

PROYECTO FIN DE GRADO

TÍTULO: Desarrollo de medidor monofásico de energía eléctrica autocalibrado con monitorización a través de aplicación móvil.

AUTOR: Volker Rivera Álvarez

TITULACIÓN: Grado en Ingeniería Electrónica de Comunicaciones

TUTOR: Manuel Vázquez Rodríguez

DEPARTAMENTO: Ingeniería Telemática y Electrónica

VºBº TUTOR

Miembros del Tribunal Calificador:

PRESIDENTE: Waldo Saúl Pérez Aguiar

TUTOR: Manuel Vázquez Rodríguez

SECRETARIO: Francisco Javier Jiménez Martínez

Fecha de lectura:

Calificación:

El Secretario

Agradecimientos

A mi tutor, Manuel Vázquez, gracias por ofrecerme la posibilidad de realizar este proyecto en el que he aprendido tanto, por la paciencia, la confianza y tus esfuerzos para que todos los conceptos quedasen lo más claros posibles. Sin tu guía y tu apoyo, este trabajo no habría sido posible.

A ti, Miriam, por todo el apoyo incondicional que me has dado, las horas de tu tiempo que me has regalado y los ratitos del mío que has mejorado. Gracias por tu paciencia y por haber creído en mí siempre. Todos mis logros son también tuyos.

A mi madre, gracias por brindarme la gran oportunidad que es crecer en España, por haber trabajado incontables horas para poder darme un futuro y, sobre todo, por inculcarme el valor del esfuerzo y el sacrificio.

A mis amigos, por estar a mi lado desde el día uno, por haberme apoyado siempre y por entender cuáles eran mis prioridades. Sois la familia que he elegido.

A los Merchán, sin vuestra ayuda probablemente no hubiera tenido la posibilidad de seguir con la carrera.

A mí, por todas las horas invertidas, por todo lo que ha costado y por no haberme rendido nunca.

Resumen

Desarrollo de un medidor monofásico de energía eléctrica autocalibrado con monitorización a través de aplicación móvil.

Este PFG (Proyecto Fin de Grado) presenta el diseño, desarrollo e implementación de un medidor de energía eléctrica monofásico autocalibrado, capaz de monitorizar la calidad del consumo energético así como su impacto económico (basado en el precio voluntario para el pequeño consumidor) a través de una aplicación móvil. El proyecto surge motivado por la creciente demanda de productos dedicados al control de la cantidad y calidad de consumo energético, con el objetivo de satisfacer esta necesidad de forma precisa y con una fácil operatividad.

El sistema desarrollado se basa en el circuito integrado ADE9153A de *Analog Devices*, montado sobre la placa de evaluación EV-ADE9153ASHIELDZ. Este circuito integrado permite medir con alta precisión diversos parámetros, de entre los cuales este proyecto recogerá valor eficaz de tensión y corriente, potencia activa, reactiva y aparente, factor de potencia, frecuencia y, además, energía activa para el cálculo del histórico de consumo. Incluye la tecnología *mSure*, que garantiza mediciones precisas sin necesidad de fuentes externas.

Dado que el elemento principal del proyecto es la placa de evaluación mencionada, se ha hecho una selección y diseño del resto de componentes en base a ella. Para lograr un producto completo, se han desarrollado elementos de *hardware* que interconectan los componentes electrónicos y permiten la comunicación dentro del medidor. También se ha tenido en cuenta el aislamiento de estos componentes del exterior, ya que el medidor se conecta directamente a la red eléctrica, lo que requiere un manejo cuidadoso para garantizar la seguridad. Teniendo en cuenta todo esto, se desarrolla una PCB que permite la conexión entre placa de evaluación, microcontrolador, memoria no volátil y alimentación, y por otro lado, una carcasa que aísla los componentes internos y protege al usuario de posibles contactos con la red eléctrica.

Para la comunicación inalámbrica entre el medidor y el teléfono móvil, así como la configuración y lectura de los datos del integrado, se hace uso de la placa de desarrollo D1 Mini, equipada con el chip ESP8266 de *Espressif*. Esta configuración permite monitorizar en tiempo real la calidad del consumo de la carga conectada al medidor. La aplicación móvil ha sido desarrollada con el fin de proporcionar una interfaz intuitiva en la que se puedan visualizar mediciones instantáneas e historial de consumo. El software del medidor se ha programado utilizando Arduino, mientras que para la aplicación móvil se ha utilizado *Flutter*, un *framework* de código abierto de *Google* para crear aplicaciones multiplataforma.

Los resultados obtenidos demuestran la eficacia del medidor desarrollado a la par que su facilidad de uso. El medidor proporciona medidas precisas y fiables, y la aplicación móvil ofrece una forma intuitiva de monitorización y análisis del consumo energético. Este proyecto contribuye al desarrollo de soluciones tecnológicas que promueven la eficiencia energética y facilitan la gestión del consumo en hogares y empresas.

Abstract

Development of a single-phase self-calibrated electric energy meter monitored through a mobile application.

This FDP (Final Degree Project) presents the design, development and implementation of a self-calibrated single-phase electricity meter, capable of monitoring the quality of energy consumption as well as its economic impact (based on the voluntary price for small consumers) through a mobile application. The project is motivated by the growing demand for products dedicated to monitoring the quantity and quality of energy consumption, with the aim of satisfying this need in an accurate and easy-to-operate way.

The developed system is based on the Analog Devices ADE9153A integrated circuit, mounted on the EV-ADE9153ASHIELDZ evaluation board. This integrated circuit allows highly accurate measurement of various parameters, among which this project will collect rms voltage and current, active, reactive and apparent power, power factor, frequency and, in addition, active energy for the calculation of the consumption history. It includes mSure technology, which guarantees accurate measurements without the need for external sources.

Since the main element of the project is the aforementioned evaluation board, the rest of the components have been selected and designed based on it. In order to achieve a complete product, hardware elements have been developed that interconnect the electronic components and enable communication within the meter. The isolation of these components from the outside has also been taken into account, as the meter is connected directly to the mains, which requires careful handling to ensure safety. Taking all this into account, a PCB is developed that allows the connection between the evaluation board, microcontroller, non-volatile memory and power supply, and on the other hand, a casing that isolates the internal components and protects the user from possible contact with the mains.

The D1 Mini development board, equipped with Espressif's ESP8266 chip, is used for wireless communication between the meter and the mobile phone, as well as the configuration and reading of data from the integrated device. This configuration allows real-time monitoring of the consumption quality of the load connected to the meter. The mobile application has been developed in order to provide an intuitive interface where instantaneous measurements and consumption history can be displayed. The meter software has been programmed using Arduino, while Flutter, an open-source framework from Google for creating cross-platform applications, has been used for the mobile application.

The results obtained demonstrate the effectiveness of the developed meter as well as its ease of use. The meter provides accurate and reliable measurements, and the mobile application offers an intuitive way to monitor and analyse energy consumption. This project contributes to the development of technological solutions that promote energy efficiency and facilitate consumption management in homes and businesses.

Índice de figuras

Figura 1. Diagrama del sistema completo	15
Figura 2. Representación de la potencia en eje cartesiano	22
Figura 3. Medidor Maxcio	25
Figura 4. Medidor TP-Link Tapo P110 [9].....	25
Figura 5. Vatímetro PCE-PA 8000 [10]	26
Figura 6. Sensor de corriente del canal A [13]	29
Figura 7. Sensor del canal de tensión [13]	29
Figura 8. Configuración de los sensores de medida EV-ADE9153A-SHIELDZ [14]	30
Figura 9. Procesado de la señal en el canal de corriente A [15]	30
Figura 10. Procesado de la señal en el canal de tensión [15]	31
Figura 11. Procesado para la obtención del valor RMS [15]	31
Figura 12. Procesado para la obtención de potencia activa [15]	32
Figura 13. Procesado para la obtención de la potencia reactiva fundamental [15].....	33
Figura 14. Procesado para la obtención de la potencia aparente [15].....	33
Figura 15. Registros de acumulación de energía [15]	34
Figura 16. Signos de la potencia según tipo de carga [16]	34
Figura 17. Detección de cruce por cero [16]	35
Figura 18. Configuración para el voltaje del SPI [14]	36
Figura 19. Posición de antena recomendada por Espressif [17]	37
Figura 20. Patrón de radiación sin planos metálicos cercanos a la antena de la placa de evaluación [18]	37
Figura 21. Parámetro $S_{1,1}$ en el primer caso [18]	38
Figura 22. Patrón de radiación con posicionamiento de la antena fuera del plano metálico [18] ...	38
Figura 23. Parámetro $S_{1,1}$ en el segundo caso [18]	38
Figura 24. Patrón de radiación con posicionamiento de la antena en el centro del plano metálico [18]	39
Figura 25. Parámetro $S_{1,1}$ en el tercer caso [18]	39
Figura 26. Enrutado de la PCB en Eagle	40
Figura 27. Diseño de la PCB adaptadora	41
Figura 28. Pinout D1 Mini [21]	41
Figura 29. Pinout microSD [22]	41
Figura 30. Relación de señales ADE9153A – pinout Arduino [14]	42
Figura 31. Boceto del medidor [23]	43
Figura 32. Medidas de la placa de evaluación [23]	44
Figura 33. Medidas de la caja del medidor [23].....	44
Figura 34. Medidas del convertidor AC/DC [23]	45
Figura 35. Elementos para sujetar el convertidor AC/DC [23].....	45
Figura 36. Colocación de placa de evaluación y convertidor [23]	45
Figura 37. Resultado final del medidor	46
Figura 38. Diagrama de flujo del inicio de la aplicación Arduino.....	47

Figura 39. Diagrama de flujo de la callback asociada al Timer 1s.....	47
Figura 40. Diagrama de flujo de la callback asociada al Timer250ms	49
Figura 41. Polaridad y fase del reloj [28]	50
Figura 42. Requisitos de tiempo del bus SPI para el ADE9153A [15]	50
Figura 43. Representación del bus SPI en el ADE9153A [15].....	50
Figura 44. Método que se encarga de configurar el ADE9153A [27]	56
Figura 45. Valor con el que se configuran los registros de las líneas 58 a 69 [25]	56
Figura 46. Cambio en la configuración para acumulación de energía [25]	57
Figura 47. Constantes de conversión teóricas para la potencia [13].....	57
Figura 48. Cambios en las constantes de conversión para las medidas [25].....	57
Figura 49. Configuración [25].....	58
Figura 50. Relación entre constante de calibración, de conversión y ganancia del canal [13]	58
Figura 51. Estructura de JSON de respuesta para una hora determinada	60
Figura 52. Ejemplo de fichero precios_hoy.txt	61
Figura 53. Ejemplo JSON Lines de energía	62
Figura 54. Configuración VMware para emular macOS Sonoma	64
Figura 55. Estructura del proyecto.....	65
Figura 56. Esquema del árbol de ficheros de la carpeta lib	66
Figura 57. Diagrama de flujo de la aplicación	67
Figura 58. Diagrama superficial del procesado y uso de los mensajes MQTT.....	68
Figura 59. Ejemplo de funcionamiento de la pantalla Consumo instantáneo en iPhone	69
Figura 60. Proceso para añadir datos a lista de consumo a partir de mensaje MQTT	70
Figura 61. Conversión de lista de datos a mapa de consumo	71
Figura 62. Ejemplo de obtención de datos para gráfico a partir de mapa de consumo	71
Figura 63. Esquemático de conexionado microSD, D1 Mini y pines Arduino.....	89
Figura 64. Plano de cobre de la capa top.....	90
Figura 65. Plano de cobre de la capa bottom	90
Figura 66. Bibliotecas del proyecto Arduino	91
Figura 67. Características del procesador	92
Figura 68. Administrador de tareas - Rendimiento	92
Figura 69. Configuración del procesador de la máquina virtual.....	93
Figura 70. Fase y neutro en el medidor	94
Figura 71. Ejemplo de uso de buscapolos.....	94
Figura 72. Ejemplo de búsqueda de fase con multímetro	95
Figura 73. Configuración de multímetro.....	95
Figura 74. Portal de configuración WiFi del medidor	96
Figura 75. Conexión a red WiFi	96
Figura 76. SSID y contraseña del router.....	97
Figura 77. Icono de la aplicación.....	97
Figura 78. Estado inicial	98
Figura 79. Estado conectado.....	98
Figura 80. Interfaz de gráficos (superior).....	99

Figura 81. Interfaz de gráficos (inferior)	99
Figura 82. Interfaz tras la petición de datos	99
Figura 83. Elección de fecha a evaluar	100
Figura 84. Elección de tipo de consumo	100
Figura 85. Consumo diario en euros	101
Figura 86. Consumo diario en Wh.....	101
Figura 87. Fichero que almacena el consumo del día evaluado	101
Figura 88. Gráfico de consumo semanal por día	102
Figura 89. Consumo mensual por semana.....	102

Lista de acrónimos

AC	Alternating Current
APIs	Application Programming Interfaces
DC	Direct Current
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FP	Factor de potencia
GPIO	General Purpose Input Output
GPS	Global Positioning System
ICs	Integrated Circuits
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
JSON	JavaScript Object Notation
LSB	Less Significant Bit
M2M	Machine to machine
mDNS	Multicast Domain Name System
microSDHC	micro Secure Digital High Capacity
MSB	Most Significant Bit
NTP	Network Time Protocol
PCB	Printed Circuit Board
PIFA	Planar Inverted-F Antenna
QoS	Quality of Service
RAM	Random Access Memory
RMS	Root Mean Square
RTC	Real Time Clock
SD	Secure Digital

SPI	Serial Protocol Interface
STL	Standard Tessellation Language
UART	Universal Asynchronous Receiver Transmitter
WiFi	Wireless Fidelity

Índice de contenidos

1.	Introducción	15
1.1.	Marco y motivación del proyecto	15
1.2.	Objetivos técnicos y académicos	16
1.3.	Estructura del resto de la memoria	17
2.	Marco tecnológico	19
2.1.	Fundamento Teórico.....	19
2.1.1	Conceptos básicos de electricidad	19
2.2.	Circuitos integrados y microcontroladores.....	23
2.2.1	ADE9153A	23
2.2.2	ATMEL-M90E26.....	23
2.2.3	MSP430AFE	24
2.2.4	STPM3X	24
2.3.	Producto final.....	24
2.3.1	Maxcio medidor de consumo	24
2.3.2	TP-Link Tapo P110.....	25
2.3.3	Analizador de potencia PCE-PA 8000.....	25
3.	Especificaciones y restricciones de diseño	27
4.	Descripción de la solución propuesta	29
4.1.	Medida de energía en EV-ADE9153A-SHIELDZ	29
4.2.	Diseño de Hardware.....	35
4.2.1	PCB adaptadora.....	35
4.2.1.1	Criterios.....	36
4.2.1.2	Conexionado	41
4.2.2	Carcasa	43
4.2.2.1	Elección de la caja	43
4.2.2.2	Elección de los shuko	44
4.2.2.3	Diseño 3D	44
4.3.	Diseño de Software	46
4.3.1	Arduino.....	46
4.3.1.1	Algoritmo	46
4.3.1.2	Comunicación SPI.....	49

4.3.1.3 Configuración de red.....	51
4.3.1.4 Configuración general y uso del ADE9153A.....	55
4.3.1.5 Lectura de medidas.....	59
4.3.1.6 Procesado de datos.....	60
4.3.1.7. Estructura de datos.....	62
4.3.2 Aplicación móvil.....	62
4.3.2.1 Requisitos para el desarrollo	63
4.3.2.2 Estructura.....	65
4.3.2.3 Funcionamiento	67
5. Resultados.....	73
5.1. Prueba con carga resistiva de 200 Ω	73
5.2. Prueba con carga resistiva e inductiva: R = 200 Ω y L = 125 mH @1.8 Ω en serie	73
5.3. Prueba con carga resistiva de 50 Ω	73
5.4. Prueba con carga resistiva en inductiva: R = 50 Ω y L = 125 mH @1.8 Ω en serie	74
5.5. Prueba con carga resistiva y capacitiva: R = 50 Ω y C = 10.7 μ F en serie.....	74
6. Presupuesto	75
7. Impacto del proyecto.....	77
8. Conclusiones	79
8.1. Conclusiones	79
8.2. Trabajos futuros	79
9. Referencias.....	81
Anexo	89
Anexo A. Esquemático y gerbers de la PCB adaptadora	89
Manual de usuario	91
1. Configuración de entorno para el desarrollador	91
1.1 Arduino.....	91
1.2 Flutter.....	91
2. Instrucciones para el usuario	93
2.1 Medidor.....	93
2.1.1 Conexión del medidor a la red.....	93
2.1.2 Conexión a red WiFi	95
2.2 Aplicación.....	97
2.2.1 Pantalla de consumo instantáneo	97

2.2.2	Pantalla de gráficos	98
-------	----------------------------	----

1. Introducción

1.1. Marco y motivación del proyecto

En el contexto actual, donde la eficiencia energética está en alza, se propone como Proyecto Fin de Grado (PFG) el desarrollo de un medidor de energía eléctrica monofásica. Para este propósito, se ha optado por utilizar el circuito integrado ADE9153A de *Analog Devices*, montado sobre la EV-ADE9153ASHIELDZ, placa de evaluación desarrollada por la misma empresa. Además, se hará uso de la placa de desarrollo D1 Mini equipada con el chip de *Espressif* ESP8266, el cual permitirá establecer una conexión inalámbrica entre el medidor y un *smartphone Android* o *iOS*.

Esta configuración permite no solo monitorizar en tiempo real el consumo de energía de las cargas conectadas al medidor, sino también evaluar la calidad de dicho consumo con un máximo de un 1% de error. Adicionalmente, se implementará en la aplicación una ventana en la que se pueda ver el consumo (tanto en euros como en unidad de energía) asociado a un rango de tiempo determinado por el usuario.

Dado que actualmente los dispositivos que proporcionan esta información o bien requieren de conocimientos muy técnicos para su configuración o bien reducen sus especificaciones técnicas en pro de la facilidad de uso, en este proyecto se pretende diseñar un producto cuyo uso sea lo más simple posible sin renunciar a la calidad de las medidas. Esta solución se ciñe al diagrama de la Figura 1.

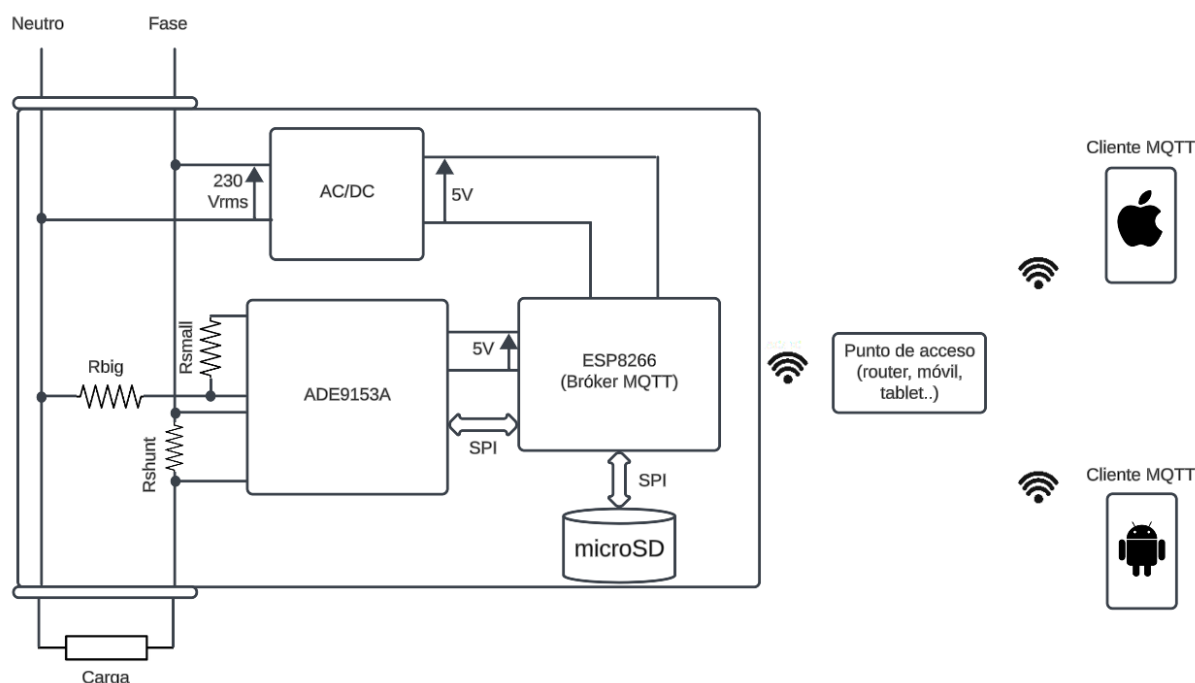


Figura 1. Diagrama del sistema completo

1.2. Objetivos técnicos y académicos

Los objetivos de este PFG son, desde el punto de vista técnico:

- Análisis de funcionamiento, configuración y uso del CI ADE9153A y de su placa de evaluación.
- Compatibilización de la placa de evaluación con el microcontrolador disponible en el departamento ESP8266 D1 Mini.
- Desarrollo de un sistema embebido basado en el chip ESP8266 que permita gestionar las medidas y comunicarse con una aplicación móvil a través de la cual se puedan visualizar las medidas instantáneas y el histórico de consumo.
- Desarrollo de la aplicación para sistemas *Android* e *iOS*.
- Concepto de producto final, por lo que en el diseño debe primar la facilidad de uso del sistema.

Desde el punto de vista académico, el proyectista adquiere las siguientes competencias y habilidades:

- Búsqueda, clasificación, análisis y redacción de documentación técnica.
- Puesta en práctica y obtención de nuevos conocimientos sobre microcontroladores ESP8266.
- Diseño, enrutamiento y envío a fábrica de una PCB.
- Efecto de radiación electromagnética en función del entorno aplicado al diseño de PCBs.
- Protocolos de comunicación y su aplicación en entornos locales y remotos.
- Diseño de arquitectura de programas para sistemas basados en microcontrolador y aplicaciones móviles.
- Adquisición de experiencia en C++, aprendizaje del lenguaje *Dart* y del *framework Flutter*.
- Análisis, configuración y uso de máquinas virtuales (*VMWare*) y emuladores (*Android* e *iOS*).
- Gestión de recursos materiales, económicos y tiempo.
- Uso de herramientas de control de versiones, APIs (*Application Programming Interfaces*) y bibliotecas.
- Inglés técnico.

1.3. Estructura del resto de la memoria

El orden establecido para esta memoria está estructurado en base a los requisitos necesarios para el funcionamiento total del proyecto, partiendo desde la concepción de este.

En primer lugar, se presentará el marco tecnológico para que el lector conozca ciertos conceptos básicos de los que se hablará en el proyecto y las herramientas existentes en la actualidad que puedan servir para realizar la misma función o una parecida a la del medidor desarrollado.

A continuación, en base al hardware del que se parte, se definen las especificaciones del proyecto, así como sus restricciones de diseño.

Posteriormente, se procederá a desarrollar la descripción del diseño, comenzando por el funcionamiento del ADE9153A dentro de la placa de evaluación que permitirá hacer las medidas. Una vez explicado el funcionamiento del integrado, se procederá a explicar el diseño de hardware (PCB y carcasa del medidor) y de software (desarrollo del programa que ejecuta el microcontrolador y el teléfono móvil).

Tras explicar la solución propuesta, se presentarán los resultados del proyecto en una serie de tablas que permitirán comparar las medidas hechas por el medidor desarrollado y el de referencia del laboratorio, así como de los valores teóricos esperados. También se evaluará el presupuesto, el impacto del proyecto y las conclusiones finales. Finalmente, dado que existe un software final enfocado al uso por parte de un consumidor, se incluye un manual de usuario.

2. Marco tecnológico

En la actualidad es común encontrar numerosos circuitos integrados y microcontroladores que permiten medir energía, así como dispositivos completamente desarrollados que brindan un producto final seguro y de calidad al consumidor. A continuación, se explicarán brevemente una serie de conceptos básicos de electricidad que facilitarán la comprensión de estos dispositivos.

2.1. Fundamento Teórico

2.1.1 Conceptos básicos de electricidad

Para poder entender el funcionamiento de un medidor, es necesario conocer una serie de conceptos básicos de teoría eléctrica [1]:

La potencia instantánea se define como el producto entre la tensión y corriente en un instante dado:

$$p(t) = v(t) \cdot i(t)$$

Ecuación 1

Si las formas de onda de v e i se repiten de forma periódica, el valor medio de la potencia se podrá calcular de la siguiente forma:

$$P_{av} = \frac{1}{T} \int_0^T p(t) dt = \frac{1}{T} \int_0^T v(t) \cdot i(t) dt$$

Ecuación 2

O lo que es lo mismo, sabiendo que por la Ley de Ohm donde R se trata de la carga resistiva $v(t) = i(t) \cdot R$:

$$P_{av} = R \cdot \frac{1}{T} \int_0^T i^2(t) dt$$

Ecuación 3

En función de la corriente media I_{rms} también se puede calcular la potencia media de la siguiente forma:

$$P_{av} = R \cdot I_{rms}^2$$

Ecuación 4

Por consiguiente, mediante igualación se obtiene la *Ecuación 5*, que define el valor *rms* (*root mean square*) de la corriente:

$$I_{rms} = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt}$$

Ecuación 5

Aplicando el mismo método a la tensión, se obtiene:

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T v^2(t) dt}$$

Ecuación 6

Dado que este medidor se enfoca en la medida de consumo de cargas conectadas a la red eléctrica, idealmente la forma de onda de la tensión será sinusoidal, quedando por tanto las siguientes ecuaciones que describen corriente y tensión:

$$v(t) = V_p \cdot \cos(\omega t) = \sqrt{2} \cdot V_{rms} \cdot \cos(\omega t)$$

Ecuación 7

$$i(t) = I_p \cdot \cos(\omega t - \phi) = \sqrt{2} \cdot I_{rms} \cdot \cos(\omega t - \phi)$$

Ecuación 8

El subíndice p corresponde a *peak*, ω a la pulsación angular y ϕ el ángulo de la fase, que en la tensión será 0 y en la corriente podrá variar en función de la carga. El desfase de la corriente respecto a la tensión será nulo cuando la impedancia sea puramente resistiva, $+\frac{\pi}{2}$ cuando sea capacitiva y $-\frac{\pi}{2}$ cuando sea inductiva.

La impedancia de tipo inductiva o capacitiva se calcula de la siguiente forma:

$$X_L = \omega L$$

Ecuación 9

$$X_C = -\frac{1}{\omega C}$$

Ecuación 10

$$Z = R + j(X_L + X_C)$$

Ecuación 11

Con esto, la corriente se puede expresar en función de la tensión y la impedancia reactiva de la siguiente forma:

$$I_L = \frac{V_L}{j\omega L} = \left(\frac{V_L}{\omega L}\right) e^{-j\pi/2}$$

Ecuación 12

$$I_C = j\omega CV_C = (\omega CV_C)e^{j\pi/2}$$

Ecuación 13

En muchos otros casos, la impedancia no será puramente resistiva ni reactiva, sino que tendrá tanto parte real como parte imaginaria, por lo que la impedancia total será la suma (compleja, fasorial) de todas las contribuciones. Por tanto:

$$Z = \sqrt{R^2 + (X_L + X_C)^2}$$

Ecuación 14

$$\text{Fase} = \phi = \arctg\left(\frac{X_L - X_C}{R}\right)$$

Ecuación 15

$$I = \frac{V \angle 0}{Z \angle \phi} = \left(\frac{V \cdot e^{j0}}{Z \cdot e^{j\phi}} \right) = \frac{V}{Z} e^{-j\phi} = I_{rms} e^{-j\phi}$$

Ecuación 16

Una vez aclarados estos conceptos, se define:

- **Potencia aparente (S):** es compleja y tiene entidad fasorial. Se trata del producto del fasor de tensión por el conjugado de fasor de la corriente. El módulo de la potencia aparente compleja es el producto de la tensión por la corriente, ambos expresados en valor eficaz. Se mide en VA.

$$S = V \cdot I^* = V e^{j0} \cdot I e^{j\phi} = V I e^{j\phi}$$

Ecuación 17

- **Potencia activa (P):** definida como la parte real de la potencia aparente. Se mide en W.

$$P = \text{Re}[S] = V \cdot I \cdot \cos\phi$$

Ecuación 18

- **Potencia aparente (Q):** definida como la parte compleja de la parte aparente. Se mide en VAR.

$$Q = \text{Im}[S] = V \cdot I \cdot \sin\phi$$

Ecuación 19

Si se representa la potencia dentro de la circunferencia de radio unidad se puede establecer la relación entre ellas de la siguiente forma:

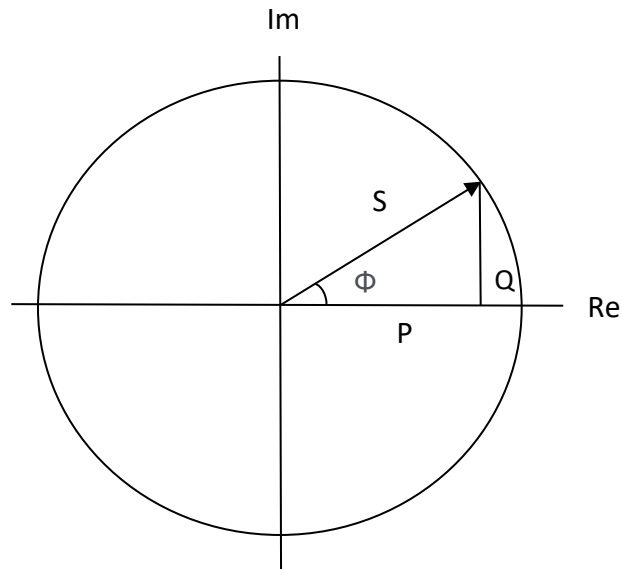


Figura 2. Representación de la potencia en eje cartesiano

$$S^2 = P^2 + Q^2$$

Ecuación 20

$$\text{Factor de potencia} = \frac{P}{S} = \frac{P}{V \cdot I} = \cos \phi$$

Ecuación 21

Este concepto de factor de potencia es importante ya que la potencia que realmente realiza un trabajo es la activa, por lo que esto indicará cómo de eficiente es el consumo de un sistema en relación a la potencia total que requiere. Idealmente tendrá un valor de 1.

Ha de mencionarse que, para cargas electrónicas conectadas a rectificadores con filtro de condensador, la forma de onda de la corriente puede desviarse significativamente de una onda sinusoidal. Esta forma de onda quedará descrita como la suma en serie de Fourier de todas sus componentes mediante la ecuación:

$$i_s(t) = i_{s1}(t) + \sum_{h \neq 1} i_{sh}(t)$$

Ecuación 22

Donde i_{s1} es la componente fundamental e i_{sh} la componente en el armónico con frecuencia $f_h = hf_1$. Su valor rms atenderá a la ecuación:

$$I_s = \sqrt{I_{s1}^2 + \sum_{h \neq 1} I_{sh}^2}$$

Ecuación 23

En este caso, el factor de potencia se calculará mediante la ecuación:

$$PF = \frac{V_s \cdot I_{s1} \cdot \cos\phi_1}{V_s \cdot I_s} = \frac{I_{s1}}{I_s} \cos\phi_1$$

Ecuación 24

Donde V_s representa el valor rms de la tensión, I_s el valor rms de la corriente total, I_{s1} el valor rms del primer armónico y ϕ_1 su desfase respecto a la fase de la tensión.

Por último, se define la energía como la integral de la potencia definida durante un período de tiempo. Dado que es dependiente de la potencia, también se tiene energía aparente, reactiva, y activa, siendo esta última la que utilizan las compañías eléctricas para la facturación al pequeño consumidor.

2.2. Circuitos integrados y microcontroladores

En el ámbito de la medición de energía, existen diferentes circuitos integrados (ICs) y microcontroladores diseñados para este propósito. Estos componentes son esenciales pues el resto del diseño viene determinado por esta pieza central. Siguiendo el análisis de [2] encontramos:

2.2.1 ADE9153A

Dispositivo de medición de energía desarrollado por *Analog Devices*. Este integrado permite medir energía y potencia aparente, activa y reactiva; factor de potencia, frecuencia y valores instantáneos y eficaces de tensión y corriente en sistemas monofásicos.

Cuenta también con la tecnología de autocalibración *mSure*, que permite que el medidor calibre automáticamente los canales de corriente y voltaje sin usar ninguna otra fuente de referencia. Esta tecnología asegura la certificación de Clase 1 y Clase 2 definida por la IEC.

Por último, el dispositivo ADE9153A es compatible con protocolos de comunicación como SPI o UART y dispone de una placa de evaluación que facilita la conexión con *Arduino Uno*, *Arduino Zero* o ESP8266. [3]

2.2.2 ATMEL-M90E26

Se trata de un chip medidor de energía monofásico de alta precisión que también cumple con los estándares IEC. Ofrece medición de energía activa y reactiva con precisión de 0.1% y 0.2% respectivamente, en un rango dinámico de 5000:1. También permite medir el valor eficaz de corriente y tensión, potencia activa, reactiva y aparente, frecuencia, factor de potencia y ángulo de fase. Es compatible con los protocolos de comunicación SPI y UART. Al igual que sucede con el dispositivo ADE9153A, la frecuencia, ángulo de fase y factor de potencia cuentan con calibración automática, sin embargo, el chip ATMEL-M90E26 necesita realizar una calibración externa para medir voltaje, corriente y energía. [4]

2.2.3 MSP430AFE

Es un microcontrolador de ultra bajo consumo desarrollado por *Texas Instruments* y diseñado para aplicaciones de medición portátiles. Puede medir corriente, voltaje, energía activa, reactiva y aparente, así como factor de potencia en sistemas de baja potencia monofásicos o trifásicos. Cuenta con interfaces SPI y UART para transmitir los datos a otros dispositivos. Otra característica importante es que cuenta con tres ADC sigma-delta independientes, dos para los dos canales de corriente y otro para el de voltaje, lo cual se traduce en una gran capacidad de resolución. La calibración, en este caso, se realiza mediante USB ya que es necesario un *software* específico y un programa que establece el fabricante. [5]

2.2.4 STPM3X

Se trata de una gama de productos de la compañía *STMicroelectronics* diseñados para la medición precisa de energía. Ofrecen alta precisión en la medición de potencia activa y reactiva, proporcionan formas de onda de voltaje y corriente instantánea, calculan el valor eficaz y provee funciones de detección de sobretensión y sobrecorriente. Además, cuenta con una interfaz serie UART y SPI para la comunicación y configuración, y utiliza ADC sigma-delta de alta resolución. STPM3X tampoco puede autocalibrarse, requiere un voltaje nominal conocido y una corriente para poder llevarlo a cabo. Además, el proceso de calibración se realiza a través de *software* modificando los registros de calibración del dispositivo en función de las medidas realizadas con los valores nominales aplicados. [6]

2.3. Producto final

Se entiende por producto final todo aquel medidor que pueda encontrarse tanto en una tienda local como en internet y cuyo uso esté dirigido al consumidor final, facilitando su uso en gran medida. Dependiendo de las necesidades, se pueden encontrar de diferentes tipos, siendo los más parecidos al desarrollado en este PFG los siguientes:

2.3.1 Maxcio medidor de consumo

Cuenta con una precisión de medición de $\pm 2\%$ sin incluir ninguna especificación más al respecto, su temperatura de funcionamiento se encuentra en el rango de 0-40°C. Soporta una corriente máxima de 16 A y mide valor eficaz de corriente y tensión, así como la potencia y energía real consumida. Adicionalmente, se puede incluir el precio de la energía para que el medidor realice el cálculo del consumo en euros. No se menciona información alguna acerca de la calibración del medidor. [7]



Figura 3. Medidor Maxcio

2.3.2 TP-Link Tapo P110

En cuanto a medidores con aplicación de monitorización, TP-Link Tapo P110 es el más vendido. Dispone de protocolo de red IEEE 802.11b/g/n, una aplicación móvil disponible para versiones de *Android* superiores a 4.4 y de *iOS* superiores a 9.0. Si bien la aplicación permite monitorear potencia y energía activa y programar encendido y apagado, no se incluyen especificaciones acerca de la precisión de las medidas, únicamente se indica que debe estar conectado a una red AC de 220-240 V a 50/60 Hz a un máximo de 16 A para un correcto funcionamiento. Tampoco se menciona información alguna acerca de la calibración. [8]



Figura 4. Medidor TP-Link Tapo P110 [9]

2.3.3 Analizador de potencia PCE-PA 8000

Este vatímetro de PCE Instruments permite registrar la potencia en circuitos monofásicos y trifásicos. Almacena los datos en una tarjeta SD en formato de archivo Excel (.xlsx), permitiendo realizar un análisis posterior detallado. Tiene una pantalla de 3.7", mide corrientes de hasta 1200 A y voltajes de hasta 600 V. También ofrece medición de energía y potencia activa, aparente y reactiva, y determina el factor de potencia junto con el ángulo de fase. La precisión en la medida de tensión AC es de $\pm 0.5\%$, en corriente AC $\pm 0.5\%$, en potencia es de $\pm 1\%$ y en la medida de energía es de $\pm 2\%$. En cuanto a la calibración se encuentra referenciado como accesorio del dispositivo, teniendo que hacerlo PCE Instruments. [10]



Figura 5. Vatímetro PCE-PA 8000 [10]

3. Especificaciones y restricciones de diseño

En vista del creciente interés por mejorar la eficiencia energética de un sistema, se ha decidido desarrollar una herramienta que permita medir la calidad de esta de forma muy precisa, así como el consumo realizado en un determinado lapso de tiempo por medio de una aplicación móvil, la placa de evaluación EVA-ADE9152A-SHIELDZ y la placa de desarrollo D1 Mini equipada con el chip de *Espressif* ESP8266 v3.0.1.

Este sistema cuenta con las siguientes especificaciones:

- Autocalibración *mSure*, con precisión de Clase 1 (error máximo de $\pm 1\%$) garantizada. [11]
- Uso de *Rshunt* como sensor de corriente.
- Capacidad para medir una corriente nominal de 5 Arms, corriente máxima de 10 Arms y tensión eficaz máxima de 240 V.
- Estándares de medidas de energía activa: IEC 62053-21, IEC 62053-22, EN50470-3, OIML R46 y ANSI C12.20.
- Estándares de medidas de energía reactiva: IEC 62053-23 e IEC 62053-24.
- Uso de tarjetas de memoria microSDHC de hasta 32 GB en formato FAT16 y FAT32 para el almacenamiento de los ficheros que se crearán durante la ejecución del programa. Utiliza el protocolo SPI para su comunicación con la ESP8266. Se guardarán datos de medidas relativos a seis meses.
- Es necesario el uso de la placa de desarrollo ESP8266 D1 Mini pues la PCB diseñada está dimensionada específicamente para esta, al igual que el código del programa de Arduino.
- Es necesario el uso de una red WiFi en la banda de 2.4 GHz a la que deban estar conectados tanto el medidor como el dispositivo móvil para poder ejecutar la comunicación entre ellos. Es posible configurar la red en forma de estrella, siendo el *router* el elemento común entre medidor y teléfono móvil, o bien se puede establecer un punto de acceso móvil y conectar el medidor, quedando conectados de forma directa.
- La puerta de enlace para configurar la primera conexión es *192.168.4.1* y será posible acceder a ella tras conectarse al punto de acceso *AutoconnectAP*. Esto permitirá configurar la red WiFi a la que se conectará el medidor, siendo automáticas las posteriores conexiones al mismo punto de acceso.
- Uso de protocolo mDNS para una conexión automática entre bróker y cliente sin la necesidad de introducir manualmente ninguna dirección IP.
- La alimentación del medidor se hará por medio de un convertidor AC/DC incluido dentro del mismo aparato.
- Se ha de colocar adecuadamente la fase y el neutro del medidor a los de la toma de red.
- La comunicación entre medidor y aplicación se llevará a cabo mediante protocolo MQTT 3.1.
- La actualización de la pantalla de medida de potencia instantánea se ejecuta cada segundo cuando la aplicación está conectada al medidor.

- La actualización de los datos de energía medidos es asíncrona pues es el usuario quien determina el instante en el que sucede, sin embargo, cada 250 milisegundos se evalúa si se ha hecho una solicitud.
- El medidor mantiene una sincronización automática con el servidor NTP para llevar un registro lo más exacto posible del instante en el que se hace cada medida, también actualiza diariamente de manera automática la tarifa regulada PVPC del mercado eléctrico español. Por todo esto, es completamente necesario que el medidor se encuentre conectado a una red WiFi con acceso a internet, pues de otra forma no funcionará de la manera esperada.
- Android 5.0 o superior.

En relación con las anteriores especificaciones, por el hecho de usar dichos componentes tenemos las siguientes limitaciones:

- Uso exclusivo del microcontrolador ESP8266 debido a la elección de las bibliotecas usadas durante la codificación del *sketch* de *Arduino*.
- Uso necesario de la placa de evaluación EV-ADE9153A-SHIELDZ para la realización de las medidas.
- Conexión entre la aplicación y medidor limitada a que ambos deben estar conectados al mismo punto de acceso a red, o bien el medidor conectado a un punto de acceso proporcionado por el dispositivo en el que se ejecuta la aplicación.
- Incompatibilidad con redes WiFi 5 GHz debido a las limitaciones del microcontrolador.
- En relación al bróker MQTT, debido a las características de la biblioteca *uMQTTBroker* [12] encontramos las siguientes limitaciones:
 - No soporta niveles QoS diferentes a 0: esto se traduce en que pese a ofrecer la mejor eficiencia en el envío de datos, tiene como contrapartida la ausencia de ACK o garantía de entrega de dichos mensajes.
 - Hasta cinco conexiones: esta limitación viene definida por el uso lwIP 2.0, el cual fue actualizado en las últimas versiones de firmware de la ESP8266. Una vez hecha la quinta desconexión el medidor se reinicia, pues de otra forma no será posible realizar una sexta conexión. En caso de usar lwIP 1.4, será necesaria la adaptación del resto de bibliotecas a una versión más antigua del firmware, en tal caso el número máximo de clientes conectados será ocho debido a las limitaciones de memoria del microcontrolador.
 - No soporta *non-clear sessions*: se entiende por este tipo de sesión aquellas en las que los mensajes no entregados se conservan para un cliente incluso después de que se desconecte. Se utilizarán, por tanto, *clear sessions*, de esta forma si el cliente se desconecta intencionalmente o no, cualquier mensaje no entregado se descarta.
- Por último, tampoco habrá soporte para TLS, por lo que las comunicaciones entre el bróker y el cliente no estarán cifradas.
- Se consultan únicamente precios de electricidad peninsulares, Canarias y Baleares.

4. Descripción de la solución propuesta

4.1. Medida de energía en EV-ADE9153A-SHIELDZ

En primer lugar, tanto corriente como tensión han de medirse individualmente.

Por parte de la corriente, si bien existen métodos que hacen uso del efecto *Hall* (medidor de referencia Fluke 43B), de un transformador de corriente (canal de corriente B del ADE9153A), la placa de evaluación únicamente hace uso del canal de corriente A por lo que se utiliza como sensor una resistencia de tipo *shunt* de 1 mΩ. Por tanto, dado que se tendrá una resistencia de valor conocido y el valor de la caída de tensión en sus bornas, mediante la Ley de Ohm se podrá calcular la corriente.

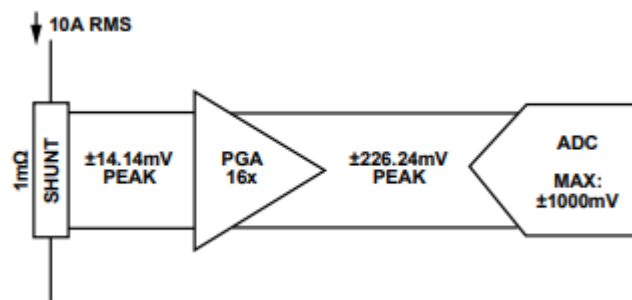


Figura 6. Sensor de corriente del canal A [13]

En cuanto a la tensión, se hace uso de un divisor de voltaje de la siguiente forma:

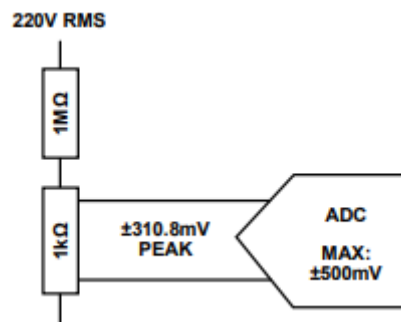


Figura 7. Sensor del canal de tensión [13]

Queda, por tanto, el siguiente conexionado entre los sensores y el integrado:

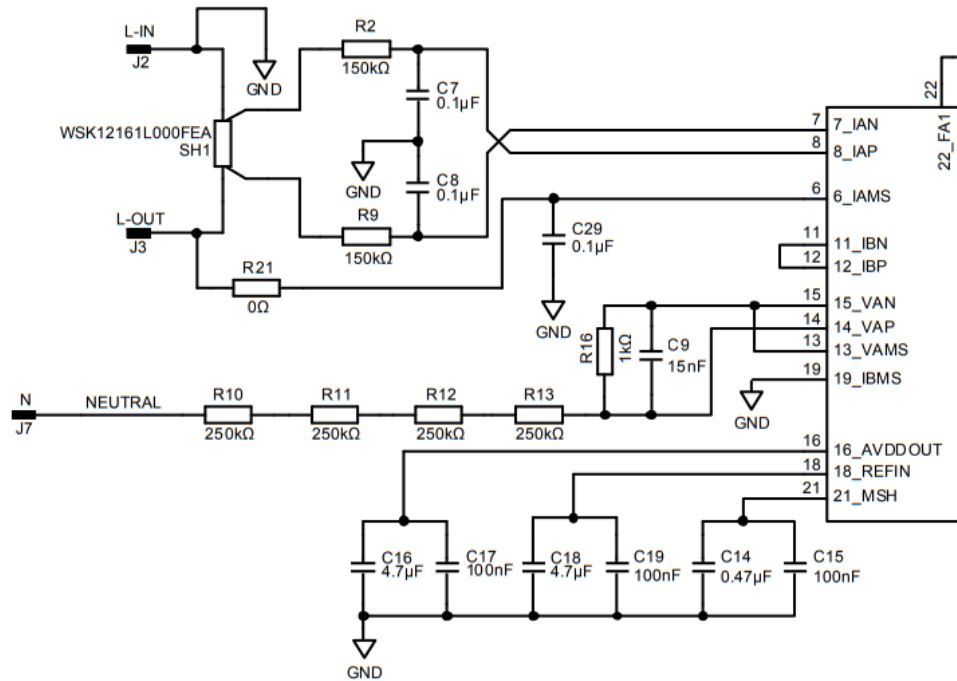


Figura 8. Configuración de los sensores de medida EV-ADE9153A-SHIELDZ [14]

Internamente, el canal de corriente se ciñe al siguiente diagrama:

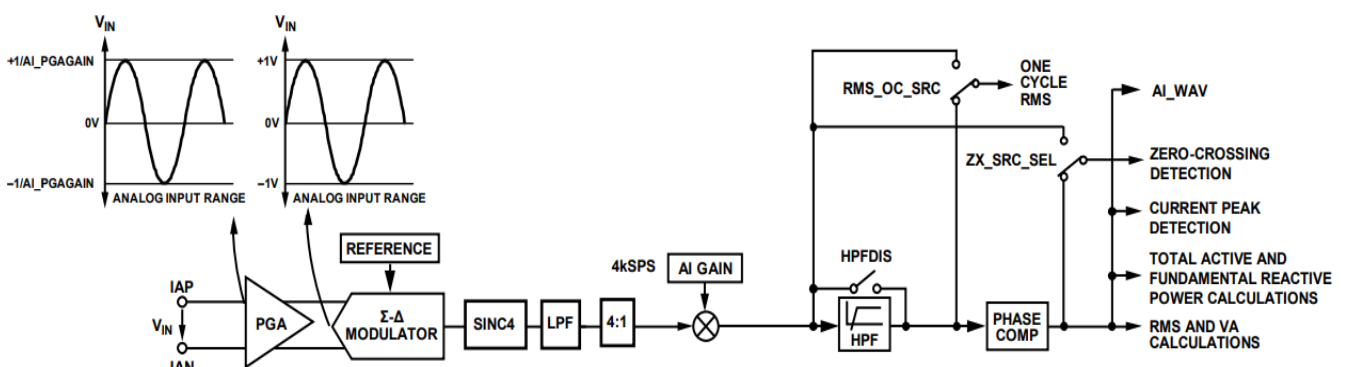


Figura 9. Procesado de la señal en el canal de corriente A [15]

Por su parte, el de tensión:

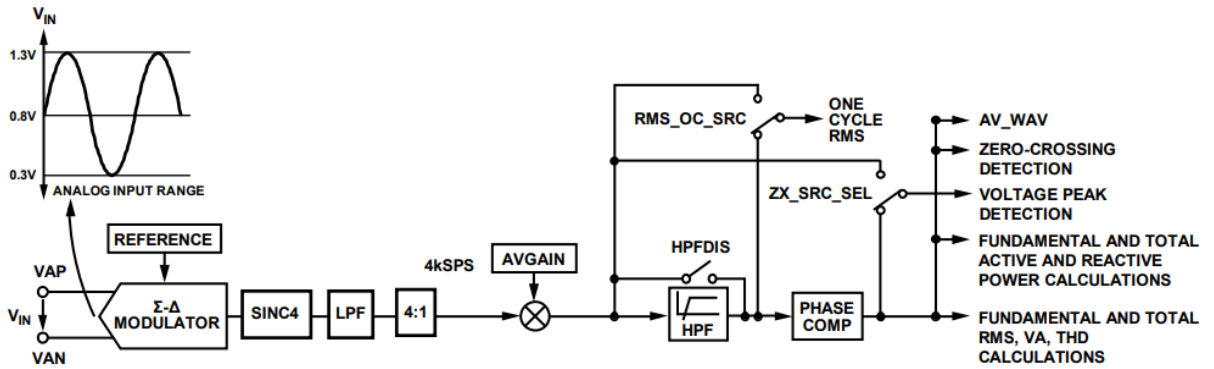


Figura 10. Procesado de la señal en el canal de tensión [15]

Como se puede ver, en ambos casos se busca un rango de escala cercano a $\pm 1V$ a la entrada del ADC. En el caso de la corriente es necesario amplificar la tensión en bornas del shunt con un amplificador de ganancia programable, en el caso del canal de tensión no.

Tras el muestreo se procede a filtrar digitalmente la señal para finalmente obtener AI_WAV y AV_WAV, es decir, las formas de onda de la corriente y la tensión en el canal A a una frecuencia de muestreo de 4 kSPS.

Una vez obtenidas estas formas de onda, en función de la magnitud a medir, se seguirá un proceso u otro:

1. **Valores eficaces (xRMS):** su valor se actualiza cada 4kSPS y el diagrama que describe su obtención es el siguiente:

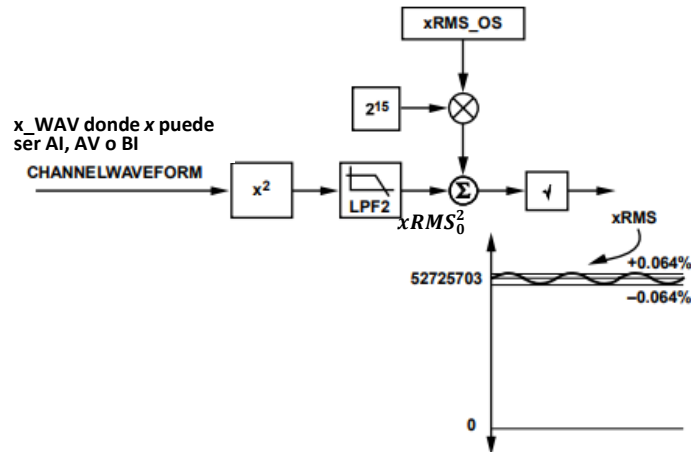


Figura 11. Procesado para la obtención del valor RMS [15]

Sabiendo que el valor *rms* se calcula de la siguiente forma:

$$xRMS_0 = \sqrt{\frac{1}{T} \int_0^T x_{WAV}^2 dt}$$

Ecuación 25

$$xRMS_0^2 = \frac{1}{T} \int_0^T x_{WAV}^2 dt$$

Ecuación 26

Y que, por otro lado, el valor medio de una señal se puede calcular mediante la siguiente ecuación:

$$\langle x_{WAV} \rangle = x_{WAV_{DC}} = \frac{1}{T} \int_0^T x_{WAV} dt$$

Ecuación 27

Se puede medir $xRMS_0^2$ como el valor medio del cuadrado de la señal. Es por ello que, en primer lugar, se eleva la señal al cuadrado y después se pasa por un filtro paso bajo, el cual atenuará las señales cuyas frecuencias queden por encima de 50 o 60 Hz que, tras elevar al cuadrado son tanto la fundamental como sus armónicos.

Con el fin de ajustar el offset, se sigue la siguiente ecuación para obtener el *rms* final:

$$xRMS = \sqrt{xRMS_0^2 + 2^{15} \cdot xRMOS_OS}$$

Ecuación 28

Donde $xRMS_0$ representa el valor inicial del registro $xRMS$ antes de la calibración de *offset*, $xRMOS_OS$ el valor del *offset* y x representa AI (canal de corriente A), BI (canal de corriente B) o AV (canal de tensión), en función del canal utilizado. El filtro LPF2 se encarga de atenuar los armónicos de la frecuencia fundamental (50 Hz) al menos 64 dB a escala completa.

2. **Potencia activa total (AWATT)**, se actualiza cada 4 kSPS y se calcula a partir del siguiente diagrama y ecuación:

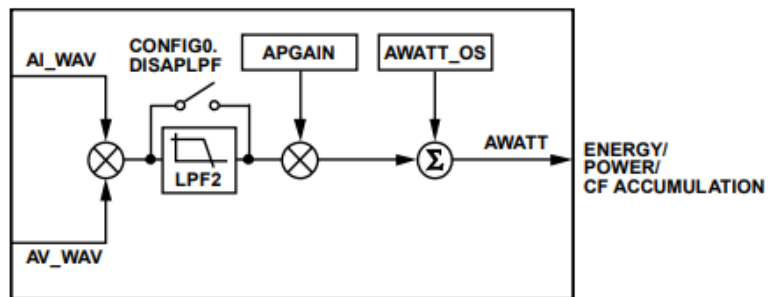


Figura 12. Procesado para la obtención de potencia activa [15]

$$AWATT = \left(1 + \frac{APGAIN}{2^{27}}\right) \cdot AWATT_0 + AWATT_OS$$

Ecuación 29

APGAIN representa una ganancia común para todos los tipos de potencia, AWATT₀ el valor inicial de AWATT antes de la calibración de *offset* y AWATT_OS el valor de *offset*. LPF2 se encarga de eliminar los armónicos de la frecuencia fundamental.

3. **Potencia reactiva fundamental (AFVAR):** se actualiza cada 4kSPS. El canal de corriente AI_WAV se desfasa 90° en la fundamental, después se multiplica por AV_WAV y se filtra con LPF2 por defecto. Después se corrige la ganancia y el *offset*:

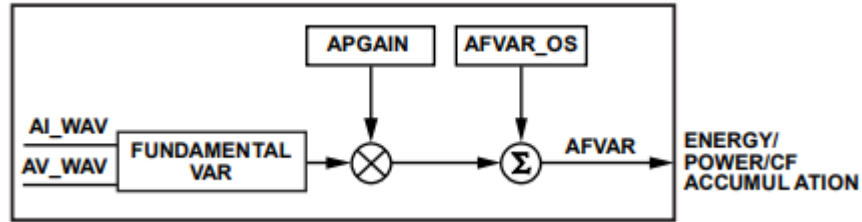


Figura 13. Procesado para la obtención de la potencia reactiva fundamental [15]

$$AFVAR = \left(1 + \frac{APGAIN}{2^{27}}\right) \cdot AFVAR_0 + AFVAR_OS$$

Ecuación 30

4. **Potencia total aparente (AVA):** se actualiza cada 4 kSPS, LPF2 está activado por defecto y para su cálculo se puede utilizar la tensión AVRMS o VNOM, esta última es configurable y se puede activar cuando falta la otra.

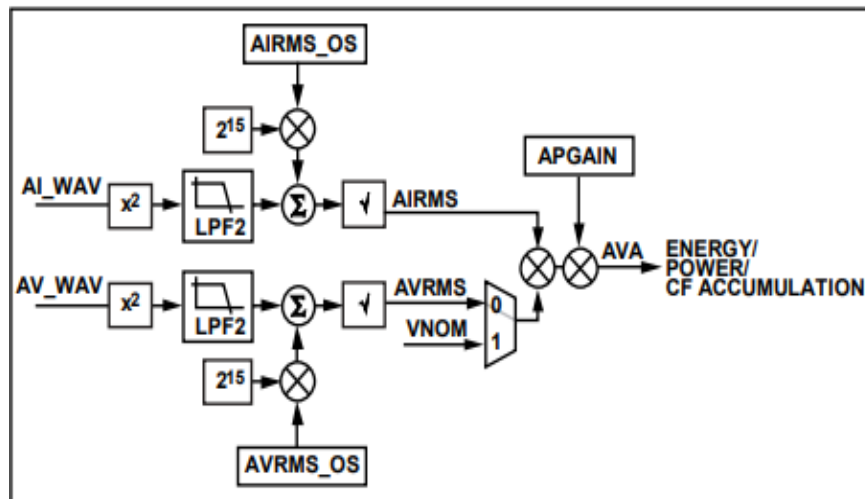


Figura 14. Procesado para la obtención de la potencia aparente [15]

5. **Energía activa (AWATTHR):** se calcula a partir de la potencia activa acumulada en registros de acumulación durante períodos de tiempo determinados por el usuario.

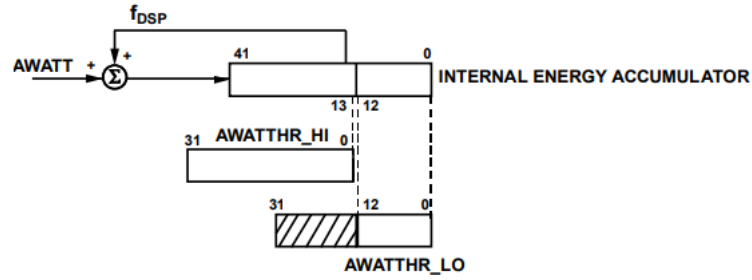


Figura 15. Registros de acumulación de energía [15]

El instante en el que se produce la acumulación viene dado por el modo de operación, siendo el primero con una frecuencia de muestreo fija de 4 kSPS y el segundo en cada cruce por cero. Cada uno de estos modos determina los tiempos máximos que tardan los diferentes registros en desbordar.

6. **Factor de potencia (APF):** se actualiza cada 1.024 segundos y se calcula mediante la siguiente fórmula:

$$APF = \frac{AWATT \text{ acumulada durante } 1.024 \text{ s}}{AVA \text{ acumulada durante } 1.024 \text{ s}}$$

Ecuación 31

El signo del factor de potencia sigue el del registro AWATT, por otro lado, para determinar a qué cuadrante pertenece sirve como base la Figura 16:

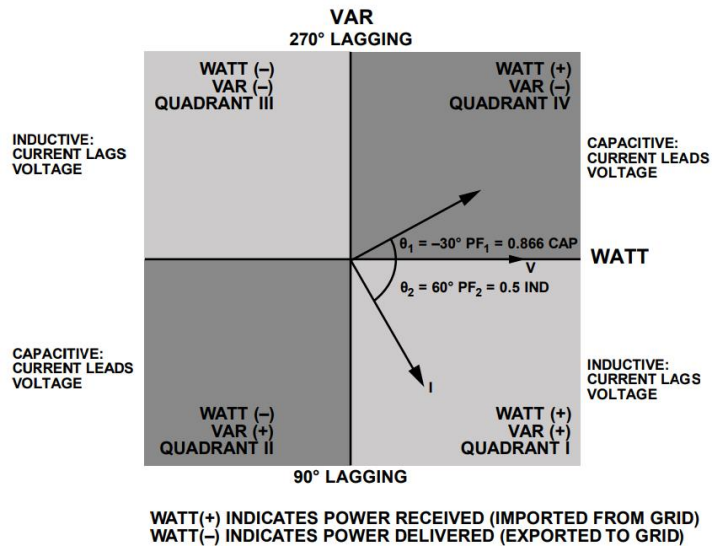


Figura 16. Signos de la potencia según tipo de carga [16]

Cabe mencionar que, en muchas aplicaciones, VAR y WATT se encontrarán en el primer y cuarto cuadrante puesto que se consumirá potencia activa de la red.

7. **Frecuencia:** se calcula a partir de la frecuencia configurada y del período (APERIOD), el cual se obtiene gracias a la detección de cruce por cero.

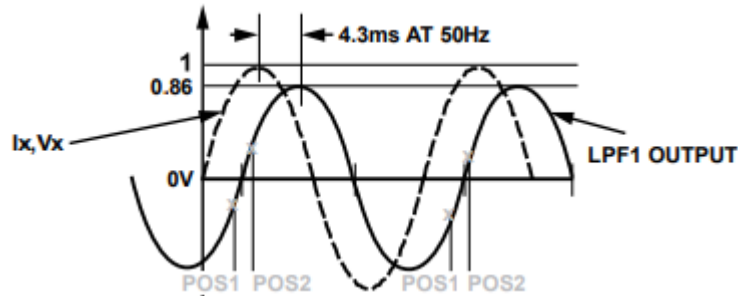


Figura 17. Detección de cruce por cero [16]

$$f_L = \frac{4000 \cdot 2^{16}}{APERIOD + 1} (Hz)$$

Ecuación 32

En todos los casos a excepción de los dos últimos hace falta el uso de una constante de conversión que permita convertir el valor almacenado en el registro en un valor decimal. Esta constante varía en función del canal usado y el hardware utilizado. En cualquier caso, se configura por medio del software que interpreta la medida almacenada en el registro.

4.2. Diseño de Hardware

Como punto de partida del proyecto se utilizó la placa de evaluación EV-ADE9153ASHIELDZ y la placa de desarrollo ESP8266 D1 Mini, ambas disponibles y proporcionadas por el Departamento de Ingeniería Telemática y Electrónica de la UPM.

La EV-ADE9153ASHIELDZ dispone tanto de *pads* como de zócalos para conectar directamente las placas Arduino Uno, Arduino Zero y ESP8266 ESP12-E Node MCU. Dado que la distribución de pines de las placas mencionadas anteriormente era completamente diferente de la distribución de la D1 Mini, se optó por diseñar una PCB que permitiera la conexión.

4.2.1 PCB adaptadora

Se llevó a cabo el diseño de la PCB adaptadora siguiendo los criterios y conexionado mencionados a continuación:

4.2.1.1 Criterios

Criterio 1: compatibilidad de conexionado

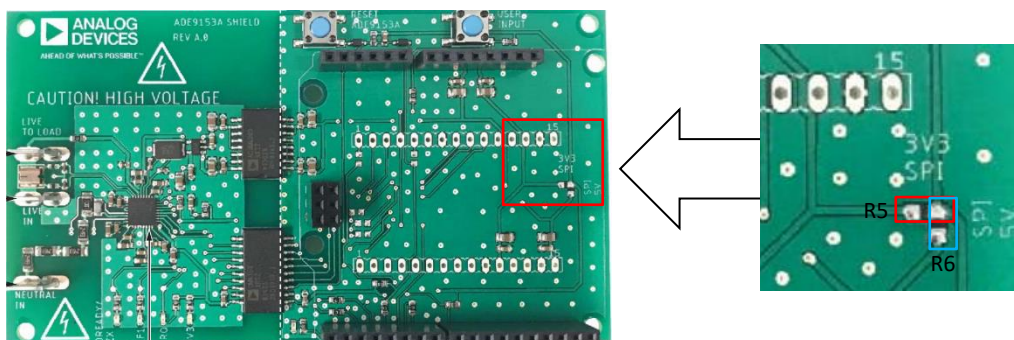


Figura 18. Configuración para el voltaje del SPI [14]

En primer lugar, para establecer la comunicación entre la EV-ADE9153ASHIELDZ y la ESP8266 D1 Mini, es necesario que los pines de ambas trabajen en el mismo rango de tensión. Para ello, se debe desoldar R6 y soldar R5 (ambas de 0 Ω , por lo que equivalen a un cortocircuito), permitiendo que las líneas SPI de la placa EV-ADE9153ASHIELDZ trabajen en un rango de 0 a 3.3 V, en lugar de 0 a 5 V. Esto es necesario ya que los pines de la D1 Mini, al igual que los de la microSDHC, únicamente soportan el rango de 0 a 3.3 V.

En segundo lugar, con el objetivo de conseguir que la PCB y la placa de evaluación encajen de manera precisa, se ha decidido utilizar la distribución de pines de la placa Arduino Zero, disponiendo así de más espacio para colocar otros componentes

Criterio 2: componentes adicionales

- **Adaptador 3M 2908-05WB-MG:** con la finalidad de ejercer de elemento de unión entre las líneas destinadas a la comunicación con la microSDHC y la propia placa.
- **Pines macho Vcc y GND:** conectar tanto la placa de evaluación como la de desarrollo a la fuente de alimentación de 5 V.

Criterio 3: posición de la D1 Mini

Con el objetivo de minimizar el impacto negativo en la radiación de la antena de la ESP8266, se ha decidido colocar la placa D1 Mini al borde de la PCB. Esto se debe a que la proximidad de un plano metálico a la antena puede afectar tanto a su patrón de radiación como a la frecuencia a la que se irradia. Hay que mencionar que, aunque la siguiente medida de precaución se menciona en el manual del chip para el caso en que la antena se encuentre directamente encima de ese plano metálico, durante el estudio de esta fase del desarrollo se ha visto cómo también afecta aun habiendo una pequeña separación entre antena y plano.

El fabricante de la ESP8266, recomienda dejar el siguiente espacio alrededor de la antena para prevenir las interferencias mencionadas anteriormente:

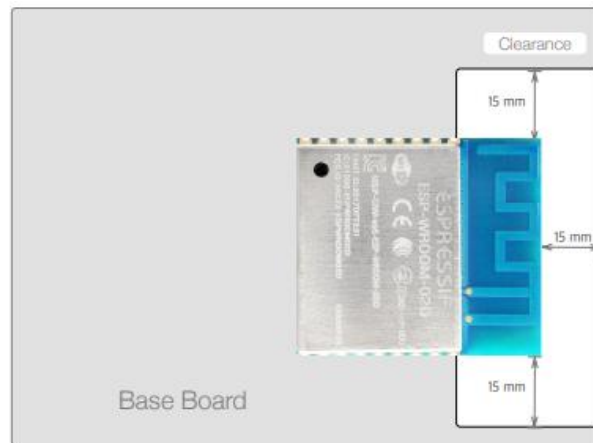


Figura 19. Posición de antena recomendada por Espressif [17]

Se ha considerado relevante asegurar esas distancias ya que, en base al siguiente estudio realizado en Ansys (software de simulación por elementos finitos) sobre la ESP32, chip del mismo fabricante, estas interferencias pueden afectar de las siguientes formas [18]:

- **Con la antena integrada en la placa de desarrollo y sin otros planos metálicos externos:** la Figura 20 presenta el patrón de radiación de una antena PIFA (*Planar Inverted-F Antenna*), que se espera con forma de *donut*, siendo la zona roja donde esta es más intensa. Por otro lado, la Figura 21 muestra el parámetro $S_{1,1}$ en dB, que indica la cantidad de potencia reflejada desde la antena. Valores cercanos a 0 dB indican alta reflexión, es decir, que la antena no está irradiando eficientemente la potencia transmitida. Valores negativos indicarán lo contrario. En este caso, la antena irradia mejor a 2.4 GHz.

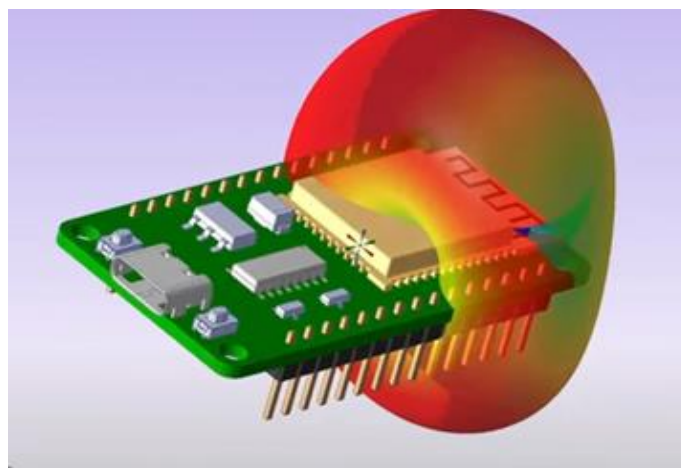


Figura 20. Patrón de radiación sin planos metálicos cercanos a la antena de la placa de evaluación [18]

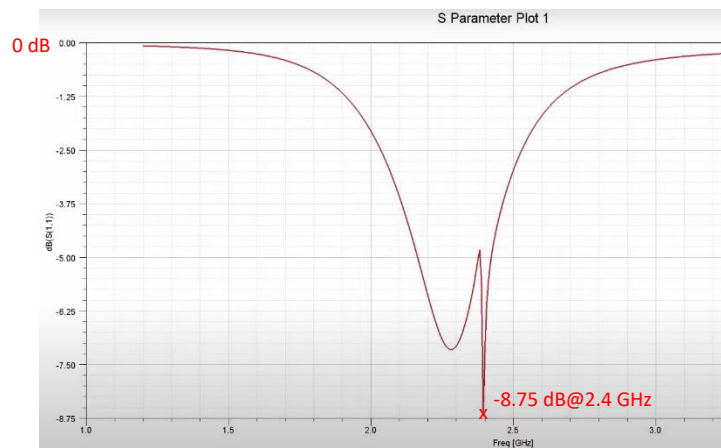


Figura 21. Parámetro $S_{1,1}$ en el primer caso [18]

- Con un plano metálico bajo la placa de desarrollo, pero un posicionamiento adecuado de la antena (fuera del plano metálico), el patrón de radiación se ve ligeramente modificado como en la Figura 22 y, aunque irradie mejor a 2.3 GHz, sigue haciéndolo también a 2.4 GHz. Seguirá funcionando correctamente, aunque de forma menos eficiente.

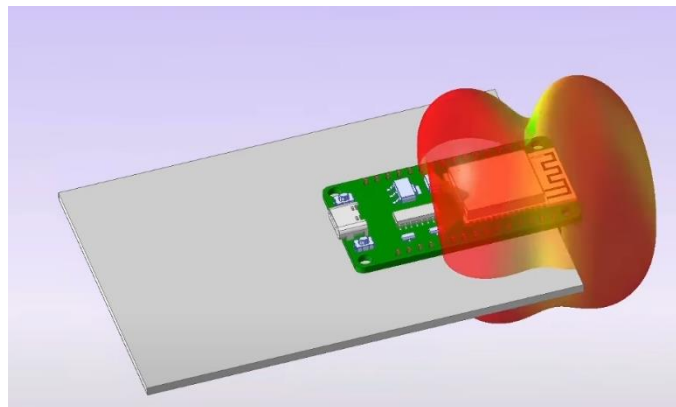


Figura 22. Patrón de radiación con posicionamiento de la antena fuera del plano metálico [18]

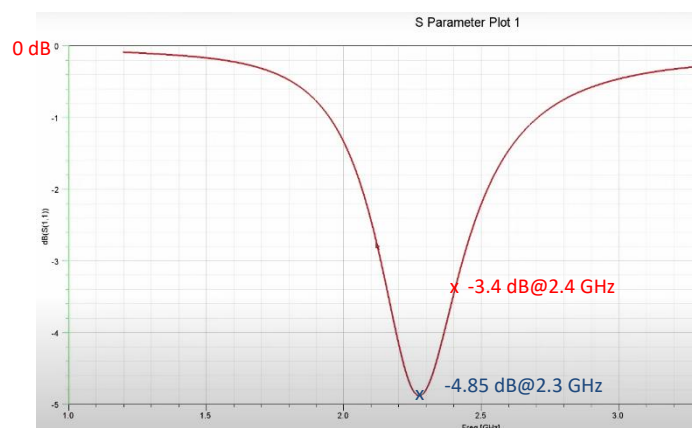


Figura 23. Parámetro $S_{1,1}$ en el segundo caso [18]

- Por último, si **se coloca la antena en el centro del plano metálico**, se obtendrá el peor resultado pues existe la posibilidad de que la frecuencia de radiación se vea modificada como en el caso del estudio (de 2.4 GHz a 1.8 GHz) y, además, el patrón será completamente distinto.

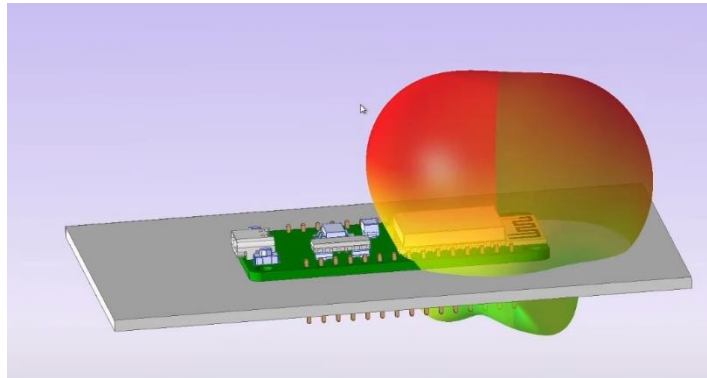


Figura 24. Patrón de radiación con posicionamiento de la antena en el centro del plano metálico [18]

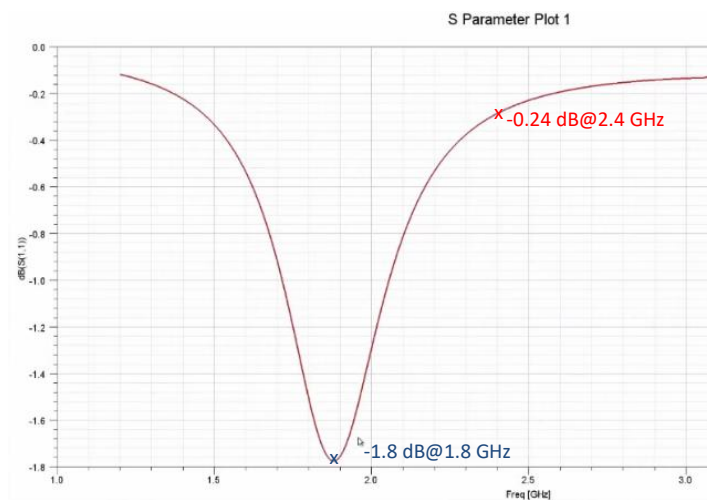


Figura 25. Parámetro $S_{1,1}$ en el tercer caso [18]

Como es de esperar, la antena no puede atravesar la placa, y si bien para algunas aplicaciones puede buscarse este tipo de radiación, en este caso se ha diseñado para que irradie con un patrón de donut como se ha podido ver en los otros dos casos. Además, aunque este patrón pudiera ser suficiente para este proyecto, la frecuencia de resonancia es totalmente distinta, lo cual carecería de funcionalidad.

Criterio 4: económico

Es un aspecto importante a valorar y, dado que las pistas no superarán los 25 MHz y la cantidad de señales y componentes no es alta, se ha decidido emplear únicamente dos capas, lo cual se traduce en un menor coste de fabricación. [19]

Criterio 5: otros aspectos

El plano de tierra ayuda a reducir el ruido y las interferencias electromagnéticas al proporcionar una vía de baja impedancia para las corrientes de retorno, mejorando y manteniendo así la integridad de la señal. Combinados con planos de alimentación mejoran el suministro de energía. Por otro lado, también ayudan a simplificar el enrutamiento de las pistas. [20]

En adición a los criterios anteriores y siguiendo [19], se le ha dado prioridad a que no hubiera roturas ni discontinuidades en el plano de masa por debajo de las pistas críticas, pues la inductancia es mucho mayor. Como se puede ver en la Figura 26 o en la Figura 66 del Anexo, el plano de masa (azul en la Figura 26) no tiene apenas discontinuidades a excepción de las generadas por la pista que une las dos vías situadas a la izquierda.

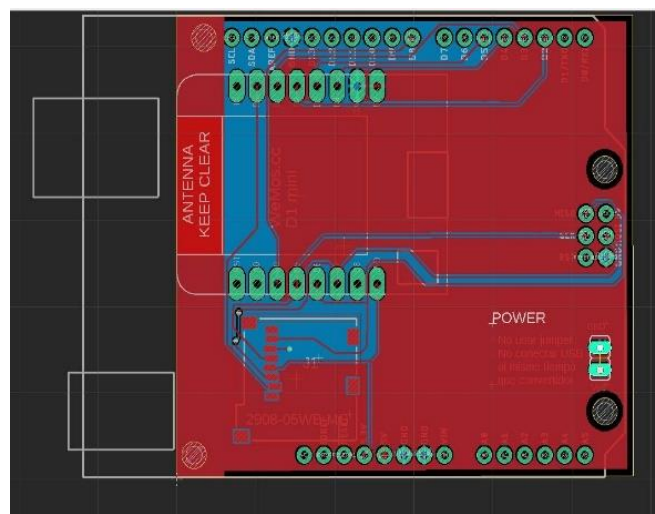


Figura 26. Enrutado de la PCB en Eagle

Finalmente, en cuanto a tamaño de *pads*, pistas, vías y espaciados, se han utilizado las reglas de diseño correspondientes a 8 mils. Teniendo en cuenta estas reglas y los criterios mencionados anteriormente se ha decidido llevar a cabo el diseño de la Figura 27, en el que no existe ningún plano metálico por debajo de la antena y además esta se encuentra posicionada a un lado.

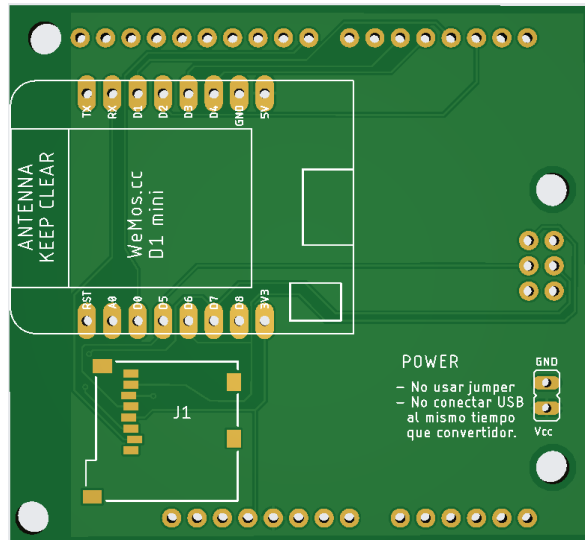


Figura 27. Diseño de la PCB adaptadora

4.2.1.2 Conexionado

El conexionado entre la D1 Mini, la microSD y la placa de evaluación se hará de la siguiente forma:

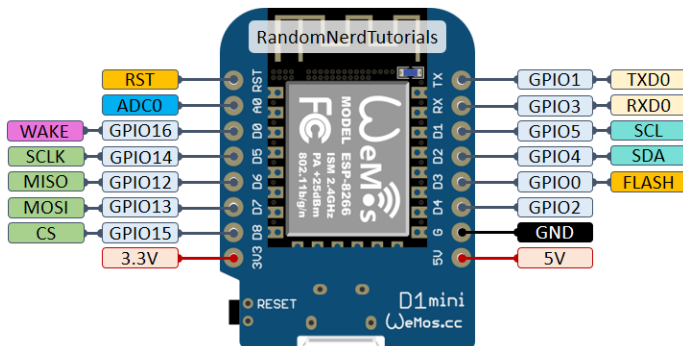


Figura 28. Pinout D1 Mini [21]

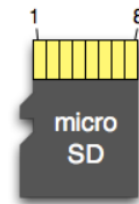


Figura 29. Pinout microSD [22]

Conexionado microSD - D1 Mini				
Señal	Pin SD	Pin D1 Mini	GPIO ESP8266	Tipo (POV ESP8266)
Chip Select microSD	2	RX	1	Salida
MOSI	3	D6	12	Salida
Vcc	4	3.3V	3.3V	Alimentación
SCLK	5	D5	14	Salida
GND	6	G	GND	Alimentación
MISO	7	D7	13	Entrada
No conectados	Resto	-	-	-

Tabla 1. Conexionado microSD-D1 Mini

Conexión EV-ADE9153A-SHIELDZ – D1 Mini			
Señal PCB	Pin D1 Mini	GPIO ESP8266	Tipo
SCLK	D5	14	Salida
MISO	D7	13	Entrada
MOSI	D6	12	Salida
Chip Select ADE	D0	16	Salida
RESET	D4	2	Salida
IRQ	D1	5	No usado
ZX/DREADY/CF2	D2	4	No usado
User Button Input	D3	0	No usado
5 V	5 V	5 V	Alimentación
3.3 V	3.3 V	3.3 V	SPI
GND	G	GND	Alimentación

Tabla 2. Conexión EV-ADE9153A-SHIELDZ- D1 Mini

En cuanto a la conexión física entre la PCB y la placa de evaluación, se seguirá el esquema de la Figura 30, pues como se mencionó anteriormente se usa la distribución de pines de la placa de Arduino Zero.

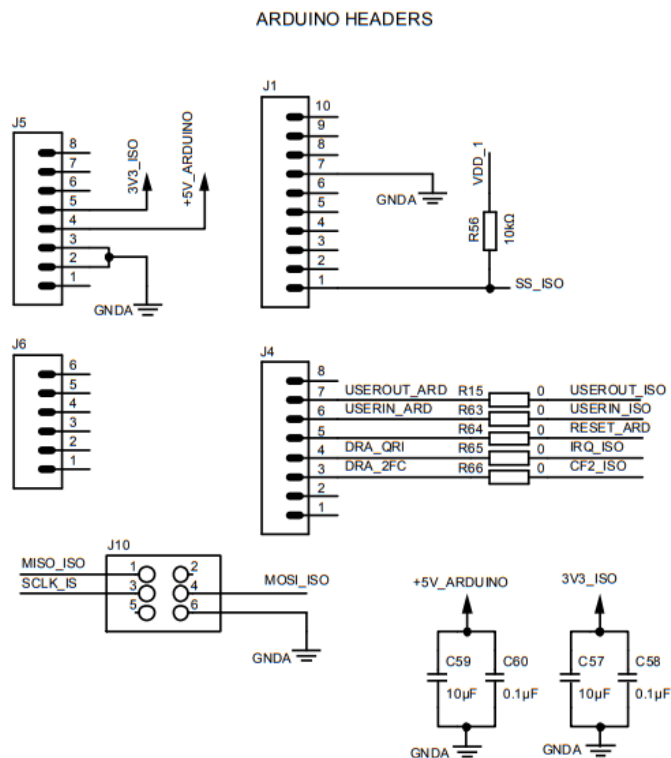


Figura 30. Relación de señales ADE9153A – pinout Arduino [14]

4.2.2 Carcasa

Además de la PCB, también ha sido necesario incluir en el diseño una carcasa para añadir robustez al medidor y evitar la manipulación de los componentes internos, conectores schuko macho y hembra para facilitar la conexión de la carga al medidor y de este a la red, un soporte para asegurar la sujeción de la placa de evaluación y otro para el convertidor AC/DC que proporcionará la alimentación al circuito.

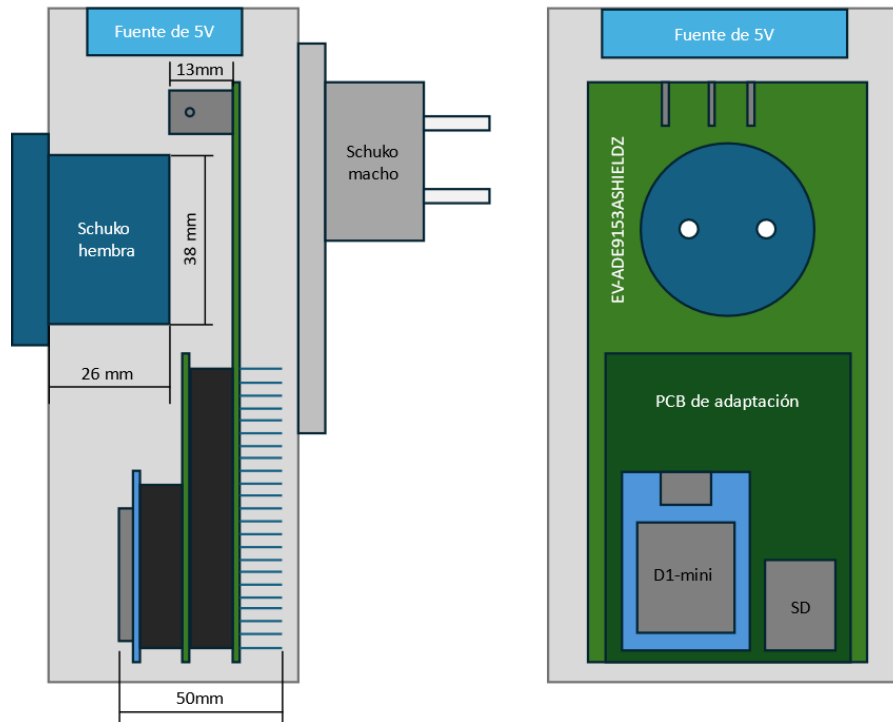


Figura 31. Boceto del medidor [23]

El proceso de desarrollo ha seguido los siguientes pasos [23]:

4.2.2.1 Elección de la caja

Para el ancho y largo han sido decisivas las medidas de la EV-ADE9153A-SHIELDZ ya que es el componente del medidor que más espacio ocupa. En cuanto a la altura, debía de ser la suficiente para que cupieran la placa de evaluación, la PCB adaptadora y la D1 Mini en su configuración final (una encima de otra).

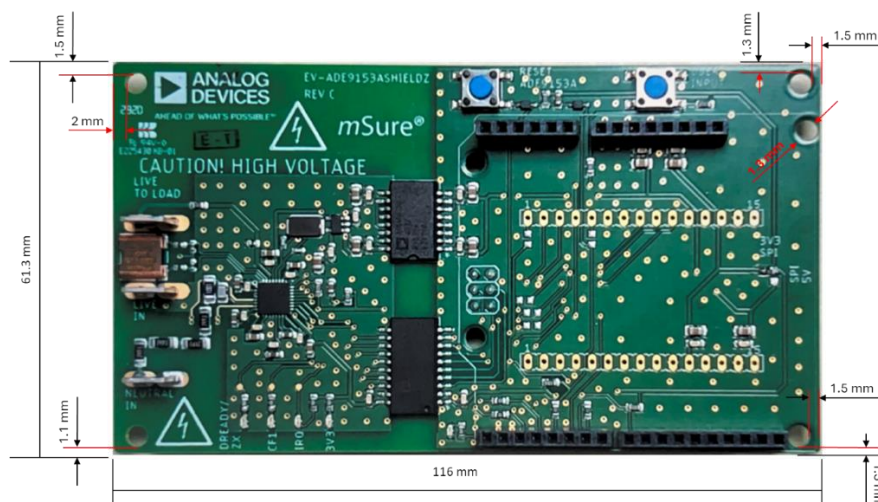


Figura 32. Medidas de la placa de evaluación [23]

4.2.2.2 Elección de los shuko

Una vez elegida la caja, era necesario elegir los conectores shuko macho y hembra. El macho se ha elegido de tal forma que su anchura fuese reducida y cuyo parte posterior fuera lo más plana posible para facilitar su posterior acoplamiento a la caja. El hembra, por su parte, debía cumplir con unas medidas y anclajes que le permitieran acoplarse a la caja mediante tornillos.



Figura 33. Medidas de la caja del medidor [23]

4.2.2.3 Diseño 3D

Con la finalidad de adaptar tanto la placa de evaluación como el convertidor AC/DC, se han diseñado una serie de elementos que permiten una correcta sujeción de estos. Para su diseño se ha utilizado *Fusion360* de *Autodesk*, posteriormente se ha exportado a STL (*Standard*

Tessellation Language) y, mediante el programa *CURA* (especializado en impresoras 3D) se ha laminado la figura para que la impresora 3D fuera capaz de imprimirla por capas.

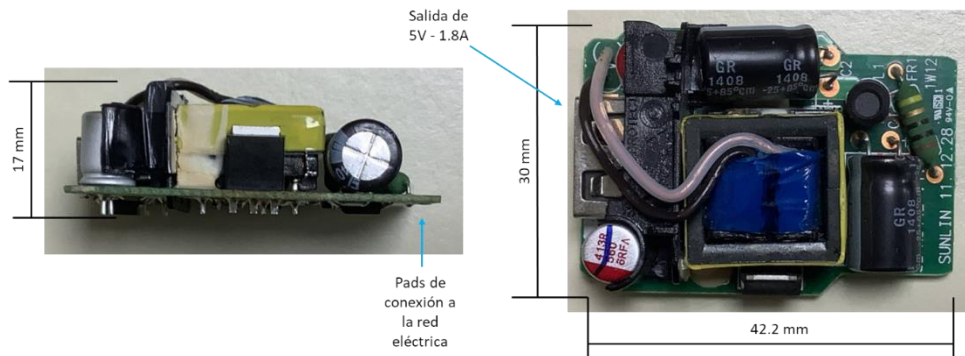


Figura 34. Medidas del convertidor AC/DC [23]

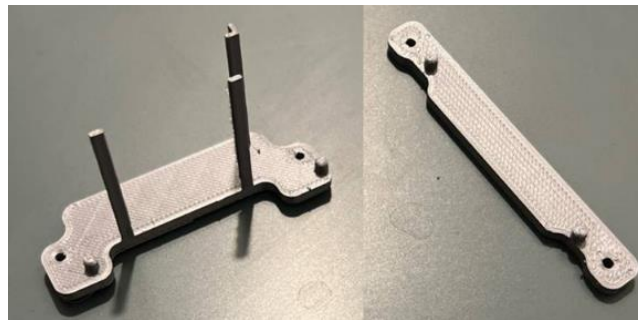


Figura 35. Elementos para sujetar el convertidor AC/DC [23]

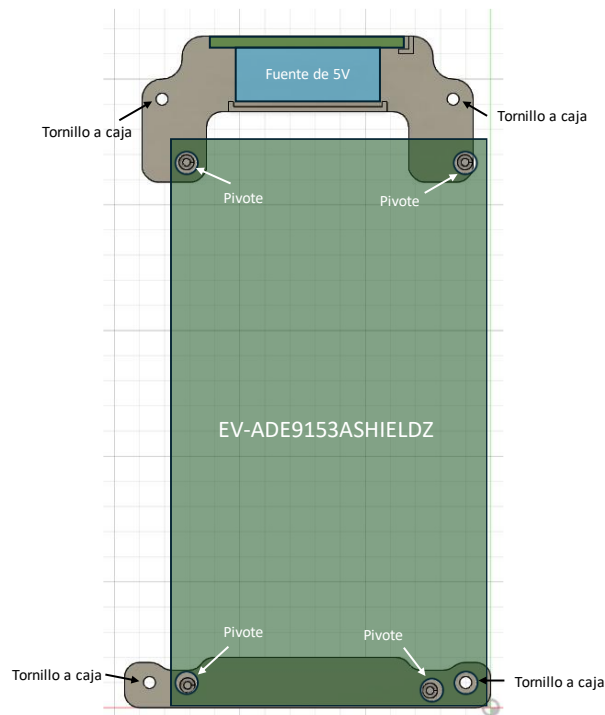


Figura 36. Colocación de placa de evaluación y convertidor [23]

Finalmente, como resultado de la parte hardware se obtiene un medidor completamente cerrado, ofreciendo solamente un conectar macho y hembra para la manipulación del usuario:



Figura 37. Resultado final del medidor

4.3. Diseño de Software

El software de este proyecto ocupa dos entornos principales: por un lado, Arduino, que será necesario para codificar el programa que ejecute el microcontrolador; y por otro lado *Flutter*, un *framework* de código abierto de *Google* que permite crear aplicaciones multiplataforma compiladas de forma nativa a partir de una única base de código.

4.3.1 Arduino

4.3.1.1 Algoritmo

En primer lugar, se ejecuta el inicio del programa, en el que se hará la configuración de pines de la D1 Mini y de registros del ADE9153A. Por otro lado, también se inician los servicios que corren en el microcontrolador. Tras conectarse con éxito a la red WiFi, se inician dos tareas que se ejecutan periódicamente y cuyas *callbacks* definirán la comunicación con la aplicación móvil.

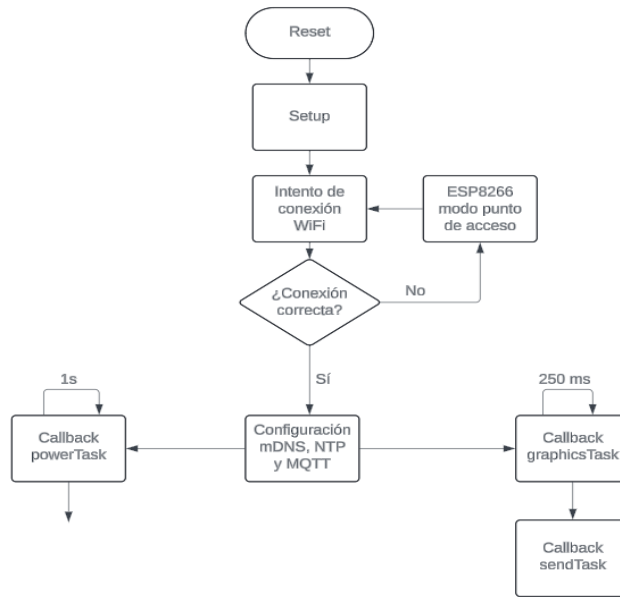


Figura 38. Diagrama de flujo del inicio de la aplicación Arduino

La tarea *powerTask* ejecuta su *callback* cada segundo, en esta se leen los registros internos de medidas del ADE9153A y se ejecuta el método *powerFunction*, el cual realiza las siguientes funciones:

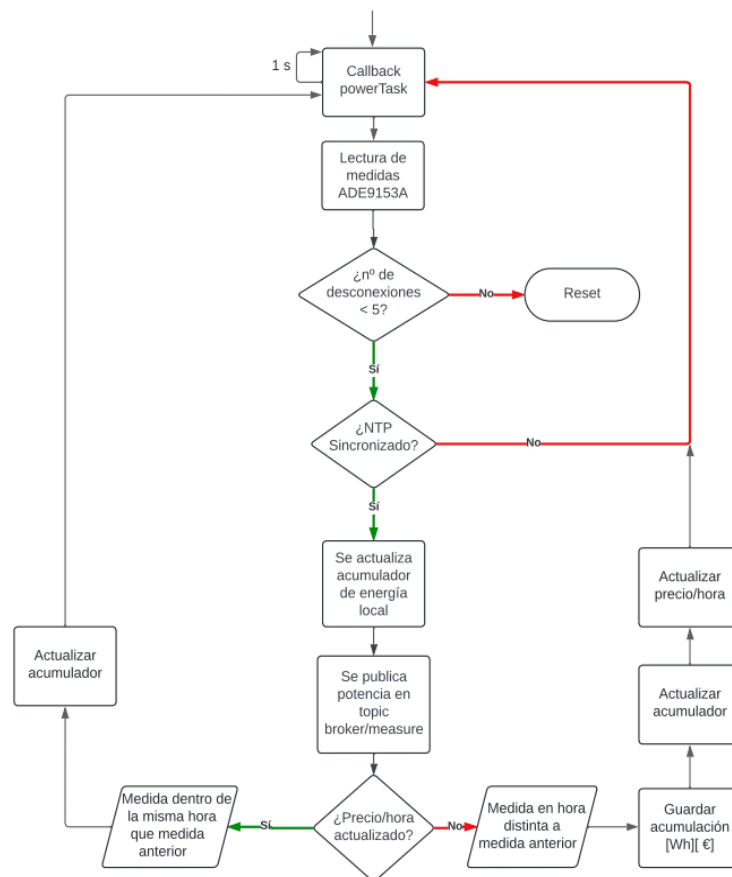


Figura 39. Diagrama de flujo de la callback asociada al Timer 1s

- **Lectura de registros de medidas**

Inicialmente se lee la información de los registros de medida del ADE9153A y después se utiliza la constante de conversión de cada magnitud (común para todas las placas de evaluación) para obtener el valor físico de las lecturas tal y como se indica en la nota de aplicación AN7571 [13].

- **Acumulación de energía localmente y en la tarjeta de memoria**

Cada segundo se lee la energía activa consumida durante el último segundo y se suma a un acumulador. Para mantener la robustez del sistema, este acumulador se encuentra almacenado en memoria, de esta forma si se desconecta el medidor o se apaga, la pérdida de datos se reduce. Cuando se realiza una medida, se lee también la última acumulación guardada en la tarjeta de memoria.

- **Si el precio [€/kWh] de la hora actual está actualizado** significa que la medida actual y la anterior se han hecho dentro del mismo rango horario, por lo que se sigue acumulando.
- **Si el precio [€/kWh] de la hora actual no está actualizado** significa que la hora actual del servidor NTP es distinta a la hora en la que se ha actualizado por última vez el precio, por lo que se procede a añadir al registro de consumo diario la acumulación hecha durante el rango horario anterior, se resetea el acumulador y se actualiza el precio para la hora actual.

El proceso de actualización de precio se explica en el siguiente apartado.

- **Asociar la energía consumida en kWh a su equivalente en euros**

El mercado eléctrico para el pequeño consumidor en España se divide en dos grupos principales, por un lado, la tarifa fija, en la que un proveedor y un cliente establecen un precio único para el kWh independientemente del momento del día; y por otro, el precio voluntario para el pequeño consumidor (PVPC), el cual varía dependiendo de la hora y cambia cada día. El caso que recoge este proyecto es el segundo, pues es el que añade más complejidad a la hora de medir el impacto económico del consumo.

Con el fin de conocer estos precios, se han implementado consultas a la API de precios de preciosdelaluz.org, cuyos datos provienen directamente de ESIOS (e-Sistema de Información del Operador del Sistema) [24], sistema informático de Red Eléctrica Española (REE) que contiene la base de datos con toda la información relacionada.

Una vez se obtienen los datos, se guardan en la tarjeta de memoria para las posteriores consultas, las cuales se harán al detectar que la hora ha cambiado, así pues, se multiplicará la energía consumida por el precio de la misma durante ese intervalo de tiempo y se obtendrá el consumo en euros ligado al consumo en vatios hora.

- **Publicar potencia instantánea en topic *broker/measure***

Aparte de la energía activa utilizada para la facturación del consumo, también se realizan lecturas del valor eficaz de tensión y corriente, potencia activa, reactiva y aparente y frecuencia, esto será de ayuda para verificar la calidad del consumo de la instalación. Estos datos se publicarán en el *topic broker/measure* en formato JSON para facilitar su manipulación en la aplicación.

De la misma manera, la tarea *graphicsTask* ejecuta su *callback* cada 250 ms, en esta se realiza la comprobación del *flag transmissionFinished*, este determinará si hay información para transmitir en el *topic broker/register* mediante la tarea *sendTask*.

La forma en la que se leen los datos a transmitir viene dada por la Figura 40.

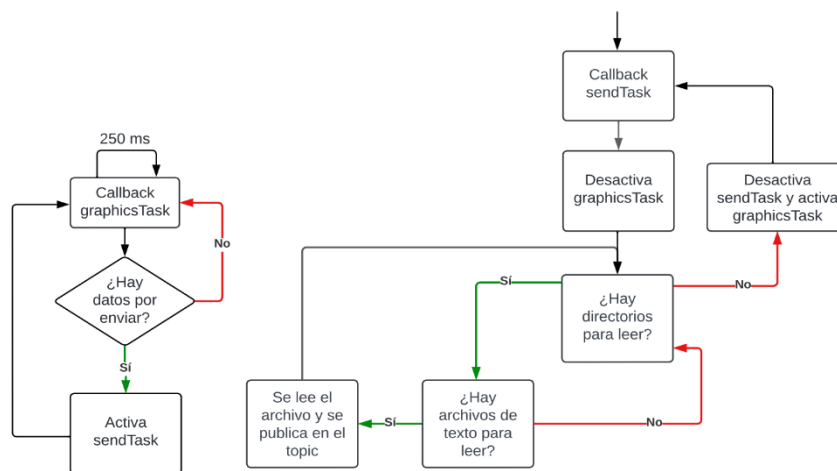


Figura 40. Diagrama de flujo de la callback asociada al Timer250ms

4.3.1.2 Comunicación SPI

La D1 Mini cuenta con dos periféricos (la microSD y la placa de evaluación), con los que se comunica mediante SPI (protocolo de comunicación síncrono que utiliza arquitectura maestro-esclavo, donde el maestro controla la comunicación y puede interactuar con varios esclavos). Ambos se encuentran conectados a las mismas líneas a excepción de las señales CS (Chip Select, indica con qué esclavo se comunica el maestro) pues cada módulo tendrá la suya propia.

Configuración bus SPI entre ADE9153A y ESP8266

El fabricante proporciona una biblioteca [25] y un ejemplo [26] que permite tanto configurar los registros del chip como realizar una primera prueba para comprobar el correcto funcionamiento.

Se hace uso de la librería SPI de Arduino para gestionar este bus, la configuración se establece en *ADE9153AAPI.cpp* [27] a una frecuencia de 1 MHz, primero el MSB y en modo SPI_MODE0, que representa el siguiente ajuste:

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Output Edge	Data Capture
SPI_MODE0	0	0	Falling	Rising

Figura 41. Polaridad y fase del reloj [28]

CPOL: a 0 significa SCLK inactivo a nivel bajo, a 1 inactivo a nivel alto.

CPHA: a 0 los datos se muestrean en el flanco de subida y se leen en el de bajada, a 1 al contrario.

Con dicha configuración se respetan los requisitos para el bus SPI indicados por el fabricante en las figuras 42 y 43:

Parameter	Symbol	Min	Typ	Max	Unit
\overline{SS} to SCLK Edge	t_{SS}	10			ns
SCLK Frequency	f_{SCLK}			10	MHz
SCLK Low Pulse Width	t_{SL}	40			ns
SCLK High Pulse Width	t_{SH}	40			ns
Data Output Valid After SCLK Edge	t_{DAV}			40	ns
Data Input Setup Time Before SCLK Edge	t_{DSU}	10			ns
Data Input Hold Time After SCLK Edge	t_{DHD}	10			ns
Data Output Fall Time	t_{DF}			10	ns
Data Output Rise Time	t_{DR}			10	ns
SCLK Fall Time	t_{SF}			10	ns
SCLK Rise Time	t_{SR}			10	ns
MISO Disable After \overline{SS} Rising Edge	t_{DIS}			100	ns
\overline{SS} High After SCLK Edge	t_{SFS}	0			ns

Figura 42. Requisitos de tiempo del bus SPI para el ADE9153A [15]

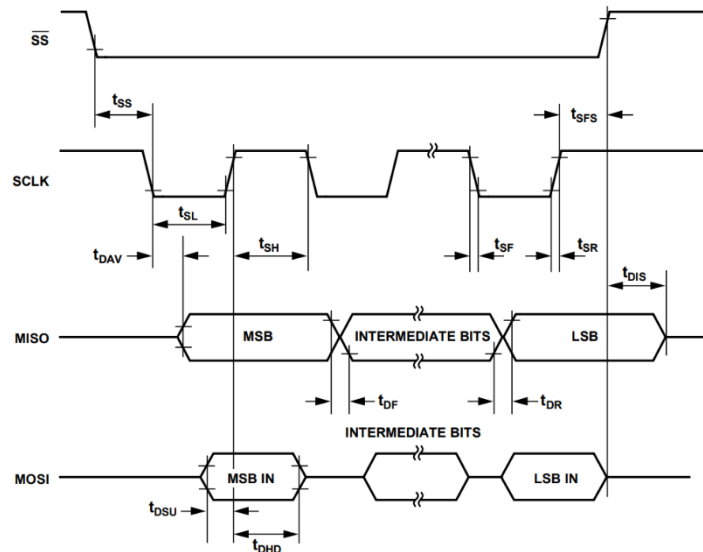


Figura 43. Representación del bus SPI en el ADE9153A [15]

Configuración bus SPI entre microSD y ESP8266

La librería SD de Arduino es la que permite llevar a cabo la comunicación de la forma más fácil posible, pues únicamente es necesario indicar el CS correspondiente a la microSD y manejar los ficheros y la información a alto nivel. La configuración del bus se realiza en el fichero *sd2card.cpp* [29], en este caso se establece una frecuencia de reloj de 250 kHz, el MSB primero y SPI_MODE0 al igual que el anterior caso.

Aunque ambos trabajan a frecuencias distintas, el *driver* que controla el bus es independiente para cada uno por lo que no existirá conflicto durante las transmisiones.

4.3.1.3 Configuración de red

Para alcanzar la funcionalidad esperada del sistema es imprescindible que este se encuentre conectado a una red WiFi con acceso a internet, pues es lo que permitirá al microcontrolador llevar a cabo los siguientes servicios:

WiFiManager

Con el fin de orientar este proyecto a un consumidor final, se ha decidido implementar la conexión a la red WiFi de la forma más fácil posible. Es por eso que se usa esta biblioteca, pues una vez introducidos el SSID y la contraseña por primera vez, queda guardada para conectarse de forma automática las siguientes veces que se inicia la aplicación.

El modo de funcionamiento es el siguiente:

Cuando se enciende el microcontrolador, se configura en modo estación e intenta conectarse a un punto de acceso previamente guardado. Si este intento de conexión resulta fallido, se configura el microcontrolador en modo punto de acceso y se crea un DNS y un servidor web, cuya ip es por defecto 192.168.4.1. Este modo de acceso se ha configurado de forma bloqueante por lo que la aplicación no seguirá ejecutándose hasta que no se introduzcan los datos de una red válida. Si los datos de la red WiFi no son correctos, se volverá a abrir el punto de acceso para un nuevo intento. El tiempo durante el cual el portal de acceso se mantiene activo es configurable mediante código, siendo por defecto cinco minutos.

Una vez conectado a la red WiFi, la asignación de la dirección IP se puede hacer de forma estática o de forma dinámica por medio del DHCP del punto de acceso. Se elige la segunda opción porque, aunque generalmente las direcciones IP de los *routers* domésticos siguen la configuración 192.168.1.x o 192.168.0.x, en este proyecto se tienen en cuenta también los puntos de acceso móviles, cuyas direcciones IP pueden seguir configuraciones distintas como en el caso del iPhone usado en este proyecto, siendo la dirección IP 172.20.10.1.

El hecho de asignar dinámicamente la dirección IP supone que no será conocida por el usuario, pues esta podrá cambiar según la red a la que se conecte el medidor. Esto plantea un problema ya que la aplicación móvil (cliente) debe conectarse al microcontrolador (bróker) y para ello necesita conocer su dirección y, si bien esta información se puede conocer inspeccionando la

lista de dispositivos conectados al punto de acceso, es algo poco práctico. Para resolver este problema se hace uso del protocolo mDNS (*Multicast Domain Name Server*), el cual se explicará más adelante. Cabe mencionar que en la documentación de esta biblioteca en el día que se escribe esta memoria, se comenta que *WiFiManger* no es compatible con el servicio mDNS, sin embargo, tras revisar cuidadosamente su código fuente se ha comprobado que esto no se aplica a la ESP8266, por lo que se han modificado las líneas necesarias que permiten el funcionamiento de ambos.

Sincronización NTP

Dado que la ESP8266 no cuenta con un RTC integrado, es necesario que esté sincronizada de la mejor manera posible al servidor NTP, es por ello que se hace uso de la biblioteca *ESPntpClient* [30], la cual proporciona una precisión cercana a 1 ms y cuyos cálculos de *offset* se implementan en base al estándar RFC5905 NTP [31].

También dispone de una *callback* programable para la gestión de eventos, siendo el generado tras una sincronización parcial el único usado para la gestión de la aplicación, y el de error de precisión con fin meramente informativo durante la depuración ya que automáticamente buscará la mayor precisión en todo momento.

La dirección IP del servidor NTP usado es 158.227.98.15, pertenece a la Escuela Técnica Superior de Ingeniería de Bilbao y la fuente es GPS (*Global Positioning System*) [32]. Quedará guardada en la memoria *flash*.

Servicio mDNS

Como se ha mencionado antes, al tratarse de un sistema en el que la manipulación del código fuente de la aplicación y la visión del puerto serie está fuera del alcance del usuario medio, para la conexión entre el microcontrolador y la aplicación móvil se ha decidido utilizar el protocolo mDNS, el cual permitirá lo siguiente:

1. Al igual que “los servidores DNS contienen tablas que convierten nombres y dominios en una dirección IP” [33], mDNS lo hará de forma local resolviendo nombres de dominio que emplean “.local” como sufijo. Siendo el caso que nos concierne *http://esp8266.local* equivalente a la dirección IP que le haya asignado el DHCP dentro de la red WiFi.
2. Ofrecer un servicio determinado identificado por el nombre, protocolo que usa y puerto donde se ofrece, siendo el servicio *mqtt* mediante el protocolo TCP en el puerto 1883 en este caso.

Con todo esto, el cliente podrá determinar qué dispositivo ofrece el servicio *mqtt* y resolver el nombre mDNS, es decir, obtener la dirección IP asociada a *http://esp8266.local* en cada caso.

MQTT

MQTT (*Message Queuing Telemetry Transport*) es un protocolo de comunicación M2M (*machine-to-machine*) de tipo *message queue* y está basado en TCP/IP como base para la comunicación [34].

El funcionamiento de MQTT es un servicio de mensajería *push* con patrón publicador/suscriptor y en este tipo de infraestructuras los clientes se conectan con un servidor central denominado bróker.

Para filtrar los mensajes que son enviados a cada cliente, se disponen en *topics*. Un cliente puede publicar en un determinado *topic*, otros clientes se pueden suscribir a este y el bróker actuará de intermediario para hacerlos llegar.

Los clientes inician una conexión TCP/IP con el bróker, el cual mantiene un registro de los clientes conectados. Esta conexión se mantiene abierta hasta que el cliente la finaliza. Por defecto, MQTT emplea el puerto 1883 y el 8883 cuando funciona sobre TLS (*Transport Layer Security*).

Dado que una red MQTT necesita un bróker, existen las siguientes formas de obtenerlo:

1. En la nube

El bróker queda alojado en un servidor remoto gestionado por el proveedor del servicio, esto elimina la necesidad de mantener *hardware* y *software* localmente, así como la capacidad de externalizar la capacidad de cómputo requerida por este. Estos ofrecen escalabilidad automática y adaptan los recursos a la demanda, también ofrecen capacidad de administración y herramientas de monitoreo, sin embargo, la personalización puede ser limitada. Por otro lado, en cuestiones de seguridad suelen tener medidas sólidas pero la privacidad de los datos transmitidos es algo que dependerá de la empresa propietaria del bróker y no del usuario. Por otro lado, al estar alojados en servidores remotos la latencia será ligeramente mayor en comparación de si el bróker estuviera dentro de la red local.

Podemos distinguir dos tipos:

a. De pago

Aunque existen diferentes opciones como Hive, EMQX o MyQttHub, todas funcionan mediante planes de suscripción mensuales o anuales, en los que el precio irá escalando en función de la capacidad soportada por el bróker, número de conexiones, de niveles de seguridad o configurabilidad del sistema.

Son las empresas mencionadas anteriormente las que ofrecen el servicio de bróker MQTT, sin embargo, los servidores que los alojan varían principalmente entre AWS (*Amazon Web Service*), Azure o Google Cloud, por lo que se debe tener en cuenta para configurar los clientes.

b. Gratuito

Al igual que existen planes de pago, también existen planes gratuitos de capacidades limitadas o versiones de prueba de tiempo limitado. Se caracterizan por no contar con servicio de atención al cliente 24 h y contar con un mínimo número de recursos en comparación con el resto de planes.

En el caso de usar la versión gratuita de Hive, ha de hacerse una configuración previa de los clientes que se conectarán puesto que solo permite conexiones TLS con autenticación.

2. Local

a. Instalado en un dispositivo externo a la ESP8266

En este caso, dependiendo del bróker elegido, se podrá instalar en un dispositivo u otro. Los principales brókers *open source* son Mosquitto, Mosca o Aedes entre otros, codificados en lenguajes distintos y por ello existiendo un amplio abanico donde elegir [35].

b. Instalado en la ESP8266

En este caso, será el propio microcontrolador el que aloje el bróker MQTT y, si bien la capacidad de procesamiento será mucho menor, la información no saldrá de la red local y la latencia será mucho menor que con un bróker alojado en la nube. Además, habrá un mayor control sobre el bróker y la información que éste maneja.

En los tres primeros casos la ESP8266 actúa como cliente publicador y envía la información a un dispositivo externo, sin embargo, en el último caso actuará directamente como bróker.

Dado que el objetivo del proyecto es ofrecer un producto que requiera la mínima configuración posible por parte del usuario, se ha decidido hacer uso de la biblioteca de Arduino uMQTTBroker [12], la cual permite llevar a cabo la última configuración.

Esta biblioteca permitirá usar las versiones 3.1 y 3.1.1 del protocolo simultáneamente, un máximo de cinco conexiones, retener mensajes, únicamente nivel 0 de QoS (*Quality of Service*) en el que el envío del mensaje se hace una única vez y posibilidad de autenticación mediante usuario y contraseña. Por otro lado, como el microcontrolador tiene recursos limitados no soporta comunicaciones TLS, por lo que no existirá esa capa de seguridad; y tampoco puede ejecutar *non-clear sessions*, es decir, una vez el cliente se desconecte se descartarán todos los mensajes e información asociada.

Consultas HTTPS

Con el fin de obtener los precios diarios de la luz, es necesario hacer consultas a la API de <https://api.preciodelaluz.org/>. Para esto se hace uso del protocolo HTTPS (*Hypertext Transfer Protocol Secure*) y el servidor devolverá la información en formato JSON.

Arduino permite hacer esta petición de tres formas en la ESP8266 [36] [37]:

1. Mediante certificado TLS:

Se trata de un certificado digital que se utiliza para establecer una conexión segura y cifrada entre un servidor (API) y un cliente (ESP8266). Este certificado proporciona autenticación, confidencialidad e integridad de los datos en su transmisión [38], lo que quiere decir que se verifica la identidad del servidor, cifrado de los datos e inalterabilidad de los mismos durante la transmisión. Estos certificados están firmados por una CA (*Certification Authority*) que determina el nivel de confianza en un certificado. Para hacer uso de este método, dicho certificado debe ser descargado y guardado, también deben hacerse cálculos de encriptación y desencriptación, para lo que se necesitan recursos de memoria y cómputo.

Se ha descartado ya que, aunque proporcione seguridad y la validez del certificado sea de alrededor de cinco años, el microcontrolador no tiene los recursos suficientes teniendo en cuenta el resto de las tareas que ejecuta.

2. Mediante *fingerprint*:

Se trata de una cadena obtenida a partir de aplicar un hash al certificado del apartado anterior. Este método tampoco es usado ya que requiere de una capacidad de procesamiento con la que el ESP8266 no cuenta cuando tiene el resto de las tareas en proceso.

3. Sin certificado ni *fingerprint*:

La conexión estará encriptada, sin embargo, no se realizará la verificación del certificado TLS del servidor. Esto significa que no habrá seguridad acerca de si el servidor al que se conecta el medidor es quien dice ser. Se opta por este método ya que es el único que hace posible la obtención de los datos, primando la funcionalidad sobre la seguridad ofrecida por el certificado. Para que funcione este método es necesario configurar el microcontrolador a modo de conexión no seguro.

Esta petición se hará siempre que se detecte que el último precio de energía guardado sea de una hora distinta a la hora establecida por el servidor NTP.

4.3.1.4 Configuración general y uso del ADE9153A

Reset inicial

Antes de hacer cualquier configuración particular, es recomendable hacer un reset al ADE9153A para que se reseteen los registros, esto permitirá hacer un arranque limpio del medidor y que no intervengan medidas de operaciones anteriores. Para esto es suficiente con una pulsación del botón *Reset ADE9153A* de la placa de evaluación, como en este caso no es posible acceder al interior del medidor para ello, se introduce un pulso a nivel bajo mediante software.

Configuración

Para hacer una primera configuración se debe escribir sobre una serie de registros. Es el fabricante quien proporciona el método que se encarga de ello por medio de la biblioteca disponible en el repositorio de *Analog Devices*.

```
5 *** void ADE9153AClass::SetupADE9153A(void)
57 {
58     SPI_Write_16(REG_AI_PGAGAIN,ADE9153A_AI_PGAGAIN);
59     SPI_Write_32(REG_CONFIG0,ADE9153A_CONFIG0);
60     SPI_Write_16(REG_CONFIG1,ADE9153A_CONFIG1);
61     SPI_Write_16(REG_CONFIG2,ADE9153A_CONFIG2);
62     SPI_Write_16(REG_CONFIG3,ADE9153A_CONFIG3);
63     SPI_Write_16(REG_ACCMODE,ADE9153A_ACCMODE);
64     SPI_Write_32(REG_VLEVEL,ADE9153A_VLEVEL);
65     SPI_Write_16(REG_ZX_CFG,ADE9153A_ZX_CFG);
66     SPI_Write_32(REG_MASK,ADE9153A_MASK);
67     SPI_Write_32(REG_ACT_NL_LVL,ADE9153A_ACT_NL_LVL);
68     SPI_Write_32(REG_REACT_NL_LVL,ADE9153A_REACT_NL_LVL);
69     SPI_Write_32(REG_APP_NL_LVL,ADE9153A_APP_NL_LVL);
70     SPI_Write_16(REG_COMPMODE,ADE9153A_COMPMODE);
71     SPI_Write_32(REG_VDIV_RSMALL,ADE9153A_VDIV_RSMALL);
72     SPI_Write_16(REG_EP_CFG,ADE9153A_EP_CFG);
73     SPI_Write_16(REG_EGY_TIME,ADE9153A_EGY_TIME);           //Energy accumulation ON
74     SPI_Write_16(REG_TEMP_CFG,ADE9153A_TEMP_CFG);
75 }
```

Figura 44. Método que se encarga de configurar el ADE9153A [27]

```
/* Configuration Registers */
#define ADE9153A_AI_PGAGAIN 0x000A    /*Signal on IAN, current channel gain=16x*/
#define ADE9153A_CONFIG0 0x00000000 /*Datapath settings at default*/
#define ADE9153A_CONFIG1 0x0300      /*Chip settings at default*/
#define ADE9153A_CONFIG2 0x0C00      /*High-pass filter corner, fc=0.625Hz*/
#define ADE9153A_CONFIG3 0x0000      /*Peak and overcurrent settings*/
#define ADE9153A_ACCMODE 0x0000      /*Energy accumulation modes, Bit 4, 0 for 50Hz, 1 for 60Hz*/

#define ADE9153A_VLEVEL 0x002C11E8    /*Assuming Vnom=1/2 of fullscale*/
#define ADE9153A_ZX_CFG 0x0000        /*ZX low-pass filter select*/
#define ADE9153A_MASK 0x00000100     /*Enable EGYRDY interrupt*/

#define ADE9153A_ACT_NL_LVL 0x000033C8
#define ADE9153A_REACT_NL_LVL 0x000033C8
#define ADE9153A_APP_NL_LVL 0x000033C8
```

Figura 45. Valor con el que se configuran los registros de las líneas 58 a 69 [25]

Con respecto a la biblioteca proporcionada se han hecho una serie de modificaciones:

```
/* Constant Definitions */
#define ADE9153A_RUN_ON 0x0001 /*DSP On*/
#define ADE9153A_COMPMODE 0x0005 /*Initialize for proper operation*/

#define ADE9153A_VDIV_RSMALL 0x03E8 /*Small resistor on board is 1kOhm=0x3E8*/

/* Energy Accumulation Settings */
#define ADE9153A_EP_CFG 0x0009 /*Energy accumulation configuration: read + reset*/
// #define ADE9153A_EGY_TIME 0x0F9F
/*Accumulate energy for (3999 + 1) samples -> 4000/4000 = 1 sec internal energy accumulation time*/
#define ADE9153A_EGY_TIME 0x0013 /* acumula energia cada (19+1) muestras, es decir, (19+1)/4000 = 5 ms*/

/* Temperature Sensor Settings */
#define ADE9153A_TEMP_CFG 0x000C /*Temperature sensor configuration*/
```

Figura 46. Cambio en la configuración para acumulación de energía [25]

1. El valor de ADE9153A_ACCMODE de 0x0010 a 0x0000 puesto que la frecuencia de la red eléctrica en España es de 50 Hz.
2. El valor de ADE9153A_EGY_TIME de 0x0F9F a 0x0013: hay dos acumuladores de energía, el pequeño acumula energía cada 5 ms (período de actualización del ADE9153A) y posteriormente, cada ADE9153A_EGY_TIME/4000 segundos los suma a un acumulador de 45 bits al que puede acceder el usuario. De esta forma, el acumulador accesible acumulará cada 5 ms y será leído cada segundo, además, en esta medida será reseteado, obteniendo por tanto en todo momento la acumulación real que se ha hecho durante ese segundo.
3. Por otra parte, la nota de aplicación [13] contiene la teoría necesaria para hallar las constantes de conversión de código almacenado en los registros de medida a valor decimal de la placa de evaluación usada, así como su proceso de calibración.

```
/* Ideal Calibration Values for ADE9153A Shield Based on Sensor Values */
#define CAL_IRMS_CC 0.838190 // (uA/code)
#define CAL_VRMS_CC 13.41105 // (uV/code)
// #define CAL_POWER_CC 1508.743 // (uW/code) Applicable for Active, reactive and apparent power
#define CAL_POWER_CC 1510.24987
// #define CAL_ENERGY_CC 0.858307 // (uWhr/xTHR_HI code)Applicable for Active, reactive and apparent energy
#define CAL_ENERGY_CC 0.859164
```

Figura 48. Cambios en las constantes de conversión para las medidas [25]

Los valores comentados pertenecen a los que venían configurados por defecto en la librería, se han cambiado por los valores teóricos reflejados en la nota de aplicación:

$TARGET_WCC = 1510.24987 \mu W/Code$
 $TARGET_VARCC = 1510.24987 \mu W/Code$
 $TARGET_VACC = 1510.24987 \mu W/Code$

Figura 47. Constantes de conversión teóricas para la potencia [13]

$TARGET_WHCC = 859.16437 nWh/Code$
 $TARGET_VARHCC = 859.16437 nVARh /Code$
 $TARGET_VAHCC = 859.16437 nVAh/Code$

Figura 46. Constantes de conversión teóricas para la energía [13]

El resto de los valores permanecen configurados por defecto.

Calibración

La nota de aplicación mencionada en el anterior punto también describe el proceso a seguir para realizar la calibración. Al usar como sensor de medida la *Rshunt*, la calibración se realizará sobre el canal A de corriente y tensión.

1. Se deben configurar las constantes de conversión para cada magnitud tal y como se indica en la figura 49:

```
/* Ideal Calibration Values for ADE9153A Shield Based on Sensor Values */
#define CAL_IRMS_CC      0.838190    // (uA/code)
#define CAL_VRMS_CC      13.41105    // (uV/code)
// #define CAL_POWER_CC    1508.743    // (uW/code) Applicable for Active, reactive and apparent power
#define CAL_POWER_CC      1510.24987
// #define CAL_ENERGY_CC    0.858307    // (uWhr/xTHR_HI code) Applicable for Active, reactive and apparent energy
#define CAL_ENERGY_CC      0.859164
```

Figura 49. Configuración [25]

La constante de conversión se mantiene para los tres tipos de potencia, al igual que para los tres tipos de energía.

Antes de cada calibración se debe comprobar si el sistema está listo, para ello se lee el registro `MS_STATUS_CURRENT` (0x0240), si el bit 0 vale 1, se puede calibrar.

2. Se ordena la autocalibración del canal de corriente A escribiendo 0x0000013 en el registro `MS_ACAL_CFG` (0x030). Los registros `MS_ACAL_AICC` (0x220) y `MS_ACAL_AICERT` (0x221) indican el resultado de la calibración.
 - a. **MS_ACAL_AICC** es la constante de conversión del canal de corriente A, que no de la medida leída del registro. Esta constante permite calcular la ganancia necesaria en el canal para obtener la constante de conversión de medida (`TARGET_AICC` o `CAL_IRMS_CC` en el código del anterior punto).

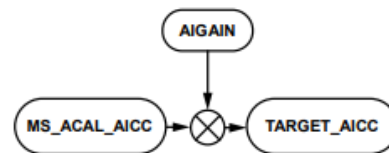


Figura 50. Relación entre constante de calibración, de conversión y ganancia del canal [13]

- b. **MS_ACAL_AICERT** indica el grado de precisión en ppm (partes por millón), por ejemplo, un valor de 3000 en ese registro equivale a una precisión superior a 0.3%.

$$\text{Precisión}(\%) = (\text{MS_ACAL_AICERT} / 10E6) \cdot 100$$

Ecuación 33

Esta calibración se puede realizar durante un tiempo fijo determinado por el usuario o revisando el valor de MS_ACAL_AICERT hasta que entre dentro de un rango específico. En este proyecto se utiliza el primer método, aplicándose un tiempo de 20 segundos.

Cabe señalar que el ADE9153A también cuenta con un modo turbo de calibración [15], el cual multiplica x4 la velocidad y x2 el consumo instantáneo, sin embargo, en el *datasheet* se recomienda el uso del modo normal.

3. Una vez obtenido MS_ACAL_AICC, se utiliza junto a TARGET_AICC para calcular AIGAIN (ganancia del canal de corriente A) mediante la siguiente fórmula:

$$AIGAIN_REGISTER_VALUE = \left(\frac{MS_ACAL_AICC}{TARGET_AICC} - 1 \right) \times 2^{27}$$

Ecuación 34

Por último, se escribe dicho valor en el registro AIGAIN (0x000).

4. Si el sistema está listo, se ordena ahora la calibración del canal de tensión, para ello se escribe 0x0000043 en el registro MS_ACAL_CFG (0x030).

Al igual que con el canal de corriente, podemos leer dos registros para leer la constante de conversión de este canal y su nivel de precisión. Estos son MS_ACAL_AVCC (0x224) y MS_ACAL_AVCERT (0x225).

Tras 40 segundos de calibración, esta se para escribiendo 0x0000001 en el registro MS_ACAL_CFG (0x030).

5. Por último, se calcula AVGAIN y se escribe en el registro AVGAIN (0x002).

$$AIGAIN_REGISTER_VALUE = \left(\frac{MS_ACAL_AICC}{TARGET_AICC} - 1 \right) \times 2^{27}$$

Ecuación 35

En el código se hace uso del método *autocalibrateADE9153A* para llevar a cabo este proceso, dicho método se define en *ADEDriver.h*.

4.3.1.5 Lectura de medidas

Aunque la frecuencia de actualización del integrado es alta, el microcontrolador únicamente leerá los valores de medida almacenados en los registros del ADE9153A una vez por segundo, esto es para que no se sobrecargue de trabajo y porque, comparado con el medidor de energía de referencia (Fluke 43B), se trata de un tiempo menor.

Además, tal y como se ha mostrado en el diagrama de flujo del *sketch*, permite aprovechar la misma *callback* para registro de consumo energético junto con la gestión de sus datos y lectura de los valores instantáneos de los registros de IRMS, VRMS, potencias, factor de potencia y frecuencia.

Para la obtención de las medidas se ordenan lecturas de 32 bits de los registros que almacenan los valores de las diferentes magnitudes, se multiplican por su respectiva constante de conversión y se almacenan en una estructura. Se mantiene el mismo procedimiento de la biblioteca de la marca en todas las magnitudes a excepción de la energía activa, que en lugar de μWh se obtendrá en dWh con la finalidad de tener la menor pérdida de resolución en el posterior cálculo de consumo en euros.

4.3.1.6 Procesado de datos

API de precios

```
{
  "date": "21-06-2024",
  "hour": "13-14",
  "is-cheap": false,
  "is-under-avg": false,
  "market": "PVPC",
  "price": 158.13,
  "units": "€/MWh"
}
```

Figura 51. Estructura de JSON de respuesta para una hora determinada

Dentro del JSON obtenido como respuesta a la petición HTTPS, tenemos una serie de datos organizados en una estructura de clave-valor. Este será procesado para que quede reflejado en *precios_hoy.txt* únicamente el valor de la clave “date” en la primera fila y el valor de “price” en €/kWh asignado a cada hora en el resto de filas. Por ejemplo, el precio correspondiente al rango “00-01” quedará almacenado en la segunda fila, el correspondiente al rango “01-02” en la tercera y así sucesivamente. Cabe mencionar que esta decisión se tomó inicialmente por la posibilidad de solicitar la lista de precios completa, sin embargo, finalmente se decidió solicitar el precio individual de cada hora ya que el microcontrolador no siempre recibía el JSON completo con la otra forma, por lo que algunas horas quedaban sin precio. Aun con precios individuales, se mantienen las veinticuatro líneas del fichero *precios_hoy.txt* ya que es una organización más polivalente que puede servir para ambos tipos de JSON modificando el mínimo número de líneas del *sketch*, esto daría a un futuro proyectista que parta de este proyecto la posibilidad de elegir el método que mejor se adapte a las capacidades del microcontrolador que esté usando.

Con esto, se obtiene el siguiente fichero:


```

2024-06-21 13:43:51
0.00000 [0]
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.15813 [13]
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000
0.00000 [23]

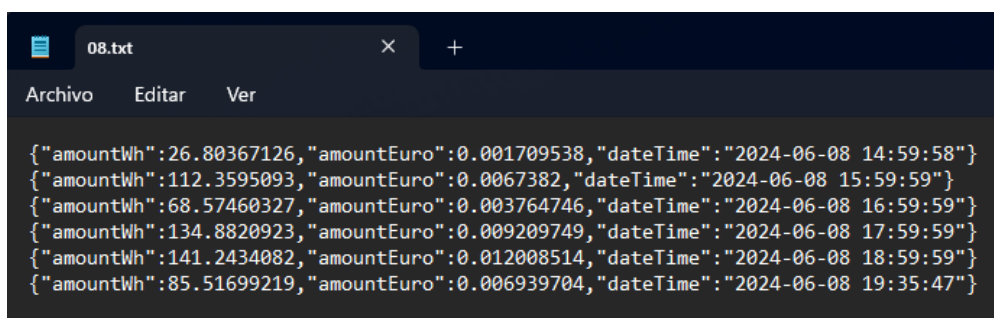
```

Figura 52. Ejemplo de fichero precios_hoy.txt

Acumulación de energía

Para llevar a cabo un registro de la energía consumida se utilizan dos ficheros:

1. El primero es **energyPerSecond.txt**, se actualiza cada segundo y almacena el valor total de la acumulación en dWh a lo largo del rango de hora en el que se encuentra el medidor. En este fichero se encuentra tanto el *timestamp* de la última vez que se actualizó el fichero como el valor de la acumulación. El motivo de uso es evitar perder el total de acumulación de energía en caso de desconexión del medidor. La razón de acumular en dWh es que en Arduino las variables de tipo float y double tienen una precisión máxima de 7 dígitos decimales [39], y dado que originalmente la lectura de energía es en μWh y el factor de conversión es €/kWh, usar una unidad de medida intermedia entre kWh y μWh permite reducir la pérdida de resolución, así como la cantidad de cómputo necesario en las operaciones. También hay información relativa a las operaciones con punto flotante que indica que se pueden evaluar valores de hasta 10^{-12} [40], por lo que al ser información contradictoria se ha decidido ajustar la acumulación a la más restrictiva.
2. El segundo es **/mes/dia.txt**, el cual se actualiza cada hora con el consumo total en Wh, euros y *timestamp*.



```
{\"amountWh\":26.80367126,\"amountEuro\":0.001709538,\"dateTime\":\"2024-06-08 14:59:58\"}
{\"amountWh\":112.3595093,\"amountEuro\":0.0067382,\"dateTime\":\"2024-06-08 15:59:59\"}
{\"amountWh\":68.57460327,\"amountEuro\":0.003764746,\"dateTime\":\"2024-06-08 16:59:59\"}
{\"amountWh\":134.8820923,\"amountEuro\":0.009209749,\"dateTime\":\"2024-06-08 17:59:59\"}
{\"amountWh\":141.2434082,\"amountEuro\":0.012008514,\"dateTime\":\"2024-06-08 18:59:59\"}
{\"amountWh\":85.51699219,\"amountEuro\":0.006939704,\"dateTime\":\"2024-06-08 19:35:47\"}
```

Figura 53. Ejemplo JSON Lines de energía

En este caso no es un archivo con formato JSON, sino que es cada línea la que tiene el formato puesto que son datos independientes y, de esta forma, se facilita la manipulación tanto en el *sketch* de Arduino como en el código de la aplicación móvil.

Adicionalmente, para evitar solapamientos entre medidas y con el fin de ajustar la aplicación a las especificaciones del anteproyecto, antes de cada escritura se comprueba si existe el fichero *mes* correspondiente a hace seis meses, en tal caso se borra. Es decir, se tendrán almacenados consumos correspondientes a un intervalo máximo de seis meses.

4.3.1.7. Estructura de datos

Finalmente, los ficheros mencionados en los puntos anteriores quedarán organizados de la siguiente forma:

```
└ ./
  └ mes/
    └ día.txt
  └ precios_hoy.txt
  └ energyPerSecond.txt
```

4.3.2 Aplicación móvil

Con el fin de desarrollar una aplicación compatible con los sistemas operativos *iOS* y *Android* se ha decidido utilizar *Flutter*, un *framework* de desarrollo de aplicaciones multiplataforma creado por *Google* [41].

Su característica más destacada es la capacidad para construir aplicaciones para distintas plataformas (*Android*, *iOS*, *Windows*, *macOS*, *Linux* y navegador) a partir de una única base de

código escrito en *Dart*, lo cual hace que se reduzcan los costos de desarrollo y mantenimiento, así como el tiempo de lanzamiento al mercado.

Por otro lado, al disponer de una amplia comunidad de desarrolladores y el soporte directo de *Google*, resulta sencillo encontrar información y soluciones en las diferentes partes del desarrollo de la aplicación.

Otras ventajas que se pueden encontrar:

- **Hot Reload:** se trata de una funcionalidad que permite al desarrollador visualizar los cambios realizados de forma instantánea, sin necesidad de reiniciar la aplicación, lo cual agiliza el proceso de desarrollo y depuración.
- **Amplia gama de *widgets* y bibliotecas:** *Flutter* ofrece *widgets* personalizables y preconstruidos que facilitan la creación de interfaces de usuario atractivas y consistentes en las distintas plataformas.

4.3.2.1 Requisitos para el desarrollo

El código desarrollado es compartido para las distintas plataformas, aunque la compilación plantea una serie de diferencias y limitaciones según el sistema operativo:

Android

Las aplicaciones para este sistema operativo pueden desarrollarse y compilarse en *Windows* (versión 10 o superior), *macOS* (versión 10.15 o superior), *Linux* (*Debian* a partir de v.11 y *Ubuntu* a partir de 20.04) y *ChromeOS* ya que se trata de un sistema de código abierto [42].

En cuanto a *hardware*, se recomienda un procesador con un mínimo de 8 núcleos, 16 GB de RAM, monitor FHD y 70 GB de espacio libre en disco [42].

Por la parte del *software*, es necesario una cuenta de *Google* y tener instalado *Flutter SDK* y *Android Studio* junto a una serie de componentes especificados en la guía de instalación correspondiente a cada sistema operativo [42] [43].

iOS

En este caso, al tratarse de un sistema cerrado, el desarrollo es más complejo. Aunque la codificación puede hacerse en entornos que corren en los mismos sistemas operativos que los mencionados anteriormente, la compilación solo puede hacerse dentro de *macOS* por lo que es requisito indispensable disponer de un equipo que cuente con ello. El motivo por el que se requiere de *macOS* es porque para la compilación se usa *XCode*, entorno que proporciona *Apple* asociado a una cuenta de *iCloud* para el desarrollo de *software* enfocado a sus productos [43]. Existen varias formas de ejecutarlo:

- **Dispositivo Mac:** es el método idóneo si se pretende desarrollar *software* de forma asidua, sin embargo, no se contempla en este proyecto debido a su alto coste.

- **Instancia remota de Mac:** equipo físico al que se accede de manera remota por medio de un servidor intermediario como pueden ser los de AWS [44]. Existen de pago y gratuitos, sin embargo, tampoco se ha tenido en cuenta dado que los pagos no los cubre el departamento y requiere introducir la contraseña personal de *iCloud* asociada al dispositivo móvil para enlazarlo con *Xcode* [45], lo cual puede suponer un riesgo en servidores gratuitos. Además, requiere de una configuración excesivamente complicada si se pretende usar el dispositivo móvil personal para depurar. La capacidad de cómputo también es insuficiente si se pretende emular (en los que no son de pago). Otro problema añadido es que se complica el requisito de que medidor y dispositivo móvil deben estar en la misma red WiFi.
- **Máquina virtual con macOS:** es el método elegido. Mediante *VMWare* se emulará *macOS 14 Sonoma*. En este caso se ejecuta *VMWare* en *Windows* y será requisito indispensable utilizar como *host* un equipo con un procesador *Intel* ya que en la documentación de *Flutter SDK* para *macOS* se indica que solo se pueden usar procesadores *Intel* o *Apple silicon* y estos últimos están reservados exclusivamente para los equipos del caso α . Si bien existen formas de emular *macOS* en equipos con procesadores *AMD*, suele haber más problemas de incompatibilidad, hay menos información relacionada con la configuración y, además, *Flutter* para *macOS* no baraja esta arquitectura, por lo que su uso en desarrollo para *iOS* no está recomendado. La configuración de la máquina virtual se muestra en la Figura 54.

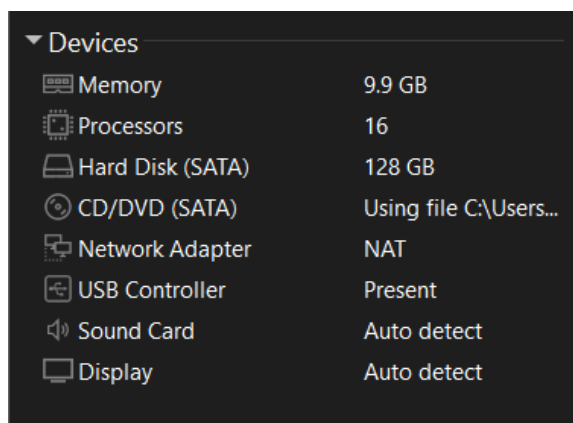


Figura 54. Configuración VMware para emular macOS Sonoma

Cabe mencionar que, por defecto, *VMWare* no está preparado para ejecutar *macOS*, por lo que habrá que realizar una configuración previa del *host* (esta se describe en el punto 1.2 del capítulo *Manual de usuario*).

Independientemente de la plataforma, no es necesaria ninguna licencia de pago durante el proceso de desarrollo. Por el contrario, si se quiere publicar la aplicación en las tiendas de cada sistema, será indispensable comprar una cuenta de desarrollador, cuyo precio varía dependiendo de la plataforma [46] [47].

4.3.2.2 Estructura

Proyecto Flutter

Tal y como se ha mencionado anteriormente, a partir de un solo código fuente (almacenado en la carpeta *lib*) se generan los ficheros correspondientes a la configuración para cada plataforma (en este caso *iOS* y *Android*), por último, el fichero *pubspec.yaml* se encarga de reunir todas las librerías usadas en el proyecto (Figura 55).

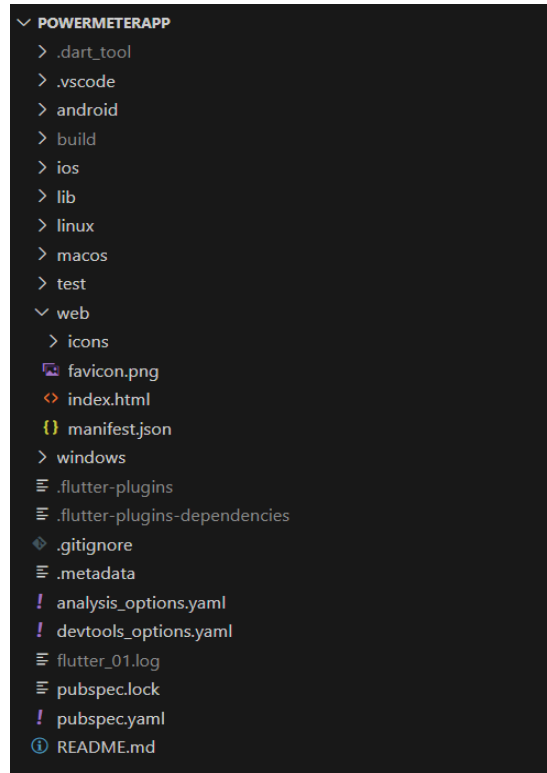


Figura 55. Estructura del proyecto

Aplicación

Se organiza en el siguiente árbol de ficheros dentro de la carpeta *lib*, tal y como se muestra en la Figura 56:

- *Data* contiene todos los ficheros relacionados con la organización de la información correspondiente al historial de consumo recibido por parte del medidor.
- *Datetime* contiene un fichero que sirve de herramienta para obtener claves para mapas y listas a partir de un instante de tiempo.
- *MQTT* contiene los ficheros correspondientes a la creación del cliente MQTT, su configuración en *mqtt_manager* y la gestión de los mensajes recibidos en cada *topic* en *mqtt_power_state* (bróker/*measure*) y *mqtt_power_state* (bróker/*register*).
- *Presentation* incluye los ficheros encargados de la creación de los diferentes elementos a representar, quedando divididos en *ítems*, para los elementos más pequeños; y *screens* para las dos ventanas de las que dispone la aplicación. Será en estas últimas donde se muestre la suma del resto de elementos.

- *Main* es el fichero que controla los elementos en la capa más alta de la jerarquía de la aplicación.

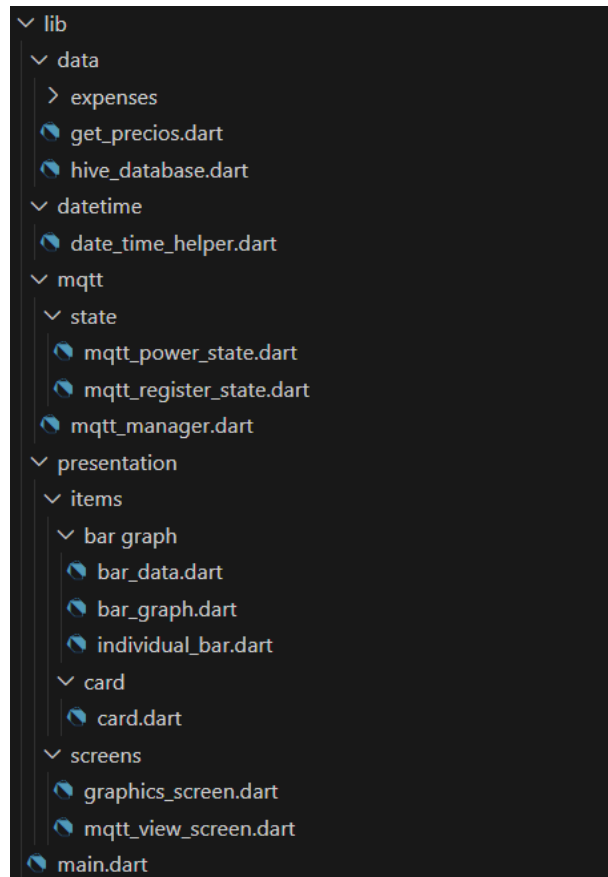


Figura 56. Esquema del árbol de ficheros de la carpeta lib

4.3.2.3 Funcionamiento

Este sigue la estructura del diagrama mostrado en la Figura 57:

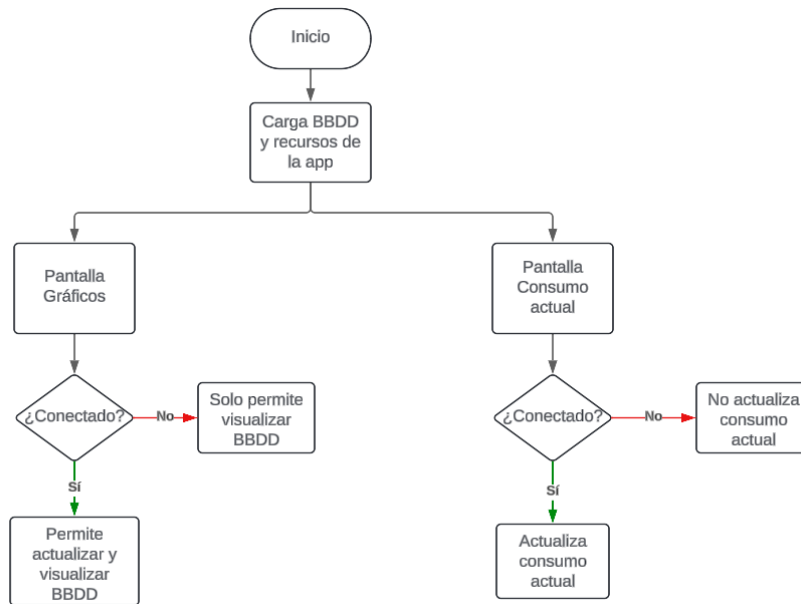


Figura 57. Diagrama de flujo de la aplicación

Para lograr este funcionamiento la aplicación ejecuta diferentes tareas, las cuales se pueden agrupar en base a:

Presentación de interfaz de usuario: permiten al usuario interactuar con la aplicación directamente mediante la visualización de datos o la orden de conexión al medidor. Tras el inicio de la aplicación se puede acceder a dos pantallas:

- **Consumo instantáneo:**

Tiene una función doble, permite conectar y desconectar la aplicación del medidor, así como visualizar el estado de conexión (conectado, conectando o desconectado) y las medidas de VRMS, IRMS, potencia activa, reactiva y aparente, factor de potencia y frecuencia.

- **Gráficos:**

También es doble su función puesto que permite visualizar el historial de consumo existente en la BBDD mediante gráficos de barras, así como ordenar la obtención del historial de medidas almacenados en la tarjeta *microSD* en caso de que la aplicación esté conectada a la ESP8266.

Obtención de datos: estos, en esencia, se obtienen únicamente por medio de comunicación inalámbrica con el bróker MQTT (ESP8266). La aplicación móvil, por su parte, actúa como cliente y dentro de la misma red podrán existir varios.

Suponiendo que el teléfono móvil y la ESP8266 están conectados a la misma red WiFi y, este último está transmitiendo con éxito, se realiza el siguiente proceso para la obtención de los datos:

- Al pulsar *Conectar* dentro de la pantalla *Consumo instantáneo*, se ejecuta un escaneo de la red WiFi en busca del servicio *_mqtt._tcp*, el cual se proporciona en la dirección *esp8266.local* tal y como se configuró en el apartado 4.3.1.3. Posteriormente se resuelve su dirección IP y, junto con la del teléfono móvil se crea el cliente MQTT. Esta última se usa para identificar al cliente.
- Una vez se ha creado el cliente, *mqtt_manager* se encarga de configurar el cliente para que se ajuste a la configuración del bróker, suscribirlo a los topics *broker/measure* y *broker/register* y crear las *callbacks* necesarias para la desuscripción de *topics*, escucha y envío de mensajes en los diferentes canales. Una vez se encuentra todo configurado, se procede a realizar la conexión.
- Si la conexión es incorrecta, se vuelve a intentar al cabo de un tiempo. Si es correcta, el cliente permanecerá escuchando en los *topics* a los que se ha suscrito.
- En *broker/measure* el cliente será únicamente oyente dado que la información se envía de forma automática por parte del bróker cada segundo.
- En el caso de *broker/register*, el cliente pasará a actuar también como publicador ya que será ese mensaje el que sirva como comando al bróker (que también es oyente del *topic*) para que ejecute el envío de datos.
- Si bien los mensajes, en primera instancia, son recibidos en *mqtt_manager*, *mqtt_power_state* será el responsable de procesar los provenientes de *broker/measure* y *mqtt_register_state* los de *broker/register*.

Los mensajes de ambos *topics* tienen estructuras JSON distintas, contenido distinto y se usan de forma diferente ya que, los de *broker/measure* se representan en la pantalla *Consumo instantáneo* nada más se reciben y, por otra parte, los de *broker/register* se guardan en una base de datos para posteriormente ser representados. El flujo de datos se puede representar de la siguiente forma:

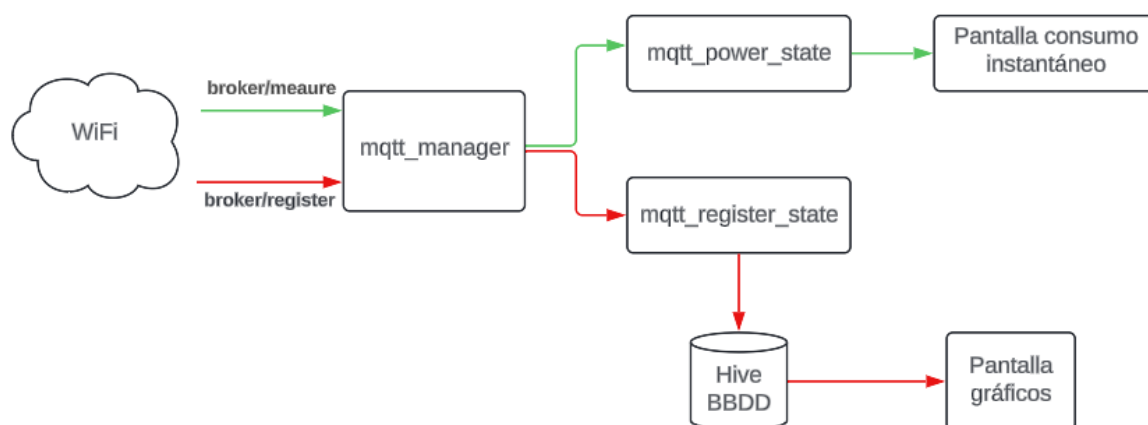


Figura 58. Diagrama superficial del procesamiento y uso de los mensajes MQTT

Gestión de datos: por último, los datos recibidos deben procesarse para poder hacer uso de ellos. En ambos *topics* se reciben JSON *Lines*, sin embargo, la estructura del contenido será distinta. Se procesará de la siguiente forma para cada pantalla:

- **Consumo instantáneo:** en *mqtt_power_state* se convertirá cada JSON *Line* recibido en un objeto de la clase *PowerData*, cuyos parámetros corresponden con las magnitudes a representar. Tras cada conversión, se notificará a los elementos que hagan uso de esta información que los datos a representar han cambiado, para lograrlo se hace uso de *Provider*, cuyo concepto se explicará más adelante.

Una vez se dispone de los datos, mediante *mqtt_view_screen* se representan en la pantalla por medio de tarjetas asignadas a cada magnitud, estas tarjetas se apilan en filas dentro de una columna, construyendo así el widget que permitirá visualizar los datos en su conjunto.



Depuración mensaje MQTT en broker/meause

Figura 59. Ejemplo de funcionamiento de la pantalla Consumo instantáneo en iPhone

- **Gráficos:** a *mqtt_register_state* pueden llegar hasta un máximo de 24 JSON *Lines* por mensaje, por lo que en primer lugar se procede a separar las líneas individualmente, después, cada línea se convierte a un objeto de la clase *ExpenseItem*, cuyos parámetros son *String amountWh*, *String amountEuro* y *DateTime dateTime*. Cada uno de estos ítems

se añade a una lista de objetos *ExpenseItem* y cuando se han añadido todos los del mensaje, se notifica a los elementos que hacen uso de esta información mediante *Provider* para que puedan saber cuándo hay nuevos datos disponibles.

Es en *graphics_screen* donde se obtiene la lista del último mensaje recibido y se añade cada *ExpenseItem* individual a la lista general *overallExpenseList*, la cual almacena todos y cada uno de los consumos de energía recogidos por el medidor a lo largo del tiempo.

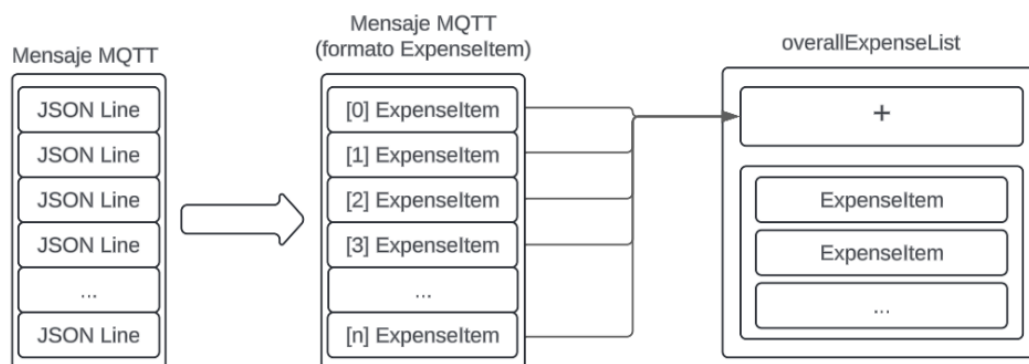


Figura 60. Proceso para añadir datos a lista de consumo a partir de mensaje MQTT

OverallExpenseList es también una copia local de los datos almacenados en la BBDD *Hive* y sirve de intermediario entre esta y la aplicación, por lo que cada vez que se añade o se elimina un gasto de *overallExpenseList*, ha de actualizarse también la BBDD para que, al cerrar la aplicación, cuando se abra de nuevo, se disponga de la misma información que había antes de cerrarla. Una vez más, cada vez que se actualice la lista general, se deberá notificar mediante *provider* a todos los elementos que hagan uso de ella pues, de otra forma, aunque los datos cambian, no se verían afectadas las representaciones visuales de los mismos hasta que se volviera a recargar el elemento que hace uso de la información.

Teniendo la totalidad de los datos, se procede a crear métodos de clasificación en función del instante de tiempo en *expense_data*, para ello se hace uso mapas clave (*tiempo*): valor (*consumo*). Esto permitirá agrupar las medidas por horas y días mediante la conversión del parámetro *dateTime* a un tipo de clave específica.

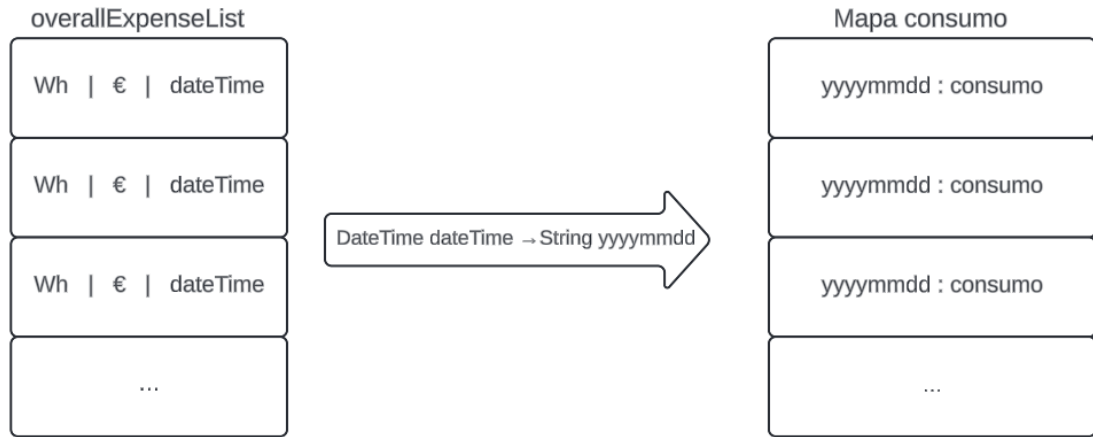


Figura 61. Conversión de lista de datos a mapa de consumo

En la Figura 61 se muestra un diagrama que representa la iteración sobre *overallExpenseList* y su posterior organización por medio de la clave *yyyyymmdd* (cuatro dígitos de año, dos dígitos de mes y dos dígitos de día), por lo que todos los consumos correspondientes al mismo día se verán agrupados. Este proceso se ejecuta tanto para el consumo en Wh como en euros y para *yyyyymmddhh* puesto que también se representará en un gráfico el consumo por hora durante un día.

Una vez se tienen los mapas, se usan sus claves para establecer el rango a evaluar, por ejemplo, para evaluar la primera semana de abril de 2024 se utilizarán como límites las claves *20240401* y *20240407*.

Finalmente, cada consumo de este rango se introduciría en una lista de *double* y dicha lista se usará como coordenadas para representar el gráfico de consumo que se muestra en la Figura 62.

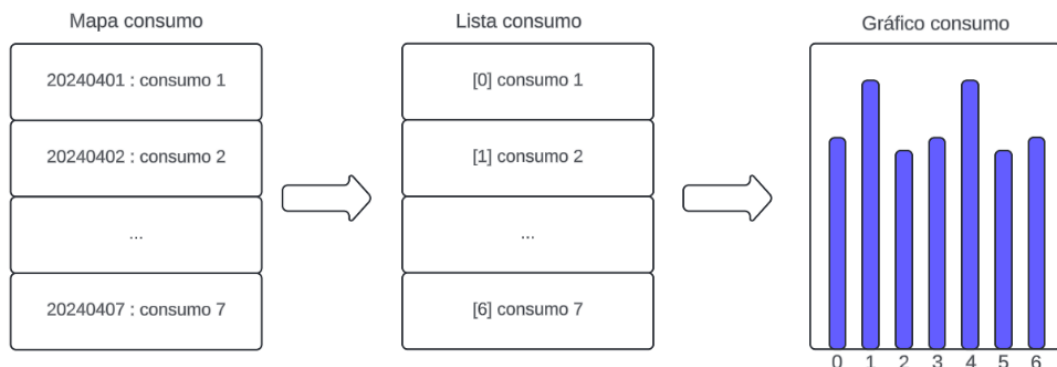


Figura 62. Ejemplo de obtención de datos para gráfico a partir de mapa de consumo

Si bien este proceso no es exactamente igual para todos los gráficos, representa el algoritmo usado para la organización de los datos. La clave reside en la correcta manipulación de los

mapas y rangos que permitirán obtener la lista de consumo pues esta podrá ser representada directamente (gráfico por horas y de una semana) o deberán agruparse de nuevo (gráfico mensual). Las listas se manipularán en *expense_summary*, los gráficos en *bar_graph* y tanto los rangos como si se quieren mostrar unidades de energía o euros en *graphics_screen*.

Tal y como se mencionó anteriormente, cuando se actualiza un dato que está siendo representado visualmente, es necesario notificarlo ya que, de otra forma, la información mostrada en la pantalla no se refresca.

Se utiliza la biblioteca *Provider* ya que permite comunicar elementos que hacen uso de una misma información sin la necesidad de que estén conectados directamente entre ellos ni mostrándose ambos por pantalla, aquí entra el concepto de árbol de *widgets*:

Todo elemento visual que se representa en pantalla se llama *widget*, y puede estar formado a su vez por la combinación de otros *widgets*. Con el fin de usar eficientemente los recursos del dispositivo móvil, *Flutter* elimina de memoria los elementos no mostrados por pantalla, así como los datos de los que se surte. *Provider* permite alojar estos datos en la capa elegida por el desarrollador dentro del árbol de *widgets*, siendo en este caso la del *main*, pues la destrucción del *widget* correspondiente a la pantalla de *Consumo instantáneo* provocaría una desconexión del cliente MQTT, gastando de esta forma una de las cinco desconexiones que soporta el bróker antes de su reinicio.

5. Resultados

El proyecto consta de diferentes partes, sin embargo, en este apartado será evaluado el funcionamiento final de la aplicación, el cual implica el correcto funcionamiento de las partes individuales como pueden ser la PCB adaptadora, el medidor o la aplicación. Como elemento de apoyo principal para la corroboración de los datos recogidos por el medidor desarrollado se ha utilizado el analizador eléctrico Fluke 43B, el cual está disponible en el departamento. Ha de mencionarse que dicho analizador no ha sido calibrado desde su adquisición por lo que no servirá como referencia de precisión sino de coherencia en los resultados.

Para realizar este estudio se han realizado pruebas con cargas resistivas y reactivas por separado, así como combinadas entre ellas. Los resultados más representativos se encuentran en las siguientes tablas:

5.1. Prueba con carga resistiva de 200 Ω

	Vrms [V]	Irms [A]	Pot. Act [W]	Pot. Reac [VAr]	Pot. Apar [VA]	FP	Frec [Hz]
Fluke 43B	226.3	1.142	258	3	258	1	50
ADE915 3A	223.48	1.12	250.55	4.94	250.58	1	49.98
Teoría	230	1.15	264.5	0	264.5	1	50

Tabla 3. Resultado de prueba con carga resistiva de 200 Ω

5.2. Prueba con carga resistiva e inductiva: R = 200 Ω y L = 125 mH @1.8 Ω en serie

	Vrms [V]	Irms [A]	Pot. Act [W]	Pot. Reac [VAr]	Pot. Apar [VA]	FP	Frec [Hz]
Fluke 43B	228.4	1.105	247	62	255	0.97	50
ADE9153 A	226.8	1.09	240.34	58.6	247.53	0.97	49.97
Teoría	230	1.11	248.39	48.38	253	0.98	50

Tabla 4. Resultado de prueba con carga R = 200 Ω y L = 125 mH

5.3. Prueba con carga resistiva de 50 Ω

	Vrms [V]	Irms [A]	Pot. Act [W]	Pot. Reac [VAr]	Pot. Apar [VA]	FP	Frec [Hz]
Fluke 43B	229.4	4.678	1060	13	1060	1	50
ADE9153A	226	4.58	1039.55	16.83	1040.29	1	50.04
Teoría	230	4.6	1058	0	1058	1	50

Tabla 5. Resultado de prueba con carga resistiva de 50 Ω

5.4. Prueba con carga resistiva en inductiva: $R = 50 \Omega$ y $L = 125 \text{ mH}$ @1.8 Ω en serie

	Vrms [V]	Irms [A]	Pot. Act [W]	Pot. Reac [VAr]	Pot. Apar [VA]	FP	Frec [Hz]
Fluke 43B	227.8	3.317	570	491	752	0.75	50
ADE9153 A	225.47	3.25	549.84	484.57	733.1	0.75	50.03
Teoría	230	3.54	648.54	491.66	813.85	0.79	50

Tabla 6. Resultado de prueba con carga $R = 50 \Omega$ y $L = 125 \text{ mH}$

5.5. Prueba con carga resistiva y capacitiva: $R = 50 \Omega$ y $C = 10.7 \mu\text{F}$ en serie

	Vrms [V]	Irms [A]	Pot. Act [W]	Pot. Reac [VAr]	Pot. Apar [VA]	FP	Frec [Hz]
Fluke 43B	230.3	0.794	33	178	180	0.18	50
ADE9153 A	227.31	0.78	31.19	172.43	176.8	0.18	50.01
Teoría	230	0.762	27.09	172.93	175.36	0.17	50

Tabla 7. Resultado de prueba con carga $R = 200 \Omega$ y $C = 10.7 \mu\text{F}$

En todos los casos se puede apreciar cómo las medidas siguen lo esperado, ha de tenerse en cuenta que las capturas no están hechas en el mismo instante de tiempo, sino que hay cierto tiempo entre una y otra, por lo que este puede ser uno de los motivos de la variación entre el medidor Fluke y el ADE9153A. También se denota una diferencia entre el resultado teórico esperado y el de los medidores, sin embargo, hay que tener en cuenta que la onda de tensión de la red no es exactamente una sinusoidal pura, sino que puede variar un poco y, además, la corriente cuenta con una serie de armónicos que pueden afectar a la medida y que no se están teniendo en cuenta para el cálculo teórico. No obstante, en términos generales, los resultados se dan por válidos ya que están alineados con lo esperado.

En cuanto a los resultados del apartado de registro de consumo en Wh y euros, durante la redacción de este PFG se ha tenido conectado un portátil, después se ha sumado la carga de un móvil y finalmente una torre PC.

Los resultados del apartado de gráficos se encuentran adjuntos en el punto 2 del capítulo *Manual de usuario* tomándose como ejemplo el consumo del día 14 de junio de 2024.

6. Presupuesto

Se puede desglosar en diferentes conceptos correspondientes al hardware, software o usado para ambos. Esta lista se representa en la Tabla 8: [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58]

Concepto	Hardware	Software	Común
ESP8266 D1 Mini	2.56 € [48]	-	-
EV-ADE9153A-SHIELDZ	177.51€ [49]		
Adaptador microSD	5.268 € [50]	-	-
MicroSDHC 32GB	15.99€ [51]	-	-
Conectores macho "low profile"	2€ [52]	-	-
Shuko hembra	5.10€ [53]	-	-
Shuko macho	1.89€	-	-
Caja	26.75€ [54]	-	-
Licencia Eagle Autodesk	-	-	Licencias UPM
Licencia Fusion360 Autodesk	-	-	Licencias UPM
Prototipo PCB	12.48€	-	-
PCB Final	16.62€	-	-
Tablet XiaomiPad5	-	213€ [55]	-
iPhone X	-	129€ [56]	-
Ordenador ASUS TUF 15	-	-	769€ [57]
Licencia VMWare Workstation 17 Pro	-	9.90€ [58]	-
Mano de obra	-	-	3462€
Total	5029.06		

Tabla 8. Desglose de gastos

En cuanto al coste de la mano de obra, se tendrán en cuenta las horas invertidas en el desarrollo de este PFG y el sueldo aproximado de un ingeniero de telecomunicaciones junior. Por lo que si se convierten los 12 ECTS a horas se obtienen 300 y, suponiendo un sueldo de 11.54€/hora (24000€ anuales), supondría un coste de 3462€.

El uso de equipos físicos para depurar el software no es estrictamente necesario ya que también se pueden emular, sin embargo, esto obliga a que el ordenador cuente con mejores especificaciones técnicas. También son un factor a tener en cuenta las bibliotecas, ya que muchas no cuentan con características de red para el emulador, que en este caso son necesarias por el uso del protocolo MQTT dentro de una red WiFi. Por todo esto, se han usado tanto emuladores como dispositivos físicos para la depuración del software.

7. Impacto del proyecto

Desde el punto de vista de los ODS (Objetivos de Desarrollo Sostenible) [59], este proyecto se alinea de forma directa con el ODS 11 (Ciudades y comunidades sostenibles), ODS 12 (Producción y consumo responsables) y ODS 13 (Acción por el clima) ya que conociendo de forma precisa cuál es la calidad del consumo de una serie de equipos, es más fácil llegar a una solución sostenible, en la que se aproveche de la mejor manera la energía genera y su canal de transporte. Un mejor aprovechamiento y un incremento en la eficiencia y calidad del consumo, se ve reflejado de forma directa en la huella de carbono, suponiendo así una acción positiva por el clima. De otra forma, también puede servir de apoyo en el ODS 7 (Energía asequible y no contaminante) en los procesos de medida o, en el ODS 4 (Educación de calidad) ya que el desarrollo de proyectos como este permite al alumno ampliar conocimientos y conocer mejor el marco económico y tecnológico.

Como tal, se trata de un proyecto que, si bien requiere cierto conocimiento acerca de distintas temáticas, no requiere un desarrollo demasiado profundo una vez planteado el algoritmo del programa, por lo que, siendo útil en el punto actual, también deja abierta la posibilidad de escalarlo tanto fuera de la red local como la posibilidad de recoger más datos.

En este caso, al ser un proyecto individual, probablemente la huella de carbono asociada a él (por los componentes, fabricación de PCB, plásticos usados, energía usada durante el desarrollo...) sea significativamente alta, sin embargo, es común que en la fase de desarrollo sea donde más recursos se gasten y que lo que más se optimice sea la fase de producción.

Por las especificaciones del sistema, no está dirigido a grandes empresas, pero si a pequeños consumidores o PYMES que quieran controlar de una manera más precisa su consumo y considerar las soluciones oportunas en base a ello. Como se ha mencionado antes, utilizando las mismas tecnologías cabe la posibilidad de escalar a una red de medida, consiguiendo de esta forma controlar el consumo de más cargas desde una misma aplicación, así como la posibilidad de mejorar la aplicación. Puede ser interesante continuar investigando en este proyecto.

Finalmente, el mayor impacto desde el punto de vista del autor del proyecto es el acercamiento del consumidor a un mayor control y entendimiento de sus consumos puesto que, tras realizar diferentes cuestiones a posibles usuarios, era en el apartado del PVPC y el de la facilidad de uso lo que más dudas generaba.

8. Conclusiones

8.1. Conclusiones

Tras la realización de este Proyecto Fin de Carrera se ha conseguido obtener un medidor que cumple con las necesidades básicas que fueron planteadas desde el principio, la posibilidad de medir la calidad del consumo de una manera fácil e intuitiva sin descuidar la precisión de estas.

Además, se ha hecho uso de los principales recursos previstos en el Anteproyecto (como por ejemplo Flutter, Eagle o el protocolo MQTT), lo que significa que se hizo un estudio previo correcto del proyecto a desarrollar. También se ha respetado el tiempo asignado para el desarrollo y escritura de la memoria ya que se han llevado a cabo dentro del plazo estimado sin aumentar el número horas dedicadas por encima de los 12 ECTS asignados.

Se trata de un proyecto muy interesante ya que abarca diferentes partes del desarrollo de un producto, desde la concepción de este hasta la presentación final en una interfaz de usuario, precisando de conocimientos técnicos de diferentes ramas de la electrónica y telemática.

En cuanto a la parte de electrónica ha sido necesario, en primer lugar, tener claros los conceptos teóricos para el cálculo de la energía con el fin de verificar la validez de los resultados del medidor. También ha sido preciso refrescar conceptos clave de cursos anteriores para entender el procesado digital de la señal que hace el integrado o de *Instrumentación Electrónica* para seguir el proceso de medida, así como la conversión analógica-digital.

Si bien no todas las tecnologías usadas han sido vistas durante la carrera, las asignaturas de telemática cursadas han supuesto una base fundamental para entenderlas. Por mencionar los principales ejemplos, el conocimiento previo en Java facilitó el aprendizaje de Dart, o en el caso de la seguridad en las peticiones HTTPS, pudo comprenderse mejor gracias a los conocimientos adquiridos en *Seguridad en redes y servicios*.

8.2. Trabajos futuros

Aunque al final de este proyecto se haya obtenido un medidor y una aplicación completamente funcionales, como se mencionó en el Apartado 7, se usan una serie de tecnologías como *MQTT* que permiten escalar la red de medidores – aplicación. Sería interesante una adaptación al microcontrolador ESP32 por su mayor capacidad de cómputo y la posibilidad de usar RTC, aliviando así la continua sincronización (para mantener la precisión de la hora) que se hace con la ESP8266.

Adicionalmente, también se puede hacer una comparación entre tener el bróker MQTT instalado en un servidor externo (*RaspberryPi* dentro de la misma red o bróker remoto) o en el mismo microcontrolador. Externalizar el bróker supondría una menor necesidad de cómputo, lo que permite aprovechar más el microcontrolador, dando así lugar al posible diseño de un analizador de red a partir de la señal *AI_WAV* y un software que permita procesar la señal (fundamental y armónicos).

En el apartado de la aplicación, si bien se cuenta con una funcionalidad básica, pueden incluirse más funcionalidades como una ventana *Perfil* en la que se pueda configurar la tarifa de cada consumidor, visualizar el PVPC diario, cambiar el idioma u otros tipos de personalización. Por último, mejorar la interfaz de usuario para una navegación más fluida entre gráficos de consumo o la representación de las medidas de *Consumo actual* en un gráfico de serie temporal.

9. Referencias

- [1] N. Mohan, T.M. Undeland y W. P. Robbins, *Power Electronics: converters, applications, and design*, 3rd ed. New York: John Wiley & Sons, 2003.
- [2] J. Tapia Niquen, «Memoria de TFG EEBE,» [En línea]. Available: https://upcommons.upc.edu/bitstream/handle/2117/394543/Memoria_de_tfg_eebe_Jemima_Niquen_Tapia%20%281%29.pdf?sequence=2&isAllowed=. [Último acceso: 28 mayo 2024].
- [3] A. D. Inc, «ADE9153A: Single-Phase, Multifunction Metering IC with autocalibration,» [En línea]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ade9153a.pdf>. [Último acceso: 28 mayo 2024].
- [4] Microchip Technology Inc., «Atmel-46002-SE-M90E26-Datasheet,» [En línea]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-46002-SE-M90E26-Datasheet.pdf>. [Último acceso: 28 mayo 2024].
- [5] Texas Instruments, «MSP430AFE253 Datasheet,» [En línea]. Available: https://www.ti.com/lit/ds/symlink/msp430afe253.pdf?ts=1717435911464&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FMSP430AFE253. [Último acceso: 28 mayo 2024].
- [6] STMicroelectronics, «STPM32 Datasheet,» [En línea]. Available: <https://www.st.com/resource/en/datasheet/stpm32.pdf>. [Último acceso: 28 mayo 2024].
- [7] Maxcio, «Power Meter,» [En línea]. Available: <https://m.media-amazon.com/images/I/A1C5Db5iVBL.pdf>. [Último acceso: 28 mayo 2024].
- [8] Tapo, «User Guide Mini Smart Wi-Fi Socket, Energy Monitoring Tapo P110,» [En línea]. Available: <https://m.media-amazon.com/images/I/A1UDI8aNMHL.pdf>. [Último acceso: 28 mayo 2024].
- [9] Tapo, «TP-Link Tapo P110 - Mini Enchufe Inteligente Wi-Fi (con Monitoreo Energético),» Amazon, [En línea]. Available: https://www.amazon.es/%E3%80%90Nuevo%E3%80%91-TP-Link-Tapo-P110-Inteligente/dp/B09BFT7NZJ/ref=sr_1_1?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&sr=8-1. [Último acceso: 28 mayo 2024].
- [10] P. Instruments, «Analizador de potencia PCE-PA 8000,» PCE Instruments, [En línea]. Available: <https://www.pce-instruments.com/espanol/instrumento->

- medida/medidor/analizador-de-potencia-pce-instruments-analizador-de-potencia-pce-pa-8000-det_100494.htm. [Último acceso: 28 mayo 2024].
- [11] Asociación Española de Normalización, «62053-21:2021, UNE-EN IEC,» Asociación Española de Normalización, [En línea]. Available: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0065594>. [Último acceso: 3 junio 2024].
- [12] M. Ger, «uMQTTBroker,» GitHub, [En línea]. Available: <https://github.com/martinger/uMQTTBroker>. [Último acceso: 3 junio 2024].
- [13] A. Heredia, «mSure® Autocalibration with the ADE9153A,» [En línea]. Available: <https://www.analog.com/media/en/technical-documentation/application-notes/an-1571.pdf>. [Último acceso: 3 junio 2024].
- [14] Analog Devices, «ADE9153A Energy Measurement Shield with mSure Autocalibration,» [En línea]. Available: <https://www.analog.com/media/en/technical-documentation/user-guides/ev-ade9153ashieldz-ug-1253.pdf>. [Último acceso: 4 junio 2024].
- [15] Analog Devices, «Data Sheet ADE9153A,» [En línea]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ade9153a.pdf>. [Último acceso: 3 junio 2024].
- [16] Analog Devices, «Functionality and Features of the ADE9153A Energy Metering IC with Autocalibration,» [En línea]. Available: <https://www.analog.com/media/en/technical-documentation/user-guides/ade9153a-technical-reference-manual-ug-1247.pdf>. [Último acceso: 4 junio 2024].
- [17] Espressif Systems, «ESP8266 Hardware Design Guidelines,» [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/esp8266_hardware_design_guidelines_en.pdf. [Último acceso: 5 junio 2024].
- [18] R. Feranec, «Don't place PCB antenna into middle of your board. Here is why | ESP32,» [En línea]. Available: <https://www.youtube.com/watch?v=vX3l0iRxi8c>. [Último acceso: 5 junio 2024].
- [19] U. P. d. C. -. L. d. i. Bioenginyeria, «Guía básica de EMC en el diseño de circuitos impresos,» [En línea]. Available: http://ieb-srv1.upc.es/gieb/tecniques/pdf/EMC_PCBs.pdf. [Último acceso: 1 junio 2024].
- [20] Advanced Circuits, «The Importance of Power and Ground Planes in PCB Design,» [En línea]. Available: <https://www.4pcb.com/blog/the-importance-of-power-and-ground-planes-in-pcb-design/#:~:text=Ground%20planes%20provide%20a%20common,and%20ensuring%20a%20steady%20supply.>+. [Último acceso: 5 junio 2024].

-
- [21] Random Nerd Tutorials, «ESP8266 Pinout Reference: Which GPIO pins should you use?,» [En línea]. Available: <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>. [Último acceso: 6 junio 2024].
- [22] Electronic Circuits Design, «SD MicroSD Card pinout,» [En línea]. Available: <https://www.electroniccircuitsdesign.com/pinout/sd-microsd-card-pinout.html>. [Último acceso: 6 junio 2024].
- [23] P. Sierra Panizo, Proyecto Fin de Grado ETSIST UPM. Desarrollo de un medidor de energía eléctrica con conexión Wifi operado a través de una aplicación de ordenador, Madrid, 2024.
- [24] Red Eléctrica, «E-sios,» Red Eléctrica, [En línea]. Available: <https://www.ree.es/es/actividades/operacion-del-sistema-electrico/e-sios#:~:text=Red%20El%C3%A9ctrica%20ha%20desarrollado%20un,espec%C3%ADficamente%20con%20el%20mercado%20el%C3%A9ctrico..> [Último acceso: 7 junio 2024].
- [25] arduino-libraries/SD, «ADE9153A Library,» GitHub, [En línea]. Available: <https://github.com/analogdevicesinc/arduino/tree/master/Arduino%20Uno%20R3/libraries/ADE9153A>. [Último acceso: 6 junio 2024].
- [26] Analog Devices, «ADE9153AAPI_Test.ino,» GitHub, [En línea]. Available: https://github.com/analogdevicesinc/arduino/blob/master/Arduino%20Uno%20R3/examples/ADE9153A_examples/ADE9153AAPI_Test/ADE9153AAPI_Test.ino. [Último acceso: 6 junio 2024].
- [27] Analog Devices, «ADE9153AAPI.cpp,» GitHub, [En línea]. Available: <https://github.com/analogdevicesinc/arduino/blob/master/Arduino%20Uno%20R3/libraries/ADE9153A/ADE9153AAPI.cpp>. [Último acceso: 6 junio 2024].
- [28] Arduino, «Arduino & Serial Peripheral Interface (SPI),» [En línea]. Available: <https://docs.arduino.cc/learn/communication/spi/>. [Último acceso: 6 junio 2024].
- [29] arduino-libraries, «Sd2Card.cpp,» GitHub, [En línea]. Available: <https://github.com/arduino-libraries/SD/blob/master/src/utility/Sd2Card.cpp>. [Último acceso: 7 junio 2024].
- [30] G. Martín, «ESPNTpClient,» GitHub, 7 junio 2022. [En línea]. Available: <https://github.com/gmag11/ESPNTpClient?tab=readme-ov-file>. [Último acceso: 8 junio 2024].
- [31] D. Mills, J. Martin, J. Burbank, W. Kasch, «IETF Datatracker,» Internet Engineering Task Force (IETF), junio 2010. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc5905>. [Último acceso: 8 junio 2024].

- [32] BandaAncha.eu, «Lista de servidores NTP stratum 1 en España,» [En línea]. Available: https://wiki.bandaancha.st/Lista_de_servidores_NTP_stratum_1_en_Espa%C3%B1a. [Último acceso: 8 junio 2024].
- [33] L. Llamas, «Cómo configurar un ESP8266 o ESP32 para acceder por mDNS,» [En línea]. Available: <https://www.luisllamas.es/como-configurar-un-esp8266-para-acceder-por-mdns/>. [Último acceso: 8 junio 2024].
- [34] L. Llamas, «¿Qué es MQTT? Su importancia como protocolo IoT,» [En línea]. Available: <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>. [Último acceso: 8 junio 2024].
- [35] L. Llamas, «Principales broker MQTT Open Source para proyectos IoT,» [En línea]. Available: <https://www.luisllamas.es/principales-broker-mqtt-open-source-para-proyectos-iot/>. [Último acceso: 8 junio 2024].
- [36] Random Nerd Tutorials, «ESP8266 NodeMCU HTTPS Requests (Arduino IDE),» [En línea]. Available: <https://randomnerdtutorials.com/esp8266-nodemcu-https-requests/#why-https-esp8266>. [Último acceso: 8 junio 2024].
- [37] E. C. Forum, «Arduino/libraries/ESP8266HTTPClient/examples,» GitHub, [En línea]. Available: <https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266HTTPClient/examples>. [Último acceso: 8 junio 2024].
- [38] IBM, «How TLS provides identification, authentication, confidentiality, and integrity,» IBM, [En línea]. Available: <https://www.ibm.com/docs/en/ibm-mq/9.1?topic=tls-how-provides-identification-authentication-confidentiality-integrity>. [Último acceso: 9 junio 2024].
- [39] Arduino, «float - Arduino Reference,» Arduino, [En línea]. Available: <https://www.arduino.cc/reference/en/language/variables/data-types/float/>. [Último acceso: 24 junio 2024].
- [40] Arduino, «Floating Point Constants,» Arduino, [En línea]. Available: <https://www.arduino.cc/reference/en/language/variables/constants/floatingpointconstants/>. [Último acceso: 24 junio 2024].
- [41] Flutter, «Flutter,» Flutter, [En línea]. Available: <https://flutter.dev/>. [Último acceso: 9 junio 2024].
- [42] Flutter, «Start building Flutter Android apps on Windows,» Flutter, [En línea]. Available: <https://docs.flutter.dev/get-started/install/windows/mobile>. [Último acceso: 9 junio 2024].

- [43] Flutter, «Start building Flutter Android apps on macOS,» Flutter, [En línea]. Available: <https://docs.flutter.dev/get-started/install/macos/mobile-android>. [Último acceso: 9 junio 2024].
- [44] A. W. S. Inc, «Instancias de Mac de Amazon EC2,» Amazon Web Services, [En línea]. Available: <https://aws.amazon.com/es/ec2/instance-types/mac/>. [Último acceso: 9 junio 2024].
- [45] F. Herrera, «[S3/L10] Flutter Móvil: De cero a experto 2023 - Mac: Probar en un iPhone físico,» YouTube, [En línea]. Available: <https://www.youtube.com/watch?v=U8XkrZxjNds&list=PLCKuOXG0bPi0sIn-nDsi7ma9OV6MEMkxj&index=41&t=274s>. [Último acceso: 10 junio 2024].
- [46] Google, «Crear una cuenta de desarrollador de Play Console,» Google, [En línea]. Available: <https://play.google.com/console/u/0/signup>. [Último acceso: 10 junio 2024].
- [47] Apple Inc, «Apple Developer Program,» Apple, [En línea]. Available: <https://developer.apple.com/programs/whats-included/>. [Último acceso: 10 junio 2024].
- [48] E. O. Store, «Placa de desarrollo WIFI Mini, placa NodeMCU Lua IOT de 3,3 V con pines, D1 Mini ESP8266, ESP-12, CH340G, V2, USB, WeMos D1,» AliExpress, [En línea]. Available: https://es.aliexpress.com/item/1005001621784437.html?spm=a2g0o.productlist.main.3.3.2a6AFgUAFgUbX&algo_pvid=84614af8-53f2-4460-bbd6-96eea5b658b1&utparam-url=scene%3Asearch%7Cquery_from%3A. [Último acceso: 11 junio 2024].
- [49] Digi-Key Electronics, «EV-ADE9153ASHIELDZ,» Digi-Key Electronics, [En línea]. Available: <https://www.digikey.es/es/products/detail/analog-devices-inc/EV-ADE9153ASHIELDZ/9484663>. [Último acceso: 11 junio 2024].
- [50] RS, «Conector para tarjeta de memoria MicroSD 3M serie 2900 de 8 contactos, paso 1.1mm, 1 fila, montaje superficial,» RS, [En línea]. Available: <https://es.rs-online.com/web/p/conectores-para-tarjeta-de-memoria/0212799?searchId=7a8d7139-4243-4cdc-b844-79b6010bb5ac&gb=s>. [Último acceso: 11 junio 2024].
- [51] Kingston, «Kingston SDC10G2/32GB - Tarjeta microSD de 32 GB,» Amazon, [En línea]. Available: <https://www.amazon.es/Kingston-SDC10G2-32GB-Tarjeta-microSD/dp/B0162YQG2I>. [Último acceso: 11 junio 2024].
- [52] RS, «Regleta de pines Samtec serie TLW de 16 vías, 1 fila, paso 2.54mm, para soldar, Montaje en orificio pasante,» RS, [En línea]. Available: <https://es.rs-online.com/web/p/conectores-macho-para-pcb/7656186>. [Último acceso: 11 junio 2024].

- [53] RS, «Conector de potencia industrial, Formato 2P + E, Orientación Recto, Azul, 230 V, 16A, IP54,» RS, [En línea]. Available: <https://es.rs-online.com/web/p/bases-y-clavijas-industriales/2144099>. [Último acceso: 11 junio 2024].
- [54] RS, «Caja para instrumentación OKW de ABS Gris, 150 x 80 x 50mm, IP40,» RS, [En línea]. Available: <https://es.rs-online.com/web/p/cajas-para-equipos/0583117?searchId=88b4c080-12d4-475e-818a-c6f4c78ae440&gb=s>. [Último acceso: 11 junio 2024].
- [55] MediaMarkt, «Tablet - Xiaomi Redmi Pad SE, 256 GB, Gris grafito, 11" Full-HD+, 8 GB RAM, Snapdragon® 680, Android,» MediaMarkt, [En línea]. Available: https://www.mediemarkt.es/es/product/_tablet-xiaomi-redmi-pad-se-256-gb-gris-grafito-11-full-hd-8-gb-ram-snapdragonr-680-android-1564784.html?gad_source=1. [Último acceso: 11 junio 2024].
- [56] S. Mobiles, «Apple iPhone X - 64GB - 256GB Unlocked Grey Silver - Gris Plata (Libre) - B,» eBay Inc., [En línea]. Available: https://www.ebay.es/itm/196056841076?chn=ps&var=495957153975&norover=1&mkrid=1185-146825-5486-0&mksid=101&itemid=495957153975_196056841076&targetid=1988336187007&device=c&mkttype=pla&googleloc=1005493&poi=&campaignid=19493756154&mkgroupid=142286609262&rls. [Último acceso: 11 junio 2024].
- [57] PC Componentes, «ASUS TUF Gaming F15 FX507ZC4-HN002 Intel Core i7-12700H/16GB/512GB SSD/RTX 3050/15.6",» PC Componentes, [En línea]. Available: https://www.pccomponentes.com/asus-tuf-gaming-f15-fx507zc4-hn002-intel-core-i7-12700h-16gb-512gb-ssd-rtx-3050-156?s_kwid=AL!14405!3!!!x!!&gad_source=1. [Último acceso: 11 junio 2024].
- [58] Codigi, «Vmware Workstation 17 Pro / 17.5 Pro Lifetime License,» Codigi, [En línea]. Available: https://www.codigi.es/producto/vmware-workstation-17-5-pro-lifetime-license/?gad_source=1. [Último acceso: 11 junio 2024].
- [59] Naciones Unidas, «Objetivos de Desarrollo Sostenible,» Naciones Unidas, [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. [Último acceso: 11 junio 2024].
- [60] ESP8266 Community, «arduino.esp8266.com,» GitHub, [En línea]. Available: https://arduino.esp8266.com/stable/package_esp8266com_index.json. [Último acceso: 13 junio 2024].
- [61] Álvarez Rivera, Volker, «PowerMeterSketch,» GitHub, [En línea]. Available: <https://github.com/VolkerRivera/PowerMeterSketch>. [Último acceso: 18 junio 2024].

- [62] F. Herrera, «Curso - Flutter Móvil: de Cero a Experto,» YouTube, [En línea]. Available: <https://www.youtube.com/playlist?list=PLCKuOXG0bPi0sIn-nDsi7ma9OV6MEMkxj>. [Último acceso: 19 junio 2024].
- [63] Álvarez Rivera, Volker, «PowerMeterApp,» GitHub, [En línea]. Available: <https://github.com/VolkerRivera/PowerMeterApp>. [Último acceso: 19 junio 2024].
- [64] Owicron, «Como instalar macOS Sonoma en VMware,» YouTube, [En línea]. Available: <https://www.youtube.com/watch?v=h8oLSDUxDEw&t=198s>. [Último acceso: 19 junio 2024].
- [65] rayw_reddit, «[Workaround/Temporary fix/solution] Intel 12th Gen Alder Lake Performance Problem/Issue on VMware Workstation (12900K/12700K/12600K and their F and non-K variants) on Windows 11,» Reddit, [En línea]. Available: https://www.reddit.com/r/vmware/comments/v0rzt8/workaroundtemporary_fixsolution_intel_12th_gen/. [Último acceso: 19 junio 2024].
- [66] Intel Corporation, «Tecnología Intel® Hyper-Threading,» Intel Corporation, [En línea]. Available: <https://www.intel.la/content/www/xl/es/architecture-and-technology/hyper-threading/hyper-threading-technology.html>. [Último acceso: 19 junio 2024].
- [67] Ferrestock, «Amazon.es,» Ferrestock, [En línea]. Available: https://www.amazon.es/Ferrestock-FSKBLP190-Destornillador-Busca-Met%C3%A1lica/dp/B074J9Z4B8/ref=sr_1_6?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&s=tools&sr=1-6. [Último acceso: 19 junio 2024].

Anexo

Anexo A. Esquemático y gerbers de la PCB adaptadora

Como se mencionó en el apartado 4.2.1 fue necesaria la creación de una PCB adaptadora, en la Figura 63 se muestra el esquemático correspondiente a las Figuras 18, 28 y 29 que describe el conexionado entre los pines de los diferentes elementos. Como se comentó en el mismo apartado, la PCB adaptadora tiene el mismo *pinout* que la placa Arduino Uno por lo que a efectos prácticos será este bloque el que la represente.

Este conexionado se rutea en la capa de *copper*, que posteriormente se exporta como *gerber* para su envío a fábrica junto a otras capas como *soldermask*, *silkscreen*, *solderpaste* y *outline*. Las del top y bottom se pueden visualizar en las Figuras 65 y 66.

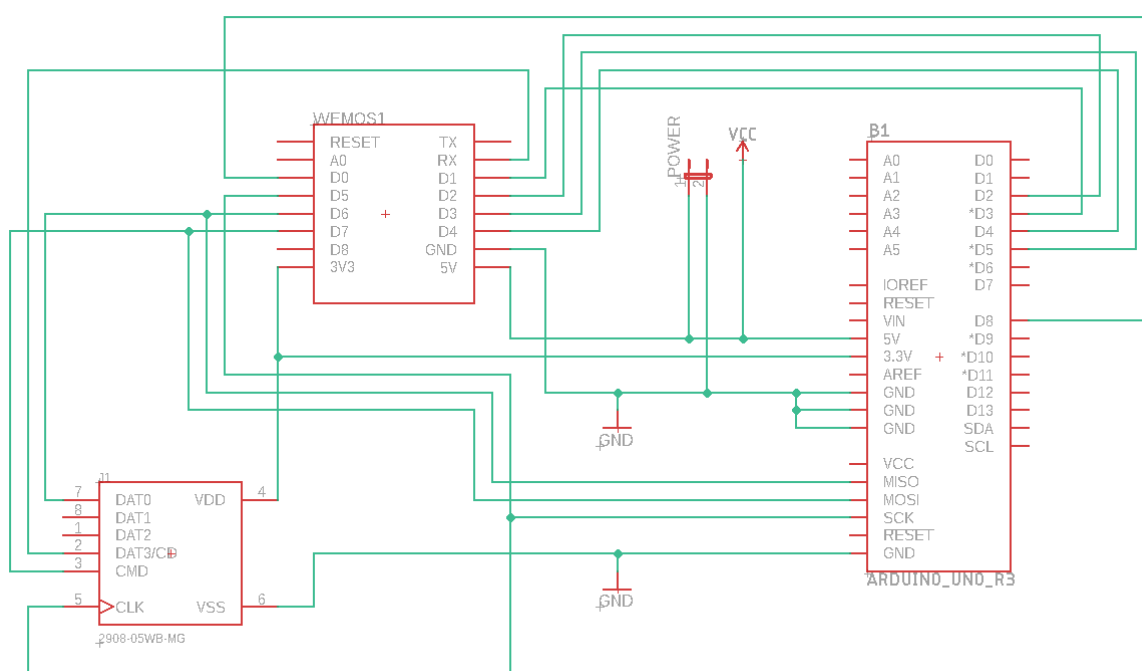


Figura 63. Esquemático de conexión microSD, D1 Mini y pines Arduino

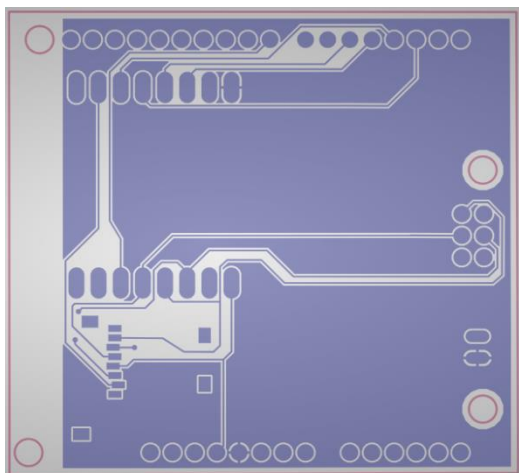


Figura 64. Plano de cobre de la capa top

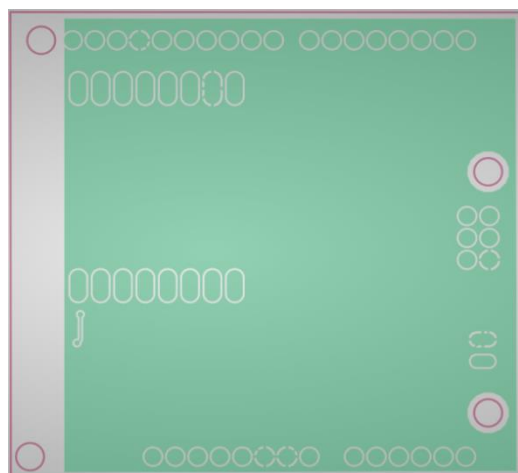


Figura 65. Plano de cobre de la capa bottom

Manual de usuario

A continuación, se explicarán los pasos principales a seguir para poner en marcha el entorno de desarrollo si se quisiera replicar o modificar la aplicación de Arduino o móvil. También se incluyen instrucciones de uso con los pasos a seguir para poner en marcha el medidor.

1. Configuración de entorno para el desarrollador

1.1 Arduino

La versión de Arduino IDE usada para el desarrollo de este proyecto es la 2.3.2 y la biblioteca para la placa de desarrollo la *LOLIN(WEMOS)D1 Mini (clone) v3.0.1*, que se puede encontrar en el repositorio de Arduino ESP8266 [60]. Junto con los ficheros del proyecto se podrán encontrar las bibliotecas utilizadas siendo únicamente necesario copiarlas en la carpeta *Documentos/Arduino/libraries* tal y como se muestra en la Figura 66.

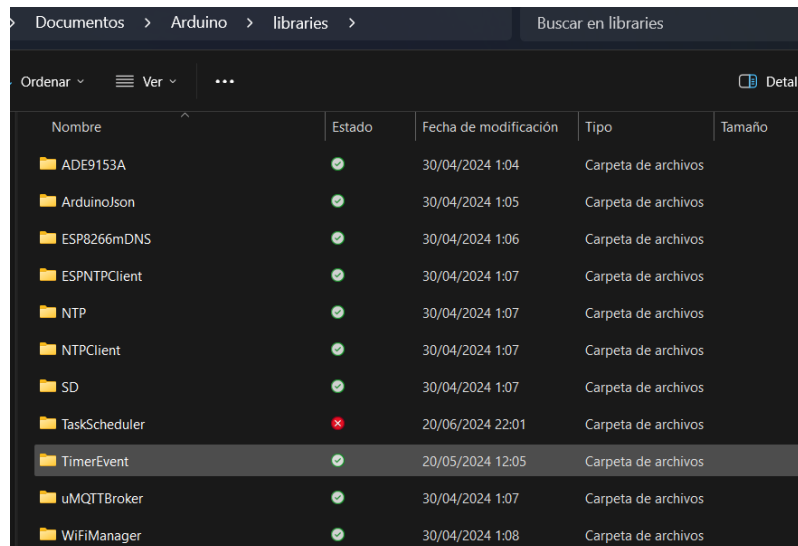


Figura 66. Bibliotecas del proyecto Arduino

Una vez instalados Arduino y las bibliotecas será necesario descargar el proyecto del repositorio de GitHub [61]. Con esto ya se podrá ejecutar la aplicación siempre y cuando se haya seguido correctamente la configuración de *hardware* del apartado 4.2.

1.2 Flutter

En cuanto a la instalación del *framework* de desarrollo para la aplicación móvil, será común tanto para *macOS* como para *Windows*. Los pasos a seguir se describen entre los capítulos 31 y 41 del curso *Flutter Móvil: de Cero a Experto* [62]. Una vez instalado, será necesario descargar el proyecto del repositorio de GitHub [63].

Adicionalmente, si se pretende emular *macOS* en una máquina virtual, es recomendable utilizar *VMWare*. Para su instalación se pueden seguir los pasos indicados en [64] y la configuración del *host*

dependerá del *hardware* del equipo. En primer lugar, es necesario obtener las características del procesador como se puede ver en la Figura 67.

Número de procesador	i7-12700H
Litografía ?	Intel 7
Precio recomendado para el cliente ?	\$502.00
Especificaciones de la CPU	
Cantidad de núcleos ?	14
Cantidad de Performance-cores	6
Cantidad de Efficient-cores	8
Total de subprocesos ?	20
Frecuencia turbo máxima ?	4.70 GHz
Frecuencia de la Tecnología Intel® Turbo Boost Max 3.0 ?	4.70 GHz
Frecuencia turbo máxima del Performance-core ?	4.70 GHz
Frecuencia turbo máxima de Efficient-core ?	3.50 GHz
Caché ?	24 MB Intel® Smart Cache

Figura 67. Características del procesador

Por defecto, para los procesadores *Intel* de duodécima generación *VMWare* prioriza los *efficient-cores* sobre los *performance-cores* cuando se ejecuta en *Windows 11*. Dado que estos están pensados para ahorrar consumo, trabajan a una menor velocidad, lo que se traduce en un peor rendimiento de la máquina virtual pudiendo llegar a ser inoperable. Para solucionar esto, es necesario desactivar los *e-cores* a través de la BIOS o bien seguir los pasos de [65](recomendable).

El resto de las generaciones en *Windows 11* al igual que esta generación en *Windows 10* no dan ningún problema, por lo que no será necesario hacer ninguna configuración en estos casos.

Posteriormente se deberá acceder al *Administrador de tareas – Rendimiento* de *Windows* para saber de cuántos procesadores lógicos dispone el *host*. Dependiendo de la tecnología del procesador cada núcleo podrá ejecutar uno (norma general) o varios procesos (la tecnología *Hyper-Threading* de *Intel* permite que se ejecuten múltiples subprocesos en cada núcleo [66]). En este caso, se dispone de 20 como se puede ver en la Figura 68.

Velocidad de base:	2,30 GHz
Sockets:	1
Núcleos:	14
Procesadores lógicos:	20
Virtualización:	Habilitado
Caché L1:	1,2 MB
Caché L2:	11,5 MB
Caché L3:	24,0 MB

Figura 68. Administrador de tareas - Rendimiento

Sabiendo esto se podrá configurar el procesador de la máquina virtual (Figura 69), donde *Number of processors x Number of cores per processor* debe ser menor que el número de *Procesadores lógicos*.

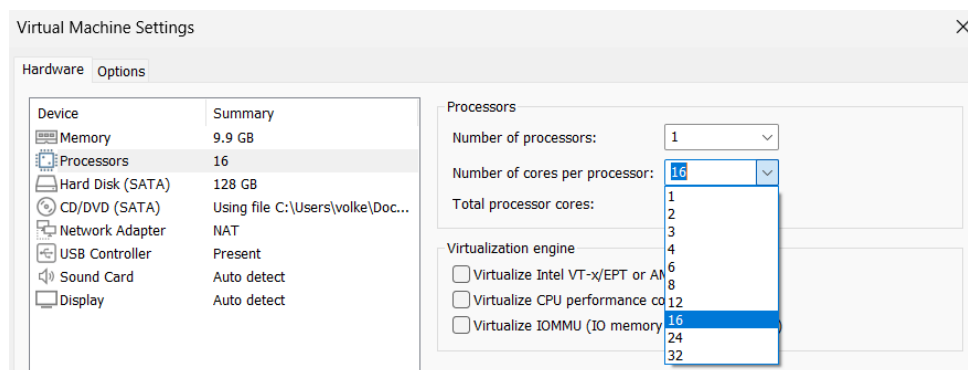


Figura 69. Configuración del procesador de la máquina virtual

En cuanto al apartado *Memmory*, se refiere a la cantidad de memoria RAM (*Random Access Memmory*) asignada a la máquina virtual. El máximo recomendable para asignar es la mitad de la memoria RAM total disponible en el *host*. En el caso de *macOS Sonoma*, que requiere 16 GB de RAM, se ha decidido asignar 9.9 GB, considerando la memoria total disponible en el hardware (16 GB).

El resto de los parámetros se configuran de la misma manera que en [64].

2. Instrucciones para el usuario

2.1 Medidor

Para lograr unas medidas correctas y una conexión exitosa entre el medidor y la aplicación han de seguirse los siguientes pasos:

2.1.1 Conexión del medidor a la red

Es imprescindible que fase y neutro del schuko macho del medidor estén conectados a la fase y el neutro de la toma de red.

En cuanto al medidor, esta distribución viene indicada en la Figura 70.



Figura 70. Fase y neutro en el medidor

Por parte de la toma de red, fase y neutro se pueden identificar mediante un buscapolos o con un multímetro. En el primer caso, bastaría con introducir la punta del buscapolos en una de las tomas mientras se toca la parte metálica del mango tal y como se muestra en la Figura 71 [67], si este se ilumina significa que esa toma es la fase.



Figura 71. Ejemplo de uso de buscapolos

En el segundo caso, se debe configurar el multímetro en modo tensión alterna con un rango de escala superior al de la tensión de red y solo podrá aplicarse en enchufes que cuenten con toma de tierra. Tal y como se representa en la Figura 72, se introducirá una punta de prueba en uno de los agujeros y la otra hará contacto con la toma de tierra, se repetirá el mismo proceso para el otro agujero y, debería haber una diferencia de tensión cercana a 230 V entre fase y tierra; y más próxima a 0 V entre neutro y tierra.

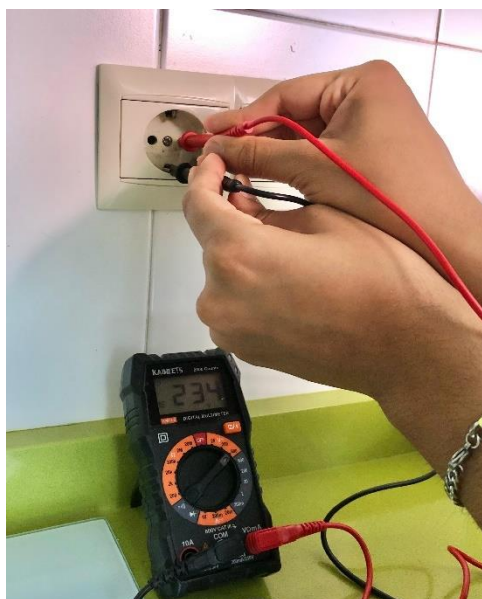


Figura 72. Ejemplo de búsqueda de fase con multímetro



Figura 73. Configuración de multímetro

2.1.2 Conexión a red WiFi

Una vez conectado a la red eléctrica, el medidor procederá a realizar la calibración durante 60 segundos. Posteriormente, procederá a escanear las redes WiFi 2.4 GHz disponibles, con lo que se darán dos escenarios:

Ninguna de las redes disponibles ha sido guardada previamente en el medidor

En este caso dejará de escanear en busca de redes disponibles y creará la suya propia para poder realizar posteriormente la configuración por medio de otro dispositivo. La red WiFi creada por el medidor tiene el SSID "AutoConnectAP" y su contraseña es "password". Mediante un dispositivo distinto al que ofrece el punto de acceso al que deberá conectarse el medidor el usuario debe conectarse a AutoConnectAP, y una vez establecida la conexión se abrirá automáticamente el portal de configuración de red durante 5 minutos. En caso de no abrirse automáticamente se podrá acceder mediante cualquier navegador web a través de la dirección 192.168.4.1 tal y como se muestra en la Figura 74.

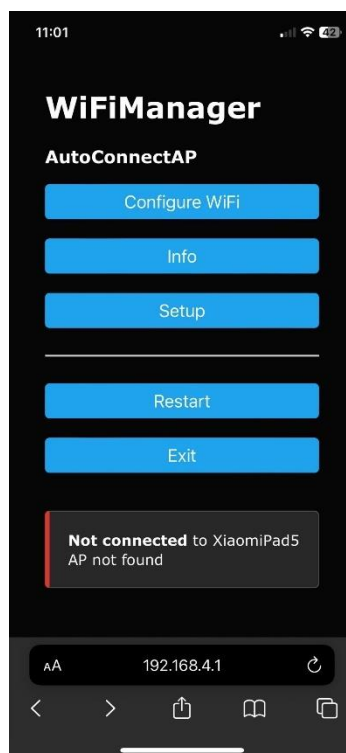


Figura 74. Portal de configuración WiFi del medidor

En el caso que se representa en la Figura 74 se ha abierto el portal de configuración de red WiFi ya que la última a la que estuvo conectado el medidor (XiaomiPad5 AP) no se encuentra disponible actualmente. Una vez abierto este portal, se ha de pulsar en *Configure WiFi* para escanear las redes disponibles y realizar la posterior conexión a la del usuario, tal y como se muestra en la Figura 75.



Figura 75. Conexión a red WiFi

Una vez realizada con éxito la conexión, el programa continuará ejecutándose.

Alguna de las redes disponibles ha sido guardada previamente en el medidor

Si al escanear las redes disponibles el medidor encuentra la última a la que se ha conectado, se conectará automáticamente y el programa seguirá con su ejecución normal.

Tras esto, el medidor no requiere de ningún otro tipo de configuración por parte del usuario. La carga conectada al shuko hembra puede estar presente o no desde el momento de conexión a red según el fabricante del circuito integrado que realiza las medidas.

2.2 Aplicación

Es indispensable que el teléfono móvil se encuentre conectado a la misma red que el medidor para poder conectarse con el medidor. En muchas ocasiones un mismo *router* facilita dos puntos de acceso, uno para la banda de 2.4 GHZ y otro para 5 GHz como se puede ver en la Figura 76.

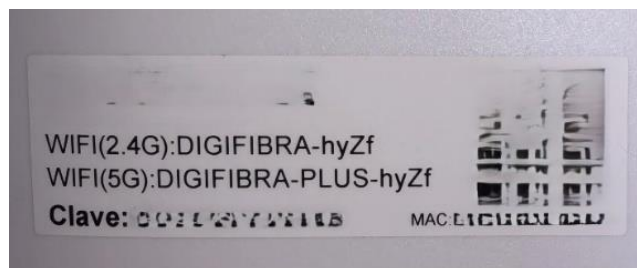


Figura 76. SSID y contraseña del router

Una vez conectado a la red WiFi, se podrá conectar al medidor. Esto abrirá la posibilidad a visualizar el consumo en tiempo real y actualizar el historial de consumo para su posterior representación en los gráficos.

2.2.1 Pantalla de consumo instantáneo

La aplicación cuenta con tres estados que se podrán visualizar en la parte superior de la pantalla, pudiendo alternar entre *Desconectado* en color rojo, *Conectando* en color azul y *Conectado* en color verde. Tras iniciar la aplicación pulsando en su icono (Figura 77) el estado será *Desconectado*, tal y como se puede ver en la Figura 78.

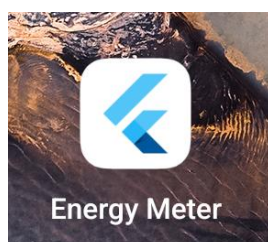


Figura 77. Icono de la aplicación



Figura 78. Estado inicial

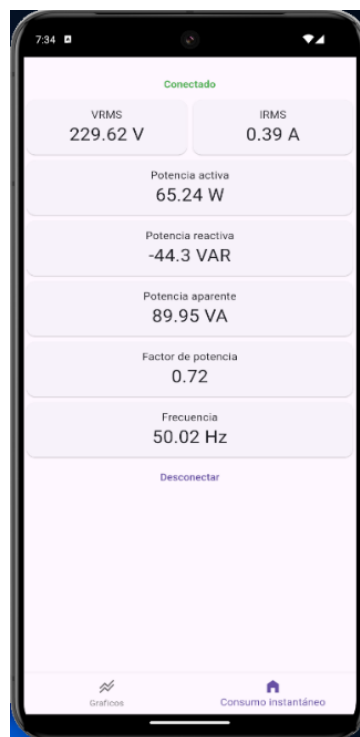


Figura 79. Estado conectado

Tras una pulsación en *Conectar*, se intentará realizar la conexión con el medidor, por lo que pasará al estado *Conectando*. Si ambos dispositivos se encuentran dentro de la misma red WiFi se pasará al estado *Conectado* (Figura 79), en caso contrario se volverá a *Desconectado*.

En el caso específico de la Figura 79 se ha conectado al medidor un ordenador portátil, una lámpara y un cargador de móvil. Diferentes cargas ofrecerán diferentes medidas.

Como se mencionó durante la explicación del desarrollo de la aplicación móvil, esta pantalla estará dividida en pequeñas tarjetas que representan las diferentes magnitudes recogidas por el medidor. Estas medidas se actualizan cada segundo siempre y cuando la aplicación se encuentre conectada al medidor.

Por último, es recomendable desconectar la aplicación antes de cerrarla. En caso de no hacerlo podría reflejarse en un peor desempeño del envío de datos del medidor a otro posible cliente que se encuentre conectado, no obstante, tras un corto período de tiempo el propio medidor desechará la conexión que el cliente dejó abierta.

2.2.2 Pantalla de gráficos

Los datos representados en los gráficos quedan almacenados en el dispositivo móvil por lo que no es necesario estar conectado al medidor para su visualización, pero sí para la obtención de estos. Si nunca se ha utilizado la aplicación o se han eliminado los datos que almacena mediante el gestor de aplicaciones, no existirán datos a representar y la interfaz se visualizará como en las Figuras 80 y 81.

El icono 1 permite solicitar el historial de consumo al medidor, el icono 2 permite alternar entre consumo en euros y unidades de energía; y el icono 3 permite abrir el calendario que permitirá al usuario escoger el día cuyo consumo se ha de representar.



Figura 80. Interfaz de gráficos (superior)



Figura 81. Interfaz de gráficos (inferior)

Tras la petición de los datos almacenados en el medidor se obtendrá una interfaz parecida a la de la Figura 82, en la que los datos representados corresponden al consumo de día actual en euros:

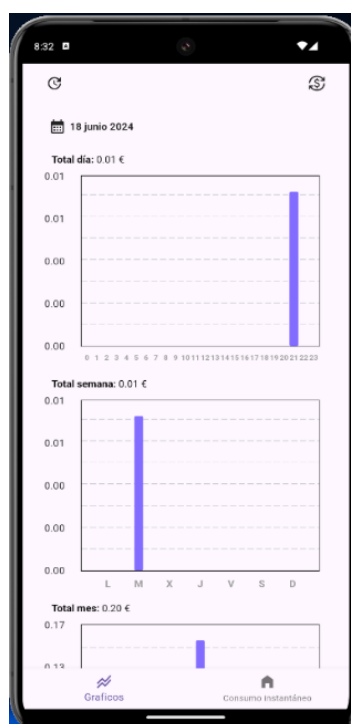


Figura 82. Interfaz tras la petición de datos

Con el objetivo de poder manejar un mayor rango de medidas, las siguientes capturas corresponderán al día 14 de junio de 2024 por la mayor cantidad de datos obtenidos. Para ello se pulsará el icono 3 para abrir el calendario y se elegirá la fecha que se quiera evaluar tal y como se muestra en la Figura 83.

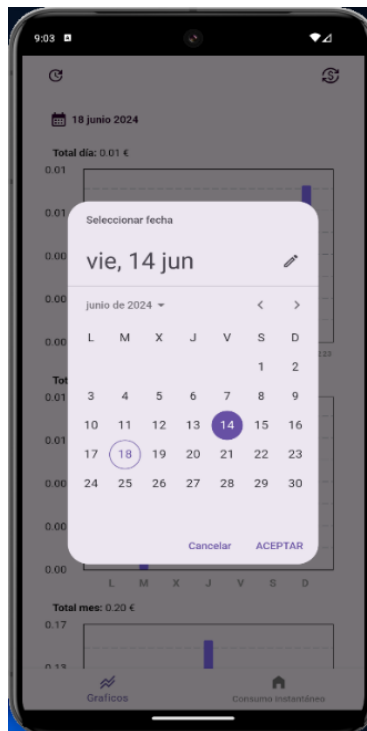


Figura 83. Elección de fecha a evaluar

Tras pulsar *ACEPTAR* se cargarán los gráficos con los datos relativos a la fecha seleccionada. Pulsando sobre el icono 2 se generará el cuadro de la Figura 84 que permitirá al usuario visualizar el consumo en euros (Figura 85) o en unidades de energía (Figura 86).



Figura 84. Elección de tipo de consumo

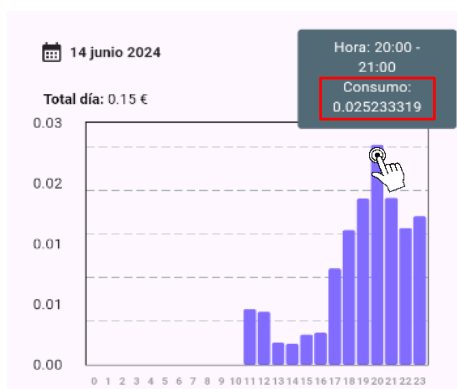


Figura 85. Consumo diario en euros

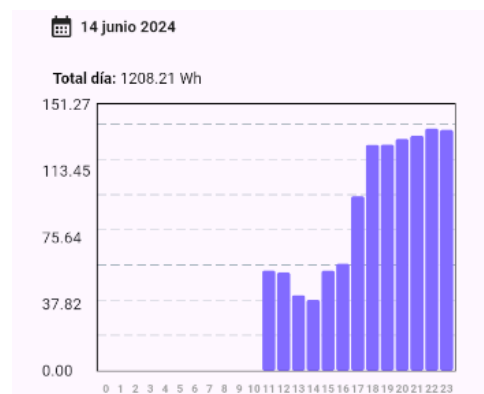


Figura 86. Consumo diario en Wh

Para conocer la información de cada barra, se ha de pulsar en ella tal y como se hace en la Figura 85.

Para navegar entre los diferentes gráficos se debe hacer *scroll*. Los gráficos que se pueden encontrar de arriba abajo son los siguientes:

Gráfico de consumo diario por horas

Son de este tipo los gráficos representados en las Figuras 85 y 86. El eje X del gráfico consta de 24 índices correspondientes a las 24 horas del día. Por su parte, el eje Y del gráfico corresponde a unas divisiones con cantidades orientativas del consumo. El valor Y de cada barra representa el consumo asociado a su índice X.

Se puede comprobar el correcto funcionamiento de los gráficos mediante su fichero asociado, que en este caso es el fichero que se encuentra en `./06/14.txt` dentro de la microSD.

```

14.txt
Archivo  Editar  Ver

{"amountWh":56.78875732,"amountEuro":0.00637397,"dateTime":"2024-06-14 11:59:59"}
{"amountWh":55.71696777,"amountEuro":0.006070921,"dateTime":"2024-06-14 12:59:59"}
{"amountWh":42.79858704,"amountEuro":0.002512705,"dateTime":"2024-06-14 13:59:59"}
{"amountWh":40.19914551,"amountEuro":0.002374563,"dateTime":"2024-06-14 14:59:59"}
{"amountWh":56.75336914,"amountEuro":0.00344493,"dateTime":"2024-06-14 15:59:59"}
{"amountWh":60.77165527,"amountEuro":0.003663315,"dateTime":"2024-06-14 16:59:59"}
{"amountWh":99.07456665,"amountEuro":0.011032944,"dateTime":"2024-06-14 17:59:59"}
{"amountWh":128.2856812,"amountEuro":0.01543405,"dateTime":"2024-06-14 18:59:59"}
{"amountWh":128.4251343,"amountEuro":0.019048016,"dateTime":"2024-06-14 19:59:59"}
{"amountWh":131.5605835,"amountEuro":0.025233319,"dateTime":"2024-06-14 20:59:59"}
{"amountWh":133.4980713,"amountEuro":0.019144958,"dateTime":"2024-06-14 21:59:58"}
{"amountWh":137.5200928,"amountEuro":0.015682791,"dateTime":"2024-06-14 22:59:59"}
{"amountWh":136.8155212,"amountEuro":0.017036269,"dateTime":"2024-06-14 23:59:59"}

```

Figura 87. Fichero que almacena el consumo del día evaluado

Gráfico de consumo semanal por día

En este caso el eje X del gráfico se divide en siete índices que representan los días de la semana en la que se encuentra el día evaluado. El valor Y de cada barra representa el consumo asociado a cada uno de estos. En este caso, si bien el consumo en euros no cambia, el consumo en unidades de energía se representa en kWh puesto que al sumarse todas las horas será necesaria una escala mayor.

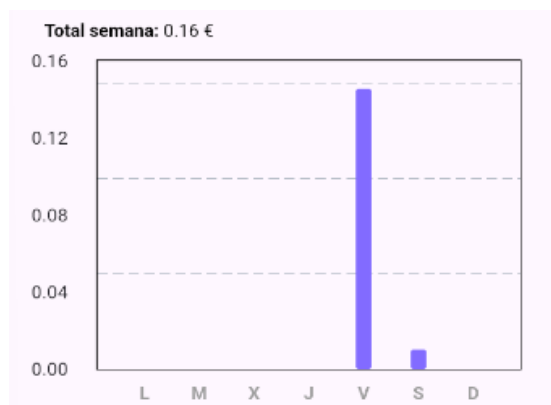


Figura 88. Gráfico de consumo semanal por día

Gráfico de consumo mensual por semanas

Por último, también se podrá visualizar el consumo mensual organizado por semanas tal y como se muestra en la Figura 89.

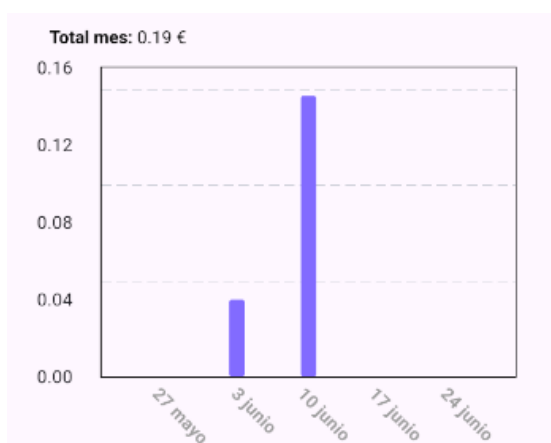


Figura 89. Consumo mensual por semana

Cada una de estas semanas corresponde a las que se pueden encontrar en el calendario (Figura 83) al pulsar el icono 3.