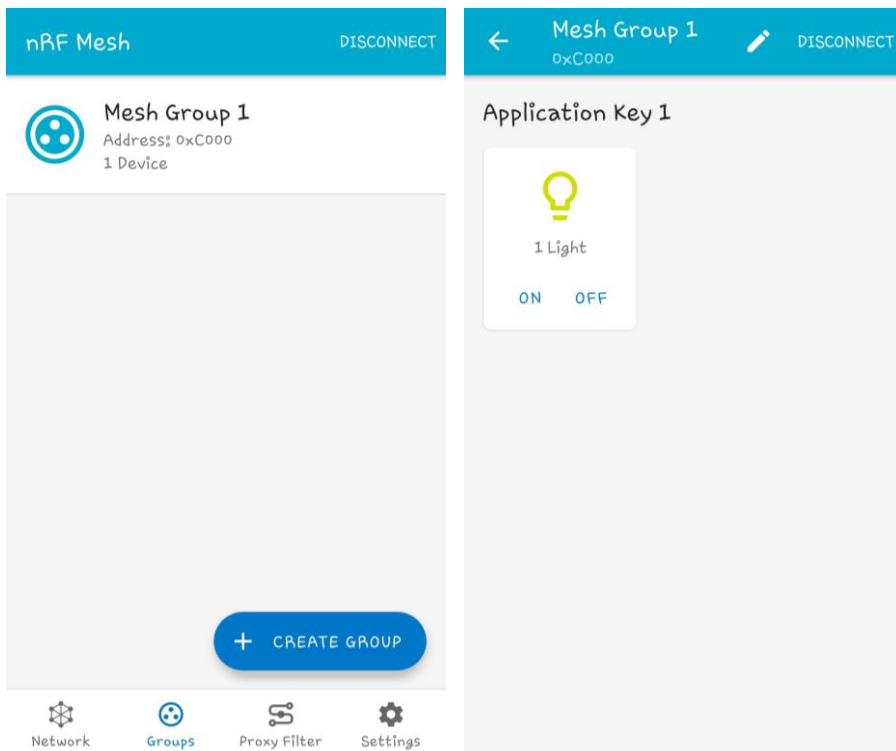


Figura 43: Grupo 1 con el elemento asignado

Fuente: Elaboración del autor

3 CAPÍTULO III: EVALUACIÓN DEL PROTOTIPO

3.1 Plan de evaluación

La evaluación es un proceso de recolección de los datos y analizarlos para tomar decisiones futuras. La propuesta tecnológica se monitorea en su fase de ejecución para medir la funcionalidad y garantizar su satisfacción. En el plan se desarrollará métodos de evaluación como: funcionalidad, usabilidad y seguridad en la red BLE Mesh.

La funcionalidad de la red con los módulos ESP32 integrados son las características funcionales que requiere para lograr su objetivo como a qué distancia se puede conectar o que tipos de servicios provee. La calidad del dispositivo es fundamental para considerar la usabilidad de la App nRF Mesh y qué seguridad ofrece para que sea considerado en su utilización.

3.2 Resultados de evaluación

3.2.1 Pruebas de distanciamiento de los nodos

Las pruebas que se realizaron son para medir el rango de distancia que emplea el módulo ESP32 para establecer una conexión de bluetooth con la aplicación nRF Mesh y que números de nodos se debe implementar para obtener una red estable. A continuación, se presentan en las Figuras 44 y 45 el circuito utilizado para las diferentes pruebas.

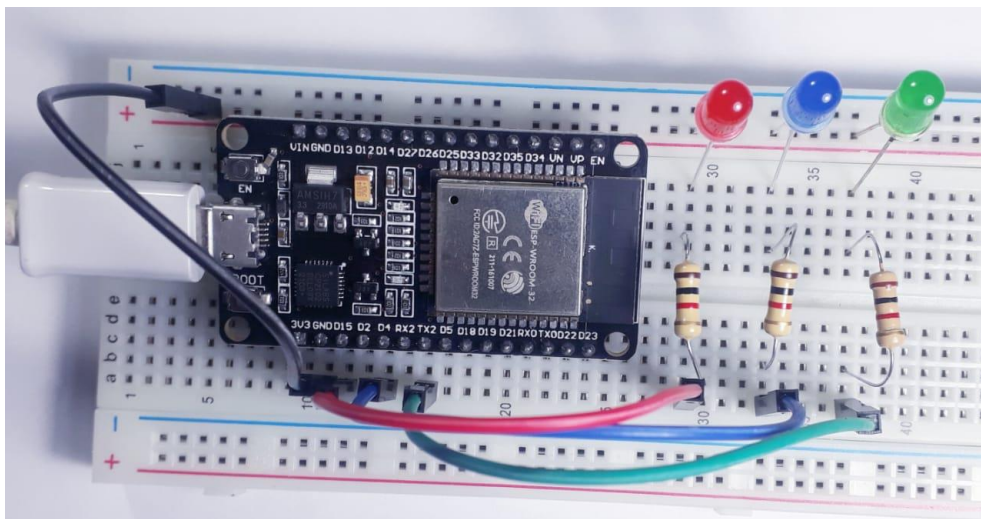


Figura 44: Circuito ESP-WROOM-32

Fuente: Elaboración del autor

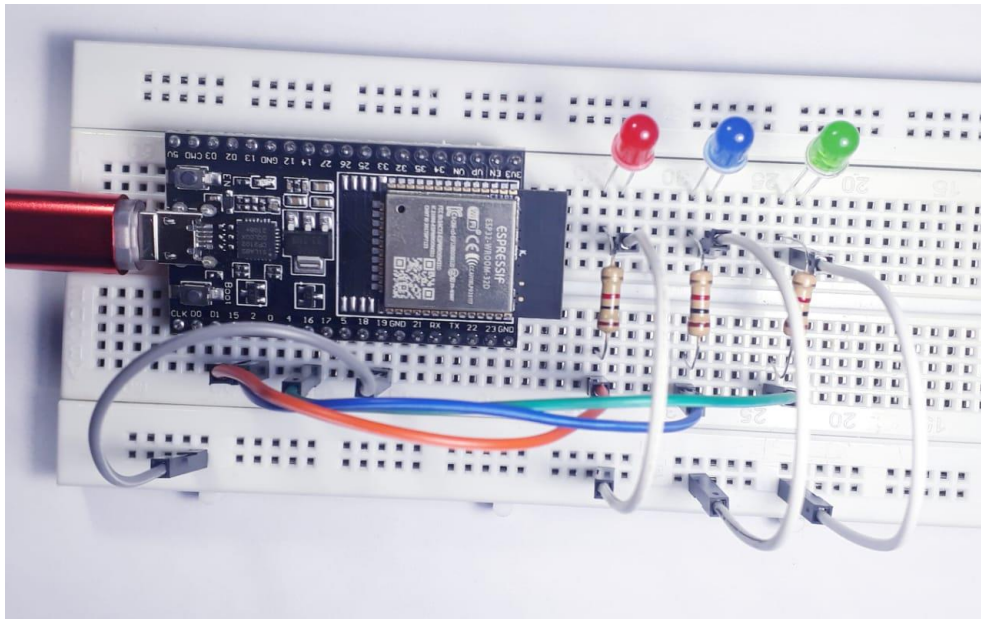


Figura 45: Circuito ESP-WROOM-32D

Fuente: Elaboración del autor

PRUEBA 1

Para que exista una conexión del módulo con la aplicación, estos tipos de dispositivos BLE aproximadamente el alcance del rango es menor a 20 metros (Figura 46) y como se puede observar en la Figura 47 el módulo está disponible para el aprovisionamiento.

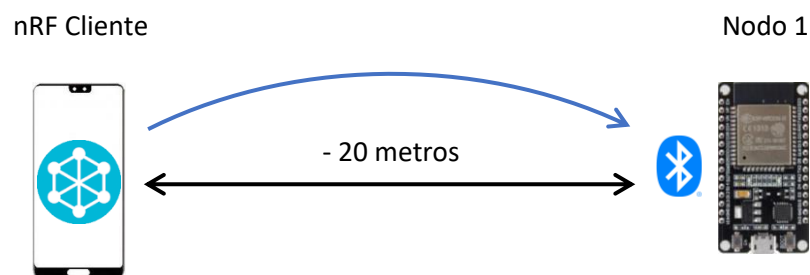


Figura 46: nRF Mes conectado a un nodo

Fuente: Elaboración del autor



Figura 47: ESP32 disponible

Fuente: Elaboración del autor

La conexión del módulo con el móvil ya vinculado se procede al aprovisionamiento del nodo como se detalló anteriormente (*Aprovisionamiento con la App nRF Mesh*) y así lograr el control del sistema de iluminación RGB, en la Figura 48 se accede al primer elemento del módulo que controla al led rojo y la Figura 49 con el segundo elemento que controla el led azul con el mismo funcionamiento el led verde.

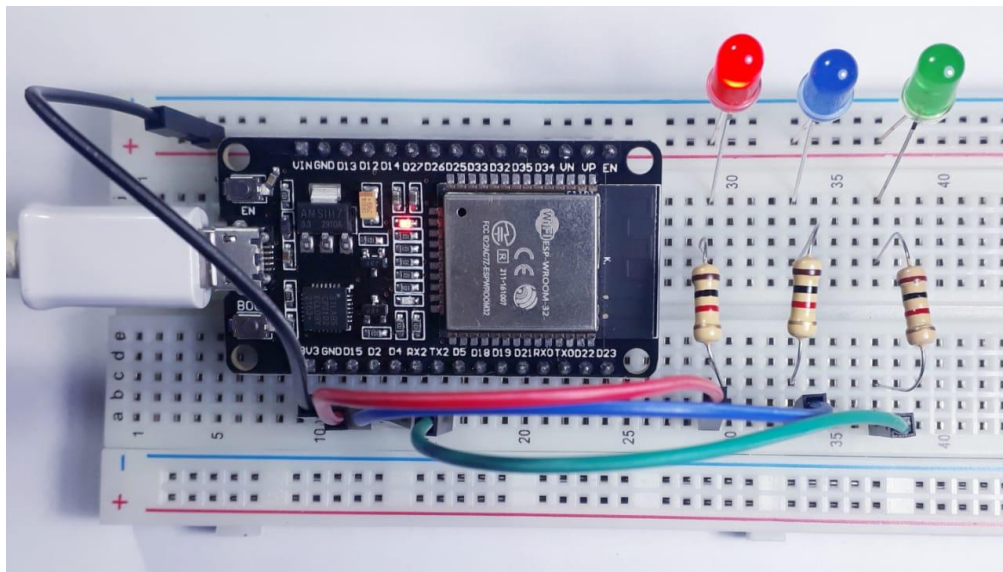


Figura 48: Control de iluminación RGB - rojo

Fuente: Elaboración del autor

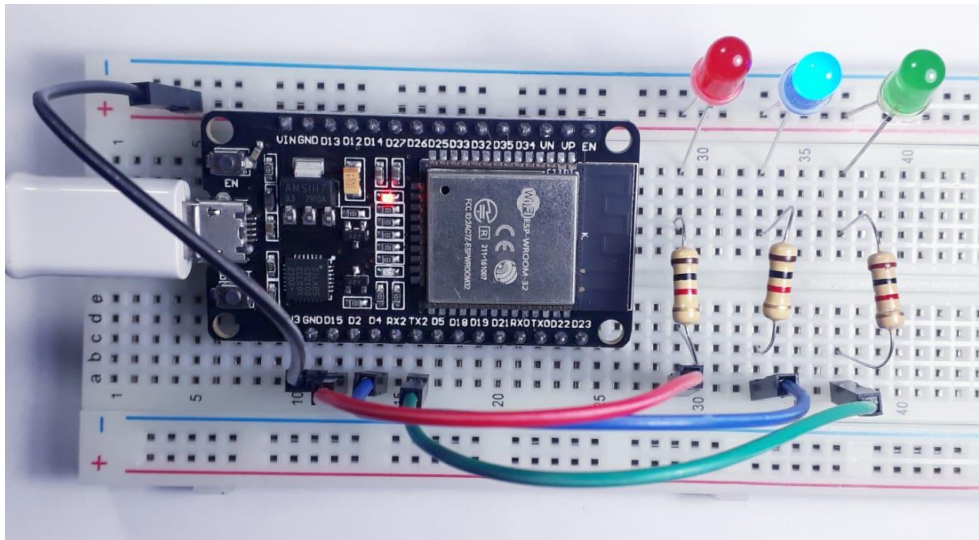


Figura 49: Control de iluminación RGB - azul

Fuente: Elaboración del autor

PRUEBA 2

Los dispositivos pertenecen a la Clase 2 del Bluetooth, es decir, en esta prueba si el móvil lo movemos a más de 20 metros ya no está en el rango para conectarse (Figura 50) y simplemente el módulo ESP32 no aparecerá.

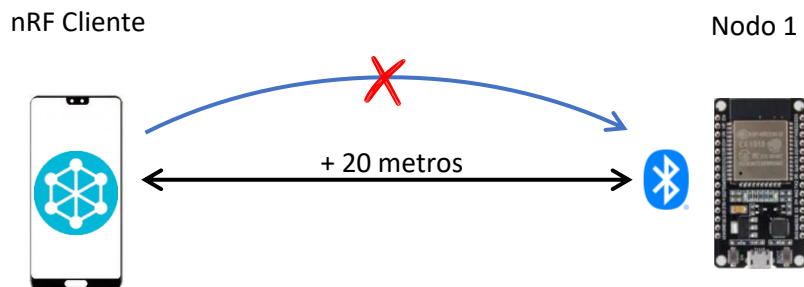


Figura 50: Limitación del rango Bluetooth

Fuente: Elaboración del autor

PRUEBA 3

Para solucionar el rango de 20 metros entre dispositivos se coloca un nodo 2 para que el rango sea mayor (Figura 51).

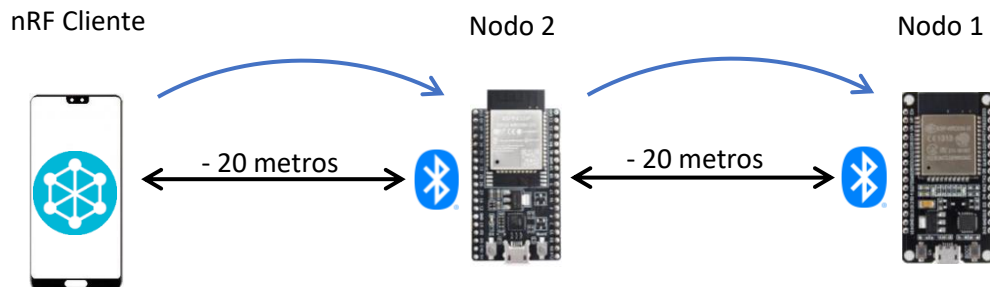


Figura 51: Conexión de 2 nodos

Fuente: Elaboración del autor

Se realizó la prueba con dos modelos de ESP32 y el nRF en donde se desea alcanzar la conexión con el nodo 1 que estaba a 25 metros aproximadamente, para ello se implementó un nodo 2 que sirve como tipo de transmisor al conectarse con el nodo 2 se puede obtener comunicación con el nodo 1. En la Figura 52 se realiza el aprovisionamiento del nodo 2.

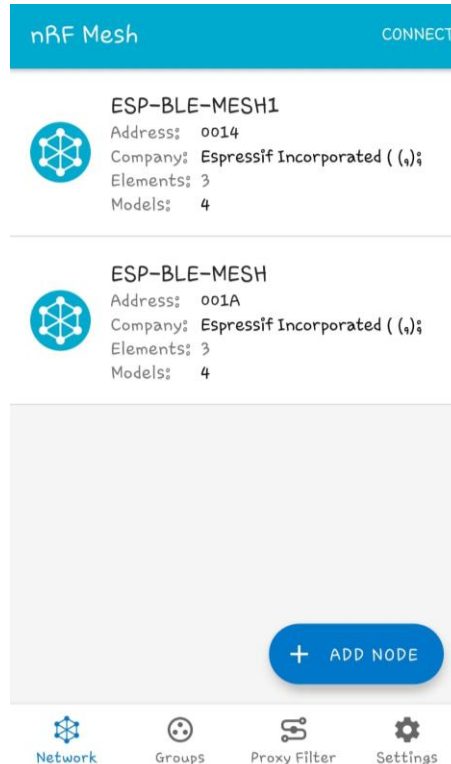


Figura 52: Aprovisionamiento del nodo nuevo

Fuente: Elaboración del autor

3.2.2 Pruebas en los Grupos de la malla

Los grupos nos ayudan en la estructuración de dispositivos según sea la función que se necesite, para esta prueba vamos a crear varios grupos como se puede observar en la Figura 53 describiendo el nombre del sistema de iluminación.

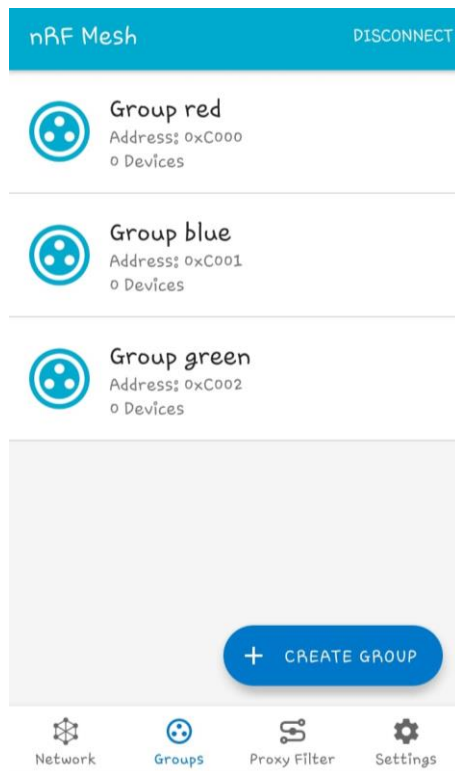


Figura 53: Grupos en la App

Fuente: Elaboración del autor

Cada nodo o módulo vinculado a la aplicación vamos a suscribir el elemento correspondiente al grupo creado (Figura 54)

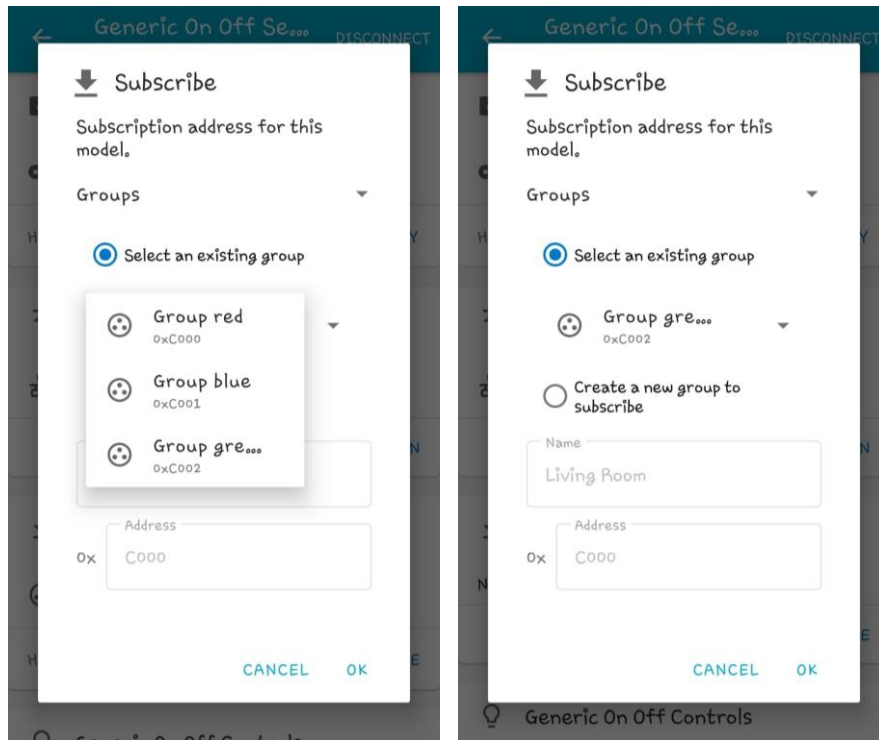


Figura 54: Suscribiendo elementos a los grupos

Fuente: Elaboración del autor

En la Figura 55 se suscribió cada elemento al grupo seleccionado y por cada grupo tiene una función de on/off para el sistema de iluminación.

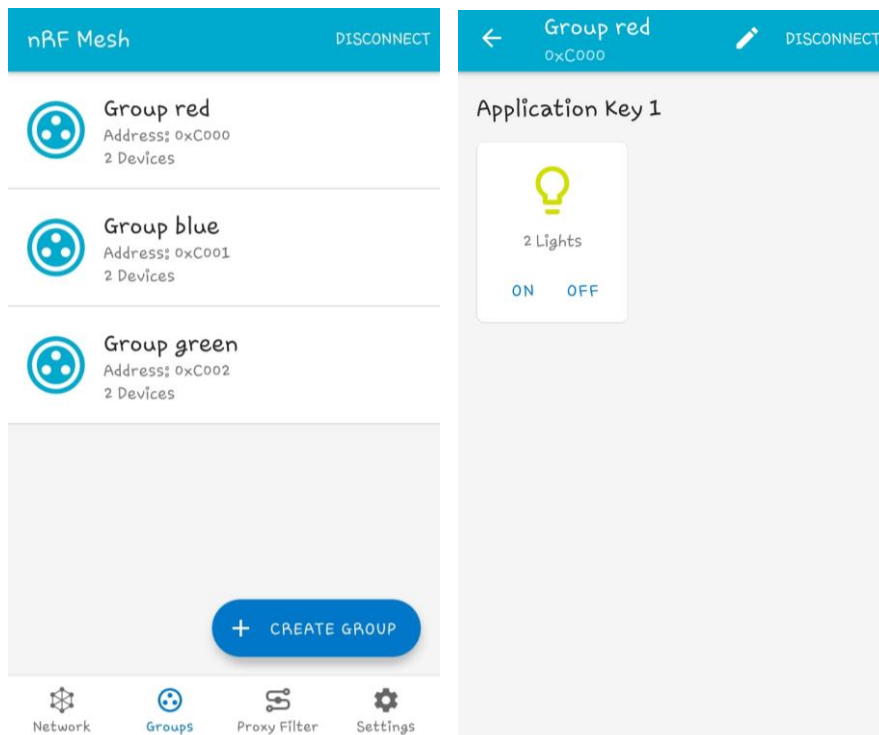


Figura 55: Grupos con sus elementos

Fuente: Elaboración del autor

En la parte izquierda de la Figura 56 se puede observar los dispositivos con el sistema apagado y en vez de abrir cada nodo para encender un led el grupo reduce el tiempo encendiendo todos los leds que pertenezcan al grupo como en la imagen derecha.

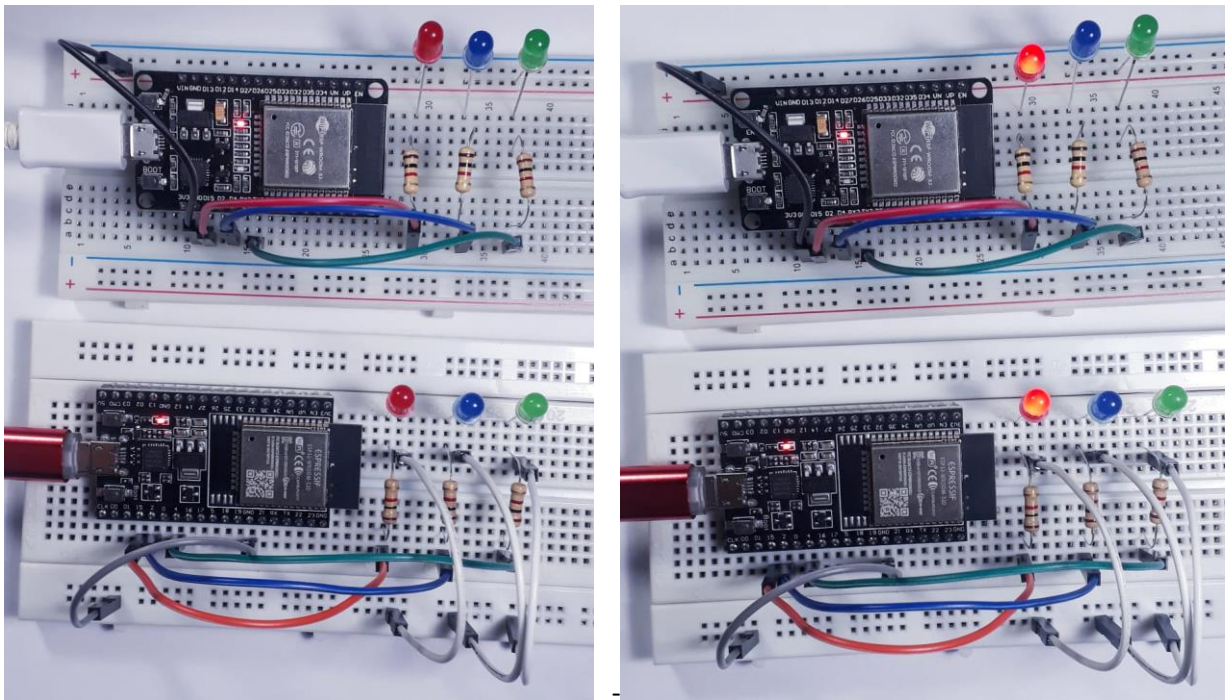


Figura 56: Led agrupados

Fuente: Elaboración del autor

La funcionalidad que ofrece la red BLE Mesh es importante cuando se tiene una estructura de muchos nodos y reduce el tiempo de espera para encender o apagar algún dispositivo y aún más cuando se emplea el sistema por grupos, este tipo de topología en malla es escalable que permite ir agregando nodos que transmiten en baja potencia.

3.2.3 Seguridad de la App nRF Mesh

Para garantizar la seguridad de la implementación de la aplicación en el móvil del usuario se comprueba los aspectos que posee esta App. Virus Total fue creada por la empresa española Hispasec Sistemas que permite un análisis en páginas web, archivos y aplicaciones en Android. Cuando se instala esta aplicación (Figura 57) y después de aceptar los términos que emplea automáticamente realiza un escaneo como se presenta en la Figura 58.



Figura 57: Virus Total

Fuente: Elaboración del autor

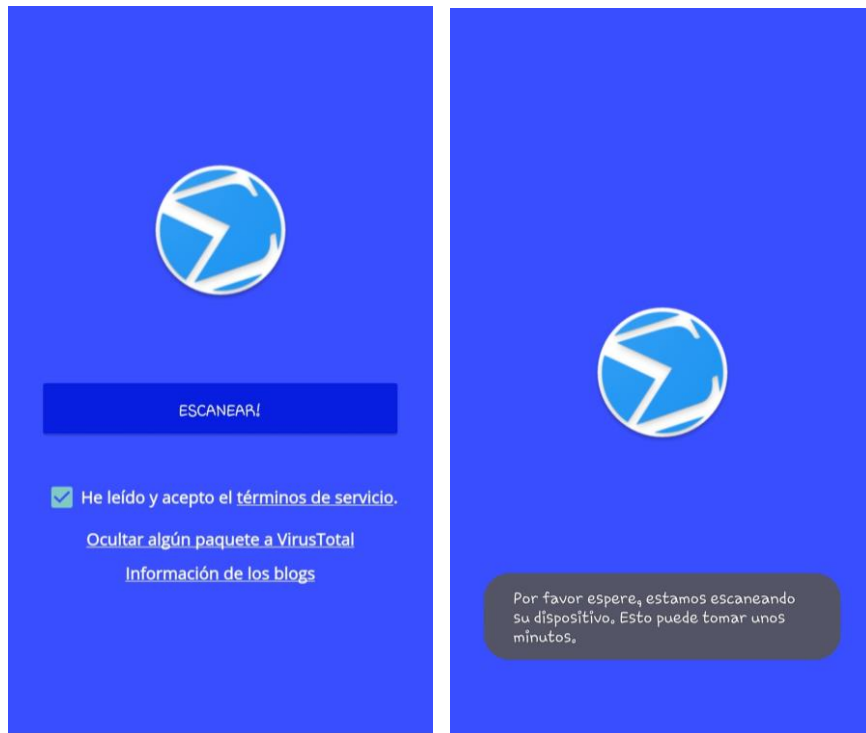


Figura 58: Escanear el dispositivo

Fuente: Elaboración del autor

En la Figura 59 la aplicación realiza el escaneo presentando así la información, permisos y reportes del antivirus.



Figura 59: Escaneo de nRF Mesh

Fuente: Elaboración del autor

La información que se detalla (Figura 60) es importante para verificar la integridad de los datos y el reporte empleado nos indica que la aplicación no genera o descarga algún tipo de virus.

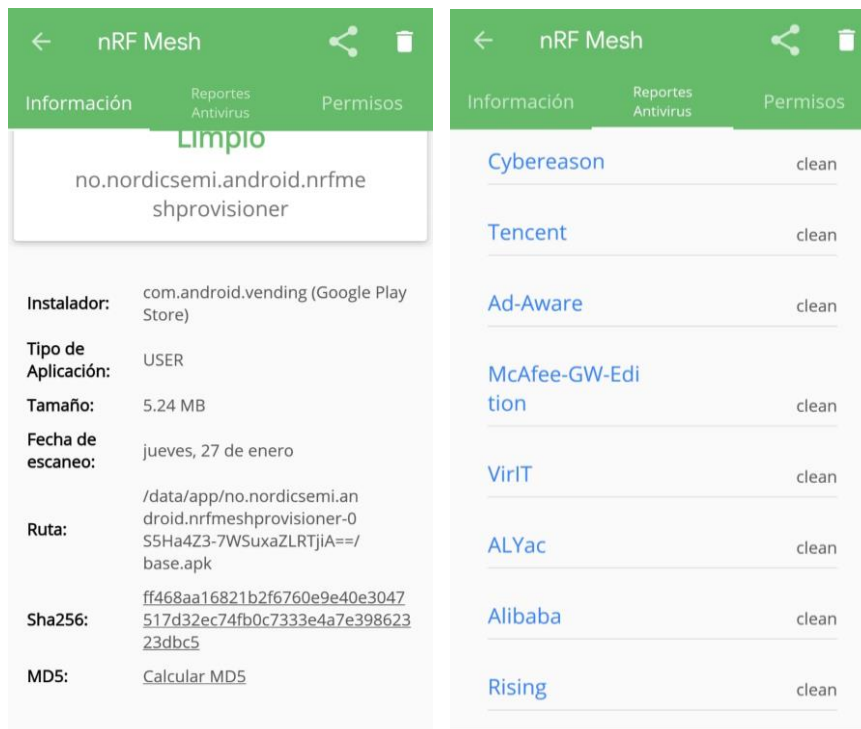


Figura 60: Información y reportes

Fuente: Elaboración del autor

Para el funcionamiento de la App nRF Mesh se debe otorgar permiso de Bluetooth ya que es necesario para la conexión con el módulo ESP32, pero los permisos que se marcan en rojo no son necesarios ya que involucran parte confidencial de nuestro móvil (Figura 61).

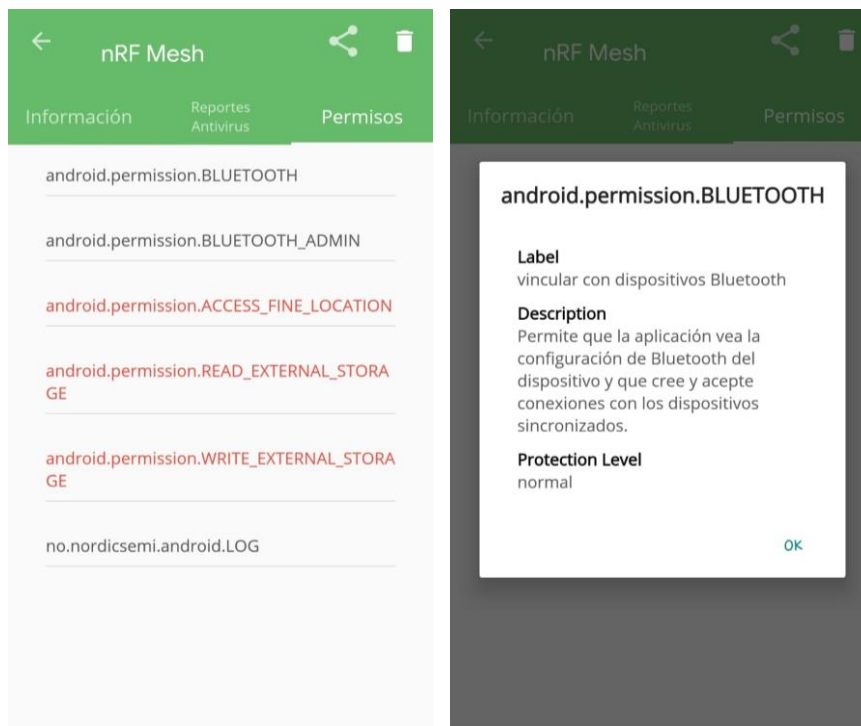


Figura 61: Permisos de la aplicación

Fuente: Elaboración del autor

3.3 Conclusiones

- En las diversas aplicaciones de IoT (Internet de las cosas) se consideran las tecnologías inalámbricas como la fase fundamental para el desarrollo de la conexión de dispositivos y la relación con el ser humano, tal y como hemos podido observar en la comparación de estas tecnologías los atributos que poseen lo definen sus características como el rango, energía, rendimiento y costo siendo importante medir estos atributos y a que tipos de proyectos futuros pueden ser implementados.
- Las aplicaciones más comunes se emplean en el área industrial, comercial y la domótica como el seguimiento de activos, redes de sensores y los sistemas de automatización y control. Los dispositivos que se adaptan a la domótica favorecen el control de un sistema on/off para lograr un hogar inteligente.
- La tecnología Bluetooth Low Energy ofrece varios tipos de topología para la comunicación, point to point es un servicio entre dos dispositivos sin tener la necesidad de administrar un sistema como es el caso de los periféricos desarrollados con el módulo ESP32.
- La red BLE Mesh tiene como ventaja ser una topología escalable que permite la creación de dispositivos o nodos, que al ser aplicada en los módulos ESP-WROOM-32 y ESP32-WROOM-32D se puede controlar diferentes sistemas de iluminación a gran distancia y con los grupos de la App nRF Mesh establece una estructura de gestión según su funcionalidad.
- De todo lo anterior, podemos deducir que el desarrollo de una red que utiliza BLE Mesh es de forma confiable, segura y su usabilidad para el control de los módulos por medio de la App es de forma intuitiva, además que este tipo de tecnología no restringe el uso de la red Wi-Fi.

3.4 Recomendaciones

- El desarrollo del proyecto se debe lograr a base de la plataforma ESP-IDF, no obstante, el módulo ESP32 soporta otros tipos de plataforma como Arduino.
- La utilización de la aplicación nRF Mesh no supone algún problema al instalarlo para su implementación con la red BLE Mesh que trae más beneficios y en el escaneo se comprueba que la seguridad del usuario no está comprometida en ningún sentido.
- Recomendamos que se empleen los grupos de trabajo que ofrece la aplicación cuando se tiene una estructura de red de muchos nodos para optimizar los recursos y reducir el tiempo de respuesta entre los dispositivos inteligentes.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Novillo-Vicuña, D. Hernández Rojas, B. Mazón Olivo, J. Molina Ríos, and O. Cárdenas Villavicencio, *Arduino y el Internet de las cosas*. Editorial Científica 3Ciencias, 2018.
- [2] D. Hernández-Rojas, T. Fernández-Caramés, P. Fraga-Lamas, and C. Escudero, "Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications," *Sensors*, vol. 18, no. 2, p. 57, Dec. 2017, doi: 10.3390/s18010057.
- [3] D. Hernandez-Rojas, B. Mazon-Olivo, J. Novillo-Vicuña, C. Escudero-Cascon, A. Pan-Bermudez, and G. Belduma-Vacacela, "IoT Android Gateway for Monitoring and Control a WSN," in *International Conference on Technology Trends*, vol. 798, 2017, pp. 18–32.
- [4] M. U.Farooq, M. Waseem, S. Mazhar, A. Khairi, and T. Kamal, "A Review on Internet of Things (IoT)," *Int. J. Comput. Appl.*, vol. 113, no. 1, pp. 1–7, 2015, doi: 10.5120/19787-1571.
- [5] A. M. Campoverde, D. L. Hernández, and B. E. Mazón, "Cloud computing con herramientas open-source para Internet de las cosas," *Maskana*, vol. 6, pp. 173–182, 2015.
- [6] J. J. Cartuche Calva, D. L. Hernández Rojas, R. F. Morocho Román, and C. D. Radicelli García, "Seguridad IoT: Principales amenazas en una taxonomía de activos," *Hamut' Ay*, vol. 7, no. 3, pp. 51–59, 2020, doi: 10.21503/hamu.v7i3.2192.
- [7] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," *10th Int. Conf. Front. Inf. Technol.*, pp. 257–260, Dec. 2012, doi: 10.1109/FIT.2012.53.
- [8] D. L. Hernández Rojas, B. Mazon-Olivo, and C. Escudero, *Internet de las cosas (IoT)*. 2018.
- [9] S. Balaji, K. Nathani, and R. Santhakumar, "IoT Technology, Applications and Challenges: A Contemporary Survey," *Wirel. Pers. Commun.*, vol. 108, no. 1, pp. 363–388, Sep. 2019, doi: 10.1007/s11277-019-06407-w.
- [10] R. Mehta, J. Sahni, and K. Khanna, "Internet of Things: Vision, Applications and Challenges," *Procedia Comput. Sci.*, vol. 132, pp. 1263–1269, 2018, doi:

10.1016/j.procs.2018.05.042.

- [11] D. Hernández Rojas, B. Mazon Olivo, and A. M. Campoverde, "Cloud computing para el internet de las cosas. Caso de estudio orientado a la agricultura de precisión," *I Congreso Int. Cienc. y Tecnol. UTMACH 2015*, vol. 1, pp. 47–53, 2015.
- [12] J. P. Novillo-Vicuña, D. L. Hernandez-Rojas, B. Mazon-Olivo, and K. D. Correa-Elizaldes, "Monitoreo inalámbrico de señales eléctricas de voltaje 110/220V a través de Arduino," *Alternativas*, vol. 19, no. 1, pp. 55–62, Jun. 2018, doi: 10.23878/alternativas.v19i1.198.
- [13] D. Hernandez Rojas and H. Villegas, "Implementación de un sistema de monitoreo remoto de una red de impresoras multi-funcionales basado en SNMP y programado con Labview.," *Rev. Tecnológica-ESPOL*, vol. 28, no. 4, pp. 78–93, 2015.
- [14] E. García Vázquez and A. I. G. Santos, "Energy efficient lighting solution for sustainable housing," *Ing. Energética*, vol. 41, no. 2, p. 1511, 2020, [Online]. Available: <http://rie.cujae.edu.cu/index.php/RIE1>.
- [15] E. Batista da Silva, P. A. Ramalho de Souza, and R. Nader, "Tendências no âmbito internet das coisas: um estudo patentário," *Innovar*, vol. 31, no. 81, pp. 49–59, May 2021, doi: 10.15446/innovar.v31n81.95572.
- [16] J. P. L. Escola, R. C. Guido, I. N. da Silva, and L. E. Soares, "IoT device coverage analysis for smart farm monitoring," *RISTI - Rev. Iber. Sist. e Technol. Inf.*, no. 42, pp. 1–11, 2021, doi: 10.17013/RISTI.42.1-11.
- [17] A. Koohang, C. S. Sargent, J. H. Nord, and J. Paliszkievicz, "Internet of Things (IoT): From awareness to continued use," *Int. J. Inf. Manage.*, vol. 62, no. September 2021, p. 102442, 2022, doi: 10.1016/j.ijinfomgt.2021.102442.
- [18] K. F. Muteba, K. Djouani, and T. Olwal, "5G NB-IoT: Design, Considerations, Solutions and Challenges," *Procedia Comput. Sci.*, vol. 198, no. 1, pp. 86–93, 2022, doi: 10.1016/j.procs.2021.12.214.
- [19] A. Haroon, M. Ali, Y. Asim, W. Naeem, M. Kamran, and Q. Javaid, "Constraints in the IoT: The World in 2020 and Beyond," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, pp. 252–271, 2016, doi: 10.14569/IJACSA.2016.071133.
- [20] M. Begum, S. T V, and F. Begum, "IoT-based Monitoring System using Tri-level

- Context Making for Smart Home Services,” *IOSR J. Electr. Electron. Eng.*, vol. 11, no. 04, pp. 01–05, Apr. 2016, doi: 10.9790/1676-1104010105.
- [21] S. Kiran and S. B. Sriramoju, “A study on the applications of IOT,” *Indian J. Public Heal. Res. Dev.*, vol. 9, no. 11, pp. 1173–1175, 2018, doi: 10.5958/0976-5506.2018.01616.9.
- [22] D.-R. Berte, “Defining the IoT,” *Proc. Int. Conf. Bus. Excell.*, vol. 12, no. 1, pp. 118–128, May 2018, doi: 10.2478/picbe-2018-0013.
- [23] E. Cubillas Hernández, C. Anías Calderón, and T. Delgado Fernández, “M2M Architecture for environmental monitoring in real time,” *ITECKNE*, vol. 18, no. 1, pp. 1–8, Mar. 2021, doi: 10.15332/iteckne.v18i1.2531.
- [24] J. A. Laverde Mena and C. G. Laverde Mena, “Internet de las cosas aplicado en la agricultura ecuatoriana: Una propuesta para sistemas de riego,” *Dilemas Contemp. Educ. Política y Valores*, vol. 8, no. 2, pp. 1–14, Jan. 2021, doi: 10.46377/dilemas.v8i2.2542.
- [25] J. Xu, B. Gu, and G. Tian, “Review of agricultural IoT technology,” *Artif. Intell. Agric.*, vol. 6, pp. 10–22, 2022, doi: 10.1016/j.aiia.2022.01.001.
- [26] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, and A. Pan, “Rules engine and complex event processor in the context of internet of things for precision agriculture,” *Comput. Electron. Agric.*, vol. 154, no. July, pp. 347–360, 2018, doi: 10.1016/j.compag.2018.09.013.
- [27] J. Berrú-Ayala, D. Hernandez-Rojas, P. Morocho-Díaz, J. Novillo-Vicuña, B. Mazon-Olivo, and A. Pan, “SCADA System Based on IoT for Intelligent Control of Banana Crop Irrigation,” in *International Conference on Applied Technologies*, 2019, pp. 243–256.
- [28] D. Hernández-Rojas, T. Fernández-Caramés, P. Fraga-Lamas, and C. Escudero, “A Plug-and-Play Human-Centered Virtual TEDS Architecture for the Web of Things,” *Sensors*, vol. 18, no. 7, p. 2052, Jun. 2018, doi: 10.3390/s18072052.
- [29] A. Quiroz Estrada, G. A. Acosta Amaya, and R. A. Torres Villa, “Diseño de un sistema internet de las cosas (IoT) para el monitoreo de la presión arterial,” *Rev. EIA*, vol. 18, no. 35, pp. 1–15, Jan. 2021, doi: 10.24050/reia.v18i35.1474.
- [30] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya,

- “Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare,” *Futur. Gener. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018, doi: 10.1016/j.future.2017.04.036.
- [31] L. Liang, L. Huang, X. Jiang, and Y. Yao, “Design and implementation of wireless Smart-home sensor network based on ZigBee protocol,” *Int. Conf. Commun. Circuits Syst. Proc.*, pp. 434–438, 2008, doi: 10.1109/ICCCAS.2008.4657808.
- [32] C. D. Escobar-Amado, J. M. Celis-Peñaranda, S. B. Sepúlveda-Mora, S. A. Castro-Casadiago, B. Medina-Delgado, and D. Guevara, “Adaptive nodes algorithm to solve the orphan nodes problem in a ZigBee Tree WSN,” *Ingeniare. Rev. Chil. Ing.*, vol. 29, no. 2, pp. 285–296, Jun. 2021, doi: 10.4067/S0718-33052021000200285.
- [33] D. D. Kalamkar, M. Chaudhuri, and M. Heinrich, “Simplifying Active Memory Clusters by Leveraging Directory Protocol Threads,” *IEEE Int. Symp. Perform. Anal. Syst. Softw.*, no. April 2007, pp. 242–253, Apr. 2007, doi: 10.1109/ISPASS.2007.363754.
- [34] C. W. Badenhop, S. R. Graham, B. W. Ramsey, B. E. Mullins, and L. O. Mailloux, “The Z-Wave routing protocol and its security implications,” *Comput. Secur.*, vol. 68, pp. 112–129, Jul. 2017, doi: 10.1016/j.cose.2017.04.004.
- [35] M. Lilli, C. Braghin, and E. Riccobene, “Formal Proof of a Vulnerability in Z-Wave IoT Protocol,” *Proc. 18th Int. Conf. Secur. Cryptogr.*, no. Secrypt, pp. 198–209, 2021, doi: 10.5220/0010553301980209.
- [36] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of LPWAN technologies for large-scale IoT deployment,” *ICT Express*, vol. 5, no. 1, pp. 1–7, Mar. 2019, doi: 10.1016/j.icte.2017.12.005.
- [37] R. Sandoval, A.-J. Garcia-Sanchez, F. Garcia-Sanchez, and J. Garcia-Haro, “Evaluating the More Suitable ISM Frequency Band for IoT-Based Smart Grids: A Quantitative Study of 915 MHz vs. 2400 MHz,” *Sensors*, vol. 17, no. 12, p. 76, Dec. 2016, doi: 10.3390/s17010076.
- [38] O. Martinez, C. Arguello, J. León, P. D. Cherres Carguacundo, and G. E. Cujilema Daga, “Prototype of automated irrigation system improves the yield of potatoes (*Solanum tuberosum* L.) in Riobamba-Ecuador using wireless network sensors-WSN and 6LoWPAN,” *MASKAY*, vol. 9, no. 2, p. 25, Mar. 2019, doi: 10.24133/maskay.v9i2.1058.

- [39] C. R. Egas, D. Viracocha, and J. Rivera, "Implementación de una red inalámbrica de sensores para la gestión de luminarias utilizando IPv6," *Enfoque UTE*, vol. 10, no. 4, pp. 45–56, Dec. 2019, doi: 10.29019/enfoque.v10n4.553.
- [40] A. Prisantama, Widyawan, and I. W. Mustika, "Tunneling 6LoWPAN protocol stack in IPv6 network," *AIP Conf. Proc.*, vol. 1755, no. July 2016, p. 070006, 2016, doi: 10.1063/1.4958501.
- [41] J. I. Vega Luna, F. J. Sánchez-Rangel, and J. F. Cosme-Aceves, "Sistema de monitorización de puertas y ventanas de un centro de datos con IoT," *Ingenius*, no. 22, pp. 72–79, Jun. 2019, doi: 10.17163/ings.n22.2019.07.
- [42] V. S. Silva *et al.*, "IEEE 802.11b/g Practical Assessment Using a Low-Cost Quasi-Yagi Rectenna for Indoor RF Energy Harvesting," *J. Microwaves, Optoelectron. Electromagn. Appl.*, vol. 20, no. 1, pp. 143–157, 2021, doi: 10.1590/2179-10.1590/2179-10742021v20i11052.
- [43] N. Todtenberg and R. Kraemer, "A survey on Bluetooth multi-hop networks," *Ad Hoc Networks*, vol. 93, p. 101922, Oct. 2019, doi: 10.1016/j.adhoc.2019.101922.
- [44] M. Bin Yaakop, I. A. Abd Malik, Z. bin Suboh, A. F. Ramli, and M. A. Abu, "Bluetooth 5.0 throughput comparison for internet of thing usability a survey," *Int. Conf. Eng. Technol. Technopreneursh.*, vol. 2017-Janua, no. September, pp. 1–6, Sep. 2017, doi: 10.1109/ICE2T.2017.8215995.
- [45] R. Abdelmoumen, "A Review of Link Layer Protocols for Internet of Things," *Int. J. Comput. Appl.*, vol. 182, no. 46, pp. 22–28, Mar. 2019, doi: 10.5120/ijca2019918618.
- [46] Á. J. Ordóñez Mendieta and Á. P. Garrochamba, "Diseño de una WSN para el monitoreo de CO2 en el aire y niveles de ruido en la ciudad de Loja," *MASKAY*, vol. 10, no. 1, p. 20, Nov. 2019, doi: 10.24133/maskay.v10i1.1522.
- [47] G. Calderón, J. H. Muñoz, and J. Y. Rivera, "Dispositivo para medir tiempo y temperatura usando un microcontrolador," *Rev. Bras. Ensino Física*, vol. 40, no. 2, Nov. 2017, doi: 10.1590/1806-9126-rbef-2017-0228.
- [48] E. Díaz Ronceros, "Relevancia de la ejecución experimental de proyectos con microcontroladores en el aprendizaje de la ingeniería electrónica," *Educación*, vol. 29, no. 56, pp. 48–72, 2020, doi: 10.18800/educacion.202001.003.

- [49] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," *Internet Technol. Appl.*, no. November, pp. 143–148, Sep. 2017, doi: 10.1109/ITECHA.2017.8101926.
- [50] D. A. Jatmiko and S. U. Prini, "Implementasi dan Uji Kinerja Algoritma Background Subtraction pada ESP32," *Komputika J. Sist. Komput.*, vol. 8, no. 2, pp. 59–65, Nov. 2019, doi: 10.34010/komputika.v8i2.2194.
- [51] Jia Liu, Canfeng Chen, and Yan Ma, "Modeling and performance analysis of device discovery in Bluetooth Low Energy networks," *IEEE Glob. Commun. Conf.*, no. December, pp. 1538–1543, Dec. 2012, doi: 10.1109/GLOCOM.2012.6503332.
- [52] H.-S. Kim, J. Lee, and J. W. Jang, "BLEmesh: A Wireless Mesh Network Protocol for Bluetooth Low Energy Devices," *3rd Int. Conf. Futur. Internet Things Cloud*, pp. 558–563, Aug. 2015, doi: 10.1109/FiCloud.2015.21.
- [53] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors (Switzerland)*, vol. 12, no. 9, pp. 11734–11753, 2012, doi: 10.3390/s120911734.
- [54] S. Darroudi and C. Gomez, "Bluetooth Low Energy Mesh Networks: A Survey," *Sensors*, vol. 17, no. 7, p. 1467, Jun. 2017, doi: 10.3390/s17071467.
- [55] A. Chiumento, B. Reynders, Y. Murillo, and S. Pollin, "Building a connected BLE mesh: A network inference study," *IEEE Wirel. Commun. Netw. Conf. Work.*, no. June, pp. 296–301, Apr. 2018, doi: 10.1109/WCNCW.2018.8369004.
- [56] Q. Wan and J. Liu, "Smart-Home Architecture Based on Bluetooth mesh Technology," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 322, no. 7, p. 072004, Mar. 2018, doi: 10.1088/1757-899X/322/7/072004.
- [57] E. Kalinin, D. Belyakov, D. Bragin, and A. Konev, "IoT security mechanisms in the example of BLE," *Computers*, vol. 10, no. 12, 2021, doi: 10.3390/computers10120162.
- [58] R. K. Kodali, G. Swamy, and B. Lakshmi, "An implementation of IoT for healthcare," *IEEE Recent Adv. Intell. Comput. Syst.*, no. January, pp. 411–416, Dec. 2015, doi: 10.1109/RAICS.2015.7488451.

[59] E. Systems, “ESP-IDF Programming Guide,” 2022.
<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>.