

# **TRABAJO ESPECIAL DE GRADO**

## **DISEÑO DE UN SISTEMA DE ADQUISICIÓN Y TRANSMISIÓN DE DATOS ORIENTADO A MEDIDORES DE ENERGÍA ELÉCTRICA, UTILIZANDO UNA RED MALLADA WIFI CON MICROCONTROLADORES ESP32**

Presentado ante la ilustre  
Universidad Central de Venezuela  
por el Br. Marco Alejandro Rodríguez Ferrer  
para optar al título de  
Ingeniero Electricista.

Caracas, julio de 2020

# **TRABAJO ESPECIAL DE GRADO**

## **DISEÑO DE UN SISTEMA DE ADQUISICIÓN Y TRANSMISIÓN DE DATOS ORIENTADO A MEDIDORES DE ENERGÍA ELÉCTRICA, UTILIZANDO UNA RED MALLADA WIFI CON MICROCONTROLADORES ESP32**

TUTOR ACADÉMICO: Ing. José Alonso

Presentado ante la ilustre  
Universidad Central de Venezuela  
por el Br. Marco Alejandro Rodríguez Ferrer  
para optar al título de  
Ingeniero Electricista.

Caracas, julio de 2020



*A mis padres  
Ayarlen Ferrer y Marco Rodríguez.*

*A la memoria de mis abuelos  
Esperanza Merecuana y Francisco Ferrer.*

*Dónde quiera que se encuentren, por todo lo que hicieron por mi y mis hermanos.  
Eternamente gracias.*

## RECONOCIMIENTOS Y AGRADECIMIENTOS

**Autor del Trabajo de Grado**

**Título del Trabajo de Grado**

**Tutor Académico:** nombre del profesor. Tesis. Caracas, Universidad Central de Venezuela. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Mención Electrónica. Año 2020, xvii, 144 pp.

**Palabras Claves:** Palabras clave.

**Resumen.-** Escribe acá tu resumen

# ÍNDICE GENERAL

RECONOCIMIENTOS Y AGRADECIMIENTOS	III
ÍNDICE GENERAL	VIII
LISTA DE FIGURAS	XII
LISTA DE TABLAS	XIV
LISTA DE ACRÓNIMOS	XV
INTRODUCCIÓN	1
CONCEPTUALIZACIÓN DEL PROYECTO	3
1.1. PLANTEAMIENTO DEL PROBLEMA . . . . .	3
1.2. JUSTIFICACIÓN . . . . .	4
1.3. OBJETIVO GENERAL . . . . .	5
1.4. OBJETIVOS ESPECÍFICOS . . . . .	5
MARCO REFERENCIAL	7
2.1. Sistema SCADA . . . . .	7
2.1.1. Definición . . . . .	7
2.2. Fundamentos de medición consumo de energía eléctrica . . . . .	7
2.2.1. Definición . . . . .	7
2.2.2. Clasificación de los medidores . . . . .	8
2.2.2.1. Según el principio de funcionamiento . . . . .	8

2.2.2.2.	Según su construcción: . . . . .	9
2.2.3.	Infraestructuras de medición de energía . . . . .	10
2.2.3.1.	Red de área local (Home Area Network) . . . . .	12
2.2.3.2.	Red de vecindad (Neighbourhood Area Network)	12
2.2.3.3.	Red de área amplia (Wide Area Network) . . . . .	13
2.2.4.	Normativa aplicada a las redes de medición . . . . .	14
2.2.4.1.	Modelo OSI . . . . .	14
2.2.4.2.	GWAC . . . . .	16
2.3.	Redes malladas . . . . .	18
2.3.1.	Definición . . . . .	18
2.3.2.	Características . . . . .	18
2.3.2.1.	Sobre los protocolos de enrutamiento en las redes Malladas . . . . .	19
2.3.2.2.	Direccionamiento por MAC . . . . .	20
2.4.	La red mallada ESP-MESH . . . . .	21
2.4.1.	Definición . . . . .	21
2.4.2.	Terminología en la ESP-MESH . . . . .	22
2.4.3.	Topología de árbol sobre la red ESP-MESH . . . . .	23
2.4.4.	Tipos de nodos en la red ESP-MESH . . . . .	25
2.4.5.	Señales de reconocimiento y límites de intensidad de señal (RSSI) . . . . .	26
2.4.6.	Tablas de enrutamiento . . . . .	28
2.5.	Sistema operativo en tiempo real (RTOS) . . . . .	29
2.5.1.	FreeRTOS . . . . .	30
2.5.1.1.	Tareas . . . . .	30



2.5.1.2.	Colas . . . . .	30
2.5.1.3.	Semáforos . . . . .	31
2.5.1.4.	Grupos de eventos . . . . .	31
2.6.	Microcontrolador ESP32 . . . . .	31
2.6.1.	Descripción . . . . .	31
2.6.2.	Características . . . . .	32
2.6.2.1.	WiFi . . . . .	32
2.6.2.2.	CPU y Memoria . . . . .	32
2.6.2.3.	Relojes y temporizadores . . . . .	33
2.6.2.4.	Periféricos . . . . .	33
2.6.2.5.	Seguridad . . . . .	34
<b>DEFINICIÓN Y DESCRIPCIÓN DEL HARDWARE</b>		<b>38</b>
3.1.	Definición y Descripción del Hardware . . . . .	38
3.1.1.	Definición . . . . .	38
3.1.2.	Descripción . . . . .	38
<b>DEFINICIÓN Y DESCRIPCIÓN DEL SOFTWARE</b>		<b>39</b>
4.1.	Definición del software . . . . .	39
4.2.	Descripción del software . . . . .	42
4.2.1.	Modo de configuración . . . . .	43
4.2.1.1.	Tablas de particiones en el ESP32 . . . . .	43
4.2.1.2.	Interfaz Gráfica de usuario . . . . .	45
4.2.2.	Modo de operación . . . . .	46
4.2.2.1.	La red mallada y el nodo central . . . . .	47
4.2.2.2.	Nodo serial RS485 . . . . .	52

4.2.2.3. Nodo de entrada por pulsos . . . . .	53
4.2.2.4. Nodo Repetidor . . . . .	60
4.2.2.5. Otras consideraciones . . . . .	60
<b>PRUEBAS Y RESULTADOS</b>	<b>61</b>
<b>CONCLUSIONES</b>	<b>62</b>
<b>RECOMENDACIONES</b>	<b>63</b>
<b>TÍTULO DEL ANEXO</b>	<b>64</b>
<b>TÍTULO DEL ANEXO</b>	<b>65</b>
<b>TÍTULO DEL ANEXO</b>	<b>66</b>
<b>REFERENCIAS</b>	<b>67</b>

## LISTA DE FIGURAS

2.1.	Infraestructura general de un sistema de medición avanzado . . .	11
2.2.	Arquitectura de red en comunicaciones para medición inteligente. J. Ekanayake and K. Liyanage, SMART GRID TECHNOLOGY AND APPLICATIONS, First edit. New Delhi, 2013. . . . .	13
2.3.	E. Hossain, Z. Han, and V. Poor, Smart Grid Communications and Networking. New York: CAMBRIDGE UNIVERSITY PRESS, 2012.	17
2.4.	Ilustración de una red Wi-Fi tradicional. Tomada desde [1] . . . .	21
2.5.	Ilustración de una red Wi-Fi ESP-MESH. Tomada desde [1] . . .	22
2.6.	Topología de árbol observada en la red ESP-MESH luego de su establecimiento. Tomada de [1] . . . . .	24
2.7.	Tipos de nodos en una red ESP-MESH. Tomada desde [1] . . . .	25
2.8.	Ejemplo ilustrativo sobre el umbral RSSI. Tomado de [1] . . . . .	27
2.9.	Ilustración de una red mallada, las tablas y subtablas de enruta- miento. Tomado de [1] . . . . .	28
2.10.	Interacción del OS con otras capas del sistema informático. . . . .	29
4.1.	Diagrama de bloques genérico del sistema . . . . .	40
4.2.	Diagrama general de funcionamiento ilustrado. . . . .	42
4.3.	Diagrama ilustrativo de la memoria flash para los modelos de tablas de partición. Columna 1: Aplicación de fábrica. Columna 2, 3, 4: Aplicaciones con arranques múltiples . . . . .	44
4.4.	Diagrama de flujo de las tareas implementadas en el nodo central.	48
4.5.	Estructura de una trama Modbus TCP/IP. . . . .	51

- 4.6. Diagrama de flujo de las tareas implementadas en el nodo RS485. 52
- 4.7. Diagrama de flujo de las tareas implementadas en el nodo de pulsos. 55

## LISTA DE TABLAS

2.1. Otra tabla . . . . .	36
2.2. Ejemplo . . . . .	36
2.3. Ejemplo . . . . .	37
4.1. ESP-IDF Tabla de Particiones por defecto (Aplicación única de fábrica, sin OTA) . . . . .	44
4.2. Tabla de particiones personalizada para la aplicación . . . . .	45
4.3. Tabla de particiones de la aplicación . . . . .	56

## LISTA DE ACRÓNIMOS

**IEC:** International Electrotechnical Comission Comisión Electrotécnica Internacional

**RTOS:** Real Time Operative Sysyem, Sistema operativo en tiempo real

**AMR:** Automatic Meter Reading, Lectura de medición automática

**HAN:** Home Area Network, Red de área doméstica

**NAN:** Neighbourhood Area Network, Red de área de vecindario

**WAN:** Wide Area Network, Red de área amplia

**ISO:** International Organization for Standardization, Organización Internacional de Normalización

**OSI:** Open System Interconnection, Interconexión de Sistemas Abiertos

**GUI:** Grafic User Interface, Interfaz Gráfica de Usuario

**NVS:** Non volatile Storage, Memoria no Volátil

**STA:** STA, Estación (Utilizado para WiFi)

**AP:** Access Point, Punto de Acceso

# INTRODUCCIÓN

Los primeros indicios del uso de la energía eléctrica sucedieron en el cuarto final del siglo XIX. La sustitución del gas y aceite por la electricidad además de ser un proceso técnico fue un verdadero cambio social que implicó modificaciones extraordinarias en la vida cotidiana de las personas, cambios que comenzaron por la sustitución del alumbrado público y posteriormente por varias clases de procesos industriales como motores, metalurgia, refrigeración y de último llegaron a las comunicaciones con la radio y la telefonía.

El siguiente cambio de paradigma en el que se vio involucrado la electricidad tuvo lugar a lo largo del siglo XX y surge desde la necesidad de facilitar las tareas realizadas a diario en casa. En ello los investigadores de la época vieron una solución adaptando equipos con energía eléctrica para su uso en el hogar. Las industrias replicaron el crecimiento tecnológico que tuvieron en sus productos, lo que trajo como consecuencia el desarrollo los electrodomésticos. La primera producción de aparatos en masa como refrigeradores, lavadoras, televisores y radios sucedieron en esta época y tuvieron una alta receptividad por parte de los compradores. La invención del transistor solo aceleró el reemplazo de aparatos dada su capacidad de minimizar los equipos.

La integración de la electrónica a la industria fomentó la creación de sistemas automatizados de adquisición de datos, supervisión y control también llamados sistemas *SCADA* por sus siglas en inglés. Estos sistemas manejan áreas críticas de las industrias y son parte de los procesos fundamentales de muchas de ellas por lo

que necesitan ser diseñados con robustez, fiabilidad y seguridad. La aparición del Internet y las comunicaciones modernas en estos sistemas permite a los usuarios, de manera inalámbrica incluso, monitorear y actuar sobre el sistema a distancia, sin presencia física en la planta.

Además de poder monitorear y realizar acciones sobre los sistemas, los instrumentos de medida de última tecnología se fabrican de modo que puedan ser compatibles con medios de comunicación inalámbricas lo que posibilita la transmisión de datos adquiridos sin necesidad de cable a la central del sistema *SCADA*. El presente trabajo de grado pretende realizar el diseño de un sistema de adquisición y transmisión de datos integrando microcontroladores ESP32 a medidores de energía eléctrica para formar una red mallada inalámbrica capaz de transmitir los datos recolectados a un punto central.

En este archivo debe escribir su introducción.

De acuerdo a Brea la transformada de Laplace debe estudiarse como una función definida en el campo de los números complejos [? ].

Otro modo de referencial es [? ]

El resto del reporte consta de: en el Capítulo ?? se describe...

En el trabajo se emplea el enfoque de [? ]

De acuerdo a la ecuación



# CAPÍTULO I

## CONCEPTUALIZACIÓN DEL PROYECTO

### 1.1. PLANTEAMIENTO DEL PROBLEMA

La energía eléctrica es diferente de otras manifestaciones de la energía, debido a que no se puede almacenar por sí sola como electricidad. Esto obliga a que la energía eléctrica consumida por un equipo u aparato tenga que generarse al momento en el cual se vaya a consumir. Los procesos para la generación de energía tienen costos altos de desarrollo e implementación a gran escala (países o estados) por lo que surtir de energía a las industrias y electrodomésticos tiene un costo que la empresa que genera la energía necesita recuperar. Como consecuencia se suele medir el consumo de cada uno de los usuarios por razones enteramente económicas.

En Venezuela se utiliza el mismo método de adquisición de datos desde que se instaló el sistema eléctrico. Este consiste en un operador que se acerca hasta el lugar donde se encuentra un medidor de energía y registra la lectura que marca el medidor, esto se hace de manera repetitiva para todos los sitios donde se quiera registrar el consumo. En ocasiones los medidores tienen una salida codificada donde comunica el valor del consumo por infrarrojo lo que permite al operador registrar el valor de ese consumo mediante un aparato compatible con este protocolo. Debido a esta problemática surge la necesidad de sustituir este sistema de adquisición de datos manual por uno que no requiera el traslado del

operador hasta el sitio, que sea económico, confiable y eficiente.

Los principales equipos de medición de energía poseen en su diseño una salida por pulsos y soportan distintos protocolos de comunicación, lo que representa una ventaja al trabajar con microcontroladores, pues estos son adaptables a la mayoría de los protocolos mediante programación lo que facilita la adquisición de los datos a partir del medidor. Por otra parte trabajar con microcontroladores ofrece la posibilidad de realizar comunicaciones inalámbricas si se adapta un módulo WiFi como periférico. Interconectar estos módulos WiFi para formar una red mallada permitiría la transmisión de los datos captados a una mayor distancia que la lograda por un único módulo y permitiría su salida hacia alguna red externa deseada sin utilizar cables entre los medidores y la central de adquisición de datos, y sin intervención presencial del operador. Ilustradas las debilidades expuestas anteriormente y las ventajas que representaría un sistema de este tipo se evidencia la necesidad de realizar el diseño.

## **1.2. JUSTIFICACIÓN**

Una red mallada WiFi utilizando microcontroladores permite adaptar a la red equipos que soportan distintos métodos de extracción de datos; otorga la posibilidad de interconectar dispositivos mediante comunicaciones inalámbricas, que no poseen dicha capacidad originalmente; además su desarrollo permitiría extender las variables a medir y los métodos de adquisición de datos del sistema; y por último, posee bajos costos de instalación al no requerir de cableado entre los elementos de la red.

Establecer la red mallada se requiere de nodos que posean la capacidad de enlazarse entre sí formando redes de comunicación, además los nodos deben ser capaces de enrutar los mensajes donde viaja la información. Para un sistema de adquisición de datos es ideal que cada nodo de la red este conformado por un microcontrolador con un módulo WiFi integrado, esto permitiría que cada uno de los nodos extrajera los datos según el método que se requiera en cada fuente de datos y al formar parte de la red mallada se facilitaría la adquisición de datos para un sistema central. El ESP32 es una opción viable para esta aplicación debido a su módulo WiFi integrado y las librerías desarrolladas en comunicación vía WiFi, utilizar dicha tarjeta representa una ventaja económica y reduce los tiempos de desarrollo respecto a otros microcontroladores.

### **1.3. OBJETIVO GENERAL**

Diseñar un sistema de adquisición y transmisión de datos orientado a medidores de energía eléctrica, utilizando una red mallada WiFi con microcontroladores ESP32.

### **1.4. OBJETIVOS ESPECÍFICOS**

- Documentar los principales métodos de extracción de datos soportados por un medidor de energía, en particular, el protocolo Modbus por RS485 y la salida por pulsos.
- Diseñar el módulo de programa para los nodos que componen la red mallada, conformados por microcontroladores ESP32
- Adaptar un nodo para ser compatible con la salida por pulsos de un medidor de energía y almacenar el valor de la medida para su adquisición mediante

la red.

- Adaptar un nodo para adquirir datos desde un medidor de energía que soporte protocolo Modbus RTU vía RS485.
- Validar el funcionamiento del sistema.

## CAPÍTULO II

### MARCO REFERENCIAL

En este capítulo se documentarán los fundamentos sobre sistemas SCADA, Medidores de energía, Redes WiFi Malladas y el protocolo Modbus.

#### **2.1. Sistema SCADA**

##### **2.1.1. Definición**

#### **2.2. Fundamentos de medición consumo de energía eléctrica**

##### **2.2.1. Definición**

Los medidores de consumo de energía eléctrica, también llamados contadores de energía debido a la tarea que desempeñan, son “Un instrumento destinado a medir la energía eléctrica integrando la potencia con respecto al tiempo” [2]. Las compañías de electricidad utilizan medidores de energía instalados en cada cliente con el propósito de monitorear y facturar el consumo. Debido a ello estos equipos suelen estar calibrados en unidades de facturación de energía, comúnmente se utiliza el kilovatio hora [kWh] y se lee el valor registrado en cada medidor una vez ha llegado el período de cobranza.

Cuando se desea ahorrar energía durante ciertos periodos de tiempo, algunos medidores pueden registrar la demanda de energía, es decir el máximo uso de potencia en un intervalo de tiempo. Aplicar esta metodología de registro de demanda brinda la capacidad de variar las tarifas de electricidad durante el día, permitiendo registrar el consumo de cada usuario durante periodos de tarifas altas (picos de demanda) o en casos de tarifas bajas donde la demanda de energía en el sistema es baja y la energía menos costosa. Incluso en algunas áreas los medidores de energía poseen relés para desprendimiento de cargas por un periodo de tiempo en caso de picos muy altos de demanda.

### **2.2.2. Clasificación de los medidores**

Tomando como referencia lo que [3] refleja en su trabajo, los medidores de consumo de energía pueden agruparse mediante las siguientes clasificaciones.

#### **2.2.2.1. Según el principio de funcionamiento**

**Medidores electromecánicos:** El tipo más común de medidor eléctrico, el registro de la energía se realiza mediante el conteo de las revoluciones de un disco metálico que es conductor eléctrico pero no conductor magnético. Este disco se hace rotar mediante la inducción electromagnética generada por la alimentación, a una velocidad proporcional a la potencia que pasa a través del medidor. El número de vueltas es entonces proporcional a el consumo de energía.

**Medidores electrónicos:** De forma general este medidor está compuesto por la alimentación, un circuito de medición, un circuito de procesamiento (usualmente microcontroladores) y comunicación además de otros módulos agregados como un RTC, una pantalla de cristal líquido, un módulo de

comunicación mediante infrarrojo, entre otros.

El circuito de medición está compuesto por muestreadores y cuantificadores conectados a las entradas de corriente y tensión, así como a la referencia de tensión. Esto viene seguido por una sección de conversión analógica-digital para encontrar el equivalente digitalizado del valor de las entradas. Dichas entradas en formato digital son tratadas utilizando un procesador digital de señal para calcular los distintos parámetros de medición.

Este tipo de medidor muestra la energía consumida en una pantalla LCD o LED, además de medir la energía consumida pueden también registrar otros parámetros de la carga y del suministro, como la tasa de demanda instantánea y máxima.

#### **2.2.2.2. Según su construcción:**

##### **Medidor monofásico bifilar (una fase y neutro):**

Está compuesto por una bobina de tensión y una de corriente. Su capacidad usualmente está entre 15 A y 60 A.

##### **Medidor bifásico bifilar (dos fases):**

Está compuesto por dos bobinas de tensión y dos bobinas de corriente. Usualmente utilizado para medir la energía eléctrica consumida por aparatos que funcionan con la tensión fase-fase residencial.

##### **Medidor bifásico trifilar (dos fases y neutro):**

Está compuesto por dos bobinas de tensión y dos bobinas de corriente. Se usa para medir la energía consumida por aparatos que requieran para su funcionamiento dos fases, con este medidor se puede medir la energía consumida por otros aparatos conectados a la misma instalación que funcionen con una sola fase.

### **Medidor trifásico tetrafilar (tres fases y neutro):**

Está compuesto por tres bobinas de tensión y tres bobinas de corriente. Se utiliza para medir la energía consumida por aparatos que requieran funcionar con tres fases.

### **2.2.3. Infraestructuras de medición de energía**

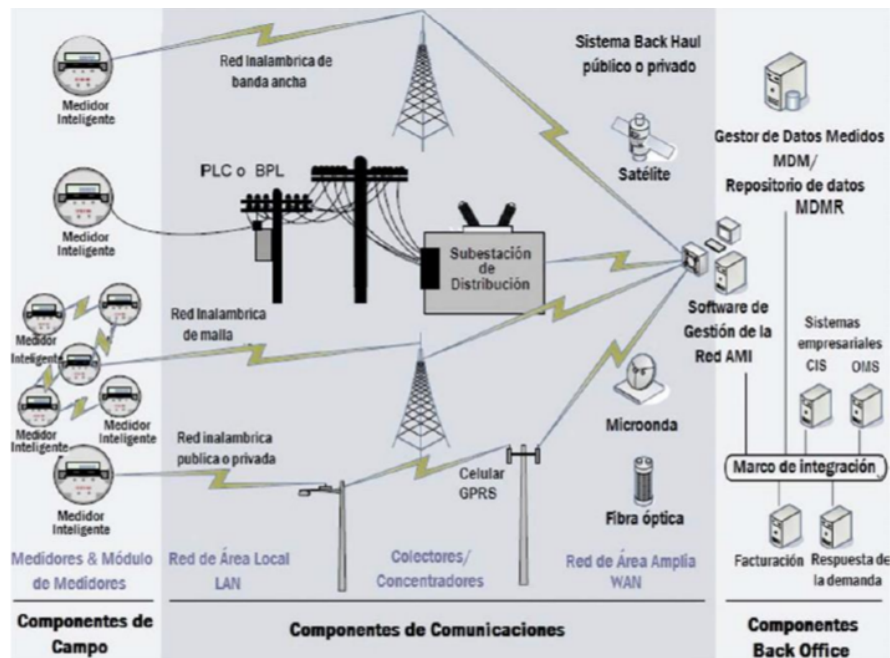
La infraestructura de cada sistema de medición es la característica que delimita sus particularidades, "El cambio de visión de la medición de energía eléctrica se realiza desde el año de 1970 con la incorporación del envío de datos, la comunicación en los años 70 era unidireccional, en una sola vía, desde el usuario hasta la empresa distribuidora. La primera etapa fue la medición AMR (Automatic Meter Reading) y duró alrededor de 30 años dando paso a la evolución, llamada medición inteligente o smart Metering. Smart metering requiere un alto grado de telecomunicaciones en ambas vías: usuario-empresa distribuidora." [4].

La meta en cuanto a estos sistemas es llegar a una medición inteligente avanzada, es decir incorporar a toda la red eléctrica dispositivos con la capacidad de operar por medio de telecomunicaciones, permitiendo equilibrar la generación de energía eléctrica al consumo real, por medio de los datos enviados por todos los medidores inteligentes.

Existen ciertos requerimientos para realizar una medición inteligente: alta confiabilidad, vida útil, interoperabilidad, rentabilidad, seguridad, consumo mínimo de energía, bajos costos de instalación y mantenimiento. La tecnología desarrollada debe cumplir con estos requerimientos para ser realmente funcional en el campo de la medición automática de consumo.



Para realizar medición inteligente se pueden emplear diferentes tipos de tecnologías de telecomunicaciones de acuerdo al área de aplicación y al canal de transmisión. La figura 2.1 muestra los tipos de arquitecturas de comunicación que son empleadas en la medición inteligente. Los medios guiados para realizar telecomunicaciones incluyen la red telefónica pública conmutada (PSTN), PLC portadora en la línea de alimentación, Ethernet y cable módems. Los medios no guiados para realizar telecomunicaciones incluyen WiFi, ZigBee, infrarrojos, RFID y GSM / GPRS / CDMA celular.



**Figura 2.1.** Infraestructura general de un sistema de medición avanzado  
**Fuente:** Adaptado de [5]

La infraestructura necesaria para la medición inteligente se divide en tres zonas de acuerdo al tipo de comunicación [5], dichas zonas son las siguientes:

- Red de área local (HAN).

- Red de área de vecindad (NAN).
- Red de área amplia (WAN).

#### **2.2.3.1. Red de área local (Home Area Network)**

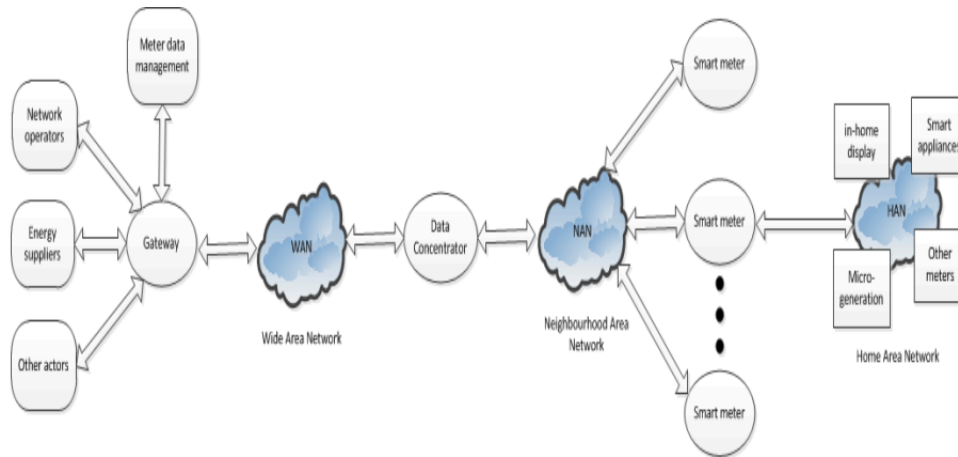
Es un sistema integrado dentro de los hogares que permite la comunicación entre diferentes dispositivos inteligentes, el dispositivo límite de esta red es el medidor de energía eléctrica. Este tipo de redes puede utilizar comunicaciones cableadas como BPL (Broadband over Power Lines) o PLC (Power Line Carrier) pero también comunicaciones inalámbricas como redes privadas, una red de banda ancha o redes malladas para comunicarse con la red NAN.

#### **2.2.3.2. Red de vecindad (Neighbourhood Area Network)**

Este tipo de red permite la conexión entre múltiples HANs, es un sistema de interconexión entre redes de medidores inteligentes. El principal elemento que lo constituye es el concentrador. El concentrador es el dispositivo que actúa como un puente entre los contadores inteligentes y la puerta de enlace. El concentrador de datos detecta y gestiona los medidores inteligentes de forma automática, realiza lecturas de los consumos y transfiere la información a los centros de control, también facilitan los mensajes de diagnóstico, actualizaciones de firmware y supervisión de las condiciones de los medidores.

### 2.2.3.3. Red de área amplia (Wide Area Network)

Es la red encargada de conectar múltiples sistemas de distribución y actúa como un puente entre redes de menor tamaño y la red del cliente. Debe ofrecer una red de retorno para conectar los servicios de recolección de datos con las instalaciones del cliente. Dicha red de retorno puede adoptar una variedad de tecnologías (Ethernet, red celular, banda ancha) para transferir la información extraída de las redes NAN a las oficinas locales de servicios públicos. Una puerta de enlace perteneciente a la red WAN puede utilizar la conexión de banda ancha o una red basada en IP para proporcionar un acceso para que el cliente puede acceder a los datos requeridos. La privacidad, fiabilidad y seguridad de la información son los principales aspectos que se evalúan en este tipo de redes.



**Figura 2.2.** Arquitectura de red en comunicaciones para medición inteligente. J. Ekanayake and K. Liyanage, SMART GRID TECHNOLOGY AND APPLICATIONS, First edit. New Delhi, 2013.

#### **2.2.4. Normativa aplicada a las redes de medición**

Los problemas en redes de medición suelen ocurrir cuando falla la interoperabilidad entre los distintos sistemas que las componen, esto es causado por la transmisión de datos que se efectúa a través de diversas redes y sistemas de comunicación. Es por ello que la interoperabilidad se convierte en algo fundamental, permitiendo a la infraestructura y la información fluir correctamente en un sistema interoperable permitiendo el intercambio de datos sin necesidad de la intervención manual. El punto más relevante de la interoperabilidad es proporcionar un sistema con la capacidad de integrar componentes plug-and-play, esto se refiere a la capacidad de configurar los componentes y el sistema principal automáticamente comienza a operar con solo conectarlo. Aunque es un concepto sencillo en muchas situaciones puede llegar a ser complejo y poco práctico mostrar una interfaz estándar entre dos sistemas diferentes. Estas consideraciones en la interoperabilidad no solo reducen los costos de instalación y de integración sino que también definen los requisitos que un componente nuevo debe tener para conectarse al sistema existente, lo que se refleja en la sustitución sencilla de componentes y en una alta escalabilidad del sistema en caso de una mayor demanda o implementación de tecnologías más eficientes.

##### **2.2.4.1. Modelo OSI**

Existen diferentes modelos y arquitecturas referenciales para el diseño de un sistema, el primer modelo creado por la ISO en 1980 es llamado OSI (Open System Interconnection) es un estándar que tiene por objetivo conseguir interconectar sistemas de procedencia distinta para que estos pudieran intercambiar información sin ningún tipo de impedimentos debido a los protocolos con los que estos operaban

de forma propia según su fabricante. El modelo OSI está conformado por 7 capas o niveles de abstracción:

1. De aplicación
2. De presentación
3. De sesión
4. De transporte
5. De red
6. De enlace de datos
7. Capa física

Cada uno de estos niveles ejerce sus propias funciones para que en conjunto el sistema sea capaz de alcanzar su objetivo final. Es por esta abstracción que el modelo hace posible la intercomunicación en diferentes sistemas al delimitar las funciones que debe realizar cada capa de operación. Es importante aclarar que el modelo OSI no es la definición de una topología ni un modelo de red en sí mismo. Tampoco especifica ni define los protocolos que se utilizan en la comunicación, ya que estos están implementados de forma independiente a este modelo. Lo que realmente hace OSI es definir la funcionalidad de ellos para conseguir un estándar.

- **Capa de aplicación:** Ofrece la posibilidad de acceder a los servicios de las demás capas. Define los protocolos que utilizan las aplicaciones para intercambiar datos. Normalmente el usuario no interactúa con esta capa sino con programas que sirven de interfaz para ocultar el nivel de complejidad subyacente.

- **Capa de presentación:** Se encarga de representar la información funcionando como un traductor entre equipos. En esta capa se trabajan más los datos en sí que cómo se establece el enlace para que estos lleguen. Se tratan aspectos como la semántica (qué significan) y la sintaxis (formato de los datos) para garantizar la compatibilidad con los equipos del sistema. También es la encargada de cifrar los datos y comprimirlos de ser necesario.
- **Capa de sesión:** Esta capa es la que se encarga de mantener y controlar el enlace establecido entre dos computadores que están transmitiendo datos de cualquier índole. Por lo tanto, el servicio provisto por esta capa es la capacidad de asegurar que, dada una sesión establecida entre dos máquinas, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción.<sup>11</sup> En muchos casos, los servicios de la capa de sesión son parcial o totalmente prescindibles.

#### 2.2.4.2. GWAC

La integración de la automatización asociada con los recursos eléctricos es importante para respaldar una mayor eficiencia e incorporar recursos renovables, variables y vehículos eléctricos en el sistema eléctrico. Los problemas de integración que enfrenta esta comunidad son análogos a los que enfrenta la industria de la salud, los servicios de emergencia y otras comunidades complejas con muchas partes interesadas. Para resaltar este problema y fomentar la comunicación y el desarrollo de una comunidad de interoperabilidad de redes inteligentes, el GridWise Architecture Council (GWAC) creó un Marco de establecimiento de contexto de interoperabilidad. Este "modelo conceptual" ha sido útil para explicar la importancia de la alineación organizativa, además de las especificaciones de interfaz técnica e informativa para dispositivos y sistemas de redes inteligentes". Como siguiente paso para construir una comunidad sensible a la interoperabilidad,

el GWAC ha estado desarrollando un Modelo de madurez de interoperabilidad de redes inteligentes (SG IMM) tomando prestado del trabajo realizado por otros para abordar circunstancias similares. El modelo proporciona un medio para medir el estado y el progreso, analizar las brechas y priorizar los esfuerzos para mejorar la situación. El objetivo es crear una herramienta, o un conjunto de herramientas, que fomente una cultura de interoperabilidad en la comunidad de redes inteligentes. En la siguiente ilustración se hace referencia a una comparación de OSI y GWAC con sus respectivos niveles o capas cada uno:

OSI	GWAC	Función de las capas
7. Aplicación	3. Sintaxis de interoperabilidad Comprensión de los datos Estructura mensajes Intercambio entre sistemas	- Traducción de los caracteres de datos de un formato a otro. - Estructura del contenido del mensaje - Patrones de intercambio de mensajes
6. Presentación		
5. Sesión		
4. Transporte	2. Interoperabilidad de las redes Intercambio de mensajes entre el sistema a través de una variedad de redes	- Traducción de direcciones lógicas y nombres a direcciones físicas. - Transferencia transparente y fiable de datos entre sistemas. - La transferencia de datos entre el origen y el destino a través de intermediarios de la red. - Gestión de la orden de entrega de mensajes
3. Red		
2. Enlace de datos	1. Conectividad básica Mecanismo para establecer conexiones físicas y lógicas del sistema	- Acceso a los medios de hardware y conectividad eléctrica - Transferencia de datos entre nodos de red - Codificación de caracteres, transmisión, recepción, decodificación y corrección de errores
1. Física		

**Figura 2.3.** E. Hossain, Z. Han, and V. Poor, Smart Grid Communications and Networking. New York: CAMBRIDGE UNIVERSITY PRESS, 2012.

## **2.3. Redes malladas**

### **2.3.1. Definición**

Una red WiFi mallada se puede definir como una red que permite la comunicación entre nodos a través de múltiples saltos en una topología mallada [6]. Un nodo es la unidad mínima de la red, estas unidades son las encargadas de generar los enlaces entre los dispositivos usuarios que construyen la red mallada.

Por lo general estas redes poseen clientes, enrutadores y puertas de enlace. Los clientes son dispositivos electrónicos, sistemas embebidos o sensores que pueden comunicarse con otros en la red. El enrutador es un dispositivo electrónico que sirve como un intermediario entre dos o más redes para transportar los datos de una red a otra. Y las compuertas de enlace son dispositivos electrónicos que conectan la red con Internet. [7]

Cuando un nodo no puede operar, el resto de los nodos en la red WiFi mallada aún pueden comunicarse con los otros, bien sea, directa o indirectamente a través de uno o más nodos intermediarios [8]

### **2.3.2. Características**

La red de Internet como la conocemos es el ejemplo más conocido y similar a una red mallada, puesto que Internet es una red con gran cantidad de nodos en los que el mensaje es enviado desde un punto y es recibido en otro mediante un enrutado inherente a la red. Tomando este ejemplo como base se pueden delimitar algunas características de esta topología de red:



- Permite que el camino recorrido por el mensaje entre un punto y otro sea dinámico, es decir, que la ruta que toma el mensaje cambie si se requiere debido a la ocurrencia de algún evento que impidan la comunicación con un nodo intermedio específico.
- Ofrece una mayor cantidad de rutas posibles para el mensaje ya que idealmente cada nodo puede conectarse con cualquier otro directamente o a través de terceros.
- Cada nodo posee un identificador único en la red que lo diferencia de los demás nodos para que no haya errores en el direccionamiento de la información.

#### **2.3.2.1. Sobre los protocolos de enrutamiento en las redes Malladas**

Esta capacidad de elegir la ruta debe estar basada en algoritmos para determinar el camino óptimo que será recorrido por el mensaje. Este algoritmo de enrutamiento debe tomar en cuenta las distintas condiciones que puede presentar el medio de transmisión, las interferencias o ruido que pueda existir según la banda de transmisión, la posible colisión de datos en un nodo, etc, para así determinar hacia dónde debe ir el mensaje en cada nodo para llegar a su destino.

La implementación de topologías malladas en sistemas embebidos ha encontrado problemas en la necesidad de procedimientos adicionales relacionados con el enrutamiento. Hay algunos ejemplos de protocolos que soportan la red mallada sobre una red IP, por ejemplo el protocolo B.A.T.M.A.N (Better Approach To Mobile Adhoc Networking), Babel (Protocolo de distancia de vector para IPv6 y IPv4 con propiedades de convergencia rápida), HWMP (Protocolo Híbrido Inalámbrico Mallado), entre otros [7]. El problema surge en la implementación de

la pila TCP/IP en estos sistemas puesto que un sistema embebido posee recursos de cómputo limitados. Es por ello que se debe seleccionar según la aplicación el poder de cómputo necesario y la topología de la red que se requiere para cumplir con cada solución. Se describirán algunas redes WiFi malladas, con funcionamiento caracterizado según la aplicación:

Para que el mensaje tenga un destinatario debe existir una manera de identificar cada nodo de una red mallada, en el caso de la red mallada WiFi se utiliza la dirección MAC para diferenciar un nodo de otro.

#### **2.3.2.2. Direccionamiento por MAC**

En una red de computadoras, la dirección MAC es un valor único asociado a un adaptador de red. La dirección MAC también es conocida como dirección de hardware o dirección física del adaptador [7]. Las direcciones MAC se componen de doce números hexadecimales (48 bits de longitud). Por convención las direcciones MAC son escritas en uno de los dos formatos siguientes:

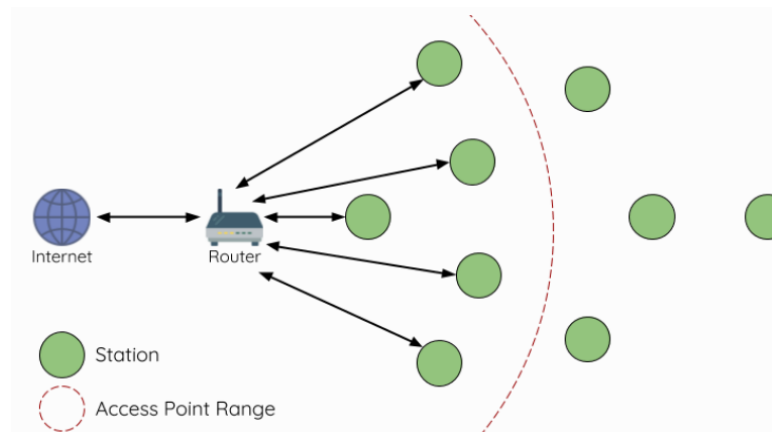
$$MM : MM : MM : SS : SS : SS \quad o \quad MM-MM-MM-SS-SS-SS \quad (2.1)$$

La primera mitad de la dirección MAC contiene el número identificador del fabricante del adaptador (i.e. 00:A0:C9:14:C8:29). Dichos identificadores están regulados por un estándar de Internet. La segunda parte de la dirección MAC representa el número de serial asignado al adaptador por el fabricante. [7]

## 2.4. La red mallada ESP-MESH

### 2.4.1. Definición

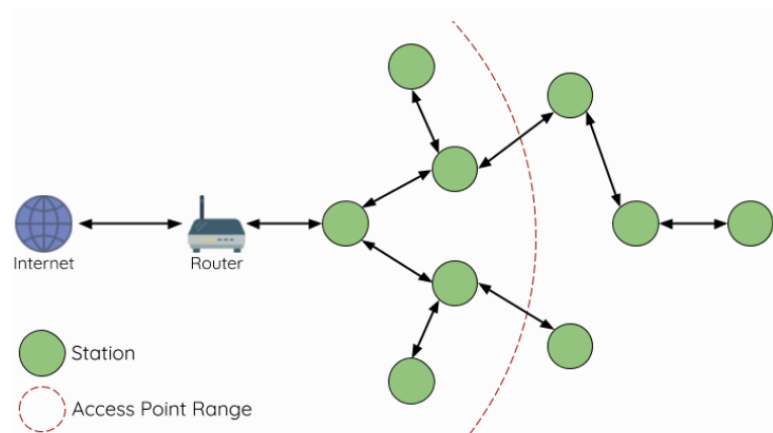
Una red Wi-Fi de infraestructura tradicional es una red punto a multipunto donde un único nodo central conocido como punto de acceso (AP) está conectado directamente a todos los demás nodos conocidos como estaciones (STA). El AP es responsable de arbitrar y reenviar las transmisiones entre las estaciones. Algunos AP también retransmiten transmisiones hacia/desde una red IP externa a través de un enrutador. Las redes Wi-Fi de infraestructura tradicional sufren la desventaja de un área de cobertura limitada debido al requisito de que cada estación debe estar dentro del alcance para conectarse directamente con el AP. Además, las redes Wi-Fi tradicionales son susceptibles de sobrecarga ya que el número máximo de estaciones permitidas en la red está limitado por la capacidad del AP.



**Figura 2.4.** Ilustración de una red Wi-Fi tradicional. Tomada desde [1]

Proporcionada por espressif la red mallada ESP-MESH, se diferencia de las redes Wi-Fi de infraestructura tradicional en que no es necesario que los nodos se

conecten a un nodo central. En cambio, los nodos pueden conectarse con nodos vecinos. Los nodos son mutuamente responsables de retransmitir las transmisiones de los demás. Esto permite que dicha red tenga un área de cobertura mucho mayor, ya que los nodos aún pueden lograr la interconectividad sin necesidad de estar dentro del alcance del nodo central. Asimismo, ESP-MESH también es menos susceptible a sobrecargas, ya que el número de nodos permitidos en la red ya no está limitado por un solo nodo central. [1]



**Figura 2.5.** Ilustración de una red Wi-Fi ESP-MESH. Tomada desde [1]

#### 2.4.2. Terminología en la ESP-MESH

- **Nodo:** Cualquier dispositivo que es o que puede ser parte de una red ESP-MESH.
- **Nodo raíz (root, root node):** Es el nodo cima de la red, este nodo se conecta a la puerta de enlace.
- **Nodo hijo (child, child node):** Es un nodo X que establece conexión con un nodo Y. Dicho nodo Y se encuentra más cerca del nodo raíz que el nodo X.
- **Nodo padre (padre, parent node):** Es un nodo Y que establece conexión

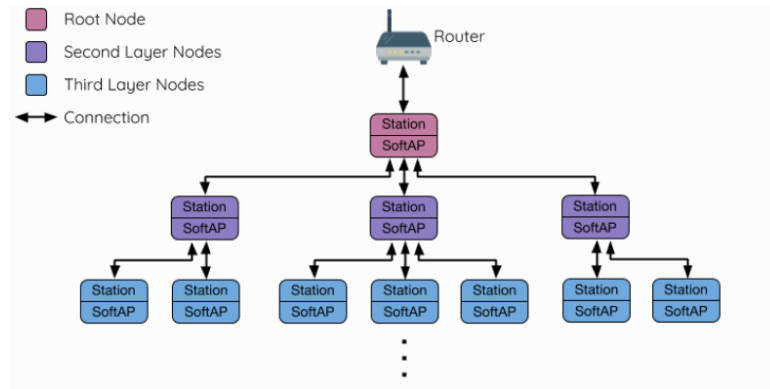
con un nodo X, el nodo X se encuentra más lejos del nodo raíz que el nodo X.

- **Nodo descendiente (descendant node):** Cualquier nodo accesible mediante la repetición del proceso padre a hijo.
- **Nodos hermanos (sibling nodes):** Nodos que comparten el mismo nodo padre.
- **Conexión:** Una asociación Wi-Fi tradicional entre un AP y una estación. Un nodo en ESP-MESH utilizará su interfaz de estación para asociarse con la interfaz softAP de otro nodo, formando así una conexión. El proceso de conexión incluye los procesos de autenticación y asociación en Wi-Fi.
- **Salto inalámbrico (wireless hop):** La parte de la ruta entre el nodo de origen y de destino para un mensaje que corresponde a una única conexión inalámbrica. Un paquete de datos que atraviesa una sola conexión se conoce como salto único, mientras que atravesar varias conexiones se conoce como salto múltiple.
- **Subred:** Subdivisión de una red ESP-MESH que consta de un nodo y todos sus nodos descendientes. Por lo tanto, la subred del nodo raíz consta de todos los nodos de una red ESP-MESH.

#### 2.4.3. Topología de árbol sobre la red ESP-MESH

ESP-MESH está construido sobre la infraestructura del protocolo Wi-Fi y se puede considerar como un protocolo de red que combina muchas redes Wi-Fi individuales en una sola WLAN. En Wi-Fi, las estaciones se limitan a una sola conexión con un AP (conexión ascendente) en cualquier momento, mientras que un AP se puede conectar simultáneamente a varias estaciones (conexiones

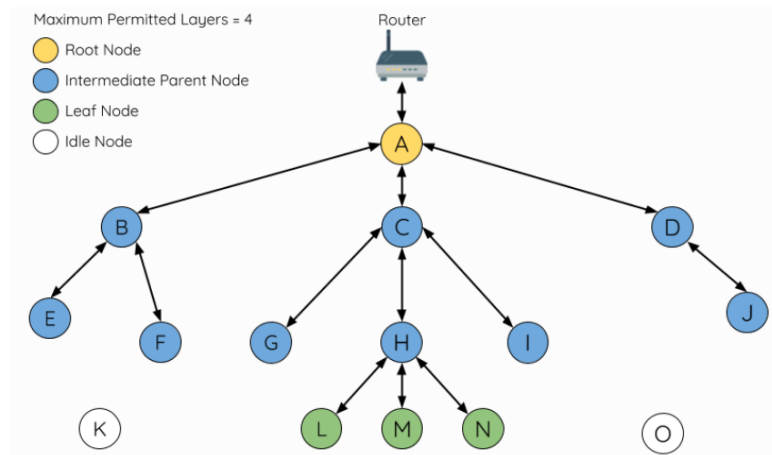
descendientes). Sin embargo, ESP-MESH permite que los nodos actúen simultáneamente como una estación y un AP. Por lo tanto, un nodo en ESP-MESH puede tener múltiples conexiones descendentes usando su interfaz softAP, mientras que simultáneamente tiene una única conexión ascendente usando su interfaz de estación. Esto, naturalmente, da como resultado una topología de red de árbol con una jerarquía de padres e hijos que consta de varias capas.



**Figura 2.6.** Topología de árbol observada en la red ESP-MESH luego de su establecimiento. Tomada de [1]

ESP-MESH es una red de múltiples saltos (multi-hop), lo que significa que los nodos pueden transmitir paquetes a otros nodos en la red a través de uno o más saltos inalámbricos. Por lo tanto, los nodos en ESP-MESH no solo transmiten sus propios paquetes, sino que también sirven como retransmisores para otros nodos. Siempre que exista una ruta entre dos nodos cualesquiera en la capa física (a través de uno o más saltos inalámbricos), cualquier par de nodos dentro de una red ESP-MESH puede comunicarse.

#### 2.4.4. Tipos de nodos en la red ESP-MESH



**Figura 2.7.** Tipos de nodos en una red ESP-MESH. Tomada desde [1]

- **Nodo raíz:** Es el nodo superior de la red y sirve como la única interfaz entre la red ESP-MESH y una red IP externa. El nodo raíz está conectado a un enrutador Wi-Fi convencional y transmite paquetes hacia / desde la red IP externa a los nodos dentro de la red ESP-MESH. Solo puede haber un nodo raíz dentro de una red ESP-MESH y la conexión ascendente del nodo raíz solo puede ser con el enrutador. Con referencia al diagrama anterior, el nodo A es el nodo raíz de la red.
- **Nodo hoja:** Es un nodo al que no se le permite tener nodos secundarios (sin conexiones descendentes). Un nodo hoja solo puede transmitir o recibir sus propios paquetes, pero no puede reenviar los paquetes de otros nodos. Si un nodo está situado en la capa máxima permitida de la red, se asignará como nodo hoja. Esto evita que el nodo forme conexiones descendentes, lo que garantiza que la red no agregue una capa adicional. Algunos nodos sin una interfaz softAP (solo estación) también se asignarán como nodos hoja debido al requisito de una interfaz softAP para cualquier conexión descendente. Con referencia al diagrama anterior, los nodos L / M / N están situados en la

capa máxima permitida de la red, por lo que se han asignado como nodos hoja.

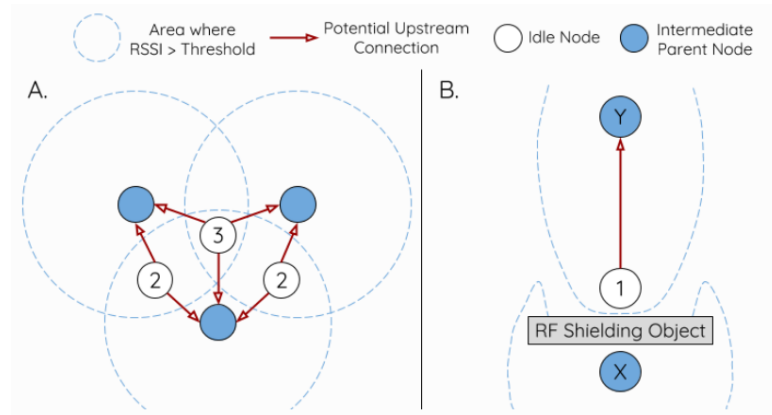
- **Nodos padres intermedios:** Los nodos conectados que no son ni el nodo raíz ni un nodo hoja son nodos padres intermedios. Un nodo intermedio debe tener una única conexión ascendente (un único nodo padre), pero puede tener de cero a varias conexiones descendentes (de cero a varios nodos hijos). Por lo tanto, un nodo padre intermedio puede transmitir y recibir paquetes, pero también reenviar paquetes recibidos desde sus conexiones ascendentes y descendentes. Haciendo referencia al diagrama anterior, los nodos B a J son nodos padres intermedios. Los nodos padres intermedios sin conexiones descendentes, como los nodos E / F / G / I / J, no son equivalentes a los nodos hoja, ya que todavía se les permite formar conexiones descendentes en el futuro.
- **Nodos ociosos:** Los nodos que aún no se han unido a la red se asignan como nodos inactivos. Los nodos inactivos intentarán formar una conexión ascendente con un nodo padre intermedio o intentarán convertirse en el nodo raíz en las circunstancias correctas. Refiriéndose al diagrama anterior, los nodos K y O son nodos inactivos.

#### 2.4.5. Señales de reconocimiento y límites de intensidad de señal (RSSI)

Cada nodo en ESP-MESH que pueda formar conexiones descendentes (es decir, tiene una interfaz AP) enviará periódicamente tramas beacon Wi-Fi. Los nodos utilizan tramas beacon para permitir que otros nodos detecten su presencia y conozcan su estado. Los nodos inactivos escucharán las tramas beacon para generar una lista de nodos padres potenciales, desde dicha lista el nodo inactivo formará una conexión ascendente.



La intensidad de la señal de una posible conexión ascendente está representada por el RSSI (Indicación de intensidad de la señal recibida) contenido en las tramas beacon del posible nodo padre. Para evitar que los nodos formen una conexión ascendente débil, ESP-MESH implementa un mecanismo de umbral RSSI para las tramas beacon. Si un nodo detecta una trama beacon con un RSSI por debajo del umbral preconfigurado, el nodo transmisor no se tendrá en cuenta para formar una conexión ascendente.

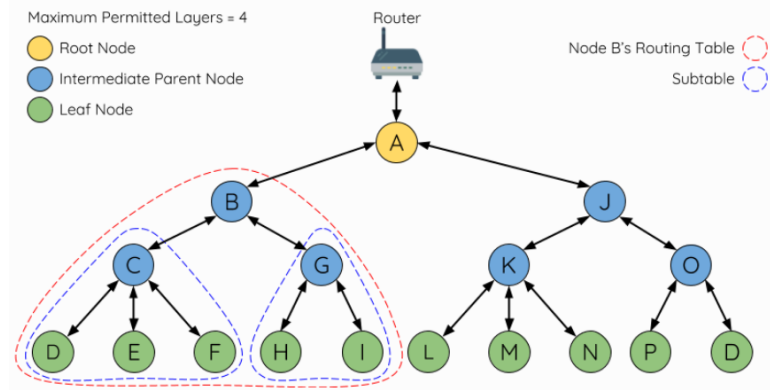


**Figura 2.8.** Ejemplo ilustrativo sobre el umbral RSSI. Tomado de [1]

En la imagen anterior 2.8 el recuadro A ilustra la cantidad de nodos padres posibles que tiene cada nodo según el RSSI que recibe. En el recuadro B de dicha imagen se observa cómo un objeto que reduce el alcance de las transmisiones del nodo X afecta en la elección del padre, pues al tener el nodo Y un mayor RSSI en el nodo acaba siendo este el elegido para la conexión ascendente a pesar de encontrarse más lejos físicamente.

#### 2.4.6. Tablas de enrutamiento

Cada nodo dentro de una red ESP-MESH mantendrá su tabla de enrutamiento individual utilizada para enrutar correctamente los paquetes ESP-MESH al nodo de destino correcto. La tabla de enrutamiento de un nodo en particular constará de las direcciones MAC de todos los nodos dentro de la subred del nodo en particular (incluida la dirección MAC del propio nodo en particular). Cada tabla de enrutamiento está dividida internamente en múltiples subtablas y cada subtabla corresponde a la subred de cada nodo hijo.



**Figura 2.9.** Ilustración de una red mallada, las tablas y subtablas de enrutamiento. Tomado de [1]

Usando el diagrama anterior como ejemplo, la tabla de enrutamiento del nodo B consistiría en las direcciones MAC de los nodos B a I (es decir, equivalente a la subred del nodo B). La tabla de enrutamiento del nodo B está dividida internamente en dos subtablas que contienen los nodos C a F y los nodos G a I (es decir, equivalentes a las subredes de los nodos C y G respectivamente).

La red utiliza las tablas de enrutamiento para determinar si un paquete debe reenviarse en sentido ascendente o descendente según las siguientes reglas:

1. Si la dirección MAC de destino del paquete está dentro de la tabla de enrutamiento del nodo actual y no es el nodo actual, seleccione la subtabla que contiene la dirección MAC de destino y reenvíe el paquete de datos en sentido descendente al nodo secundario correspondiente a la subtabla.
2. Si la dirección MAC de destino no se encuentra dentro de la tabla de enrutamiento del nodo actual, reenvíe el paquete de datos en sentido ascendente al nodo principal del nodo actual. Si lo hace repetidamente, el paquete llegará al nodo raíz, donde la tabla de enrutamiento debe contener todos los nodos dentro de la red.

## 2.5. Sistema operativo en tiempo real (RTOS)

Un sistema operativo es un conjunto de órdenes y programas de un sistema informático que gestiona los recursos de hardware y provee servicios que permiten el funcionamiento de otros programas.



**Figura 2.10.** Interacción del OS con otras capas del sistema informático.

La ventaja de utilizar sistemas operativos es que estos permiten gestionar los recursos de modo que se puedan ejecutar múltiples órdenes al mismo tiempo.

Aunque en realidad cada procesador solo está ejecutando una orden a la vez, el sistema operativo es esa capa de abstracción que simula la ejecución en paralelo de varias tareas.

### **2.5.1. FreeRTOS**

FreeRTOS es una implementación de un sistema RTOS que está diseñado para ser lo suficientemente liviano para correr en un microcontrolador. Este posee interfaces de programación que proveen planificadores en tiempo real, comunicación entre tareas, temporizadores y sincronización. Lo que quiere decir que una descripción más exacta sería tratarlo como un kernel en tiempo real. El planificador usado en el FreeRTOS, logra determinismo, proporcionando a los usuarios la capacidad de asignar prioridad a cada tarea de ejecución.

#### **2.5.1.1. Tareas**

Las tareas se implementan como funciones. Cada tarea es un pequeño programa en sí mismo. Tiene un punto de entrada, se ejecuta normalmente de forma continua en un bucle infinito y no se cierra.

#### **2.5.1.2. Colas**

Las colas proporcionan mecanismo de comunicación de una a otra tarea, de una tarea a una interrupción y de una interrupción a una tarea. La comunicación se basa en la transferencia en un dato ubicado en la cola.

#### **2.5.1.3. Semáforos**

Los semáforos son muy similares a variables booleanas con la diferencia de que estos son utilizados para bloquear o desbloquear tareas, esperar a que ocurran eventos o cambios de un valor lógico y pueden ser activados desde una interrupción.

#### **2.5.1.4. Grupos de eventos**

Los grupos de eventos son espacios de memoria de hasta 23 bits en los que cada bit representa una variable booleana que puede utilizarse como una bandera para sincronizar tareas, bloquearlas o desbloquearlas. También puede ser llamada desde interrupciones.

### **2.6. Microcontrolador ESP32**

#### **2.6.1. Descripción**

El ESP-32 es un chip con integración WiFi y Bluetooth diseñado con la tecnología de ultra bajo consumo de 40 nm. Este microcontrolador está diseñado para aplicaciones móviles, electrónicos personales, y de Internet de las cosas (IoT). Posee características de bajo consumo, incluyendo reloj de alta precisión, múltiples estados de energía, y escalamiento de consumo dinámico.

Además es una solución integrada, que ya posee WiFi, Bluetooth, junto con alrededor de 20 componentes externos. El chip incluye una interruptor de antena, acoplador de radio-frecuencia, amplificador de potencia, amplificador de recepción de bajo ruido, filtros, y módulos de administración de consumo.

Este microcontrolador posee dos núcleos CPU que pueden ser individualmente controlados, con un reloj ajustable entre 80 MHz y 120 MHz. Además, usa el sistema operativo en tiempo real freeRTOS y tensión de alimentación nominal de 3,3 V.

## **2.6.2. Características**

### **2.6.2.1. WiFi**

- 802.11 b/g/n.
- 802.11 n (hasta 150Mbps).
- WMM.
- TX/RX A-MPDU, RX A-MSDU.
- Bloque de ACK inmediato.
- Defragmentación.
- Monitorización de faro automático (Hardware TSF).
- 4 interfaces virtuales WiFi.
- Soporte simultaneo para estación, Punto de acceso y modo promiscuo.
- Diversidad de antena.

### **2.6.2.2. CPU y Memoria**

- Doble núcleo Xtensa de 32 bits, hasta 600 MIPS.
- 448 kB ROM.

- 520 kB SRAM.
- 16 kB SRAM en RTC.
- Memoria flash embebida de hasta 16MB

#### **2.6.2.3. Relojes y temporizadores**

- Oscilador interno con calibración de 8MHz, RC
- Oscilador externo de cristal desde 2 a 60MHz.
- Dos grupos de temporizadores, incluyendo 2x64-bits con un perro guardián en cada uno.
- TX/RX A-MPDU, RX A-MSDU.
- Un temporizador y un perro guardián RTC.

#### **2.6.2.4. Periféricos**

- 34 GPIO mapeables.
- ADC de 12 bits con hasta 18 canales.
- Do convertidores de digital a analógico
- 10 sensores táctiles.
- 4 SPI, 2  $I^2C$ , 3 UART
- 1 host (SD/eMMC/SDIO)
- Interfaz de MAC Ethernet con DMA dedicado y soporte IEEE 1588.
- CAN 2.0.

- Soporte simultaneo para estación, Punto de acceso y modo promiscuo.
- IR (TX/RX).
- Motor PWM.
- LED PWM hasta de 16 canales
- Sensor Hall.

#### 2.6.2.5. Seguridad

- Modo Boot seguro.
- Encriptación de flash.
- 1024-bits OTP, hasta 768-bit clientes
- Aceleración de criptografía por hardware:
  - AES.
  - SHA-2
  - RSA
  - ECC
- 4 SPI, 2  $I^2C$ , 3 UART
- 1 host (SD/eMMC/SDIO)
- Interfaz de MAC Ethernet con DMA dedicado y soporte IEEE 1588.
- CAN 2.0.
- Soporte simultaneo para estación, Punto de acceso y modo promiscuo.
- IR (TX/RX).



- Motor PWM.
- LED PWM hasta de 16 canales
- Sensor Hall.

Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

**Tabla 2.1.** Otra tabla

Tabla 2.2. Ejemplo		
Item		
Animal	Description	Price (\$)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

Un ejemplo de tabla es

**Tabla 2.3.** Ejemplo

Character	Success	Detected	Recognized
0	1.0	12	12
1	1.0	7	7
2	1.0	14	14
3	1.0	11	11
4	1.0	11	11
5	1.0	10	10
6	1.0	14	14
7	1.0	12	12
8	1.0	12	12
9	1.0	15	15
A	1.0	49	49
B	1.0	18	18
C	1.0	6	6
D	1.0	13	13
E	0.857	7	6
F	1.0	10	10
G	1.0	12	12
H	1.0	6	6
I	1.0	4	4
J	1.0	4	4
K	0.8	5	4
L	1.0	2	2
M	0.933	15	14
N	1.0	3	3
O	1.0	1	1
P	1.0	3	3
Q	0	0	0
R	1.0	1	1
S	1.0	1	1
T	1.0	1	1
U	1.0	4	4
V	1.0	3	3
W	1.0	2	2
X	1.0	4	4
Y	1.0	2	2
Z	0	0	0

## **CAPÍTULO III**

### **DEFINICIÓN Y DESCRIPCIÓN DEL HARDWARE**

#### **3.1. Definición y Descripción del Hardware**

##### **3.1.1. Definición**

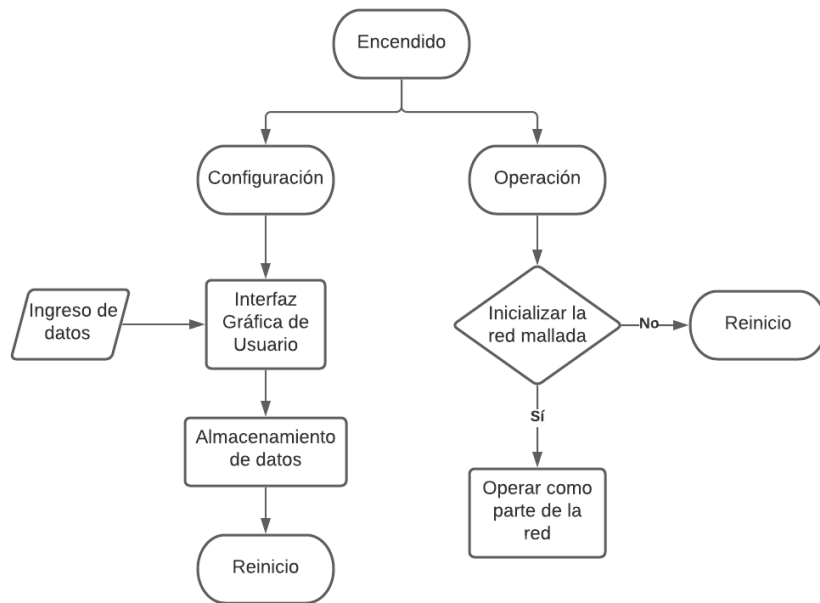
##### **3.1.2. Descripción**

## CAPÍTULO IV

### DEFINICIÓN Y DESCRIPCIÓN DEL SOFTWARE

#### 4.1. Definición del software

Para el objetivo planteado en este proyecto, se requiere el uso de un microcontrolador por lo que se debe implementar un programa o firmware para configurar y utilizar, de manera correcta, los periféricos correspondientes a las conexiones inalámbricas y el manejo de datos seriales; además es necesario un protocolo de conexión hacia un agente externo que permita enviar los datos recolectados por la infraestructura, una infraestructura que debe ser creada mediante este firmware y como método de configuración de los nodos una interfaz gráfica de usuario (GUI) que le permita a la red adaptarse de manera sencilla al lugar donde se instale.



**Figura 4.1.** Diagrama de bloques genérico del sistema

El firmware en el microcontrolador (ESP32) debe ser capaz de iniciar en dos modos: el modo de operación y el modo de configuración. El modo de inicio debe ser controlado por el usuario.

La rutina de configuración debe ofrecer al usuario una interfaz gráfica donde se pueda modificar todos los parámetros que sean necesarios para establecer la conexión con el punto de acceso local, del mismo modo se debe conectar la red con el agente externo y realizar el envío de datos; los parámetros de funcionamiento de la parte serial de cada nodo para extraer desde los medidores la información y los necesarios en la red mallada para establecer la cantidad de conexiones y el registro de nodos en la red; además del usuario y contraseña de acceso a dicha página.

Por otra parte, la rutina de configuración debe detenerse en caso de que faltase algún parámetro fundamental para el funcionamiento, en caso contrario debe ir activando etapa por etapa los procesos para: configurar los periféricos necesarios, activar la conexión inalámbrica, conectarse con el punto de acceso y registrarse en la red mallada.

Las operaciones anteriores son compartidas en todos los nodos y son necesarias para entrar en la red. Las siguientes rutinas a ejecutarse dependen de la jerarquía que posee el nodo en la red mallada y de la función que se haya configurado para el nodo en la interfaz gráfica. La jerarquía que tiene un nodo dependerá de la intensidad de señal de este respecto al punto de acceso.

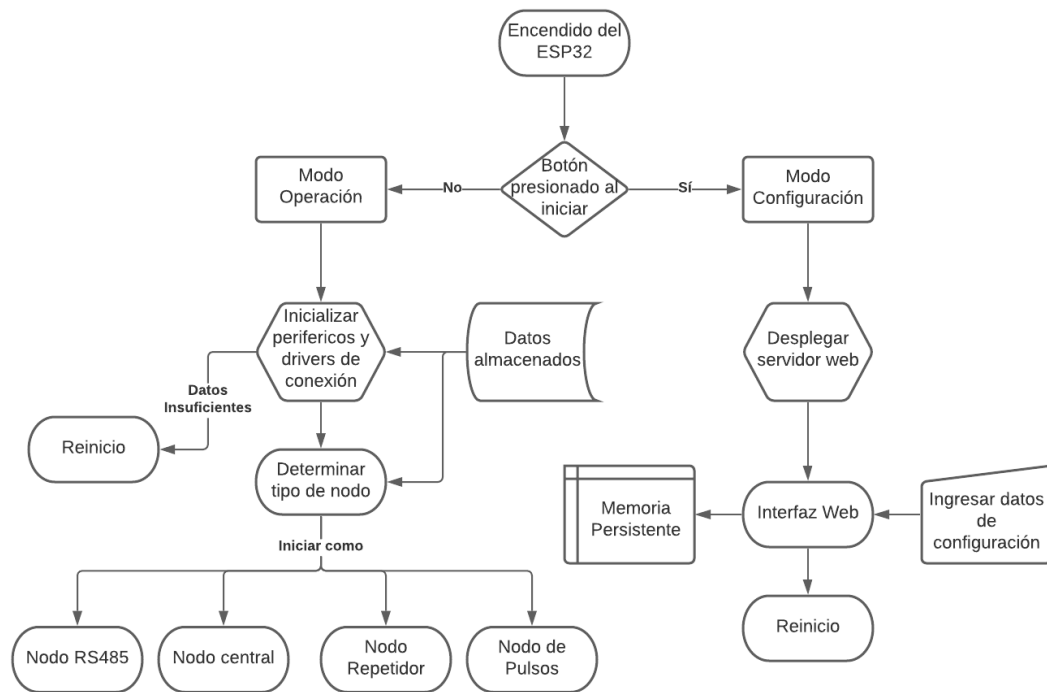
En caso de ser el nodo de mayor jerarquía, dicho nodo será el encargado de gestionar la comunicación de la red con el exterior, este nodo servirá como el enlace para enviar y recibir la información para toda la red. Además a partir de este nodo se formará el resto de la topología de red mallada.

Los demás nodos deben ser repetidores para que la señal pueda abarcar el espacio físico necesario o nodos conectados a medidores para la extracción de datos. Estas dos rutinas no deben ser excluyentes, los nodos deben tener la capacidad de ser repetidores y extraer datos al mismo tiempo.

Hay dos rutinas principales para la extracción de datos, una mediante la salida de calibración del medidor y otra que se conecta a un bus serial RS485. La rutina de extracción de datos debe estar configurada para que se utilice el método necesario según el medidor. En caso de que los datos sean extraídos por la salida

de calibración del medidor se debe diseñar una rutina para contar y almacenar los pulsos que esta genera considerando la cantidad inicial de kWh, esto por ser el registro que se desea transmitir debe estar almacenado de manera persistente y debe estar protegido contra pérdidas de alimentación. Por el contrario si en el nodo los datos son adquiridos mediante un bus serial, la persistencia de datos es realizada por el medidor, entonces la rutina en este caso solo debe encargarse de la trama serial. En ambos casos una vez adquiridos los datos se deben enviar a la red exterior cuando sean solicitados.

## 4.2. Descripción del software



**Figura 4.2.** Diagrama general de funcionamiento ilustrado.



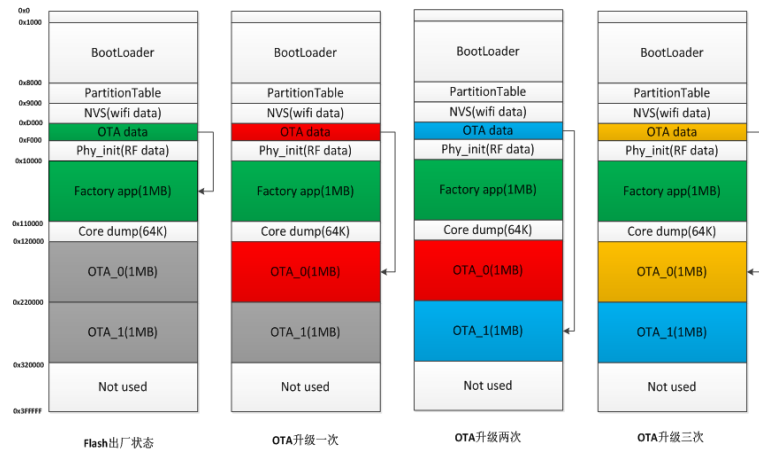
#### 4.2.1. Modo de configuración

El modo de configuración utiliza un sistema de archivos para implementar un servidor web que soporta protocolos http. Este sistema de archivos en el microcontrolador se trabaja con la API de *SPIFFS* que ofrece el fabricante, esto permite grabar en la memoria flash del microcontrolador los diferentes archivos necesarios para implementar una página web como interfaz gráfica de usuario. Sin embargo para que esto ocurra se requiere en la configuración de inicio definir un espacio en una tabla de particiones para toda esta estructura.

##### 4.2.1.1. Tablas de particiones en el ESP32

Un código cualquiera en el ESP32 puede contener múltiples aplicaciones, así como muchos tipos diferentes de datos (datos de calibración, sistemas de archivos, almacenamiento de parámetros, etc.) es por esto que las tablas de particiones se encuentran con un offset para cada código en cuestión en la memoria flash.

En cada grabado se utiliza un archivo de tabla de particiones por defecto que es definida en el menú de configuraciones también llamado *menuconfig*, hay dos tablas modelos para utilizar en este menú: *Aplicación única de fábrica* (4.1) y *Aplicación de dos definiciones*. La diferencia entre ambas tablas radica en la cantidad de particiones en la memoria flash y cómo son utilizadas: La aplicación única solo posee un código para arranque de tipo *app* en la tabla de particiones, mientras que la tabla de dos definiciones posee dos códigos para iniciar el micro y es el gestor de arranque mediante ciertas reglas el que decide cuál data de tipo *app* poner en marcha. Este último modelo de tabla es útil cuando se quiere incorporar actualizaciones inalámbricas del programa en el micro también llamadas *Over The Air updates (OTA)*.



**Figura 4.3.** Diagrama ilustrativo de la memoria flash para los modelos de tablas de partición. Columna 1: Aplicación de fábrica. Columna 2, 3, 4: Aplicaciones con arranques múltiples

En este proyecto se utiliza una aplicación única sin actualizaciones inalámbricas por lo que se utilizó el primer modelo de tabla, cuyas estructuras en archivo CSV se especifican a continuación:

**Tabla 4.1.** ESP-IDF Tabla de Particiones por defecto (Aplicación única de fábrica, sin OTA)

Name	Type	SubType	Offset	Size	Flags
nvs	data	nvs	0x9000	0x6000	
phyinit	data	phy	0xf000	0x1000	
factory	app	factory	0x10000	1M	

Con un desplazamiento de 0x10000 (64KB) en la flash, se encuentra la aplicación llamada "fábrica". El gestor de arranque ejecutará esta aplicación de forma predeterminada. También hay dos regiones de datos definidas en la tabla de particiones para almacenar la partición de la biblioteca *NVS* y los datos de inicio *PHY* (datos de calibración de periféricos).

Con la finalidad de adaptarse a la aplicación fue necesario modificar la tabla de particiones puesto que la interfaz gráfica de modo de configuración y las diferentes particiones para datos requerían de un tamaño mayor al que posee la tabla de particiones original. Para crear o modificar los tamaños en una partición basta con configurar en el *menuconfig* del proyecto que se utilizará una tabla de particiones personalizada.

**Tabla 4.2.** Tabla de particiones personalizada para la aplicación

Name	Type	SubType	Offset	Size	Flags
nvs	data	nvs	0x9000	0x6000	
phyinit	data	phy	0xf000	0x1000	
factory	app	factory	0x10000	1M	
www	data	spiffs		1M	
.	.	.		.	
.	.	.		.	
.	.	.		.	

La columna *Name* y la fila *www* de este archivo separado por comas representado en la tabla 4.2 se define el espacio de memoria donde se guardaran todos los archivos necesarios para la página web, el *Type* es data y el *SubType* es *spiffs* debido a que así se debe definir en la memoria el tipo de datos para un sistema de archivos. Y el espacio en la memoria es de 1MB como lo dice en la columna *Size*.

El servidor *HTTP* está implementado por el fabricante en los ejemplos de aplicaciones, se tomo dicha API y se adaptó a las necesidades de esta aplicación.

#### 4.2.1.2. Interfaz Gráfica de usuario

La interfaz gráfica consta de 6 pantallas, cada una con un formulario que son parámetros necesarios para conectarse a la red mallada, para su funcionamiento

como algún tipo de nodo o para la conexión con el enrutador. A la interfaz se accede mediante la dirección IP del nodo o mediante un DNS relacionado con la MAC. También se incluye la configuración de usuario y clave para proteger al acceso, de usuarios no deseados. Las pantallas por las que puede ingresar datos el usuario son:

- Inicio de sesión
- Parámetros de red mallada
- Parámetros seriales
- Configuración de red local
- Configuración de inicio de sesión

#### **4.2.2. Modo de operación**

En este modo el primer periférico en configurarse en cualquiera de los nodos es la memoria no volátil (NVS), también llamada memoria flash, que se utiliza para almacenar información persistente entre reinicios del microcontrolador. El fabricante ofrece una interfaz de programación de aplicaciones () que permite configurarla y operar fácilmente con esta.

Esto se debe a que ahí se almacenan todos los datos que se transmiten al microcontrolador mientras este se encuentra en el modo de configuración.

Se toma desde la memoria flash la información sobre el tipo de nodo (serial RS485, entrada por pulsos o repetidor) y todos los parámetros que vienen asociados a este (tasa de baudios, ID serial, kWh iniciales, etc.), así como los parámetros necesarios para registrarse en la red mallada y poder comunicarse a través de la red.

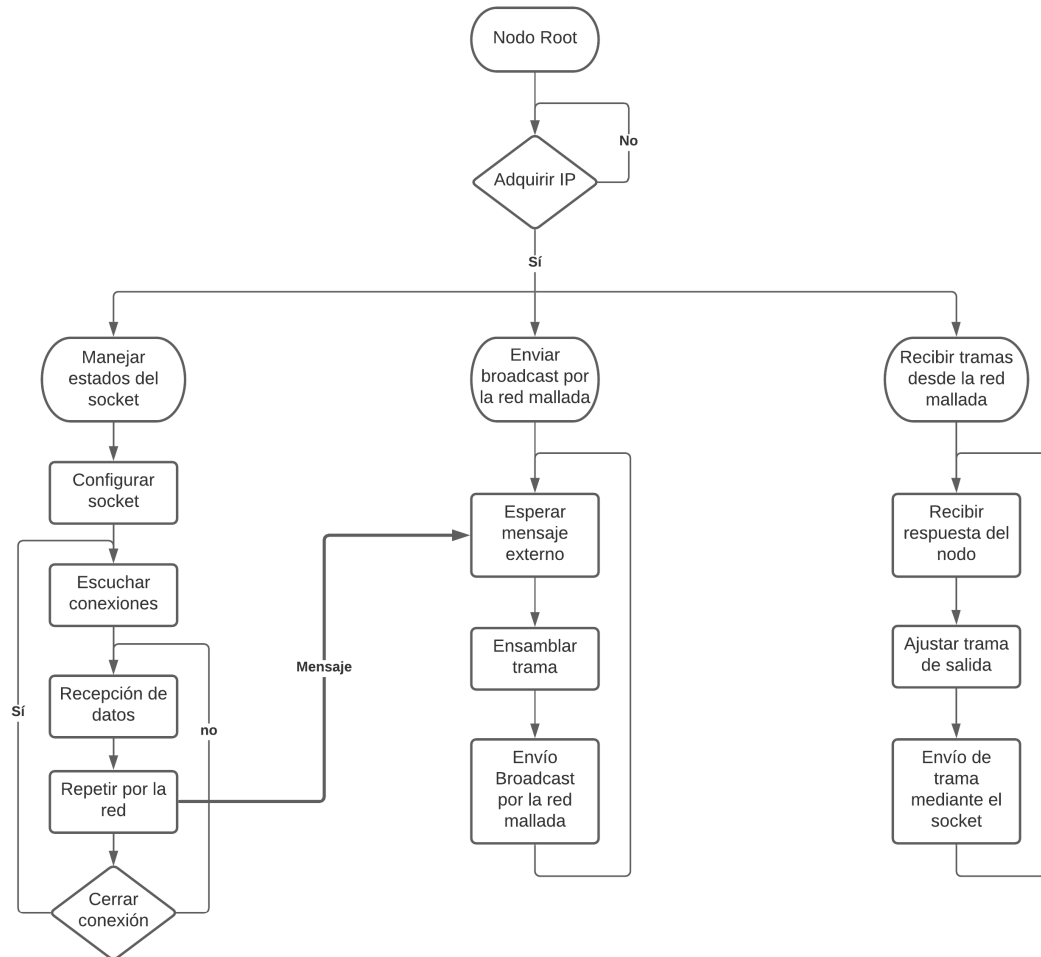
Luego se debe configurar el WiFi en cada uno de los medidores para poder unirse a la red mallada. Este periférico en el ESP32 posee también una interfaz de programación de aplicaciones para programarlo de manera más sencilla (). Mediante esta interfaz, se calibra el WiFi, se asigna un lugar para almacenar sus parámetros de calibración y se coloca el modo de operación. El ESP32 posee tres modos de operación para este periférico: estación (STA), punto de acceso (AP) y un modo híbrido (AP/STA). Este último es el que se suele utilizar en la red mallada. Una vez configurado el WiFi se procede con la configuración e inicio de la red mallada.

#### **4.2.2.1. La red mallada y el nodo central**

La topología de red mallada utilizada en el proyecto está basada en la que provee *Espressif* para ser utilizada en conjunto con el ESP32. Dicha topología es llamada ESP-MESH y el fabricante proporciona una interfaz de programación de aplicaciones () para interactuar y modificar dicha topología con la intención de que sea adaptable a las soluciones que se deseen implementar. Según el tipo de medidor que haya sido configurado en la interfaz web serán activadas unas tareas u otras, puesto que cada medidor requiere de diferentes procesos para extraer y transmitir sus datos.

El objetivo es implementar una red mallada que transmita una información representativa de la cantidad de energía consumida, tomándola a partir de medidores de energía. Esta arquitectura de red requiere de un nodo particular para comunicarse hacia redes externas, llamado nodo raíz o nodo *root*. Dicho nodo se elige mediante parámetros de calidad de señal, específicamente se tiene en cuenta la intensidad de señal (RSSI) de cada nodo con el enrutador. Considerando esto se efectúa una votación en la cual se elige qué nodo será asignado como nodo

raíz, este suele ser el nodo de mayor intensidad de señal pues esto garantiza una comunicación estable.



**Figura 4.4.** Diagrama de flujo de las tareas implementadas en el nodo central.

Una vez elegido este nodo comienza a ejecutarse el código con las tareas implementadas en la programación para él. Son tres tareas, sin embargo debe comenzar estableciendo la conexión WiFi con el enrutador y obteniendo la IP de parte del DHCP del enrutador. Esta dirección asignada es la que será utilizada para comunicarse con la red mallada desde una red exterior.

Una vez se obtiene la IP se crean en el RTOS las tareas necesarias para el nodo central:

- **Manejar estados del socket:** El sistema requiere un punto de conexión que sirva como entrada y salida para los datos requeridos. Un socket ofrece una manera de intercambiar un flujo de datos y permite enviar mensajes a otra dirección IP.

Dicho socket requiere de un proceso de configuración para el que se utiliza inicialmente un número de puerto. Este puerto será habilitado para permitir conexiones entrantes de clientes que soporten esta estructura de socket y el protocolo inherente a este bien sea TCP o UDP. Se utilizó el protocolo TCP ya que este protocolo de transporte garantiza la integridad de los datos.

Una vez establecidos el puerto y el protocolo a soportar, el socket queda en espera de conexiones entrantes. Cuando un cliente solicita conectarse y el socket no está ocupado por otro cliente, esta conexión es aceptada y se cambia el estado a *esperar mensaje*.

Al ser recibido un mensaje este es copiado a una estructura de datos llamada *cola* del sistema RTOS que funciona como un buzón de mensajes, en el que se puede acceder desde otra tarea que se encuentre esperando por dicha información, esta tarea toma la información y la retira del buzón y la utiliza para su ejecución.

El socket vuelve al estado de *esperar mensaje* pero no por un tiempo ilimitado, el socket para mantener la conexión requiere recibir mensajes cada cierto tiempo y si no se cumple con el tiempo máximo entre mensajes el socket

acabará con la conexión establecida con ese cliente y se pondrá a la espera de clientes nuevos, esto para que un cliente no bloquee permanentemente el puerto.

- **Enviar mensajes broadcast por la red mallada:** Esta tarea se encuentra a la espera del mensaje que se recibe en el socket, una vez hecho esto la tarea toma de la *cola* del sistema RTOS la información depositada. Con estos datos se procede a repetir este mensaje mediante WiFi para todos los nodos conectados a este nodo central utilizando la función de envío de mensajes internos que ofrece el API de la red mallada.

Para que esto funcione el nodo central debe tomar la dirección MAC de cada nodo desde su tabla de enrutamiento, ya que en la tabla del nodo central se encuentran todos los nodos de la red. La API del fabricante ofrece dicha opción y los mensajes que son enviados mediante WiFi van dirigidos con la dirección MAC de cada nodo. Repetidos según las reglas de la red hasta llegar a su destino.

Una vez los nodos que conforman la red mallada reciben el mensaje deben interpretarlo pero para que la información contenida en este tenga sentido, debe haber un protocolo estándar para comunicarse. Se escogió el protocolo Modbus TCP/IP como lenguaje interno de la red mallada por la poca modificación que requiere la trama para comunicarse con los medidores que tienen interfaces Modbus RS485 y por la existencia de software que emula un maestro modbus mediante un socket TCP/IP.

El protocolo Modbus TCP/IP es un protocolo sincronizado por silencios, de un solo maestro y múltiples esclavos. En este caso se utilizan los mensajes enviados por el socket para simular el maestro Modbus. Por lo que la red



diseñada se puede ver de manera análoga como un bus serial inalámbrico, que repite tal cual el mensaje hacia los demás nodos. En la trama se envía:

- **Identificador del mensaje:** Evita repeticiones de mensajes.
- **Identificador de protocolo:** Se refiere a la versión del protocolo Modbus.
- **Longitud de la trama:** Cantidad de bytes de la trama sin contar el mensaje.
- **Identificador del esclavo:** Numero en hexadecimal del esclavo al que se dirige el mensaje.
- **Código de función:** Las funciones están identificadas con códigos hexadecimales. Por ejemplo: 0x03 es la función para leer un registro estático y 0x04 es la encargada de leer un registro de entrada.
- **Datos:** Aquí se encuentran los números de registros y la información contenida en ellos.

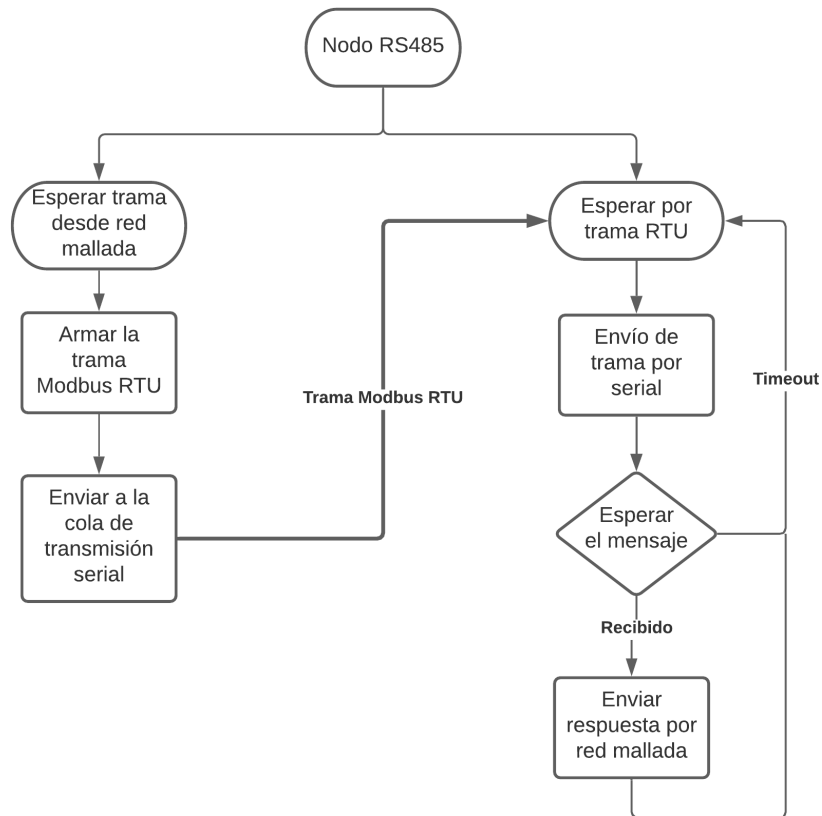
ID de Transición 2 bytes	Protocolo ID 2 bytes	Longitud 2 bytes	Unit ID 1 byte	Cód. Función 1 byte	Datos n bytes
-----------------------------	-------------------------	---------------------	-------------------	------------------------	------------------

**Figura 4.5.** Estructura de una trama Modbus TCP/IP.

- **Recibir tramas desde la red mallada y enviarlas hacia una dirección externa:** Luego de que se envía la trama mediante la red WiFi esta tarea espera la respuesta del esclavo al que iba dirigido el mensaje. Cuando la respuesta llega desde la red mallada se utilizan las partes de la trama correspondientes a la longitud para enviar el mensaje mediante el socket TCP/IP hacia el cliente conectado. El tiempo que el mensaje toma en llegar desde el maestro al esclavo correspondiente y de vuelta debe ser menor al tiempo máximo de retardo del mensaje que establezca el maestro modbus para que no hayan errores de sincronización.

#### 4.2.2.2. Nodo serial RS485

Este tipo de nodo se debe inicializar con los parámetros seriales mediante el modo de configuración en la interfaz gráfica para el usuario. El código en este nodo fue pensado para que este nodo funcione como una conversión de inalámbrico a serial, puesto que es el encargado de comunicarse con medidores que cumplen la norma RS485 y soportan el protocolo Modbus. Pero también fue pensado para transformar la trama modbus TCP/IP que llega desde la red mallada en una trama modbus RTU que es la que soportan la mayoría de medidores con modbus.



**Figura 4.6.** Diagrama de flujo de las tareas implementadas en el nodo RS485.

- **Esperar la trama desde la red mallada:** Esta tarea se encarga tomar el mensaje que llega y convertir el modbus TCP/IP en un Modbus RTU,

esta conversión consiste en quitar los 6 primeros bytes de la trama modbus TCP/IP que serían los bytes de cabecera y luego colocar una comprobación de errores mediante un CRC de 2 bytes al final de la trama.

Una vez realizado este procedimiento, el resultado es puesto en una *cola* del sistema RTOS para comunicarlo hacia la tarea encargada de manejar el bus serial del microcontrolador.

- **Esperar por trama RTU:** En esta tarea se maneja el bus de datos para la comunicación serial con los medidores. Se toman los datos recibidos en la *cola* del sistema y se envían mediante el UART del microcontrolador hacia el bus serial con las API del UART que ofrece el fabricante. A partir del envío comienza a correr un tiempo máximo de respuesta que tiene el dispositivo serial conectado al bus para devolver la información solicitada.

Todos los nodos configurados como este tipo de nodo repiten el mensaje independientemente de su contenido pero al ser protocolo modbus solo responderá el esclavo que corresponda con el id contenido en el mensaje. Si la respuesta serial llega, esta se transforma de modbus RTU a modbus TCP/IP para ser enviada al nodo central como respuesta y el nodo queda en espera del siguiente mensaje.

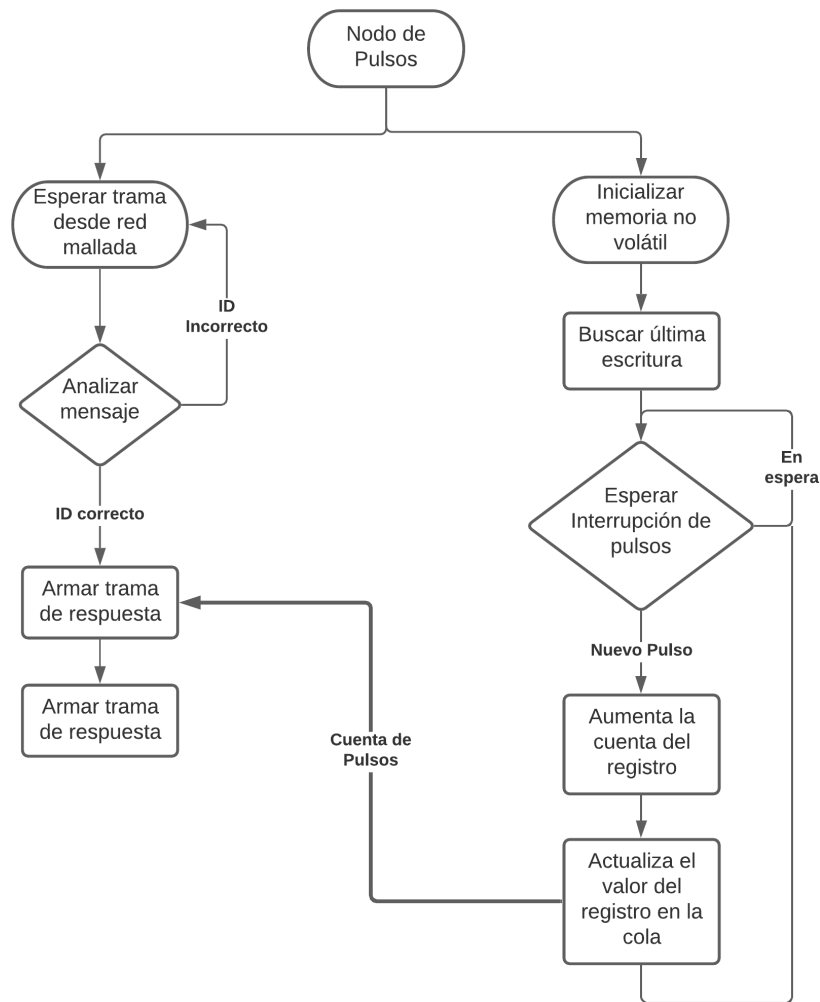
#### 4.2.2.3. Nodo de entrada por pulsos

Este nodo se inicializa mediante la interfaz web gráfica para configuración. Este nodo se creó con la finalidad de adaptarse a los medidores que se encuentran ya colocados en la mayoría de conjuntos residenciales y comerciales. Estos medidores poseen una salida binaria que es la que se utiliza para calibrarlos, en esta

salida se puede observar un cantidad de impulsos por cada kWh consumido que registra el medidor.

La idea de este nodo es tomar esos impulsos y almacenarlos para contar el total de los kWh consumidos, teniendo en consideración la condición inicial del medidor y para evitar que la cuenta se pierda por cualquier razón este nodo debe almacenar dicho valor en una memoria que sea persistente.

Además el nodo debe tener la capacidad de responder a los mensajes enviados por la red mallada y en caso de que dicho mensaje contenga la trama indicada para solicitar el valor de energía, responder con dicho registro mediante la red mallada WiFi hacia el nodo central.



**Figura 4.7.** Diagrama de flujo de las tareas implementadas en el nodo de pulsos.

- **Esperar trama desde la red mallada:** El nodo debe poder comunicarse por la red mallada, por lo que se implementó en este la comunicación mediante la red utilizando la API del fabricante. Esta tarea estará en espera hasta que llegue el mensaje proveniente de la red mallada, una vez se tenga dicho mensaje se procede a comprobar los campos respectivos al identificador del esclavo, el código de función, el número de registro y el CRC. De comprobarse que todos estos campos son correctos se procede a armar la trama de respuesta con el valor actual de la cuenta de pulsos y

se envía de vuelta al nodo central encapsulado en la trama modbus TCP/IP.

- **Inicializar memoria no volátil:** En el microcontrolador utilizado el fabricante utiliza una API llamada NVS (*Non volatile Storage*) para referirse a la memoria flash, esta interfaz de programación ofrece facilidades para tratar y almacenar datos en la memoria flash, así como la capacidad de manejar las particiones, páginas y registros que la comprenden.

Para inicializar la memoria no volátil se utiliza dicha interfaz ofrecida por el fabricante. Cada partición de la memoria no volátil requiere de un nombre, el nombre por defecto es *nvs* pero en este caso serán utilizadas varias particiones por lo que se les asignó el nombre de *app1* , *app2*, *app3*. Estos nombres son definidos en la tabla de particiones y serán utilizados en el código del programa.

**Tabla 4.3.** Tabla de particiones de la aplicación

Name	Type	SubType	Offset	Size	Flags
nvs	data	nvs	0x9000	0x6000	
phyinit	data	phy	0xf000	0x1000	
factory	app	factory	0x10000	1M	
www	data	spiffs		1M	
app1	data	nvs		64K	
app2	data	nvs		64K	
app3	data	nvs		64K	

- **Tamaños y tablas de particiones**

Las páginas de partición poseen 64kB de la memoria flash cada una, esto debido a que cuando se inicializa una partición con el API *NVS* dicha partición consume una cantidad de memoria *RAM* proporcional

al tamaño de la partición inicializada, por lo que para no consumir memoria *RAM* excesiva se escogió este tamaño de partición.

La partición a su vez se divide en estructuras llamadas páginas, que también poseen nombres, estas páginas tienen un tamaño de 4kB (por defecto) y el nombre por defecto de la página inicial es *storage*, el resto de las 15 páginas fueron nombradas con los nombres *pv0* - *pv14*.

Y las páginas están compuestas por registros, 126 para ser exactos, llamados por el fabricante como entradas. Estas llevan también un nombre y fueron asignadas como (*e0-e125*). Las entradas son capaces de almacenar diferentes tipos de datos: enteros con y sin signo, cadenas de caracteres y arreglos.

Es importante que se consideren los límites físicos de la memoria flash, un mismo espacio de memoria puede sobrescribirse hasta unas 100000 veces antes de dañarse. Por lo que la cantidad de veces que se reescribe en una entrada en el algoritmo se fijó como 50000, para preservar el estado físico de la memoria. Para el algoritmo, una entrada se encuentra llena cuando alcanza ese número de veces escrita.

- **Buscar la última escritura en la memoria**

El algoritmo para almacenar los pulsos comienza en la partición *app1* y comprueba si esta partición se encuentra 'llena' mediante una bandera lógica en la pagina principal *storage*, de ser así cierra la partición actual y busca en la siguiente hasta encontrar una que no esté llena, de no encontrar una disponible se emite una alerta mediante el serial.

Una vez en la partición se busca la última página en la que se escribió mediante un entero almacenado en la página principal *storage*. En la última página escrita se cuenta la cantidad de entradas escritas, con este numero se puede saber la última entrada que se escribió.

Se toma el valor de esa última entrada y se le suman todas las entradas, páginas y particiones anteriores además de la condición inicial de kWh para saber la cantidad de pulsos que se llevan en la cuenta y se almacena en el registro para estar disponible en caso de que se deba enviar mediante la red WiFi.

- **Nuevo pulso detectado**

Por último cuando se produce un nuevo pulso y se detecta mediante una interrupción en el pin asignado, el programa aumenta la cuenta de pulsos total que se encuentra en el registro para enviar por la red y también aumenta la cuenta en la entrada actual para que la información sea persistente.

En caso de llenarse una entrada se cambia a la siguiente hasta llegar a la entrada *e125*, luego se cambia a la siguiente página y se modifica el entero correspondiente a la última página escrita en *storage*, hasta llegar a la página *pv14*, en este caso se coloca una bandera lógica en la página *storage* para avisar que se encuentra llena la partición y se cambia a la siguiente. Luego vuelve al estado de espera de la siguiente interrupción por pulsos.

- **Cálculos de vida útil estimada según el uso de la memoria flash**



Tomando la como referencia el medidor de una casa promedio en España, al año se alcanza la cifra de 9922 kWh consumidos por lo que haciendo una estimación para la duración de la memoria tendríamos:

$$52 \frac{\text{semana}}{\text{anio}} \times 7 \frac{\text{dia}}{\text{semana}} \times 24 \frac{\text{horas}}{\text{dia}} = 8736 \frac{\text{horas}}{\text{anio}} \quad (4.1)$$

$$\frac{9922 \frac{\text{kWh}}{\text{anio}}}{8736 \frac{\text{horas}}{\text{anio}}} = 1,1358 \frac{\text{kWh}}{\text{hora}} \quad (4.2)$$

La cantidad de pulsos emitidos por kWh depende enteramente del medidor, puesto que todos tienen valores diferentes. Para un medidor de 1800  $\frac{\text{imp}}{\text{kWh}}$  que es el que se utiliza para las pruebas de sistema, se tiene que:

$$1,1358 \frac{\text{kWh}}{\text{hora}} \times 1800 \frac{\text{impulsos}}{\text{kWh}} = 2045 \frac{\text{impulsos}}{\text{hora}} \quad (4.3)$$

En el cálculo de la vida útil es necesario tomar en cuenta la cantidad de impulsos que se puede almacenar en las particiones:

$$126 \text{ entradas} \times 15 \text{ paginas} \times 50k \frac{\text{impulsos}}{\text{entrada}} = 94,5 \text{ Mimpulsos} \quad (4.4)$$

$$94,5 \text{ Mimpulsos} \times 2045 \frac{\text{impulsos}}{\text{hora}} = 46210 \text{ horas o } 5 \text{ anios} \quad (4.5)$$

Esto significa que cada partición duraría para el consumo de una casa promedio 5 años, al tener 3 particiones inicialmente se estarían garantizando según los cálculos unos 15 años de vida útil.

Esto sin considerar que el número de particiones de 64kB se puede aumentar y solo se ve limitado por el tamaño de la memoria flash del micro que en muchos casos llega a ser entre 2 y 16 MB.

#### **4.2.2.4. Nodo Repetidor**

El nodo repetidor es un nodo que ha sido configurado para ser parte de la red mallada sin tener conexiones físicas con alguna remota mediante un bus serial o con un medidor de energía mediante su salida de calibración.

Este tipo de nodos es útil para ampliar la cobertura de la red en caso de que sea necesario abarcar un espacio o tener mayor densidad de nodos en una zona específica. Este nodo no posee tareas implementadas en sus núcleos, solo posee el código necesario para ser parte de la red y repetir en ambos sentidos los mensajes que lleguen a este.

#### **4.2.2.5. Otras consideraciones**

Las tareas se crean en líneas de código contiguas de modo tal que no hay un orden de creación de ellas, pues el objetivo es que estas se ejecutan en paralelo y el RTOS sea el encargado de esperar por los datos necesarios para su ejecución sincronizada y correcta.

## **CAPÍTULO V**

### **PRUEBAS Y RESULTADOS**

## **CAPÍTULO VI**

## **CONCLUSIONES**

## CAPÍTULO VII

### RECOMENDACIONES

## Apéndice I

### TÍTULO DEL ANEXO

## Apéndice II

### TÍTULO DEL ANEXO

## Apéndice III

### TÍTULO DEL ANEXO



## REFERENCIAS

- [1] E. Systems. (2020) "fundamentos de esp-mesh". Consultado en diciembre 2020. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/mesh.html#mesh-introduction>
- [2] IEC. (2007) 313-01-35. Consultado en diciembre 2020. [Online]. Available: <http://www.electropedia.org/iev/iev.nsf/display?openform&ievref=313-01-35>
- [3] D. Londono, "Desarrollo de una guía enfocada a medidores de energía y conexiones de medidores," 2013.
- [4] M. Ruiz, "Interoperabilidad entre medidores inteligentes de energía eléctrica residencial," 2015, obtenido desde <https://dspace.ups.edu.ec/bitstream/123456789/8435/6/UPS-KT01062.pdf> , en agosto 2020.
- [5] M. Alvarado, "Servicios de medición avanzada (AMI) para redes inteligentes y su adaptabilidad en el marco de la legislación ecuatoriana," 2011.
- [6] M. Bahr, "Update on the hybrid wireless mesh protocol of IEEE 802.11s," *Siemens Corporate Technology, Information and Communications*, p. 1, 10 2016.
- [7] A. Vazquez, "Diseño e implementación de una red wifi mallada que soporte protocolo modbus para equipos de control industrial." 2020, consultado en Octubre, 2020.
- [8] M. B. Rifki Muhendra, Aditya Rinaldi, "Development of WiFi mesh

infrastructure for internet of things applications,” *Engineering Physics International Conference, EPIC 2016*, p. 331, 2016.