



# FACULTAD DE INGENIERIA

Universidad de Buenos Aires

## CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

### Monitor para medidores de energía eléctrica

**Autor:**

**Ing. Mauricio Barroso Benavides**

Director:

Mg. Ing. Gonzalo Sanchez (F.A.A.)

Jurados:

Mg. Ing. Martín Nicolás Menéndez (pertenencia)  
Mg. Ing. Christian Yañez Flores (pertenencia)  
Esp. Ing. Esteban Daniel Volentini (pertenencia)

*Este trabajo fue realizado en la Ciudad de Salta,  
entre agosto de 2019 y agosto de 2020.*



## *Resumen*

La presente memoria tiene como objetivo describir el desarrollo de un dispositivo electrónico, capaz de monitorear la información de consumo de energía eléctrica proveniente de medidores de uso domiciliario. El trabajo fue propuesto por la cooperativa de servicios eléctricos de la ciudad boliviana de Tupiza, COPELECT, para automatizar la lectura de los medidores que tiene instalados en los hogares de sus más de diez mil abonados.

En la elaboración de este trabajo se ven plasmados los conocimientos adquiridos en la carrera de especialización, sobre ingeniería de software, sistemas operativos en tiempo real I y II, protocolos de comunicación y diseño de circuitos impresos.



## *Agradecimientos*

Esta sección es para agradecimientos personales y es totalmente **OPCIONAL**.



# Índice general

|  |            |
|--|------------|
| <b>Resumen</b>   | <b>III</b> |
| <b>1. Introducción general</b>                             | <b>1</b>   |
| 1.1. Medición del consumo eléctrico domiciliario . . . . . | 1          |
| 1.2. Medición inteligente . . . . .                        | 3          |
| 1.3. Soluciones disponibles en el mercado . . . . .        | 4          |
| 1.4. Motivación . . . . .                                  | 6          |
| 1.5. Objetivos y alcance . . . . .                         | 6          |
| <b>2. Introducción específica</b>                          | <b>7</b>   |
| 2.1. Requerimientos . . . . .                              | 7          |
| 2.1.1. Requerimientos funcionales . . . . .                | 7          |
| 2.1.2. Requerimientos no funcionales . . . . .             | 7          |
| 2.2. Esquema general del sistema . . . . .                 | 8          |
| 2.2.1. Conversor óptico-eléctrico . . . . .                | 8          |
| 2.2.2. Microcontrolador . . . . .                          | 9          |
| 2.2.3. Transceptor Wi-Fi . . . . .                         | 10         |
| 2.2.4. Transceptor LoRa . . . . .                          | 12         |
| 2.2.5. Reloj en tiempo real . . . . .                      | 14         |
| 2.2.6. Memoria no volátil . . . . .                        | 14         |
| 2.3. Planificación . . . . .                               | 16         |
| <b>3. Diseño e implementación</b>                          | <b>19</b>  |
| 3.1. Prototipo de pruebas . . . . .                        | 19         |
| 3.1.1. Microcontrolador + Wi-Fi . . . . .                  | 20         |
| 3.1.2. Transceptor LoRa . . . . .                          | 21         |
| 3.1.3. RTC + EEPROM . . . . .                              | 22         |
| 3.1.4. Conversor óptico-eléctrico . . . . .                | 22         |
| 3.2. Diseño de firmware . . . . .                          | 23         |
| 3.2.1. DATA LOGGER . . . . .                               | 27         |
| 3.2.2. DATA COMMUNICATION . . . . .                        | 29         |
| 3.2.3. WEB SERVER . . . . .                                | 31         |
| 3.3. Interfaz web . . . . .                                | 34         |
| 3.4. Prototipo comercial . . . . .                         | 36         |
| <b>4. Ensayos y resultados</b>                             | <b>41</b>  |
| 4.1. Pruebas unitarias . . . . .                           | 41         |
| 4.2. Pruebas funcionales de firmware . . . . .             | 41         |
| 4.3. Pruebas de la interfaz web . . . . .                  | 43         |
| 4.4. Pruebas de laboratorio . . . . .                      | 46         |
| 4.5. Pruebas del transceptor LoRa . . . . .                | 49         |
| <b>5. Conclusiones</b>                                     | <b>51</b>  |

|                                       |           |
|---------------------------------------|-----------|
| 5.1. Conclusiones generales . . . . . | 51        |
| 5.2. Próximos pasos . . . . .         | 52        |
| <b>Bibliografía</b>                   | <b>55</b> |

# Índice de figuras

|   |    |
|---|----|
| 1.1. Medidor de consumo eléctrico analógico <sup>1</sup> . . . . .  | 2  |
| 1.2. Medidor de consumo eléctrico digital <sup>2</sup> . . . . .  | 2  |
| 1.3. Smart meter de la firma emlite <sup>3</sup> . . . . .  | 3  |
| 1.4. Registrador de pulsos PA-FL de la firma SyxthSense <sup>4</sup> . . . . .  | 5  |
| 1.5. Registrador de pulsos AirTM-100S de la firma iNES <sup>5</sup> . . . . .   | 5  |
| <br>2.1. Diagrama en bloques general del dispositivo. . . . .   | 8  |
| 2.2. Fotoresistencia GL5528 <sup>6</sup> . . . . .  | 9  |
| 2.3. Fototransistor IR333C <sup>7</sup> . . . . .   | 9  |
| 2.4. Tarjeta de desarrollo del fabricante STMicroelectronics basada en el microcontrolador STM32F030R8T6 <sup>8</sup> . . . . . | 10 |
| 2.5. Ubicación de Wi-Fi en el modelo OSi <sup>9</sup> . . . . .   | 11 |
| 2.6. Arquitectura de una red Wi-Fi <sup>10</sup> . . . . .  | 11 |
| 2.7. Módulo Wi-Fi basado en el circuito integrado EMW3162 <sup>11</sup> . . . . .   | 12 |
| 2.8. <i>Stack</i> LoraWAN <sup>12</sup> . . . . .   | 12 |
| 2.9. Arquitectura de una red LoraWAN <sup>13</sup> . . . . .  | 13 |
| 2.10. Módulo LoRa basado en el circuito integrado RF96 <sup>14</sup> . . . . .  | 14 |
| 2.11. Módulo RTC basado en el circuito integrado DS1307 <sup>15</sup> . . . . .   | 14 |
| 2.12. Módulo EEPROM basado en el circuito integrado 24C256 <sup>16</sup> . . . . .  | 15 |
| 2.13. Módulo flash basado en el circuito integrado W25Q16BVSIG <sup>17</sup> . . . . .  | 15 |
| 2.14. Diagrama AON del trabajo. . . . .   | 16 |
| 2.15. Primera parte del diagrama de Gantt. . . . .  | 17 |
| 2.16. Segunda parte del diagrama de Gantt. . . . .  | 17 |
| 2.17. Tercera parte del diagrama de Gantt. . . . .  | 18 |
| <br>3.1. Diagrama en bloques del prototipo de pruebas. . . . .  | 19 |
| 3.2. Tarjeta de desarrollo NodeMCU de la firma Amica <sup>18</sup> . . . . .  | 20 |
| 3.3. Módulo LoRa PM1280 <sup>19</sup> . . . . .   | 21 |
| 3.4. Módulo RTC + EEPROM <sup>20</sup> . . . . .  | 22 |
| 3.5. Módulo detector de luz <sup>21</sup> . . . . .   | 23 |
| 3.6. Diagrama de capas del firmware. . . . .  | 24 |
| 3.7. Diagrama de capas para DATA LOGGER. . . . .  | 27 |
| 3.8. Diagrama de conexión con las herramientas de FreeRTOS de DATA LOGGER. . . . .  | 27 |
| 3.9. Diagrama de flujo de la tarea pulses_task. . . . .   | 28 |
| 3.10. Diagrama de flujo de la tarea alarm_task. . . . .   | 28 |
| 3.11. Diagrama de flujo de la función de inicialización de DATA LOGGER. . . . .   | 29 |
| 3.12. Diagrama de capas para LORA COMMUNICATION. . . . .  | 30 |
| 3.13. Diagrama de flujo de la tarea lora_task. . . . .  | 30 |
| 3.14. Formato de los paquetes enviados y recibidos por DATA COMMUNICATION. . . . .  | 31 |
| 3.15. Diagrama de flujo de la función de inicialización del módulo LORA COMMUNICATION. . . . .                                  | 31 |

|  |    |
|--|----|
| 3.16. Diagrama de capas para WEB SERVER. . . . .   | 32 |
| 3.17. WEB SERVER en modo punto de acceso. . . . .  | 32 |
| 3.18. WEB SERVER en modo estación. . . . .   | 32 |
| 3.19. Diagrama de flujo de la función de inicialización del módulo WEB SERVER. . . . .   | 33 |
| 3.20. Pantalla principal de la interfaz web. . . . .   | 35 |
| 3.21. Pantalla de configuración de la interfaz web. . . . .  | 35 |
| 3.22. Carcasa VG-S43 de la firma Vange <sup>22</sup> . . . . .   | 36 |
| 3.23. Módulo de alimentación HLK-PM03 de la firma Hi-Link <sup>23</sup> . . . . .  | 37 |
| 3.24. Diagrama esquemático del prototipo comercial. . . . .  | 38 |
| 3.25. Capa top del PCB. . . . .  | 39 |
| 3.26. Capa bottom del PCB. . . . .   | 39 |
| 3.27. Modelo 3D del PCB montado del prototipo comercial. . . . .   | 40 |
| 3.28. PCB montado del prototipo comercial. . . . .   | 40 |
| <br>   |    |
| 4.1. Banco de pruebas para evaluar el funcionamiento del firmware. . . . .   | 42 |
| 4.2. Captura de pantalla de idf-monitor cuando el dispositivo inicia. . . . .  | 42 |
| 4.3. Captura de pantalla de idf-monitor cuando el dispositivo ejecuta sus funciones normales. . . . .  | 43 |
| 4.4. Banco de pruebas para verificar el funcionamiento de la interfaz web cuando el dispositivo está en modo punto de acceso. . . . .                      | 43 |
| 4.5. Captura de pantalla de las redes Wi-Fi disponibles en la laptop. . . . .  | 44 |
| 4.6. Captura de pantalla de la página principal de la interfaz web con la utilidad de depuración funcionando. . . . .                                      | 44 |
| 4.7. Captura de pantalla del idf-monitor después de enviar los archivos solicitados por el navegador web y el dispositivo en modo punto de acceso. . . . . | 45 |
| 4.8. Captura de pantalla de la página de configuración de la interfaz web con la utilidad de depuración funcionando. . . . .                               | 45 |
| 4.9. Captura de pantalla del idf-monitor después de configurar el dispositivo en modo estación con los datos enviados por la interfaz web. . . . .         | 46 |
| 4.10. Banco de pruebas para el conversor óptico-eléctrico. . . . .   | 46 |
| 4.11. Salida de la pantalla del osciloscopio. . . . .  | 47 |
| 4.12. Banco de pruebas para el conversor óptico-eléctrico. . . . .   | 47 |
| 4.13. Gráfico de líneas del comportamiento de la fuente de alimentación. . . . .   | 48 |
| 4.14. Captura de pantalla de idf-monitor después de enviar los archivos para la interfaz web. . . . .  | 49 |
| 4.15. Captura de pantalla de la ubicación del gateway LoRa y el prototipo comercial. . . . .   | 49 |

# Índice de Tablas

|   |    |
|---|----|
| 2.1. IEEE 802.11 . . . . .                                  | 10 |
| 3.1. Contenido SPIFFS . . . . .                             | 26 |
| 3.2. Consumo de corriente del prototipo comercial . . . . . | 37 |
| 4.1. Pruebas unitarias . . . . .                            | 41 |
| 4.2. Análisis de cobertura . . . . .                        | 41 |
| 4.3. Prueba de la fuente de alimentación 176 VAC . . . . .  | 47 |
| 4.4. Prueba de la fuente de alimentación 220 VAC . . . . .  | 48 |
| 4.5. Prueba de la fuente de alimentación 264 VAC . . . . .  | 48 |
| 4.6. Parámetros del transceptor LoRa . . . . .              | 50 |



*Dedicado a... [OPCIONAL]*



# Capítulo 1

## Introducción general

En este capítulo se presenta una descripción del proceso que COPELECT realiza para obtener información sobre el consumo eléctrico de sus abonados, nociones sobre medidores inteligentes, una comparación de las soluciones comercialmente disponibles en esta temática, las razones que motivaron al desarrollo del trabajo junto con sus objetivos y alcances.

### 1.1. Medición del consumo eléctrico domiciliario

En los hogares se dispone de diversos dispositivos eléctricos y electrónicos que son utilizados para entretenimiento, labores domésticas, trabajo, etc. La energía eléctrica consumida por estos dispositivos es medida en vatio-hora, simbolizado Wh [1]. El kWh, equivalente a 1000 vatios-hora, se utiliza para la facturación del consumo de energía eléctrica por parte de las compañías prestadoras del servicio [1]. Para este fin, las compañías instalan en los hogares de sus abonados dispositivos llamados medidores, que se encargan de contar la cantidad de kWh consumidos. También, los medidores proporcionan una interfaz para que los funcionarios de dichas compañías puedan registrar la información de consumo eléctrico.

Las mayor parte de compañías prestadoras del servicio eléctrico utilizan principalmente dos tipos de medidores para medir el consumo eléctrico domiciliario. Estos son:

1. Medidores analógicos: contienen un disco giratorio metálico y un contador analógico que indica el total de kWh consumidos. Cuando la corriente fluye a través del medidor, se genera un campo eléctrico que impulsa el disco a girar. Entonces, la velocidad angular del disco está relacionada linealmente con el consumo eléctrico. Cada medidor analógico tiene un valor que indica el número de revoluciones que representan exactamente 1 kWh [2].
2. Medidores digitales: tienen una interfaz que consiste en una pequeña pantalla digital para mostrar la cantidad total de kWh consumidos y una salida de pulso óptico compuesta por un LED (*Light-Emitting Diode*). Cada cierta cantidad de transiciones entre el estado apagado y encendido del LED representa exactamente 1 kWh consumido, esta cantidad es una constante indicada por el medidor como impulsos/kWh. Por lo tanto, monitorear el parpadeo del LED brinda la capacidad obtener el consumo eléctrico en el tiempo. El valor de los impulsos/kWh difiere según el fabricante del medidor y generalmente se encuentra debajo del LED [2].

En las figuras 1.1 y 1.2, se pueden observar un medidor de consumo eléctrico analógico y otro digital, respectivamente.



FIGURA 1.1. Medidor de consumo eléctrico analógico<sup>1</sup>.



FIGURA 1.2. Medidor de consumo eléctrico digital<sup>2</sup>.

Cuando la compañía prestadora del servicio eléctrico quiere obtener la información de consumo de sus medidores, lo hace registrando el valor que exhibe la interfaz del medidor, que posee un contador analógico en el caso de un medidor analógico o una pantalla digital en el caso de un medidor digital, ambas exhiben el total de kWh consumidos por el abonado.

La cooperativa de servicios eléctricos Tupiza Ltda., COPELECT, de la ciudad de Tupiza, Bolivia, tiene instalados alrededor de diez mil medidores de consumo eléctrico analógicos y digitales de uso domiciliario en los hogares de sus abonados, que son monitoreados para determinar el consumo eléctrico de cada uno de ellos. El monitoreo lo realizan funcionarios que se desplazan por toda la ciudad y registran el valor que exhibe la interfaz de los medidores junto con el nombre del abonado al que corresponde. Esta información es recopilada y utilizada para

---

<sup>1</sup>Imagen tomada de: <https://expansionba.com.ar/2020/05/23/medidas-para-amortiguar-el-costo-energetico-en-pymes/>

<sup>2</sup>Imagen tomada de: <https://www.casasbahia.com.br/medidor-de-energia-monofasico-nansen-220v-100a-lumen-mc/>

emitir la factura correspondiente de cada abonado. Finalmente, la factura emitida es impresa y llevada por los funcionarios a los hogares de los abonados para que tengan conocimiento del monto que deben pagar por su consumo eléctrico.

El proceso de monitoreo antes descrito es llevado a cabo una vez al mes por doce funcionarios, quienes tardan alrededor de ocho días en registrar toda la información de los medidores. Posteriormente, esa información es introducida a una base de datos que funciona en un servidor local ubicado en las oficinas centrales de COPELECT. La cantidad de kWh consumidos que deben ser facturados se determinan al restar el conteo de kWh del mes anterior con el actual. En casos particulares donde los funcionarios no pueden acceder al medidor para registrar el conteo de kWh consumidos, se emite la factura con los datos del mes anterior.

## 1.2. Medición inteligente

La mayoría de los medidores de consumo eléctrico utilizados por parte de las compañías que prestan dicho servicio, sean estos analógicos o digitales, son dispositivos cuya única función es medir y exhibir mediante su interfaz la cantidad de kWh consumidos. Esta información únicamente es útil para la compañía y no brinda otros datos de relevancia. Existen también en el mercado otro tipo de medidores cuyas prestaciones son beneficiosas tanto para la compañía como para el abonado.

Los medidores inteligentes o *smart meters*, son dispositivos que graban información como el consumo eléctrico, niveles de voltaje, corriente y factor de potencia. Esta información es comunicada a la compañía eléctrica para generar la facturación de sus servicios y a los abonados para que tengan mayor conocimiento sobre el comportamiento de su consumo eléctrico. Los smart meters típicamente graban la información eléctrica en tiempo real o en intervalos cortos a lo largo del día. En la figura 1.3 se observa un smart meter.



FIGURA 1.3. Smart meter de la firma emlite<sup>3</sup>.

<sup>3</sup>Imagen tomada de: <https://www.jwsmartmeters.co.uk/brand/emlite/>

Para mejorar el proceso de monitoreo y adquisición de información sobre consumo eléctrico, los smart meters representan una solución idónea, pero, el costo de su implementación los vuelve inviables para muchas compañías que ofrecen el servicio eléctrico. Entonces, debido a la problemática antes planteada, existe un mercado creciente para medidores no inteligentes, ampliados con un dispositivo que transfiera la información sobre el consumo eléctrico a la compañía y al abonado.

El dispositivo añadido a los medidores eléctricos de uso convencional puede utilizar distintos tipos de sensores para obtener la información de consumo eléctrico. Estos son:

- Pinza de corriente: es una bobina sujetada alrededor de un conductor eléctrico. Cuando la corriente pasa a través del conductor, se genera un campo eléctrico. La bobina medirá este campo eléctrico y lo traducirá a un flujo de corriente [3].
- Cámara: podría ser situada en frente de del medidor y periódicamente tomar obtener fotografías del contador o pantalla. Las lecturas del consumo pueden ser extraídas de estas fotografías con técnicas de procesamiento de imágenes [4].
- Foto-reflector: consiste en un LED y un fototransistor en una sola carcasa. Este sensor es posicionado en frente del disco que poseen los medidores analógicos, cuando el LED emite luz es reflejada por el disco y medida por el fototransistor [5].
- Fototransistor: en conjunto con la salida de pulso óptico de los medidores digitales, se puede contar la cantidad de veces que el LED pasa de estado bajo a alto, para determinar el consumo eléctrico [6].

### 1.3. Soluciones disponibles en el mercado

Como se mencionó en la subsección anterior, dotar a los medidores convencionales de un dispositivo que amplíe sus funciones, es una manera de mejorar el proceso de monitoreo y adquisición de información de consumo eléctrico que realizan las compañías prestadoras de servicio.

Comercialmente existen dispositivos que cumplen esta función y utilizan alguno de los sensores adecuados para este fin. La fabricación de estos dispositivos se realiza sobre todo en Estados Unidos y algunos países europeos. A continuación se listan algunos dispositivos que utilizan la salida de pulso óptico de los medidores digitales para registrar el consumo de kWh:

- PA-FL [7] es un contador de pulsos con comunicación inalámbrica de la firma SyxthSense. Es alimentado mediante baterías o una fuente de tensión de 24 V y trabaja como parte de un sistema propietario de SyxthSense. Puede ser instalado en medidores de electricidad, agua o gas y transmitir inalámicamente los datos que registra utilizando una modulación de tipo FSK (*Frequency Shift Keying*, modulación por desplazamiento de frecuencia) en la banda de 868,3 MHz. También, posee dos salidas de potencia de 1 A y 60 V que pueden ser utilizadas para interactuar con otros dispositivos eléctricos o electrónicos. Dispone de una carcasa con certificación IP54. En la figura 1.4 se muestra una fotografía del dispositivo.



FIGURA 1.4. Registrador de pulsos PA-FL de la firma SysthSense<sup>4</sup>.

- AirTM-100S [8]: creado por la firma iNES, es un dispositivo diseñado para adquirir datos de medidores de energía eléctrica, agua y gas. Utiliza la salida de pulso óptico de medidores digitales para registrar el consumo del servicio. Es alimentado por una batería de 3,6 V que le brinda un tiempo de vida de aproximadamente cinco años, tiene carcasa con certificación IP65 y puede transmitir utilizando redes Sigfox [9] a una frecuencia de 868 MHz. El dispositivo puede observarse en la figura 1.5.



FIGURA 1.5. Registrador de pulsos AirTM-100S de la firma iNES<sup>5</sup>.

<sup>4</sup>Imagen tomada de: [7]

<sup>5</sup>Imagen tomada de: [8]

## 1.4. Motivación

Hoy en día, no solo las compañías de servicio eléctrico están interesadas en los números que proporcionan los medidores domiciliarios, sino también los propios abonados. Con la introducción del *smart meter*, la cantidad de electricidad consumida se puede comunicar en tiempo real al abonado. Este consumo se presenta en un dispositivo, por ejemplo, un teléfono inteligente o una tableta, que brinda más información a los abonados y los motiva a reducir su consumo de energía hasta en un 9 % [10]. Entonces, el trabajo se originó como una propuesta de COPELECT, para contar con una alternativa tecnológica que optimice el proceso de monitoreo de los medidores que tiene instalados en la ciudad boliviana de Túpiza y proporcione a sus usuarios una manera de conocer su consumo eléctrico de manera oportuna.

Otra motivación importante para la realización de este trabajo fue la aplicación de los conocimientos adquiridos en la carrera de Especialización, para desarrollar e implementar un dispositivo basado en buenas prácticas de desarrollo de *firmware* y *hardware*, que sea lo suficientemente robusto y eficiente para que puedan reproducirlo por cientos o miles de unidades.

## 1.5. Objetivos y alcance

El objetivo principal de este trabajo fue desarrollar e implementar un dispositivo electrónico, capaz de monitorear un medidor de consumo eléctrico de uso domiciliario mediante la salida de pulso óptico incorporada, para proporcionar la información obtenida a la compañía prestadora del servicio de manera remota y permitir al abonado conocer su consumo eléctrico en el momento que realiza la consulta a través de una interfaz gráfica web.

El alcance de este proyecto incluye:

- Un prototipo comercial que pueda ser instalado en los medidores de consumo eléctrico de COPELECT.
- Manual de uso e instalación.

## Capítulo 2

# Introducción específica

Este capítulo presenta los requerimientos del dispositivo, una descripción de los bloques que lo componen y la planificación que se siguió para lograr satisfactoriamente el desarrollo.

### 2.1. Requerimientos

El dispositivo tiene dos tipos de requerimientos, funcionales y no funcionales. Los funcionales, se refieren a la capacidad para cumplir con ciertas tareas impuestas, que garantizan un correcto desempeño del dispositivo en general. Los no funcionales, tienen relación con temas de carácter económico e informativo.

#### 2.1.1. Requerimientos funcionales

- El dispositivo deberá poseer conexión Wi-Fi<sup>1</sup>
- El dispositivo deberá funcionar como servidor web local.
- El dispositivo deberá contar con la hora y fecha exactas.
- El dispositivo deberá interpretar los pulsos ópticos provenientes de un medidor de consumo de energía eléctrica domiciliario.
- El dispositivo deberá poseer una memoria no volátil para registrar datos como la hora, fecha, conteo de pulsos e ID del usuario; durante al menos tres meses.
- El dispositivo deberá contar con un sistema de adquisición, procesamiento, transmisión y recepción de datos, que podrá ser implementado en un microcontrolador con Wi-Fi integrado.
- El dispositivo deberá poseer una interfaz web para que los usuarios puedan observar un registro histórico de su consumo de energía eléctrica.
- El dispositivo deberá poder establecer conexión con un gateway LoRa, para enviar diariamente en formato hexadecimal la hora, fecha, consumo de energía eléctrica e ID del usuario.

#### 2.1.2. Requerimientos no funcionales

- El dispositivo deberá tener un precio menor a 50 \$us.

---

<sup>1</sup>Wi-Fi. Es una tecnología inalámbrica para la interconexión de dispositivos electrónicos.

- El dispositivo deberá contar con manuales de uso e instalación.

## 2.2. Esquema general del sistema

Para cumplir con todos los requerimientos funcionales expuestos en la sección anterior, los componentes mínimos necesarios y su interconexión se muestran en el diagrama en bloques de la figura 2.1.

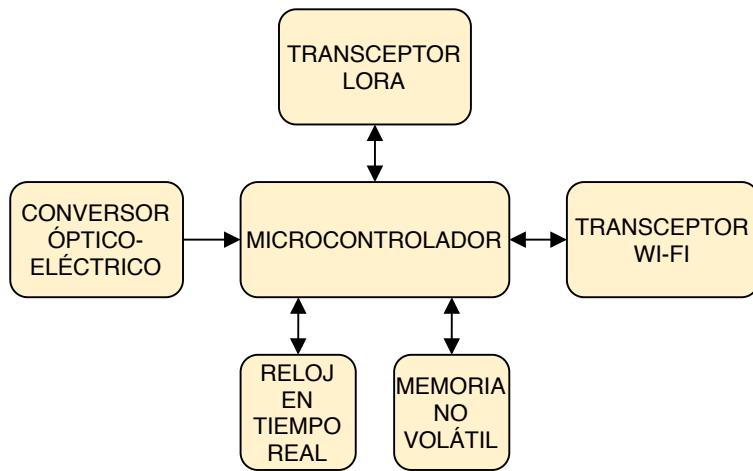


FIGURA 2.1. Diagrama en bloques general del dispositivo.

En el diagrama de la figura 2.1, el conversor óptico-eléctrico transforma los pulsos de luz provenientes del LED de un medidor de consumo eléctrico a pulsos eléctricos y los entrega al microcontrolador. El microcontrolador procesa estos pulsos y realiza el cálculo del consumo eléctrico. Esta información junto con la hora y fecha provenientes del reloj en tiempo real son almacenados en la memoria no volátil para su posterior utilización. El transceptor Wi-Fi se comunica con el microcontrolador para obtener los datos que serán utilizados para generar la interfaz gráfica mostrada al usuario. El transceptor LoRa tiene la función de establecer comunicación bidireccional con un dispositivo concentrador LoRa, para enviar la información de la memoria no volátil y recibir parámetros de funcionamiento.

### 2.2.1. Conversor óptico-eléctrico

Es el encargado de convertir la salida de pulso óptico de medidores eléctricos digitales a pulsos eléctricos, para que puedan ser interpretados por un microcontrolador. Esta información determina el consumo eléctrico que registra el medidor.

La salida de pulso óptico de los medidores eléctricos digitales está compuesta por un LED de color rojo, que emite luz cuando se ha consumido una cierta cantidad de kWh. El valor de la relación entre los pulsos emitidos y el consumo eléctrico es un parámetro intrínseco del medidor, que varía según el modelo y la firma que lo fabrica.

Para realizar la conversión de pulsos de luz a pulsos eléctricos, existen principalmente dos transductores que cumplen cabalmente esta función:

- Fotoresistencia: es una resistencia cuyo valor se modifica en función a la intensidad de luz incidente. También es conocida como LDR (*Light-Dependent*

*Resistor, resistencia dependiente de la luz) [11]. En la figura 2.2 se observa una fotoresistencia.*



FIGURA 2.2. Fotoresistencia GL5528<sup>2</sup>.

- Fototransistor: es un transistor sensible a la luz, normalmente a los infrarrojos. La cantidad de luz incidente es proporcional a la corriente de base generada. Generalmente tiene el factor de forma de un LED [11]. Un fototransistor de uso común se observa en la figura 2.3.



FIGURA 2.3. Fototransistor IR333C<sup>3</sup>.

### 2.2.2. Microcontrolador

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las instrucciones que tiene almacenadas. Dispone de los tres componentes básicos de una computadora: memoria, CPU (*Central Processing Unit*, unidad central de procesamiento) y periféricos de entrada/salida.

Los microcontroladores son específicos de la aplicación y se utilizan para tareas predefinidas. Por ejemplo, se puede usar un microcontrolador para controlar un motor en un sistema robótico. Por el contrario, una PC que utiliza un microprocesador es utilizada para aplicaciones que requieren un procesamiento intenso, como ejecutar grandes programas de gráficos en computadoras portátiles [12] . Por esa razón generalmente no se utilizan PCs para realizar el trabajo de microcontroladore .

Los fabricantes más populares de microcontroladores son: Analog Devices, Cypress Semiconductor, Infineon, Maxim Integrated, Microchip, NXP, On Semiconductor, Panasonic, ROHM Semiconductor, STMicroelectronics y Texas Instruments [13].

<sup>2</sup>Imagen tomada de: <https://www.devobox.com/en/photosensors/38-photoresistor-ldr07.html>

<sup>3</sup>Imagen tomada de: <https://www.steren.com.gt/fototransistor-de-5-mm-transparente.html>

En el mercado se pueden encontrar microcontroladores en diferentes factores de forma, pero, para el desarrollo de sistemas embebidos, como el que se describe en este trabajo, resulta conveniente utilizar tarjetas de desarrollo debido a que estas tienen toda la electrónica necesaria para funcionar, además de que ofrecen conectores estándar para simplificar la interacción con otros dispositivos. Una de estas tarjetas de desarrollo es la que muestra en la figura 2.4.

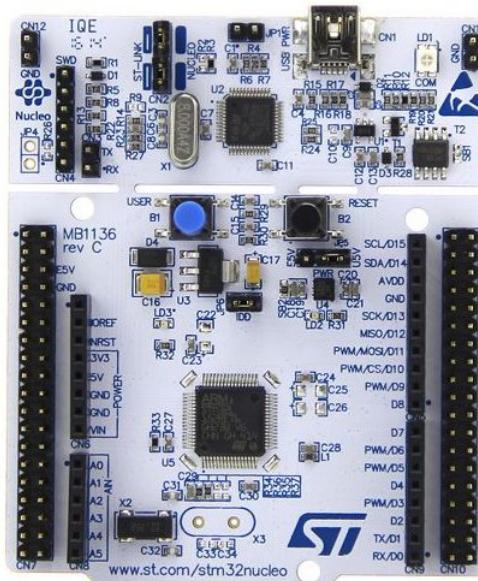


FIGURA 2.4. Tarjeta de desarrollo del fabricante STMicroelectronics basada en el microcontrolador STM32F030R8T6<sup>4</sup>.

### 2.2.3. Transceptor Wi-Fi

Wi-Fi es una tecnología de red inalámbrica, que permite a dispositivos como computadoras y teléfonos celulares conectarse entre sí para formar una red, o conectarse a un enrutador por el que se disponga de conexión a Internet. Está basado en la familia de estándares IEEE 802.11, que definen los protocolos que permiten la comunicación entre dispositivos compatibles con Wi-Fi [14]. Según la versión de Wi-Fi, puede funcionar en las bandas de 2.4 GHz o 5 GHz[14].

En la tabla 2.1 muestran las características técnicas de las distintas versiones del estándar IEEE 802.11, donde se puede apreciar claramente que la diferencia más grande es la velocidad de datos entre cada uno de los protocolos.

TABLA 2.1. Tabla comparativa de características del estándar IEEE 802.11<sup>5</sup>

| Protocolo 802.11 | Frecuencia      | Ancho de banda  | Velocidad de datos (Mb/s)       |
|------------------|-----------------|-----------------|---------------------------------|
| a                | 5 GHz           | 20 MHz          | 5, 9, 12, 18, 24, 36, 48, 54    |
| b                | 2,4 GHz         | 20 MHz          | 1; 2; 5,5; 11                   |
| g                | 2,4 GHz         | 20 MHz          | 6, 9, 12, 18, 24, 36, 48, 54    |
| n                | 2,4 GHz y 5 GHz | 20 MHz y 40 MHz | 7,2; 28,9; 43,3; 57,8; 65; 72,2 |

<sup>4</sup>Imagen tomada de: <https://www.seeedstudio.com/NUCLEO-L152RE-Development-Board-for-STM32-p-1934.html>

Dentro del modelo OSI [15], Wi-Fi se encuentra en la capa física y de enlace de datos. En la figura 2.5 se ve el modelo OSI.

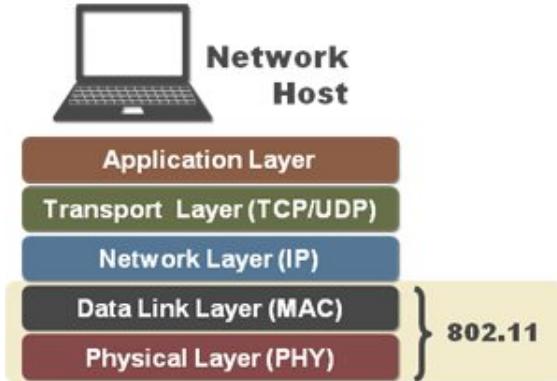


FIGURA 2.5. Ubicación de Wi-Fi en el modelo OSI<sup>6</sup>.

Una red Wi-Fi tiene una arquitectura de tipo estrella, donde las estaciones están conectadas directamente a un punto central y todas las comunicaciones se hacen necesariamente a través de ese punto. Esta red se ilustra en la figura 2.6.

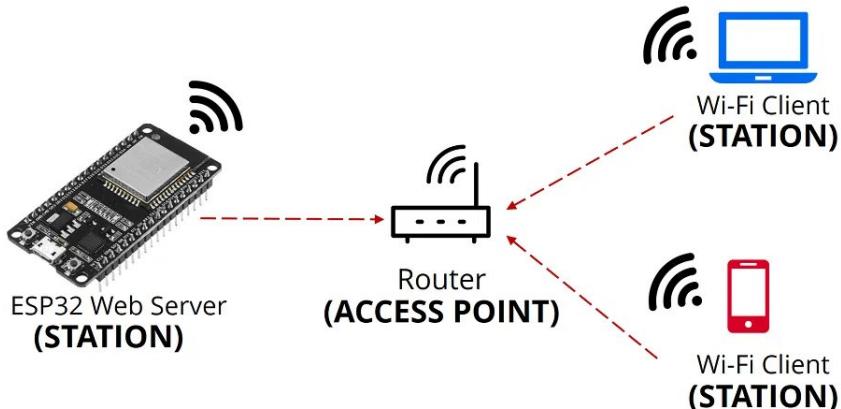


FIGURA 2.6. Arquitectura de una red Wi-Fi<sup>7</sup>.

Los elementos principales de una red Wi-Fi son:

- Estaciones: son dispositivos electrónicos que se conectan entre sí a través de enrutadores inalámbricos. Son más conocidos como *hosts* y pueden ser computadoras, tabletas, teléfonos celulares o sistemas embebidos.
- Puntos de acceso: también conocidos como *access points*, son los elementos de la red que enrutan la información proveniente de las estaciones dentro de la red local o hacia otras redes.

Dentro de lo referido al desarrollo de sistemas embebidos, comercialmente pueden encontrarse módulos Wi-Fi como el de la figura 2.7. Estos módulos generalmente incorporan un microcontrolador de uso general para manejar el transceptor Wi-Fi.

<sup>5</sup>Datos obtenidos de: <https://microchipdeveloper.com/wifi:a-b-g-n-explained>

<sup>6</sup>Imagen tomada de: <https://microchipdeveloper.com/wifi:80211-osi>

<sup>7</sup>Imagen tomada de: <https://randomnerdtutorials.com/esp32-access-point-ap-web-server/>



FIGURA 2.7. Módulo Wi-Fi basado en el circuito integrado EMW3162<sup>8</sup>.

#### 2.2.4. Transceptor LoRa

LoRa (*Long Range*, largo alcance), es una técnica de modulación de espectro extendido derivada de la tecnología CSS (*Chirp Spread Spectrum*, espectro extendido de tipo chirp) [16]. Fue desarrollado por la firma Semtech y es utilizada principalmente en dispositivos orientados a IoT (*Internet of Things*, Internet de las cosas) y dispositivos alimentados por baterías. Opera en las bandas de 433 Mhz, 868 Mhz y 915 MHz, según el país.

Las comunicaciones LoRa son del tipo punto a punto, es decir, que un dispositivo con esta tecnología debe establecer un enlace directo con otro para intercambiar información. Para formar redes LoRa, es necesaria una capa MAC (*Media Access Control*, control de acceso al medio), la cual es llamada LoRaWAN (*Long Range Wide Area Network*, red de área amplia LoRa).

LoRaWAN, es una especificación de redes LPWAN (*Low Power Wide Area Network*, red de área amplia de baja potencia) y LoRa Alliance es la encargada de su estandarización. Está diseñada para conectar dispositivos de bajo consumo energético a Internet a través de redes regionales, nacionales o globales. Además proporciona comunicación bidireccional, seguridad, movilidad y servicios de localización[17].

En la figura 2.8 se puede observar el modelo de capas de una red de dispositivos LoRa, donde el protocolo LoRa define la capa física (PHY) y LoRaWAN la capa de acceso al medio (MAC). Este modelo tiene muchas similitudes con el modelo capas OSI.

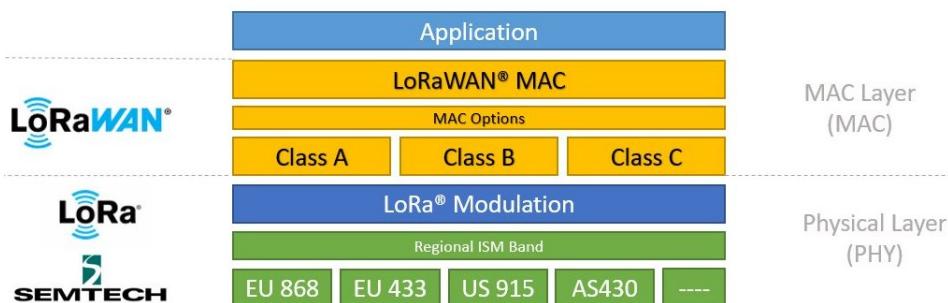


FIGURA 2.8. Stack LoraWAN<sup>9</sup>.

<sup>8</sup>Imagen tomada de: <https://www.seeedstudio.com/EMW3162-WiFi-Module-External-IPEX-antenn-p-2235.html>

<sup>9</sup>Imagen tomada de: <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>

Al igual que en una red Wi-fi, la arquitectura de una red LoRaWAN es de tipo estrella y permite una conexión multipunto entre dispositivos, como se muestra en la figura 2.9.

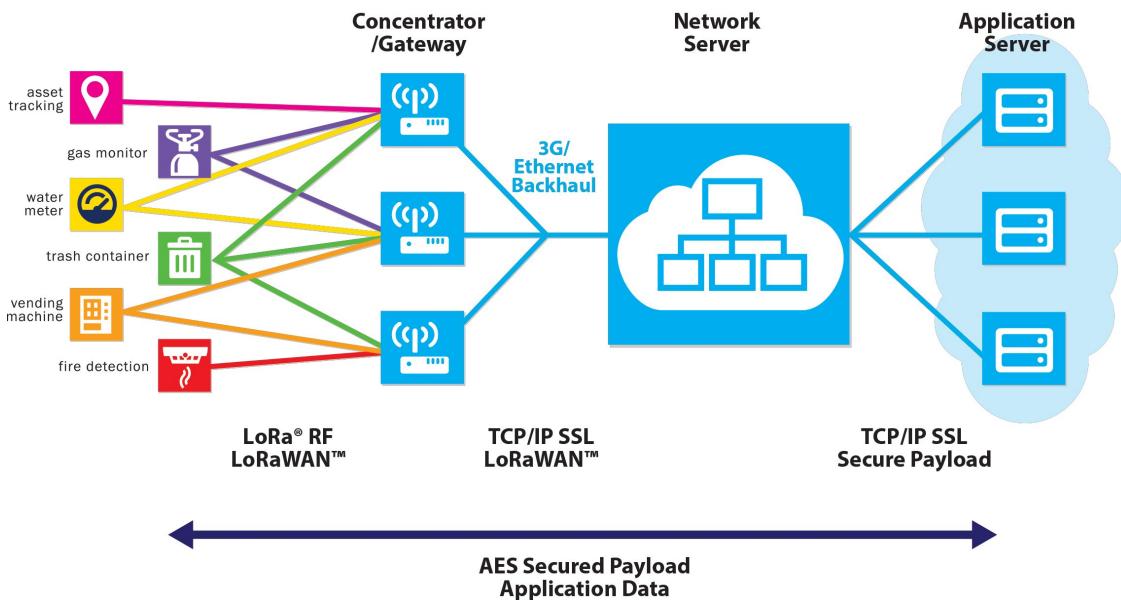


FIGURA 2.9. Arquitectura de una red LoraWAN<sup>10</sup>.

De la figura 2.9, se distinguen cuatro tipos diferentes de elementos que conforman la red LoRaWAN. Estos son:

- Nodos: son los dispositivos que utilizan la tecnología LoRa como método de transmisión de datos. Son utilizados para obtener datos de sensores o para interactuar con actuadores. Generalmente son dispositivos de bajo consumo energético y alimentados por baterías.
- Concentradores: también conocidos como *gateways*, son los encargados de recibir la información de los nodos y reenviarla a un servidor de red. Estos dispositivos tienen acceso a Internet mediante redes celulares, Wi-Fi o Ethernet.
- Servidores de red: son los responsables del enrutamiento de los mensajes al dispositivo adecuado, seleccionar el mejor gateway para el mensaje de enlace descendente, eliminar mensajes duplicados y descifrar los mensajes que vienen cifrados desde los nodos.
- Servidores de aplicación: es donde se realizan los procesos útiles sobre los datos obtenidos de los nodos. Típicamente se ejecutan en una nube privada o pública.

En el desarrollo de nodos para redes LoRaWAN, se utilizan módulos que llevan embebido un circuito integrado con tecnología LoRa y todos los componentes electrónicos necesarios para que este funcione correctamente, como el de la figura 2.10. Cabe resaltar que muchos de estos módulos no pueden cumplir ninguna función si no son manejados por un microcontrolador que se comunique con ellos para configurarlos para mandar y recibir paquetes de datos. Además, es mandatorio conectarles una antena adecuada antes de ser energizados.

<sup>10</sup>Imagen tomada de: <https://www.aprendiendoarduino.com/2018/03/05/redes-lpwan/>

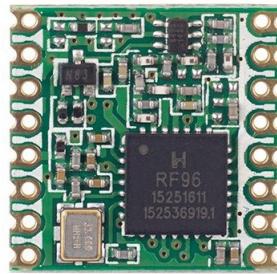


FIGURA 2.10. Módulo LoRa basado en el circuito integrado RF96<sup>11</sup>.

### 2.2.5. Reloj en tiempo real

Más conocido como RTC (*Real-Time Clock*, reloj en tiempo real), es un circuito integrado que tiene la capacidad de llevar con precisión la hora y fecha. Para contar con exactitud los segundos, utiliza un oscilador de cristal de cuarzo de 32,768 kHz, que puede o no estar embebido en el encapsulado del RTC.

La principal aplicación de un RTC es brindar a un sistema electrónico la hora y fecha exactas, también puede ofrecer otras funciones como alarmas, salidas de reloj de 1 Hz o medición de temperatura.

Algunos RTCs tienen una fuente de poder alternativa basada en baterías, que mantiene funcionando la parte del circuito que lleva la cuenta de la hora y fecha. Esta fuente de tensión normalmente son baterías de litio o supercapacitores [18]. Comercialmente un RTC puede adquirirse como parte de un módulo, como el que se ve en la figura 2.11, que tiene instalada la fuente de alimentación alternativa y brinda mayor facilidad para acceder a los pines del circuito integrado.

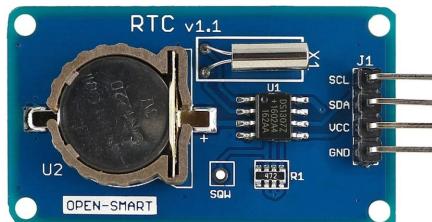


FIGURA 2.11. Módulo RTC basado en el circuito integrado DS1307<sup>12</sup>.

### 2.2.6. Memoria no volátil

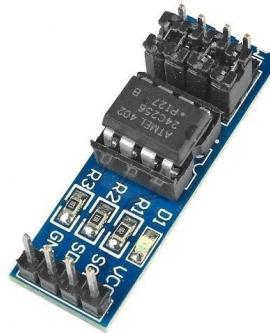
Es un tipo de memoria de lectura y escritura, en la que los datos que tiene almacenados se mantienen intactos cuando la fuente de alimentación deja de funcionar, es decir, que no necesita energía para mantener guardada la información grabada en ella [19].

En sistemas embebidos, existen principalmente dos tipos de memorias no volátiles:

<sup>11</sup>Imagen tomada de: <https://www.antratek.com/rfm95-lora-module>

<sup>12</sup>Imagen tomada de: <https://www.antratek.com/rfm95-lora-module>

- EEPROM (*Electrically Erasable Programmable Read-Only Memory*, ROM borrable y programable eléctricamente): es un tipo de memoria ROM que puede ser programada y borrada mediante métodos eléctricos. Aunque puede ser leída un número ilimitado de veces, las operaciones de escritura o borrado de datos solo se pueden realizar entre cien mil y un millón de veces. Este tipo de memorias pueden encontrarse como circuitos integrados que generalmente disponen de comunicación I<sup>2</sup>C (*Inter-Integrated Circuit*, circuito inter-integrado) o SPI (*Serial Peripheral Interface*, interfaz periférica serial). Comercialmente se pueden encontrar módulos EEPROM como el de la figura 2.12.



### 2.3. Planificación

De acuerdo a los requerimientos planteados en la sección 2.1 y en función del diagrama en bloques general del dispositivo mostrado en la sección 2.2, se confeccionó una planificación de este trabajo como parte de la materia de gestión de proyectos de la Carrera de Especialización en Sistemas Embebidos.

El trabajo fue dividido en distintas actividades, cada una cumple con uno o varios de los requerimientos planteados previamente. En la figura 2.14 se observa el diagrama AON (*Activity On Node*, actividad en el nodo).

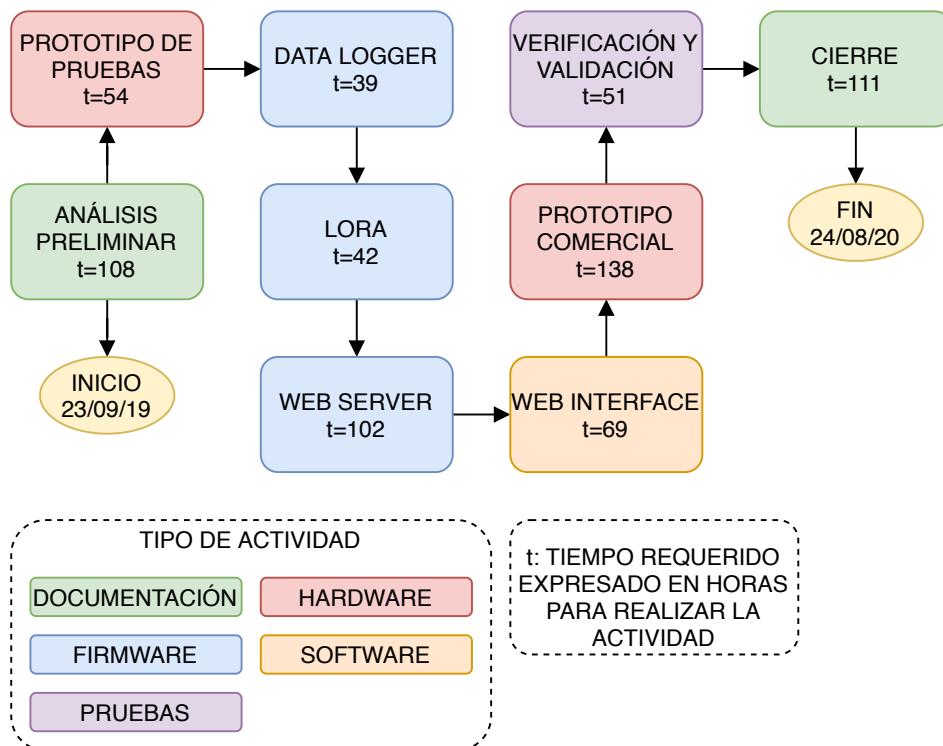


FIGURA 2.14. Diagrama AON del trabajo.

Resalta que la cantidad de horas destinadas al desarrollo de firmware y hardware son aproximadamente el 62 % del tiempo previsto para el desarrollo del trabajo en general. Esto guarda relación con el esfuerzo destinado para obtener resultados que garanticen un buen desempeño técnico del dispositivo desarrollado.

Para mejorar el control del tiempo en el desarrollo de todas las actividades del trabajo, estas fueron desglosadas en tareas. Estas tareas fueron planificadas y programadas según el diagrama de Gantt de las figuras 2.15, 2.16 y 2.17.

Los entregables del proyecto son los siguientes:

- Diagrama esquemático.
- Código fuente.
- Prototipo comercial.
- Manual de uso e instalación.
- Informe final.



FIGURA 2.15. Primera parte del diagrama de Gantt.



FIGURA 2.16. Segunda parte del diagrama de Gantt.

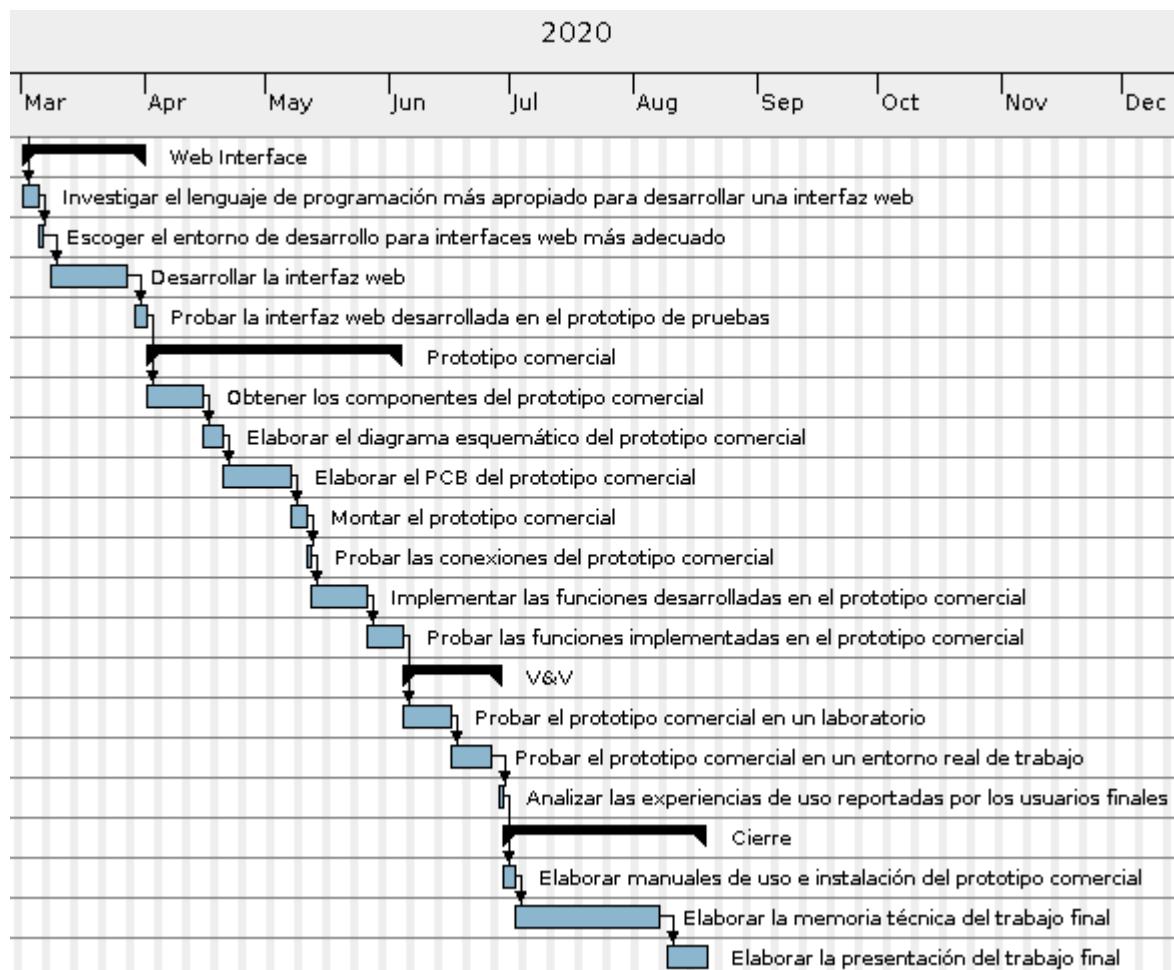


FIGURA 2.17. Tercera parte del diagrama de Gantt.

## Capítulo 3

# Diseño e implementación

En este capítulo se explica el proceso que se siguió para desarrollar e implementar el prototipo de pruebas, el firmware, la interfaz web y el prototipo comercial.

### 3.1. Prototipo de pruebas

El prototipo de pruebas fue desarrollado con la finalidad de probar todas las funciones de firmware que componen el trabajo, para brindar una primera aproximación al prototipo comercial del dispositivo.

Como se vio en el diagrama de la figura 2.1, el dispositivo está compuesto por los siguientes bloques funcionales: microcontrolador, transceptor Wi-Fi, transceptor LoRa, memoria no volátil, reloj en tiempo real y conversor óptico-eléctrico.

La construcción del prototipo de pruebas se realizó en una *breadboard*, que permitió realizar cambios en las conexiones de los componentes de una manera sencilla cuando estos se requerían. Se eligieron componentes de hardware acordes con los bloques que constituyen el dispositivo, en su mayor parte módulos de desarrollo con circuitos integrados embebidos que disponen de conectores apropiados para una breadboard. En la figura 3.1 se muestra el diagrama en bloques general con los componentes del prototipo de pruebas.

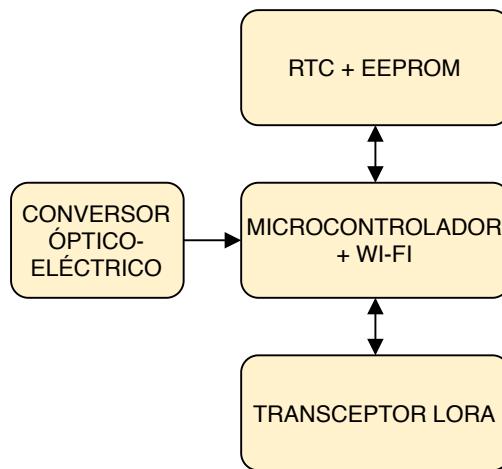


FIGURA 3.1. Diagrama en bloques del prototipo de pruebas.

Para garantizar un tiempo corto en la obtención de los componentes del prototipo de pruebas, el criterio predominante para la elección de los componentes fue la

disponibilidad en el mercado local. Además, la elección de proveedores locales aseguró la restitución eficaz de los componentes que se malograron durante el desarrollo.

### 3.1.1. Microcontrolador + Wi-Fi

Este bloque fusiona los bloques microcontrolador y transceptor Wi-Fi. El desarrollo de dispositivos con conexión Wi-Fi ha tenido un gran crecimiento en los últimos años [20], por lo que existen algunos fabricantes de circuitos integrados que ofrecen soluciones que integran microcontroladores y transceptores Wi-Fi en un solo encapsulado.

El componente elegido para este bloque es la tarjeta de desarrollo NodeMCU de la firma Amica, basado en el módulo ESP-12F de la firma Ai-Thinker. Las características más atractivas de esta tarjeta en lo referente al desarrollo son la alimentación y programación a través de un puerto micro USB, factor de forma adecuado para ser montado sobre un breadboard e incorporación de LEDs y pulsadores en la misma tarjeta. En la figura 3.2 se muestra la NodeMCU.

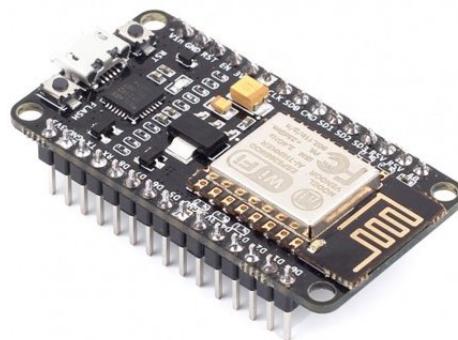


FIGURA 3.2. Tarjeta de desarrollo NodeMCU de la firma Amica<sup>1</sup>.

El módulo ESP-12F monta sobre sí un SoC (*System on a Chip*, sistema en un chip) de la firma Espressif Systems, el ESP8266, que funciona como microcontrolador y transceptor Wi-Fi. Otros componentes instalados sobre este módulo son condensadores, resistencias, oscilador, memoria flash y una antena impresa; todos ellos necesarios para que el ESP8266 pueda desempeñar correctamente sus funciones.

El ESP8266 es un chip de bajo costo que incorpora un microcontrolador y un transceptor Wi-Fi, además de contar con un *stack* TCP/IP. Sus características técnicas más relevantes son:

- Procesador Tensilica LX106 de arquitectura RISC (*Reduced Instruction Set Computer*, computador con conjunto de instrucciones reducido) de 32 bits a una frecuencia de 80 MHz.
- RAM de 64 KB para instrucciones y 96 KB para datos.
- ROM externa, puede soportar hasta 16 MB de memoria flash con conexión QSPI (*Quad SPI*, SPI cuádruple).
- IEEE 802.11 b/g/n.

---

<sup>1</sup>Imagen tomada de: <https://www.amazon.com/-/es/KeeYees-Internet-Development-Wireless-Compatible/dp/B07PR9T5R5>

- Periféricos GPIO (*General Purpose Inputs/Outputs*, entradas/salidas de propósito general), SPI, I<sup>2</sup>C, UART y ADC.

### 3.1.2. Transceptor LoRa

Para la elección del componente de este bloque hubo varias consideraciones. La más importante fue la frecuencia de transmisión y recepción. LoRa trabaja en las frecuencias de 433 MHz, 868 MHz y 915 MHz; de acuerdo al país donde se implementa. Esto en Bolivia el espectro electromagnético está normado por la Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes, ATT, a través del documento de plan de frecuencias [21]. Allí se determina la frecuencia de 915 MHZ como la banda destinada para las aplicaciones ISM (*Industrial, Scientific and Medical*; industrial científica y médica) que es usada en otros países para comunicaciones LoRa. Este tipo de comunicaciones no están contempladas en dicho documento, pero en el decreto supremo 4272 en su artículo 73[22] lanzado en junio de 2020, se especifica el procedimiento para la utilización de la frecuencia de 915 MHz para redes LPWAN (*Low Power Wide Area Network*, redes de área amplia y bajo consumo) de manera libre.

En el mercado local no se pudieron encontrar módulos LoRa que funcionen a la frecuencia de 915 MHz. Se adquirieron los módulos disponibles, que trabajan en la frecuencia de 433 MHz, lo que según el plan de frecuencia boliviano [21] está destinado a radioaficionados. El módulo utilizado para el prototipo de pruebas fue el PM1280, que está basado el circuito integrado SX1278. En la figura 3.3 se observa una fotografía del módulo PM1280.

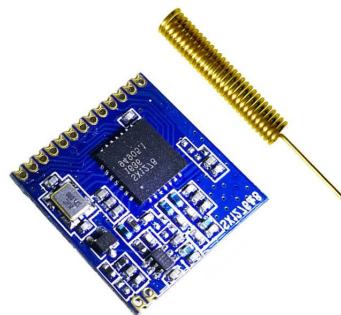


FIGURA 3.3. Módulo LoRa PM1280<sup>2</sup>.

El circuito integrado SX1278 es un transceptor LoRa de la firma Semtech que provee comunicación de espectro ensanchado de largo alcance y alta inmunidad a las interferencias. Su principales características son:

- Potencia de transmisión de 100 mW.
- Alta eficiencia del amplificador de potencia.
- Frecuencia de operación: 137 MHZ a 525 MHZ.
- Velocidad de bit programable hasta 300 Kbps.
- Bajo consumo de corriente, 9,9 mA en modo de recepción y 200 nA en la retención de datos en sus registros.

---

<sup>2</sup>Imagen tomada de: <https://www.todomicro.com.ar/arduino/910-modulorf-lora-sx1278-chip-pm1280-con-antena.html>

- Soporta paquetes de hasta 256 bytes.
- Sensor de temperatura e indicador de batería incorporados.

### 3.1.3. RTC + EEPROM

Los bloques memoria no volátil y reloj en tiempo real fueron fusionados en un único bloque, ya que comercialmente existen módulos que cumplen ambas funciones. Estos módulos tienen embebidos circuitos integrados de memoria y RTC, además de otros componentes como resistencias, condensadores, osciladores, zócalos para baterías y conectores apropiados para un breadboard. Estos módulos en su gran mayoría poseen una EEPROM como medio de almacenamiento de datos, esta tecnología es preferible sobre las memorias flash en aplicaciones de adquisición de datos, ya que proporciona un número mayor de ciclos de escritura y borrado.

La mayor parte de los módulos que existen en el mercado local cumplen cabalmente con las funciones que requiere este bloque, pero debido a la cantidad de pines utilizables de la NodeMCU se tuvo preferencia por los módulos que tenían integrados chips con interfaz I<sup>2</sup>C. Asimismo, al haber muchos módulos que cumplían el requisito de la interfaz, se buscó uno que tuviera un RTC con la capacidad de generar alarmas en función de la hora. En la figura 3.2 se observa el módulo de RTC + EEPROM elegido.



FIGURA 3.4. Módulo RTC + EEPROM<sup>3</sup>.

Los circuitos integrados que componen el módulo son el DS3231 y el AT24C32, un RTC y una EEPROM, respectivamente. El DS3231 es un RTC de alta precisión de la firma Maxim Integrated, que cuenta con una interfaz I<sup>2</sup>C para conectarse con otros dispositivos, también tiene la capacidad de generar alarmas y medir la temperatura. El AT24C32 es una EEPROM de la firma Microchip, con interfaz I<sup>2</sup>C y 32 KB de capacidad de almacenamiento.

### 3.1.4. Conversor óptico-eléctrico

Para este bloque, el componente elegido es un módulo detector de luz, compuesto por un fototransistor PT333-3C de la firma Everlight y un comparador de voltaje LM393 de la firma Texas Instruments. El módulo genera como salida un pulso eléctrico acotado al nivel de tensión con el que se alimenta. Cuando la cantidad

<sup>3</sup>Imagen tomada de: <https://electropeak.com/extremely-accurate-rtc-module>

de luz incidente en el fototransistor provoca un nivel de tensión igual o mayor al nivel de tensión del potenciómetro que viene incluido. En la figura 3.5 se puede observar el módulo.



FIGURA 3.5. Módulo detector de luz<sup>4</sup>.

## 3.2. Diseño de firmware

El desarrollo del firmware fue la actividad que requirió más esfuerzo en el trabajo, debido a que los objetivos principales del autor fueron lograr modularización y reutilización del código escrito. Para lograr dichos objetivos, el firmware fue estructurado en capas y se utilizó control de versiones para documentarlo. De esta manera, se logró un desarrollo de carácter más profesional que podría ser reutilizado en futuros proyectos que requieran funciones similares.

Antes de realizar la descomposición del firmware en capas, fue necesario elegir las herramientas de desarrollo implicadas, que fueron imprescindibles al momento de escribir el código fuente del dispositivo. Estas herramientas fueron un SDK (*Software Development Kit*, kit de desarrollo de software) que proporcionó una API (*Application Programming Interface*, interfaz de programación de aplicaciones) para facilitar el desarrollo de código fuente para el ESP8266 y un IDE (*Integrated Development Environment*, Entorno de Desarrollo Integrado) que proporcionó un entorno con herramientas que agilizaron la escritura de código con el SDK elegido. Estos fueron:

- ESP8266RTOS\_SDK: este SDK fue desarrollado por la firma Espressif Systems para la programación del SoC ESP8266 y facilita un conjunto de funciones para la creación de código fuente. Está basado en el RTOS (*Real-Time Operating System*, sistema operativo en tiempo real) de uso gratuito FreeRTOS [23], que fue utilizado en las materias sobre sistemas operativos en tiempo real de la Carrera de Especialización y brindó funciones que ayudaron a lograr determinismo en la ejecución de las tareas del dispositivo. Asimismo, contiene un documentación completa sobre las funciones que incorpora y ejemplos de uso.

<sup>4</sup>Imagen tomada de: <https://www.roboter-bausatz.de/en/diy-electronics/extension-modules/sensors/optics-light/149/light-sensor-module>

- Eclipse: el aspecto más importante en la elección de este IDE fue que en la documentación de instalación y uso del ESP8266RTOS\_SDK [24], se indicaba el proceso de configuración que permitió utilizar ambos en conjunto. Otro aspecto de importancia fue la experiencia previa del autor con este IDE, fue utilizado en varias materias de la Carrera de Especialización.

Entonces, una vez definidas las herramientas utilizadas, fue posible dividir el firmware en capas para facilitar el desarrollo y reducir la complejidad del código escrito para el dispositivo. La división en capas del firmware puede observarse en el diagrama de la figura 3.6, donde existen tres capas claramente diferenciadas: APP, DRIVERS y BASE.

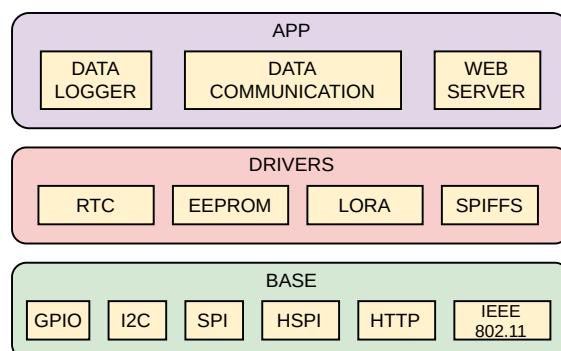


FIGURA 3.6. Diagrama de capas del firmware.

BASE es la capa de menor nivel y está compuesta por la API del ESP8266RTOS\_SDK. Proporciona a las capas de niveles superiores la capacidad de interactuar con los periféricos y protocolos del ESP8266 a través de funciones en lenguaje C. Los periféricos y protocolos que fueron utilizados en el presente trabajo fueron:

- GPIO: este periférico fue utilizado por la capa APP para gestionar los pines disponibles en el ESP8266, ya que algunos de ellos tienen funciones específicas y no pueden ser utilizados para propósitos generales. La API posee funciones para definir los pines como entradas o salidas, configuración de interrupciones por flanco positivo o negativo y resistencias de *pull-up* internas.
- I<sup>2</sup>C: se utilizó este periférico para que la capa DRIVERS interactúe con el RTC y la EEPROM. Al tener pocos pines disponibles en el ESP8266, este componente se hizo muy importante, ya que la comunicación I<sup>2</sup>C solo requiere dos pines, uno para datos y otro el reloj de sincronización.
- SPI: la capa DRIVERS utiliza este periférico para comunicarse con el transceptor LoRa. El módulo LoRa elegido interacciona a través del protocolo SPI con el microcontrolador que lo maneja para transmitir o recibir datos.
- HSPI: el ESP8266 no posee memoria ROM embebida en el SoC, por tanto, utiliza una memoria flash externa para almacenar las instrucciones del programa y los datos del usuario. Esta memoria flash se comunica con el ESP8266 mediante el protocolo HSPI. Este periférico se utilizó para que la capa DRIVERS configure la flash como un sistema de archivos.
- HTTP (*HyperText Transfer Protocol*, protocolo de transferencia de hipertexto): la API ofrece funciones para ejecutar este protocolo. Fue de utilidad para

proporcionar a la capa APP las funciones necesarias para implementar un servidor web capaz de responder a los métodos HTTP, GET y POST [25].

- IEEE 802.11: el ESP8266 tiene embebida toda la electrónica necesaria para implementar los protocolos IEEE 802.11 en sus versiones b, g y n. La capa APP utilizó las funciones disponibles de este módulo para lograr que el dispositivo funcionara como punto de acceso y/o estación.

La capa DRIVERS está compuesta por módulos que son bibliotecas de firmware, que le permiten al ESP8266 interactuar con los periféricos de hardware externos a los que está conectado. Se desarrollaron bibliotecas para los módulos EEPROM, RTC, LORA y SPIFFS; todos basados en la capa BASE. Estas bibliotecas se encuentran en el repositorio del trabajo y constan de un archivo de código fuente y otro de cabecera, cada una.

La biblioteca para la EEPROM se desarrolló con ayuda del *datasheet* [26] del AT24C32, donde se indican todos los pormenores técnicos del funcionamiento de este circuito integrado. Además, se utilizaron las funciones de la capa BASE para gestionar correctamente la comunicación I<sup>2</sup>C. Las funciones que proporciona esta biblioteca sirven para:

- leer de valores de 8, 16 y 32 bits de una dirección determinada de la EPROM.
- escribir de valores de 8, 16 y 32 bits de una dirección determinada de la EPROM.

Para el módulo RTC se desarrolló una biblioteca que sirvió para configurar la hora, fecha y otras funciones incorporadas en el DS3231. La herramienta principal en el desarrollo de esta biblioteca fue el *datasheet* [27] de dicho circuito integrado. De este se obtuvo información sobre las direcciones de los registros que manejan sus funciones y la forma adecuada de configurarlos. Igual que para la biblioteca de la EEPROM, las funciones de la capa BASE para el protocolo I<sup>2</sup>C permitieron que se disponga de una manera para que el ESP8266 pueda intercambiar datos con el DS3231 con la menor cantidad de pines posible. Esta biblioteca permite:

- leer y configurar las horas, minutos y segundos.
- leer y configurar el día, fecha, mes y año.
- leer y configurar las dos alarmas disponibles.
- leer y configurar las salidas digitales.

El desarrollo de la biblioteca para el módulo LORA permitió manejar el circuito integrado SX1278, para establecer la comunicación de este elemento con el ESP8266 a través del periférico SPI. Esto permitió configurar sus parámetros para lograr la transmisión y recepción de datos con dispositivos de tecnología LoRa de manera exitosa. Está basada en la biblioteca Arduino LoRa de Sandeep Mistry [28] y en la información del *datasheet* [29] del SX1278. Asimismo, utiliza las funciones proporcionadas por la capa BASE para la comunicación SPI. Las funciones más importantes que proporciona son:

- configurar la frecuencia del módulo.
- transmitir un *buffer* de tamaño variable.
- recibir datos en el buffer interno.

- leer el valor del RSSI (*Received Signal Strength Indication*, indicador de fuerza de la señal recibida) de los datos recibidos en el buffer interno.
- establecer el modo de funcionamiento en bajo consumo.
- configurar la potencia de transmisión.
- configurar el ancho de banda.
- habilitar/deshabilitar el CRC (*Cyclic Redundancy Check*, verificación de redundancia cíclica).

Por último, se desarrolló una biblioteca para establecer un sistema de archivos muy reducido llamado SPIFFS (*SPI Flash File System*, sistema de archivos flash SPI), que está alojado en la memoria flash externa utilizada para almacenar el programa del ESP8266. Esta biblioteca requirió menos esfuerzo en su desarrollo que las anteriores, debido a que la mayoría de las funciones necesarias para configurar el sistema de archivos son parte de la API del ESP8266\_RTOS\_SDK y para el manejo de archivos se utilizaron las funciones estándar de C. Solo posee una función para inicializar el sistema de archivos, que configura la cantidad máxima de elementos y su capacidad de almacenamiento.

El tamaño de este sistema de archivos es de 1 MB y fue configurado de acuerdo al tamaño total de la memoria flash, que en el módulo ESP-12F es de 4 MB. El restante se utilizó para el programa, datos de fábrica y datos de configuración de la interfaz física. El detalle de los archivos almacenados en SPIFFS puede observarse en la tabla 3.1.

TABLA 3.1. Tabla de detalle del contenido del sistema de archivos SPIFFS.

| Nombre             | Tamaño (KB) | Descripción   |
|--------------------|-------------|---|
| ajax-loadergif.gif | 6,2         | Imagen de carga de la interfaz web                      |
| favicon.ico        | 1,1         | Ícono de la interfaz web                                |
| highchartsjs.gz    | 92          | Biblioteca JavaScript Highcharts comprimida             |
| highchartsmap.gz   | 235,6       | Archivo de mapeo para highchartsjs.gz                   |
| index.html         | 7,3         | Documento HTML de la interfaz web                       |
| jqueryjs.gz        | 33,2        | Biblioteca JavaScript jQuery comprimida                 |
| jquerymobilecss.gz | 25,1        | Hoja de estilos CSS de la biblioteca jQuery Mobile      |
| jquerymobilejs.gz  | 55,5        | Biblioteca JavaScript jQuery Mobile comprimida          |
| jquerymobilemap.gz | 88,8        | Archivo de mapeo para jquerymobilejs.gz                 |
| config.txt         | 0,6         | Archivo de configuración del dispositivo                |
| pulses.csv         | 1           | Archivo con el registro histórico del consumo eléctrico |

La mayoría de los archivos almacenados en SPIFFS son utilizados para generar la interfaz web, excepto config.txt y pulses.csv. El tamaño de memoria utilizado por todos los archivos es de 546,4 KB, que ocupa aproximadamente un 54 % del tamaño total del sistema de archivos. Hay que notar que los archivos de mayor tamaño fueron comprimidos antes de ser almacenados, ya que sin este proceso el tamaño total hubiera sido de 1,6 MB, que superaba aproximadamente en un 60 % el tamaño del sistema de archivos.

La capa APP, está compuesta por los módulos que ejecutan las tareas del dispositivo. Se basa en las capas inferiores para interactuar con los periféricos del

ESP8266 y con el hardware externo. Sus módulos son DATA LOGGER, DATA COMMUNICATION y WEB SERVER. Los archivos de estos módulos se encuentran en el repositorio del trabajo.

### 3.2.1. DATA LOGGER

Este módulo tiene la función principal de adquirir, procesar y almacenar la información de consumo eléctrico del medidor al que está instalado el dispositivo. Para este fin, se comunica con los módulos de las capas inferiores como se muestra en el diagrama de la figura 3.7.

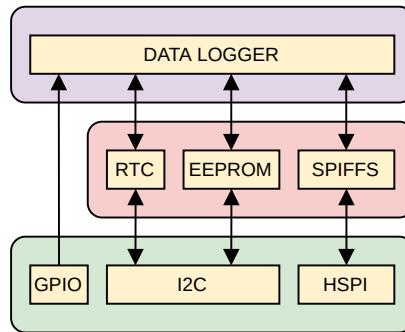


FIGURA 3.7. Diagrama de capas para DATA LOGGER.

Utiliza el RTC y la EEPROM para mantener un registro histórico de la información adquirida por GPIO. Modifica el archivo pulses.csv almacenado en SPIFFS para actualizar la información de consumo eléctrico cuando se registran nuevos datos. Esto es utilizado posteriormente por WEB SERVER. Asimismo, en función de las alarmas generadas por el RTC, se envían los datos de la EEPROM a DATA COMMUNICATION.

Dentro del sistema operativo utilizado existen dos tareas para este módulo. Una para registrar los pulsos del medidor eléctrico y otra para manejar las alarmas del RTC, pulses\_task y alarm\_task. Estas tareas utilizaron algunas herramientas proporcionadas por FreeRTOS para gestionar la comunicación de los módulos. En la figura 3.8 se observa un diagrama que muestra la manera en que se realiza la comunicación con ayuda de las herramientas de FreeRTOS.

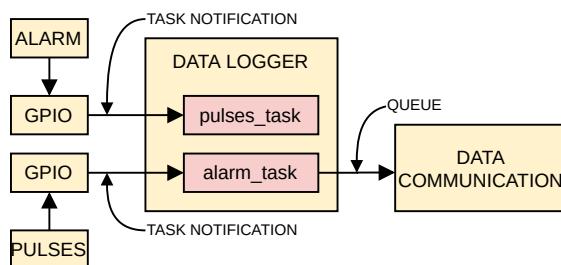


FIGURA 3.8. Diagrama de conexión con las herramientas de FreeRTOS de DATA LOGGER.

De la figura 3.8, ALARM representa las alarmas generadas por el RTC y PULSES los pulsos eléctricos provenientes del conversor óptico-eléctrico. PULSES y ALARM son conectados cada uno a un pin manejado por GPIO, que utiliza interrupciones por flanco de subida para generar notificaciones a pulses\_task y alarm\_task. Una de las funciones de la tarea alarm\_task es enviar por una cola

los datos de consumo eléctrico a DATA COMMUNICATION. Mediante los diagramas de flujo de las figuras 3.9 y 3.10, se puede apreciar el funcionamiento de estas tareas.

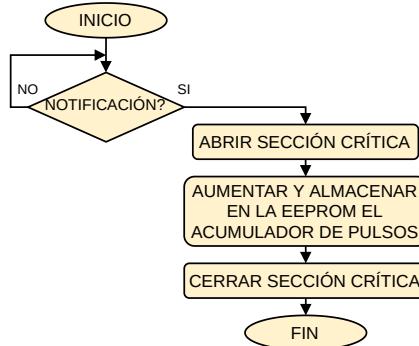


FIGURA 3.9. Diagrama de flujo de la tarea pulses\_task.

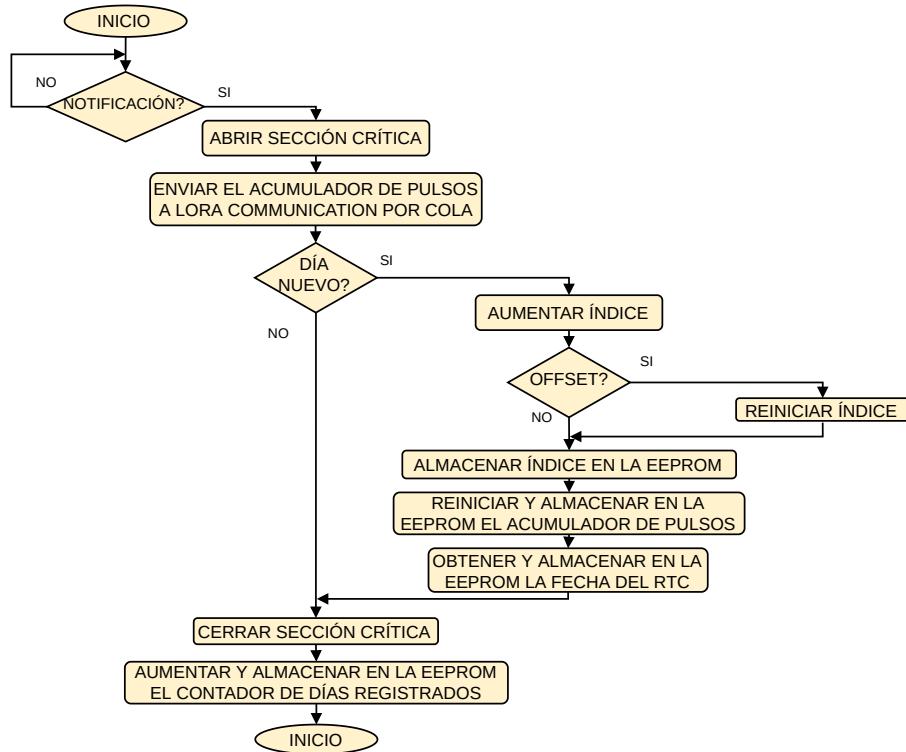


FIGURA 3.10. Diagrama de flujo de la tarea alarm\_task.

Según el diagrama de flujo de la figura 3.9, la tarea pulses\_task espera por una notificación provocada por el flanko de subida de los pulsos eléctricos del conversor óptico-eléctrico. Cuando esto ocurre, se abre una sección crítica para prevenir que existan cambios de contexto dentro del sistema operativo que modifiquen los datos implicados antes de que estos puedan ser utilizados. Una vez en la sección crítica, en una variable de 16 bits se acumulan la cantidad de pulsos detectados y se almacenan en la EEPROM, en una dirección de memoria definida por una variable que hace referencia al índice. Finalmente, se cierra la sección crítica y este proceso se lleva a cabo mientras el dispositivo funcione. Esta es la función más importante del dispositivo, por lo que se le dedicó mayor tiempo y esfuerzo.

En el diagrama de la figura 3.10, los pulsos eléctricos generados por el RTC crean una notificación a la tarea `alarm_task`. Si se detecta una notificación, se abre una sección crítica donde mediante una cola se envía el valor de la variable acumuladora de pulsos al módulo DATA COMMUNICATION. Con ayuda del RTC, si se detecta un cambio de fecha, se ejecutan instrucciones para que la cantidad de pulsos contada a partir de ese momento se reinicie y se almacene en un posición diferente de la EEPROM, lo que evita que los datos en esta memoria se sobrescriban mientras exista espacio suficiente para almacenar más información. Si no se detecta un cambio en la fecha o en caso contrario, se ejecutó todo el proceso antes descrito para la modificación del índice de la EEPROM, la tarea termina, pero vuelve a repetirse cada vez que ocurre una nueva notificación.

Para que este módulo funcione correctamente cuando el dispositivo es encendido se ejecuta una función de inicialización. En el diagrama de flujo de la figura 3.11 se ilustra su comportamiento.

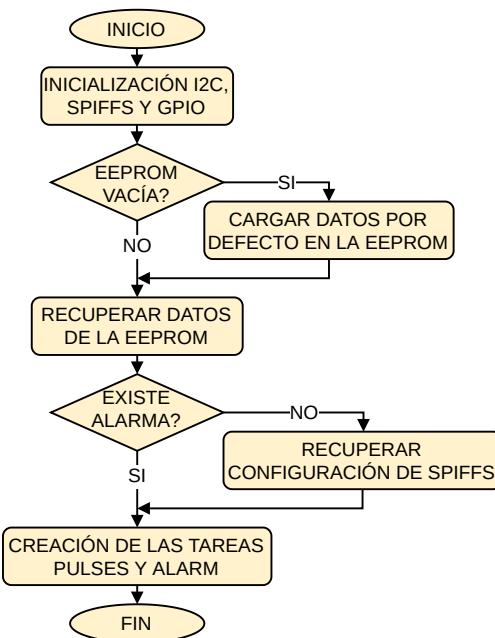


FIGURA 3.11. Diagrama de flujo de la función de inicialización de DATA LOGGER.

El procedimiento de inicialización del módulo, empieza con la configuración de los periféricos I<sup>2</sup>C y GPIO para utilizar sus funciones. También se inicializa el sistema de archivos SPIFFS para tener la capacidad de leer y modificar todos los elementos almacenados. Se hace una lectura de la EEPROM para verificar si esta tiene datos de un funcionamiento anterior y en caso positivo almacenar los datos por defecto. Se recuperan los datos previamente cargados y se configura la alarma del RTC con los datos obtenidos del archivo `config.txt` almacenado en SPIFFS. Finalmente, se crean las tareas `pulses_task` y `alarm_task`.

### 3.2.2. DATA COMMUNICATION

La función de este módulo se basa en utilizar el transceptor LoRa para intercambiar información con un dispositivo concentrador de datos de la misma tecnología. Sus tareas principales son enviar la cantidad de pulsos registrados y recibir

parámetros de funcionamiento. Para esto, se comunica con otros módulos de las capas inferiores como se muestra en la figura 3.12.

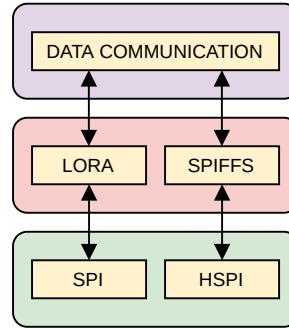


FIGURA 3.12. Diagrama de capas para LORA COMMUNICATION.

Para que este módulo pueda enviar o recibir información utiliza las funciones proporcionadas por LORA, que a su vez utiliza el periférico SPI, para commutar el modo de operación. Cuando recibe información del dispositivo concentrador de datos se accede a SPIFFS para modificar el archivo config.txt, lo que actualiza los parámetros de funcionamiento del dispositivo.

Este módulo posee una sola tarea que se ejecuta en el sistema operativo, nombrada lora\_task. En la figura 3.13 puede observarse su comportamiento mediante un diagrama de flujo.

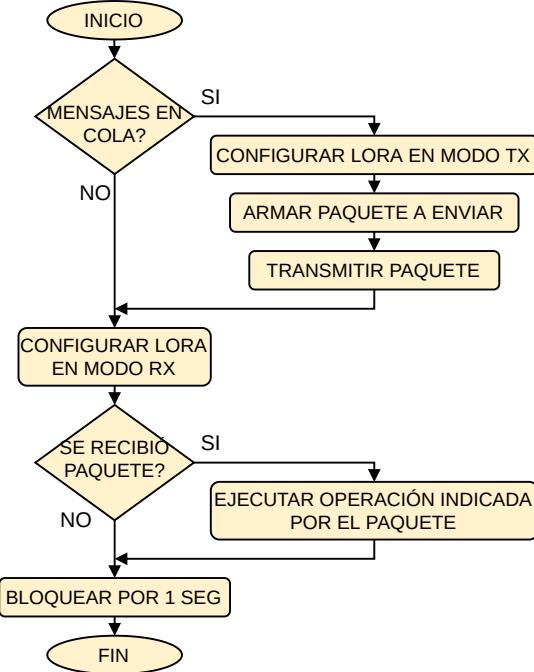


FIGURA 3.13. Diagrama de flujo de la tarea lora\_task.

Del diagrama de la figura 3.13, esta tarea consulta la cola de mensajes para determinar si existe algún elemento pendiente de atención. Si existen mensajes pendientes en la cola se configura el transceptor LoRa en modo de transmisión, se arma un paquete con la información necesaria y se envía. Si la cola está vacía o

se envió un paquete anteriormente, se configura el transceptor LoRa en modo de recepción y cuando se recibe un paquete se ejecuta la acción requerida por este. Finalizado todo este proceso la tarea se bloquea por un segundo y vuelve a repetirse mientras el dispositivo esté en funcionamiento.

El formato de los paquetes es el que se muestra en la figura ???. Donde SRC es un campo de 8 Bytes, identificador de la fuente del paquete. DEST tiene una longitud de 4 Bytes e indica el destinatario del paquete. OP es de 1 Byte y define las operaciones solicitadas al dispositivo, que pueden ser la modificación del archivo config.txt, configuración de la hora y fecha, solicitud de los datos de la EEPROM, por citar algunas. PARAM1 y PARAM2 son campos de 32 Bytes cada uno, contienen los parámetros con los que se deben ejecutar las operaciones requeridas por el campo OP.



FIGURA 3.14. Formato de los paquetes enviados y recibidos por DATA COMMUNICATION.

Este módulo tiene una función de inicialización que debe ser ejecutada cuando el dispositivo es energizado y el ESP8266 empieza a ejecutar el código que tiene grabado. Su comportamiento se muestra en el diagrama de flujo presentado en la figura 3.15

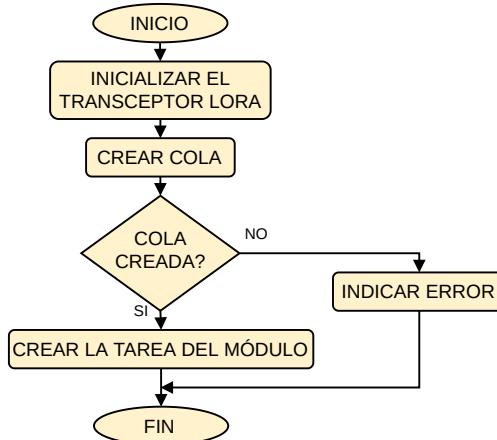


FIGURA 3.15. Diagrama de flujo de la función de inicialización del módulo LORA COMMUNICATION.

Esta función de inicialización ejecuta todos los procesos necesarios para configurar el transceptor LoRa antes de utilizarlo. Posteriormente, intenta crear una cola para recibir información del módulo DATA LOGGER. Si esta no puede ser creada termina la función e indica un error. Finalmente, si el proceso anterior se realizó exitosamente se crea la tarea lora\_tasl que deberá ejecutarse para transmitir y recibir paquetes durante el funcionamiento del dispositivo.

### 3.2.3. WEB SERVER

El objetivo de este módulo es establecer un servidor web con la capacidad de conectarse con dispositivos que dispongan de conexión Wi-Fi, para permitirles leer o modificar lo contenido del sistema de archivos. Para cumplir con lo planteado

anteriormente, se utilizan los componentes de las capas inferiores como indica la figura 3.16.

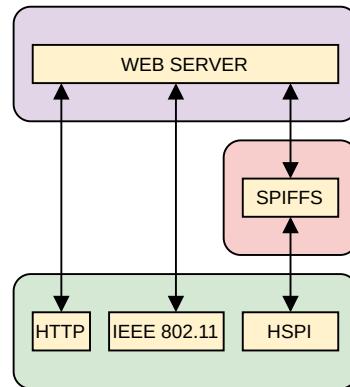


FIGURA 3.16. Diagrama de capas para WEB SERVER.

WEB SERVER utiliza las funciones del protocolo HTTP para establecer un servidor que puede comunicarse con múltiples clientes HTTP mediante los métodos GET y POST, para la transferencia y modificación de los archivos almacenados en SPIFFS. El módulo IEEE 802.11 proporciona funciones para que WEB SERVER configure todas las funciones disponibles del transceptor Wi-Fi del ESP8266.

Este módulo puede configurar el dispositivo como punto de acceso o como estación. Esto se hace de manera automática y depende de la información contenida en el archivo de configuración almacenado en SPIFFS, config.txt. Si existe información de red, el dispositivo se configura como estación, en caso contrario, como punto de acceso. En cualquiera de los dos modos citados los clientes pueden acceder al servidor a través de su dirección de red como indican las figuras 3.17 y 3.18.

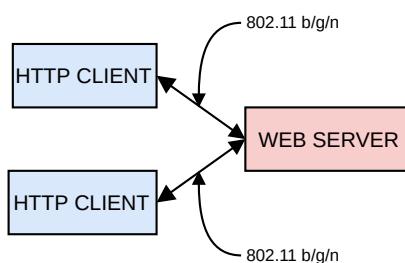


FIGURA 3.17. WEB SERVER en modo punto de acceso.

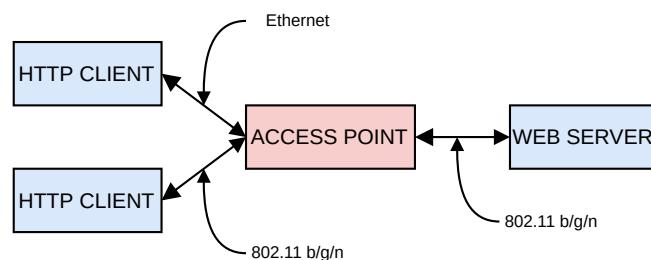


FIGURA 3.18. WEB SERVER en modo estación.

En la figura 3.17, el dispositivo está configurado en modo punto de acceso y el servidor web puede ser accedido directamente por un cliente HTTP que cuente con conectividad Wi-Fi. Por otro lado, en la figura 3.18 el dispositivo está configurado en modo estación y los clientes HTTP solo podrán acceder a este a través de un punto de acceso con conectividad Wi-Fi que enrute las conexiones.

WEB SERVER tiene la capacidad de responder a peticiones GET y POST provenientes de los clientes HTTP gracias a una tarea propia del ESP8266\_RTOS\_SDK' que se ejecuta todo el tiempo en el sistema operativo. El método GET es utilizado para solicitar los archivos necesarios para generar la interfaz web, mientras que el método POST se utiliza para modificar el archivo config.txt almacenado en SPIFFS. Para esto, WEB SERVER utiliza funciones conocidas como *handlers* que se ejecutan para transferir los recursos cuyos nombres coinciden con la URI (*Uniform Resource Identifier, identificador de recursos uniforme*) de la petición con el método GET. En el caso del método POST se lee el cuerpo del mensaje recibido para extraer los parámetros con los que debe ser modificado config.txt y actualizar la información de conexión de red Wi-Fi.

Como los módulos DATA LOGGER y DATA COMMUNICATION, WEB SERVER también posee una función de inicialización que configura todos los módulos de capas inferiores de los que depende, para que pueda cumplir su propósito. El diagrama de flujo de la figura 3.19 es utilizado para explicar su funcionamiento.

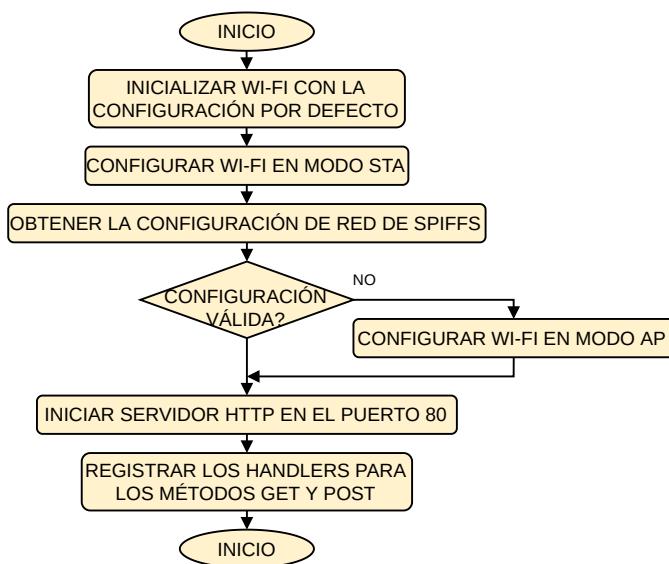


FIGURA 3.19. Diagrama de flujo de la función de inicialización del módulo WEB SERVER.

En esta función el primer paso es inicializar la interfaz física de Wi-Fi con la configuración por defecto del ESP8266. Se configura el dispositivo en modo estación y se busca en el archivo config.txt configuración de red que permita conectarse a una red Wi-Fi existente. Si no existe dicha información o es inválida, se configura el dispositivo en modo punto de acceso. En cualquiera de los dos casos el siguiente paso es iniciar un servidor HTTP en el puerto 80 y finalmente registrar todos los handlers para los métodos GET y POST.

### 3.3. Interfaz web

El diseño e implementación de una interfaz web tiene como objetivo proporcionar a los usuarios, es decir, a los abonados de las compañías eléctricas, la capacidad de interactuar con el dispositivo para visualizar gráficamente información relativa a su consumo eléctrico y configurar parámetros de la conexión Wi-Fi.

Para el desarrollo se utilizó el IDE Visual Studio Code, que ofrece un entorno de desarrollo muy intuitivo y también brinda la posibilidad de descargar *plugins* que facilitan la escritura de código. Asimismo, se utilizaron distintos lenguajes enfocados en el desarrollo web, para brindar a la interfaz una estructura bien definida, estética y funcionalidad. Estos fueron:

- HTML: se utilizó para definir todos los aspectos estructurales de la interfaz, como la ubicación de los elementos, las llamadas a bibliotecas externas y otros parámetros informativos. La versión utilizada fue HTML 5.
- CSS: brindó control sobre la presentación, formato y el diseño de la interfaz.
- JavaScript: permitió dotar de funcionalidad a los elementos de la interfaz. Fue necesaria para realizar el procesamiento de los datos provenientes del dispositivo.
- jQuery Mobile: con esta biblioteca fue posible darle a la interfaz un aspecto de aplicación para teléfonos móviles, además de la capacidad de adaptarse a cualquier tamaño de pantalla sin que la información mostrada se vea alterada.
- Highcharts: a través de esta biblioteca se logró exhibir la información de consumo eléctrico en un gráfico de barras, de esta manera es más comprensible para el usuario.

La interfaz web está dividida en dos pantallas, principal y de configuración. La primera es meramente informativa y es donde se muestra el consumo eléctrico al usuario. La segunda permite conectar el dispositivo a un red Wi-Fi existente.

La pantalla principal fue diseñada pensando en brindarle al usuario la información de su consumo eléctrico de la manera más simple posible. En la mayor parte del área de la pantalla se muestra un gráfico de barras que presenta el consumo eléctrico de los últimos tres meses y en la esquina superior izquierda, un pequeño botón que dirige a la pantalla de configuración.

Al cargar la interfaz en un navegador web, se obtiene mediante el método GET el archivo pulses.csv que contiene los valores de consumo eléctrico que están almacenados en el dispositivo. Estos son procesados con instrucciones escritas en JavaScript para que la biblioteca Highcharts los utilice y genere el gráfico de barras. En la figura 3.20 se observa la pantalla principal de la interfaz web.

Se diseñó la pantalla de configuración para que la única configuración que puede realizarse sea la conexión del dispositivo a una red Wi-Fi existente a través de su SSID y contraseña. Esta pantalla es imprescindible, debido a que el dispositivo no debería ser manipulado manualmente bajo ninguna circunstancia por el usuario y se necesitaba una forma de realizar esta configuración.

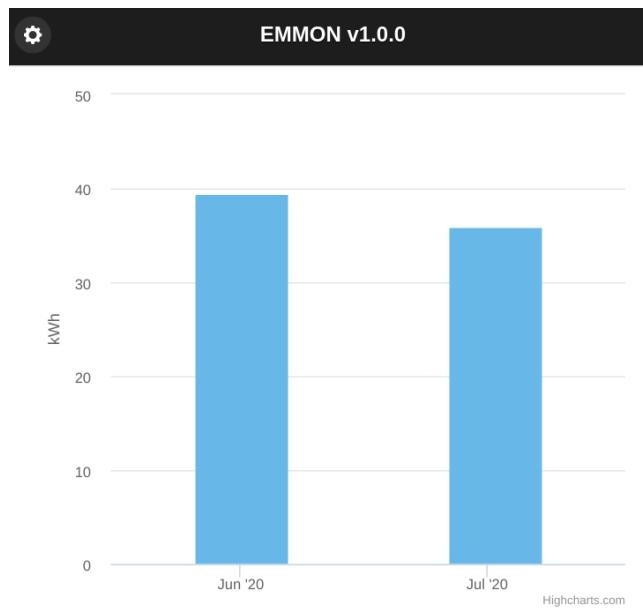


FIGURA 3.20. Pantalla principal de la interfaz web.

El componente principal es un formulario para ingresar el SSID y la contraseña de la red a la que el usuario desea conectar el dispositivo. En la esquina superior izquierda se encuentra un botón para retornar a la pantalla principal y en la esquina superior derecha, un botón para enviar por el método POST el contenido del formulario al dispositivo. En la figura 3.21 se muestra la pantalla de configuración de la interfaz web.

The figure shows a configuration form titled "Access Point". At the top left is a back arrow icon, and at the top right is a checkmark icon. Below the title, there is a placeholder text: "Ingrese el el nombre ya la contraseña de la red Wi-Fi a la que quiera conectarse.". The form has two input fields: "SSID:" and "Password:", each with a corresponding text input box below it. Below these fields is a large, empty gray rectangular area, likely a placeholder for additional content or a status message. The entire form is contained within a white rectangular frame.

FIGURA 3.21. Pantalla de configuración de la interfaz web.

### 3.4. Prototipo comercial

El desarrollo de un prototipo para ser comercializado fue necesario para una primera implementación del dispositivo en un entorno real de trabajo y la realización de pruebas a nivel físico. Consta de una carcasa y un PCB (*Printed Circuit Board*, tarjeta de circuito impreso).

El primer paso fue elegir una carcasa de dimensiones adecuadas para que pueda ser montada directamente sobre un medidor de consumo eléctrico domiciliario. Para este fin, se estudió la posibilidad de diseñar una carcasa personalizada, pero, debido a los altos costos de producción a nivel de prototipo, esta idea fue rápidamente descartada. Entonces, después de realizar un análisis de las dimensiones de los medidores utilizados por COPELECT, se eligió una carcasa disponible en el mercado internacional, la VG-S43 de la firma Vange. La elección de esta carcasa sobre otras similares, fue debido a los zócalos que tiene, que se adecuaban perfectamente para que el fototransistor estuviera descubierto y tuviera vista directa con el LED del medidor eléctrico. En la figura 3.22 se puede apreciar la carcasa elegida.

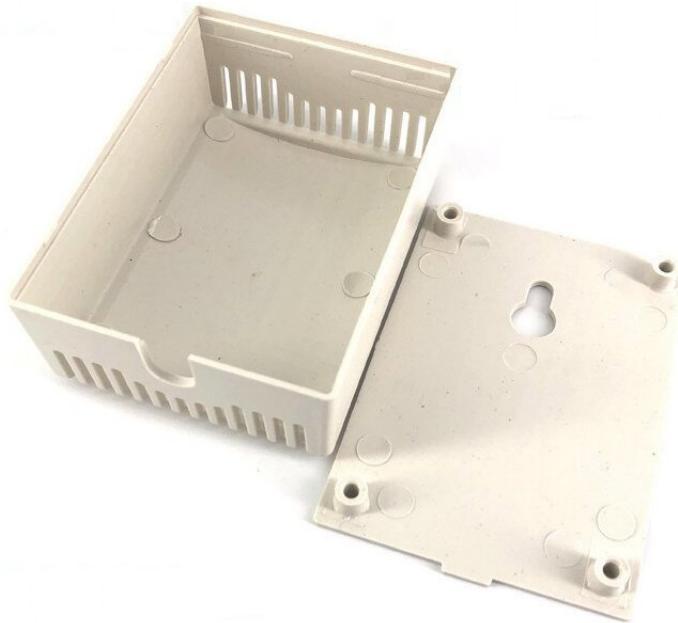


FIGURA 3.22. Caja VG-S43 de la firma Vange<sup>5</sup>.

Antes de empezar con el diseño del PCB, se realizó la elección de los componentes que serían parte del mismo. En el prototipo de pruebas se utilizaron módulos y tarjetas de desarrollo que con el firmware implementado en ellos, cumplieron todos los requerimientos planteados. Entonces, para que el firmware desarrollado pudiera ser utilizado exitosamente en el prototipo comercial, se utilizaron los circuitos integrados principales de los módulos y tarjetas de desarrollo, también se descartaron los componentes electrónicos que no resultaban necesarios para este trabajo. Existen dos componentes que se implementaron como módulos, el ESP-12S que es una variante del ESP-12F, componente principal de la NodeMCU, y el RA-01, que es un transceptor LoRa basado en el mismo circuito integrado que

<sup>5</sup>Imagen tomada de: <https://es.aliexpress.com/item/33004284623.html?spm=a2g0o.cart.0.0.50483c00xuS0Xo&mp=1>

el PM1280, el SX1278. Además, el PT333-3C fue sustituido por el PT11-21C, que también es un fototransistor de similares características, pero es un SMD (*Surface-Mount-Device*, dispositivo de montaje superficial).

Una vez elegidos los componentes implicados, se realizó un análisis del consumo de corriente de cada uno de ellos para implementar una fuente de alimentación adecuada. Cabe resaltar que la tensión de alimentación de todos los componentes es 3,3 V. En la tabla 3.2 se muestran los valores máximos de consumo de corriente de los componentes, estos datos fueron obtenidos de los respectivos datasheets.

TABLA 3.2. Tabla de consumo de corriente eléctrica de los componentes del prototipo comercial.

| Componente | Consumo de corriente (mA)             |
|------------|---------------------------------------|
| ESP-12S    | 500 (en modo de transmisión continua) |
| RA-01      | 93 (en modo transmisor)               |
| DS3231     | 0,2 (en modo activo)                  |
| AT24C32    | 3 (cuando se escribe un dato)         |
| LM393      | 20 (cortocircuitado a tierra)         |
| PT11-21C   | 20                                    |

De la tabla 3.2, se determinó que el consumo total de todos los componentes es de 636,2 mA. Al momento de elegir la fuente de alimentación al consumo total se le añadió un margen de seguridad del 50 %, que dio un nuevo valor de 954,43 mA. Por lo tanto, la fuente de alimentación elegida debió ser de 3.3 V y 1 A.

Para reducir la cantidad de componentes de la fuente de alimentación, se escogió un módulo conversor de energía alterna a directa. De esta forma, el prototipo comercial podría conectarse directamente a la misma línea eléctrica del medidor. El componente elegido fue el módulo HLK-PM03 de la firma Hi-Link, que proporciona 3,3 V y 1 A a su salida, cuando a la entrada existen 90 V - 240 V alternos. En la figura 3.23 puede observarse el módulo para la fuente de alimentación.



FIGURA 3.23. Módulo de alimentación HLK-PM03 de la firma Hi-Link<sup>6</sup>.

Con ayuda del software KiCAD, se realizó el dibujo de un diagrama esquemático del prototipo comercial, que interconecta todos los componentes y brinda información relacionada a aspectos importantes sobre el funcionamiento y diseño del PCB. En la figura 3.24 se muestra el diagrama esquemático del prototipo comercial.

<sup>6</sup>Imagen tomada de: <https://es.aliexpress.com/item/33004284623.html?spm=a2g0o.cart.0.0.50483c00xuS0Xo&mp=1>

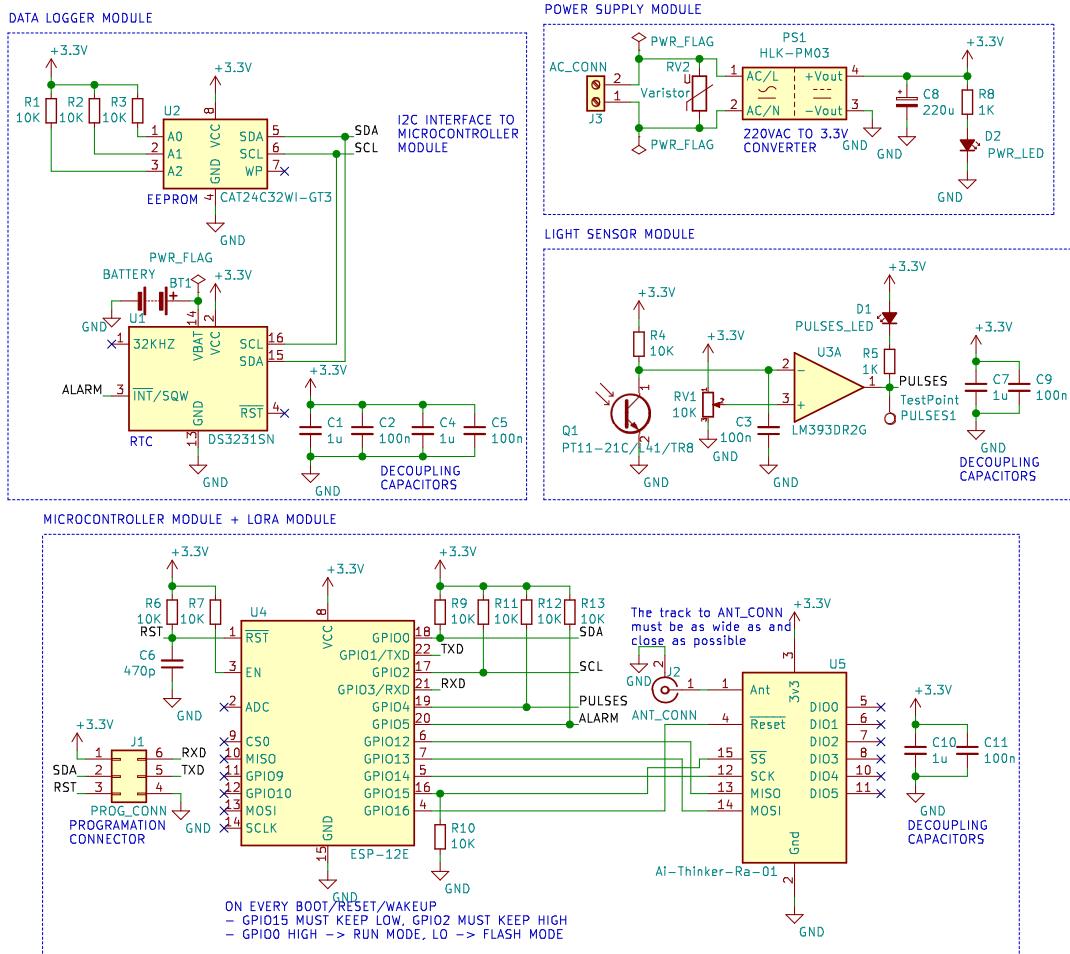


FIGURA 3.24. Diagrama esquemático del prototipo comercial.

Del diagrama anterior, se puede notar que se añadieron *test points* para poder probar la respuesta del sensor de luz mediante instrumentación especializada. Se añadieron también un conector destinado a la depuración del código almacenado en el ESP8266, junto con LEDs para monitorear el estado de la fuente y el sensor de luz.

Con el diagrama esquemático finalizado, se realizó la ERC (*Electrical Rule Check*, comprobación de reglas eléctricas) en busca de posibles cortocircuitos, conexiones ilegales y contactos flotantes entre otras comprobaciones. Posteriormente, se dibujó el circuito impreso, donde se tuvieron en consideración las restricciones físicas impuestas por la elección de la carcasa. Se hizo especial énfasis en la ubicación de los conectores, para que quedaran al borde del PCB y pudieran ser accedidos con mayor facilidad. El fototransistor quedó ubicado en una posición tal que coincidiera con el zócalo inferior de la carcasa. Otra consideración de importancia fue la distancia entre el transceptor LoRa y el conector coaxial, ambos componentes fueron ubicados muy cerca, de tal forma que la pista que los conectaba tuviera una distancia muy corta. Asimismo, se dibujó la pista lo más ancha posible y se pusieron vías conectadas a tierra para lograr una mejor respuesta a las interferencias electromagnéticas.

Las capas *top* y *bottom* del PCB, pueden apreciarse en las figuras 3.25 y 3.26, respectivamente. Por otro parte, en las figuras 3.27 y 3.28, se muestran el modelo 3D renderizado del PCB y una fotografía del PCB montado.

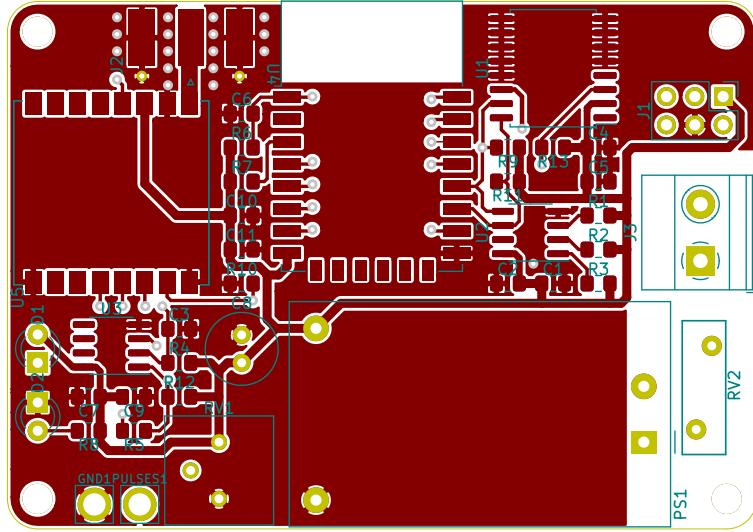


FIGURA 3.25. Capa top del PCB.

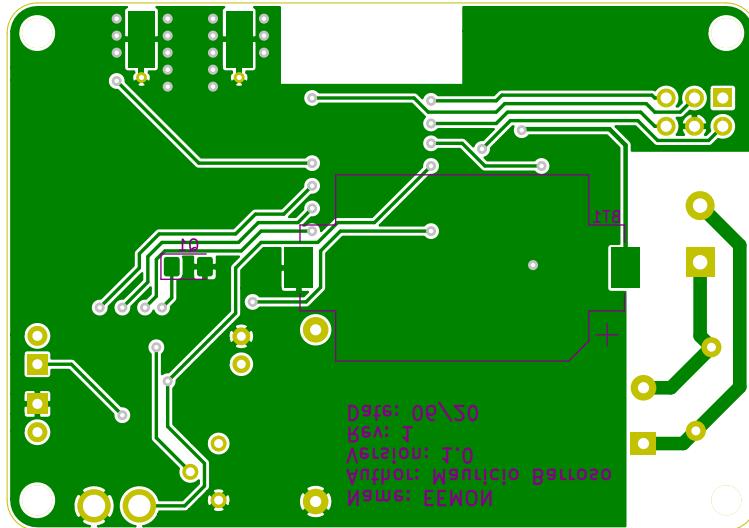


FIGURA 3.26. Capa bottom del PCB.

La manufactura del PCB fue realizada por el fabricante JLCPCB y los componentes fueron adquiridos de la firma LCSC. Ambos fueron elegidos por los costos reducidos que ofrecen en sus productos, además de que JLCPCB ofrece el servicio de PCBA (*Printed Circuit Board Assembly*, montaje de PCB) con los componentes que tiene disponibles LCSC.

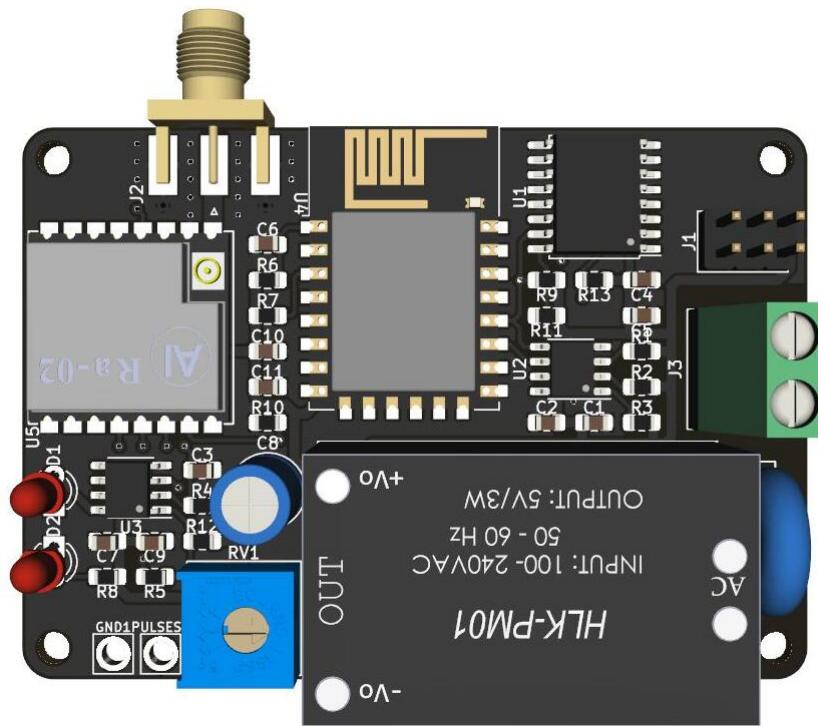


FIGURA 3.27. Modelo 3D del PCB montado del prototipo comercial.

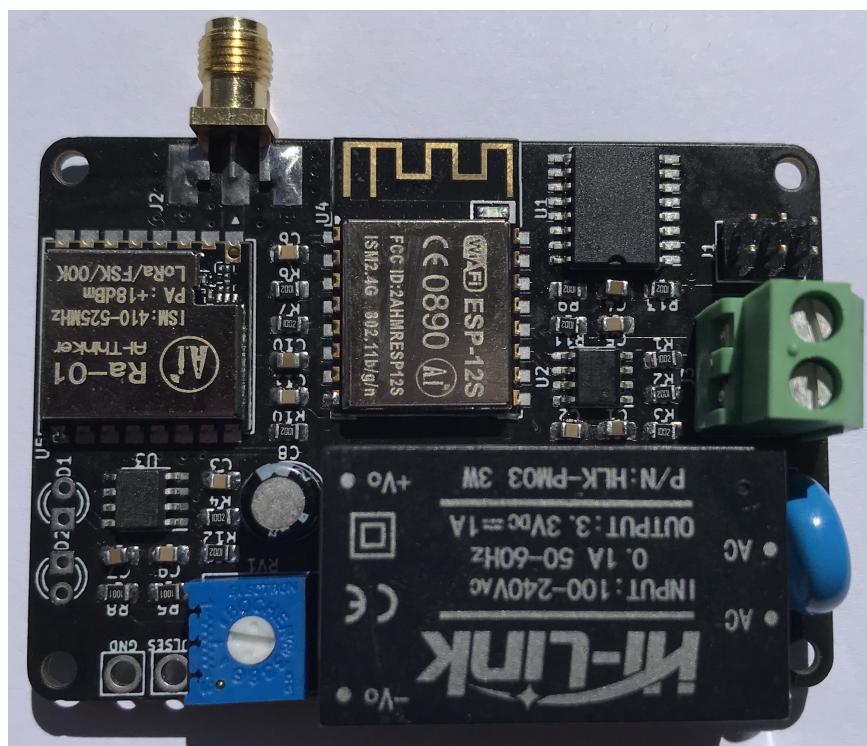


FIGURA 3.28. PCB montado del prototipo comercial.

## Capítulo 4

# Ensayos y resultados

En este capítulo se presentan las pruebas realizadas sobre los prototipos de pruebas y comercial. Además se exhiben los resultados obtenidos que validan su correcto funcionamiento. Las pruebas fueron realizadas sobre el firmware y hardware expuestos en el capítulo 3.

### 4.1. Pruebas unitarias

Se hicieron pruebas unitarias sobre las bibliotecas desarrolladas para el manejo de los circuitos integrados DS3231, AT24C32 y SX1278. Se utilizó Ceedling para ejecutar dichas pruebas en combinación con Gcov para generar los análisis de cobertura correspondientes. En la tabla 4.1 se pueden observar los resultados de las pruebas unitarias y en la tabla 4.2 se exhibe el análisis de cobertura.

TABLA 4.1. Tabla de resultados de las pruebas unitarias.

| Biblioteca | Cantidad de tests | Exitosos | Fallidos |
|------------|-------------------|----------|----------|
| AT24C32    | 8                 | 8        | 0        |
| DS3231     | 11                | 11       | 0        |
| SX1278     | 14                | 14       | 0        |

TABLA 4.2. Tabla de resultados del análisis de cobertura.

| Archivo   | Líneas ejecutadas | Funciones ejecutadas |
|-----------|-------------------|----------------------|
| at24c32.c | 52/52             | 6/6                  |
| ds3231.c  | 54/62             | 11/13                |
| sx1278.c  | 172/220           | 26/31                |

### 4.2. Pruebas funcionales de firmware

Se probaron los módulos DATA LOGGER, LORA COMMUNICATION y WEB SERVER de la capa superior del firmware, APP. Durante la etapa de desarrollo del firmware, estos módulos fueron probados para garantizar su correcto funcionamiento, de acuerdo con la planificación del trabajo descrita en el capítulo 2. El banco de pruebas utilizado consiste en el prototipo de pruebas conectado a una PC por medio de un cable micro USB. También se utilizó un medidor eléctrico modelo LUMEN 2 MC de la firma Nansen, que fue facilitado por COPELECT. El banco de pruebas se muestra en la figura 4.1.

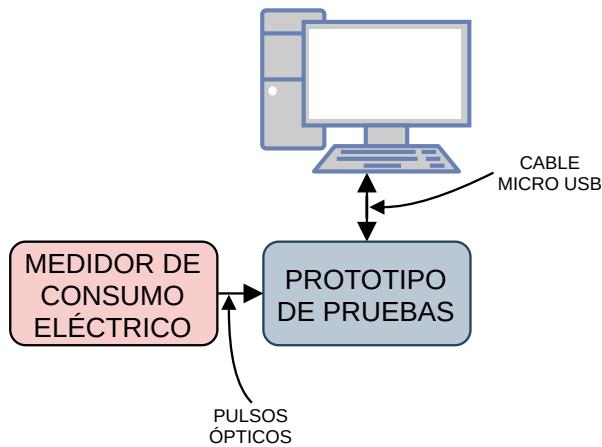


FIGURA 4.1. Banco de pruebas para evaluar el funcionamiento del firmware.

Las pruebas consistieron en monitorear a través de la PC el funcionamiento de los módulos que componen la capa APP. Para esto, se añadieron instrucciones en el código fuente de estos módulos, que sirvieron para imprimir mensajes por el puerto serial. En la PC se ejecutó la utilidad idf-monitor, que es una terminal para puerto serial incluida en el ESP8266\_RTOS\_SDK. A medida que se desarrollaron los módulos, estos fueron probados individualmente verificando su correcto funcionamiento.

Con todos los módulos funcionando individualmente, se realizó la prueba de integración de la capa APP. En la figura 4.2 se observa una captura de pantalla del idf-monitor cuando el dispositivo inicia su operación.

```

I (547) gpio: GPIO[2]| InputEn: 0| OutputEn: 1
| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0

I (561) i2c_module: Configured I2C parameters
I (569) i2c_module: I2C interface configured!
I (577) spiffs: Initializing SPIFFS
I (750) spiffs: Partition size: total: 1920401
, used: 556467
I (758) data_logger: index:0x123
I (763) data_logger: pulses:468
I (769) data_logger: logged days:55
I (803) data_logger: date:31/07/20
I (823) data_logger: Alarm configured!
I (828) data_logger: Configuring GPIOs...
I (833) gpio: GPIO[4]| InputEn: 1| OutputEn: 0
| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:2

I (843) gpio: GPIO[5]| InputEn: 1| OutputEn: 0
| openDrain: 0| Pullup: 1| Pulldown: 0| Intr:2

I (854) data_logger: GPIOs configured!
I (861) data_logger: Tasks created!
phy_version: 1163.0, 665d56c, Jun 24 2020, 10:
00:08, RTOS new
I (1421) phy_init: phy ver: 1163_0
I (1565) web_server: wifi set to STA mode
I (1770) wifi: state: 0 -> 2 (b0)
I (1784) wifi: state: 2 -> 3 (0)
I (1790) wifi: state: 3 -> 5 (10)
I (3880) tcpip_adapter: sta ip: 192.168.1.104,
mask: 255.255.255.0, gw: 192.168.1.1
  
```

FIGURA 4.2. Captura de pantalla de idf-monitor cuando el dispositivo inicia.

Las funciones que se ejecutan en el sistema operativo del dispositivo también generaron mensajes informativos. En la captura de pantalla de la figura 4.3, se observan los mensajes que imprimen las tareas de los módulos cuando funciona normalmente.

```
I (213761) data_logger: [0x123]=509
I (213766) data_logger: 31/07/20,18:20:39
I (215671) data_logger: [0x123]=510
I (215678) data_logger: 31/07/20,18:20:41
I (217582) data_logger: [0x123]=511
I (217587) data_logger: 31/07/20,18:20:43
I (219493) data_logger: [0x123]=512
I (219498) data_logger: 31/07/20,18:20:45
I (220321) web_server: Sending file : /favicon.ico (1150 bytes)...
I (220343) web_server: File sending complete
I (221403) data_logger: [0x123]=513
I (221410) data_logger: 31/07/20,18:20:47
I (221923) lora_communication: packet received!
I (221929) lora_communication: 15320101000000
I (221935) lora communication: RSSI: -31
```

FIGURA 4.3. Captura de pantalla de idf-monitor cuando el dispositivo ejecuta sus funciones normales.

Con ayuda de todos los mensajes generados, además de los diagramas de flujo presentados en el capítulo 3, se pudo probar que los módulos de firmware del dispositivo funcionan correctamente.

### 4.3. Pruebas de la interfaz web

Las pruebas realizadas sobre la interfaz web tuvieron la finalidad de corroborar su funcionalidad. De acuerdo a lo expuesto en el capítulo 3, el dispositivo puede ser configurado mediante el módulo WEB SERVER en dos modos de operación. Entonces, se realizaron dos tipos de pruebas distintas, una con el dispositivo como punto de acceso y la otra como estación. Para estas pruebas se utilizó una PC, un cable micro USB, un router Wi-Fi TL-WR940N de la firma TP-Link y una laptop con el navegador web Chrome instalado. En la figura 4.4 se puede ver un diagrama del banco de pruebas montado.

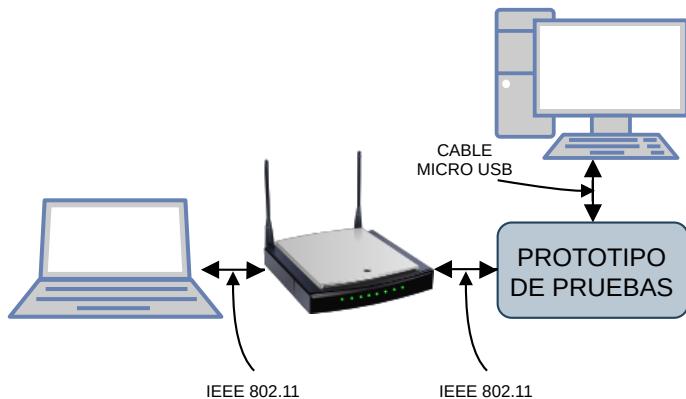


FIGURA 4.4. Banco de pruebas para verificar el funcionamiento de la interfaz web cuando el dispositivo está en modo punto de acceso.

El primer paso fue eliminar todas las configuraciones existentes en el sistema de archivos del dispositivo, lo que provocó que al iniciar se ejecuten las instrucciones por defecto del mismo. Entonces, el dispositivo se configuró automáticamente como punto de acceso y se conectó la laptop a la red Wi-Fi que generó. En la figura 4.5 se observa la red Wi-Fi generada por el dispositivo en el administrador de redes de la laptop.

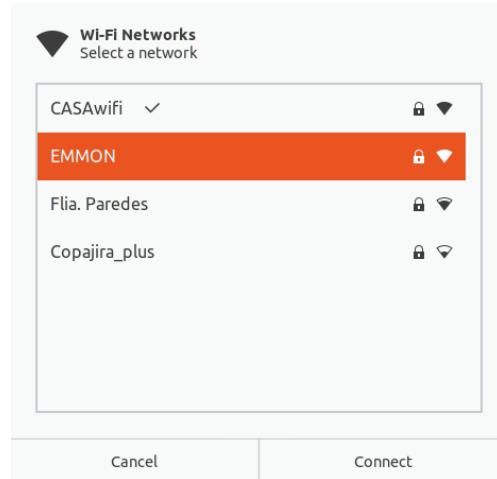


FIGURA 4.5. Captura de pantalla de las redes Wi-Fi disponibles en la laptop.

El siguiente paso fue ingresar a la dirección de red del dispositivo mediante el navegador web de la laptop, que dio como resultado la transferencia del archivo index.html. Este archivo HTML solicitó automáticamente al dispositivo mediante el método GET todos los elementos restantes para generar la interfaz web. Para verificar que las transferencias de estos archivos se hicieran correctamente, para el lado del prototipo de pruebas se utilizó el idf-monitor y para el lado de la laptop, se hizo uso de la herramienta de depuración del navegador. En las figuras 4.6 y 4.7, se muestran capturas de pantalla de la utilidad de depuración del navegador y la salida del idf-monitor, respectivamente.

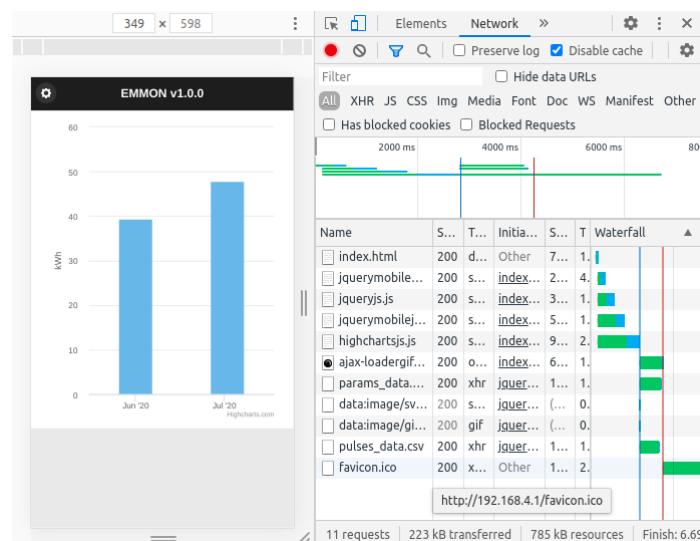


FIGURA 4.6. Captura de pantalla de la página principal de la interfaz web con la utilidad de depuración funcionando.

```
I (6430) web_server: Sending file : /index.html
(7292 bytes)...
I (6650) web_server: File sending complete
I (6874) web_server: Sending file : /jquerymobile.css (25107 bytes)...
I (7103) web_server: File sending complete
I (7177) web_server: Sending file : /jquery.js (33189 bytes)...
I (8009) web_server: File sending complete
I (8042) web_server: Sending file : /jquerymobile.js (55533 bytes)...
I (13120) web_server: File sending complete
I (13244) web_server: Sending file : /highcharts.js (91994 bytes)...
I (17146) web_server: File sending complete
I (17292) web_server: Sending file : /ajax-loader.gif (6240 bytes)...
I (17645) web_server: File sending complete
I (17713) web_server: Sending file : /pulses_data.csv (935 bytes)...
I (17735) web_server: File sending complete
I (17866) web_server: Sending file : /params_data.txt (0 bytes)...
I (17876) web_server: File sending complete
I (17993) web_server: Sending file : /favicon.ico (1150 bytes)...
I (18021) web_server: File sending complete
```

FIGURA 4.7. Captura de pantalla del idf-monitor después de enviar los archivos solicitados por el navegador web y el dispositivo en modo punto de acceso.

La siguiente prueba consistió en ingresar a la página de configuración de la interfaz web a través el botón ubicado en la esquina superior izquierda de la página principal. Ahí se llenó el formulario con los datos de la red Wi-Fi generada por el router, es decir, su SSID y su contraseña. Se utilizó el botón ubicado en la esquina superior derecha para enviar estos datos al prototipo de pruebas con el método POST. Con esta información el módulo WEB SERVER cambió la configuración al modo estación y pudo conectarse al router, que le proporcionó una dirección de red. Por último, la laptop también se conectó a la red del router y se utilizó el navegador web junto con la nueva dirección de red del prototipo de pruebas para solicitar los archivos de la interfaz web. En las figuras 4.8 y 4.9, se pueden observar una captura de pantalla con los campos del formulario llenados y la salida del idf-monitor, respectivamente.

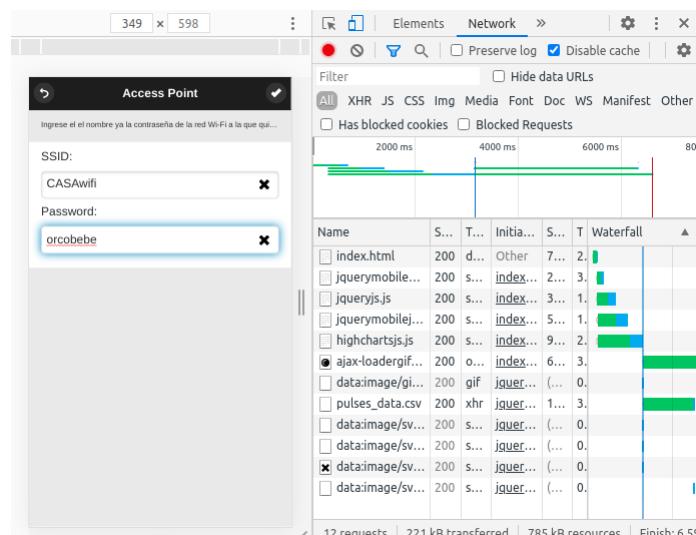


FIGURA 4.8. Captura de pantalla de la página de configuración de la interfaz web con la utilidad de depuración funcionando.

```
I (756749) web_server: wifi set to STA mode
I (756759) web_server: station 9c:fc:e8:75:28:3
S leave, AID=1
I (758908) wifi: state: 0 -> 2 (b0)
I (758914) wifi: state: 2 -> 3 (0)
I (758920) wifi: state: 3 -> 5 (10)
I (760418) tcpip_adapter: sta ip: 192.168.1.104
, mask: 255.255.255.0, gw: 192.168.1.1
I (760438) web_server: Connected to AP with SSI
D:CASAwifi_and_password:orcobebe
```

FIGURA 4.9. Captura de pantalla del idf-monitor después de configurar el dispositivo en modo estación con los datos enviados por la interfaz web.

Al finalizar estas pruebas se pudo evidenciar el correcto funcionamiento de las dos páginas de la interfaz web. Asimismo, implícitamente se verificó que el módulo de firmware WEB SERVER respondía las peticiones con los métodos GET y POST según lo esperado.

#### 4.4. Pruebas de laboratorio

Estas pruebas tuvieron como objetivo principal utilizar instrumentación especializada para verificar el buen funcionamiento del conversor óptico-eléctrico y la fuente de alimentación.

El propósito de la prueba del conversor óptico-eléctrico fue observar la forma de onda que genera, para implementar un algoritmo en el firmware que evite la detección de pulsos falsos consecuencia de las características intrínsecas del LED del medidor de consumo eléctrico proporcionado por COPELECT. Para llevar a cabo esta prueba se utilizó un osciloscopio TDS2000C de la firma Tektronix, el prototipo comercial y el medidor proporcionado por COPELECT. El banco de pruebas puede observarse en el diagrama de la figura 4.10.

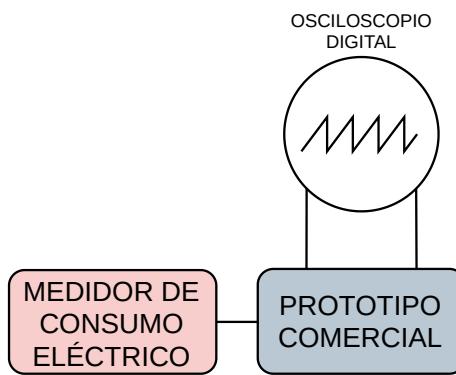


FIGURA 4.10. Banco de pruebas para el conversor óptico-eléctrico.

De la figura 4.11, se puede observar que la forma de onda producida por el medidor tiene elementos que pueden ocasionar que el módulo DATA LOGGER registre erróneamente los pulsos y generar un reporte erróneo del consumo de energía eléctrica. Para solucionar esto, se implementó una función similar a la utilizada para detectar rebotes en los pulsadores en DATA LOGGER. Con esto se evitó en gran medida el error antes mencionado.

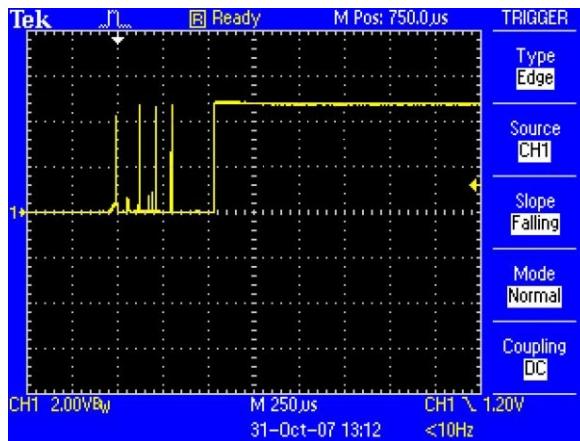


FIGURA 4.11. Salida de la pantalla del osciloscopio.

La prueba de la fuente de alimentación tuvo como propósito excitar este elemento con una fuente de tensión alterna, que simuló el comportamiento de la línea de alimentación cuando existen cambios en su valor nominal. Los elementos utilizados fueron una fuente de tensión alterna variable modelo 1653A de la firma BK precisión, un reóstato como carga variable y dos multímetros MUT-39 de la firma Truper. El banco de pruebas utilizado se ilustra en la figura 4.12

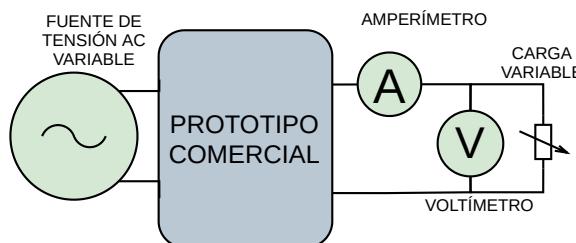


FIGURA 4.12. Banco de pruebas para el conversor óptico-eléctrico.

El procedimiento consistió en establecer el nivel de tensión de entrada en un valor determinado y variar la carga conectada a la salida, para registrar los datos obtenidos del amperímetro y el voltímetro conectados en serie y paralelo, respectivamente. Los valores de tensión de entrada fueron el valor nominal de la fuente de alimentación, el valor nominal menos el 20 % y el valor nominal más el 20 %. En las tablas 4.3, 4.4 y 4.5, se pueden apreciar los resultados obtenidos de estas pruebas.

TABLA 4.3. Tabla de resultados de la prueba de la fuente de alimentación con 176 VAC

| Voltaje (V) | Corriente (A) |
|-------------|---------------|
| 3.27        | 0.2           |
| 3.26        | 0.4           |
| 3.24        | 0.6           |
| 3.21        | 0.8           |
| 3.15        | 1             |

TABLA 4.4. Tabla de resultados de la prueba de la fuente de alimentación con 220 VAC

| Voltaje (V) | Corriente (A) |
|-------------|---------------|
| 3.33        | 0.2           |
| 3.32        | 0.4           |
| 3.3         | 0.6           |
| 3.28        | 0.8           |
| 3.24        | 1             |

TABLA 4.5. Tabla de resultados de la prueba de la fuente de alimentación con 264 VAC

| Voltaje (V) | Corriente (A) |
|-------------|---------------|
| 3.38        | 0.2           |
| 3.36        | 0.4           |
| 3.33        | 0.6           |
| 3.31        | 0.8           |
| 3.28        | 1             |

Para visualizar más fácilmente los resultados de estas pruebas y tener una perspectiva más clara sobre la variación de la tensión de salida en función de la corriente que circula por la carga, en la figura 4.13 se presentan gráficamente los resultados de las pruebas anteriores. Donde la línea roja representa la prueba con 264 VAC, la línea verde la prueba con 220 VAC y la línea azul la prueba con 176 VAC.

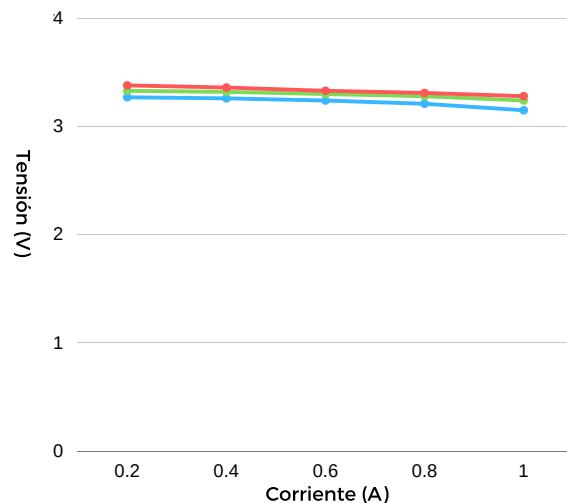


FIGURA 4.13. Gráfico de líneas del comportamiento de la fuente de alimentación.

Entonces, según los valores necesarios para alimentar los componentes del prototipo comercial expuestos en la tabla 3.2 y con ayuda de las pruebas realizadas sobre la fuente de alimentación, se concluye que la fuente fue elegida correctamente para brindar los niveles de tensión y corriente adecuados cuando el valor de tensión de la línea eléctrica varíe en más o menos 20 %.

## 4.5. Pruebas del transceptor LoRa

Estas pruebas fueron realizadas para determinar los parámetros adecuados del transceptor LoRa, para intercambiar información con un gateway de la misma tecnología que está ubicado en el edificio central de COPELECT. Para esto se utilizaron principalmente el prototipo comercial del dispositivo y un gateway LoRa basado en la plataforma Arduino y en el módulo LoRa PM1280. Otros elementos utilizados fueron una PC, una laptop y cables micro USB. El banco de ensayos puede observarse en la figura 4.14.

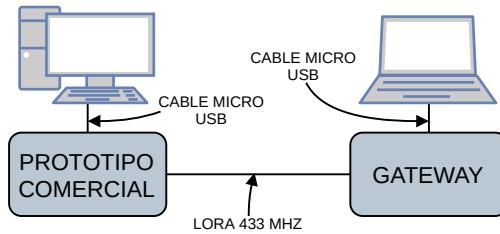


FIGURA 4.14. Captura de pantalla de idf-monitor después de enviar los archivos para la interfaz web.

El gateway LoRa fue ubicado en la azotea del edificio central de COPELECT, que es el lugar donde debería instalarse un gateway LoRaWAN finalmente. El prototipo comercial se dispuso en el domicilio del autor, más precisamente en el mismo gabinete donde se encuentra instalado el medidor eléctrico. En la figura 4.15 se muestra la ubicación del gateway LoRa y el prototipo comercial.

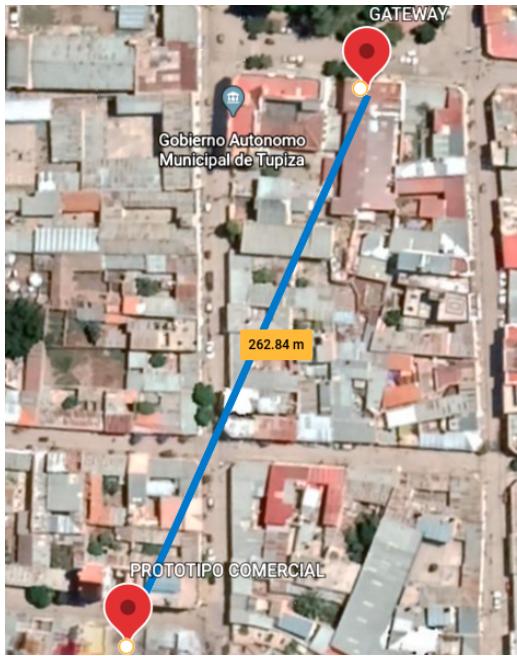


FIGURA 4.15. Captura de pantalla de la ubicación del gateway LoRa y el prototipo comercial.

Básicamente la prueba realizada consistió en el envío de un paquete como que tiene la estructura expuesta en la figura ?? por parte del prototipo comercial. Una vez que el gateway lo recibe y procesa, devuelve como respuesta un paquete con la estructura que exhibe la figura 3.14. Con el serial monitor de Arduino instalado

en la laptop se monitoreó el gateway. Mientras que para monitorear el prototipo comercial se utilizó el idf-monitor instalado en la PC.

Se probaron distintos tipos de configuraciones para lograr una comunicación exitosa entre ambos dispositivos. Los parámetros que fueron modificados en el transceptor LoRa fueron el SF (*Spreading Factor*, factor de propagación), el BW (*Band Width*, ancho de banda) y el CR (*Coding Rate*, tasa de codificación). En la tabla 4.6 se muestran los valores utilizados de los parámetros antes citados.

TABLA 4.6. Tabla de parámetros de configuración por software del transceptor LoRa.

| Frecuencia | BW (MHz) | SF                     | CR  |
|------------|----------|------------------------|-----|
| 433        | 41.7     | 12 (4096 chips/symbol) | 4/5 |

De acuerdo a los parámetros de la tabla 4.6 se determina lo siguiente:

- Entre mayor sea el BW, mayor tiempo tomará la comunicación y esto se debe a que la frecuencia es inversamente proporcional al tiempo. Sin embargo, entre menor sea la frecuencia, mayor será el alcance de transmisión esperado.
- El valor de SF determina el rendimiento en la transmisión de datos, es decir, que entre mayor sea este valor, el dispositivo tendrá menor probabilidad de recibir datos incorrectos y tendrá mayor radio de cobertura.
- El CR asegura la fiabilidad de los datos, pero entre mayor sea este valor se sobrecarga el tiempo de transmisión.

## Capítulo 5

# Conclusiones

### 5.1. Conclusiones generales

En este trabajo se logró diseñar e implementar el prototipo comercial de un dispositivo electrónico, que tiene la capacidad de utilizar la salida de pulso óptico de medidores de consumo eléctrico domiciliario para obtener, procesar y transmitir información sobre la cantidad de kWh consumidos por los abonados de la compañía eléctrica COPELECT.

Para este fin, se diseñaron distintos módulos de firmware y hardware, que permiten transmitir diariamente la información obtenida a un gateway LoRa instalado en el edificio central de COPELECT. Asimismo, el dispositivo brinda a los abonados de COPELECT una interfaz gráfica web para conocer su consumo eléctrico de los últimos tres meses.

Durante el desarrollo del trabajo se presentó el riesgo de demora al conseguir los componentes electrónicos requeridos. Se aplicó el mecanismo de mitigación descrito en la planificación y se destinaron más recursos económicos de los previstos para poder cumplir con los plazos establecidos. El motivo de la demora, fue la pandemia global provocada por la enfermedad infecciosa COVID-19, que demoró el arribo de componentes a los proveedores locales y encareció la importación de componentes de proveedores internacionales. A pesar de que el motivo de la demora fue insalvable y de fuerza mayor, en futuros trabajos se estimarán tiempos en la obtención de componentes menos optimistas, para manejar un margen de tiempo que no complique otras tareas implicadas.

Otro punto importante fue el lanzamiento del decreto supremo que regula el uso de redes LPWAN en la frecuencia de 915 MHz [22], que servirá como punto de partida para que los proveedores locales de componentes electrónicos comercien licencias módulos LoRa de la frecuencia adecuada para el Bolivia.

Los requerimientos del trabajo fueron cubiertos de acuerdo con la planificación, con las siguientes modificaciones:

- Se eliminó la implementación de WPS (*Wi-Fi Protect Setup*, configuración de Wi-Fi segura) para suprimir cualquier tipo de interacción física del abonado con el dispositivo y evitar posibles manipulaciones incorrectas.
- La cantidad de meses visualizados en la interfaz web fue reducida de seis a tres, para exhibir más claramente los gráficos en dispositivos de pantallas pequeñas.

- La comunicación de los prototipos con un gateway LoRaWAN no se logró por que COPELECT no pudo adquirir uno en el mercado local. Entonces, para una primera aproximación con esta tecnología se realizó un intercambio de información estable con un gateway LoRa basado en Arduino. Esto permitió conocer la factibilidad técnica y los beneficios de LoRa.

Para desarrollar exitosamente el trabajo, se aplicaron los conocimientos obtenidos de varias de las materias cursadas en la Carrera de Especialización en Sistemas Embebidos. Estos fueron:

- Metodología de trabajo con repositorios locales y en la nube.
- Programación orienta a objetos en lenguaje C.
- Programación con sistemas operativos en tiempo real.
- Protocolos de comunicación I<sup>2</sup>C y SPI.
- Pruebas de software para sistemas embebidos.
- Diseño de esquemáticos y circuitos impresos basados en normas internacionales.

Por otra parte, para concluir exitosamente el trabajo también fue necesario adquirir algunos conocimientos sobre:

- Diseño de páginas web: los conocimientos adquiridos fueron útiles para crear la interfaz web embebida en el dispositivo, se obtuvieron conocimientos sobre HTML, CSS y JavaScript.
- jQuery: se aprendió a utilizar la biblioteca jQuery Mobile para suministrar funcionalidad y un aspecto sobrio a la interfaz web.
- Highcharts: utilizando esta biblioteca se pudo generar de una manera sencilla un gráfico de barras que ayuda al abonado a visualizar el consumo de kWh registrado por el dispositivo.

## 5.2. Próximos pasos

Como se especificó en esta memoria, el trabajo desarrollado es un prototipo comercial del dispositivo, que debe ser probado durante varios meses en un entorno real de trabajo para encontrar y solucionar posibles errores de firmware y hardware que no se presentaron en ninguna de las pruebas realizadas. Por lo tanto, posterior al periodo de pruebas del prototipo comercial el paso a seguir es la fabricación de una versión final del dispositivo siguiendo buenas prácticas de manufacturabilidad.

Debido a las limitaciones para obtener módulos LoRa de 915 MHz, se utilizaron los módulos disponibles en el mercado local que funcionaban a 433 MHz. Una tarea pendiente de este trabajo es implementar módulos con el circuito integrado SX1276 que funciona a 433 MHz, en lugar del SX1278 en los prototipos. Debido a que ambos circuitos integrados solo difieren en la frecuencia de transmisión y recepción, la biblioteca desarrollada en este trabajo podrá ser utilizada sin ningún tipo de inconveniente.

También, existen algunas características que deben ser incorporadas para mejorar la calidad del dispositivo. Estas son:

- Implementar un mecanismo de actualización de firmware remoto, OTA (*Over The Air*).
- Implementar algoritmos de *wear leveling* para incrementar el tiempo de vida de la memoria EEPROM.
- Adecuar el dispositivo para que pueda ser utilizado en medidores de agua y gas.



# Bibliografía

- [1] Wikipedia. *Vatio-hora - Wikipedia, la enciclopedia libre*. Visitado el 2020-07-02. 2020. URL: <https://es.wikipedia.org/wiki/Vatio-hora>.
- [2] Wikipedia. *Electricity meter - Wikipedia*. Visitado el 2020-07-011. 2020. URL: [https://en.wikipedia.org/wiki/Electricity\\_meter](https://en.wikipedia.org/wiki/Electricity_meter).
- [3] Wikipedia. *Current clamp - Wikipedia*. Visitado el 2020-07-011. 2020. URL: [https://en.wikipedia.org/wiki/Current\\_clamp](https://en.wikipedia.org/wiki/Current_clamp).
- [4] Manisha V Shinde ; Pradip W Kulkarni. «Camera click energy meter reading system». En: *IEEE* (2015).
- [5] François GUILLIER's blog RSS Feed. *Electricity meter*. Visitado el 2020-07-010. 2020. URL: [www.guillier.org/blog/2014/08/electricity-meter/](http://www.guillier.org/blog/2014/08/electricity-meter/).
- [6] OpenEnergyMonitor. *Learn | OpenEnergyMonitor*. Visitado el 2020-07-06. 2020. URL: <https://learn.openenergymonitor.org/electricity-monitoring/pulse-counting/introduction-to-pulse-counting>.
- [7] SyxthSense. *Wireless Pulse Counter for Metering (PA-FL)*. Visitado el 2020-07-14. 2020. URL: [www.syxthsense.com/wireless/pa-fl/wireless-pulse-counter-for-metering//pulse-counting/introduction-to-pulse-counting](http://www.syxthsense.com/wireless/pa-fl/wireless-pulse-counter-for-metering//pulse-counting/introduction-to-pulse-counting).
- [8] ElkoEP. *Wireless pulse converter - AirTM-100S bull ElkoEP*. Visitado el 2020-07-14. 2020. URL: <https://www.elkoep.com/airtm-100s/>.
- [9] Sigfox. *Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)*. Visitado el 2020-07-19. 2020. URL: <https://www.sigfox.com/en>.
- [10] Energy - European Commission. *Smart grids and meters - Energy European Commission*. Visitado el 2020-07-14. 2020. URL: <https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters>.
- [11] Juan Carlos Rico Noguera Antonio Serna Ruiz Francisco Antonio Ros García. *Guía Práctica de Sensores*. CREACIONES COPYRIGHT, 2010. ISBN: 9788492779499. URL: <https://www.casadellibro.com/libro-guia-practica-de-sensores/9788492779499/1799582>.
- [12] Elektor Magazine. *What Is a Microcontroller? | Elektor Magazine*. Visitado el 2020-07-27. 2020. URL: <https://www.elektormagazine.com/news/what-is-a-microcontroller>.
- [13] BISinfotech. *Top 10 Microcontrollers (MCU) Manufacturers for 2020*. Visitado el 2020-07-19. 2020. URL: <https://www.bisinfotech.com/top-10-microcontrollers-mcu-manufacturers-2020/>.
- [14] CISCO. *¿Qué es la tecnología wifi? Definición y tipos - Cisco*. Visitado el 2020-07-18. 2017. URL: [https://www.cisco.com/c/es\\_mx/products/wireless/what-is-wifi.html](https://www.cisco.com/c/es_mx/products/wireless/what-is-wifi.html).
- [15] Departamento de Informática y Sistemas - Universidad de Murcia. *El modelo OSI*. Visitado el 2020-07-28. 2015. URL:

- [http://dis.um.es/~lopezquesada/documentos/IES\\_1213/LMSGI/curso/xhtml/xhtml22/index.html](http://dis.um.es/~lopezquesada/documentos/IES_1213/LMSGI/curso/xhtml/xhtml22/index.html).
- [16] Semtech. *Semtech LoRa Technology Overview* | Semtech. Visitado el 2020-07-17. 2018. URL: <https://www.semtech.com/lora>.
- [17] LoRa Alliance®. *About LoRaWAN®* | LoRa Alliance®. Visitado el 2020-07-16. 2019. URL: <https://lora-alliance.org/about-lorawan>.
- [18] Explain that Stuff. *How do supercapacitors work?* - *Explain that Stuff*. Visitado el 2020-07-28. 2011. URL: <https://www.explainthatstuff.com/how-supercapacitors-work.html>.
- [19] Thomas L. Floyd. *Fundamentos de Sistemas Digitales - 6 Edicion*. Prentice Hall, 2000. ISBN: 8489660212. URL: <https://www.amazon.com/-/es/Thomas-L-Floyd/dp/8489660212>.
- [20] Wikipedia. *Wi-Fi - Wikipedia*. Visitado el 2020-07-16. 2020. URL: <https://en.wikipedia.org/wiki/Wi-Fi>.
- [21] Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes ATT. *Plan Nacional de Frecuencias*. Visitado el 2020-07-28. 2012. URL: <https://att.gob.bo/sites/default/files/archivospdf/Plan%20Nacional%20de%20Frecuencias%20-%2008%20-%2011%20-%202012.pdf>.
- [22] Bolivia emprende. *Decreto supremo 4272*. Visitado el 2020-07-31. 2020. URL: <https://boliviaemprende.com/wp-content/uploads/2020/06/D.S.-Programa-Nacional-de-Reactivaci%C3%B3n-23-06-20.pdf>.
- [23] FreeRTOS. *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions*. Visitado el 2020-07-28. 2019. URL: <https://www.freertos.org>.
- [24] Espressif Systems. *Build and Flash with Eclipse IDE - ESP8266 RTOS SDK Programming Guide documentation*. URL: <https://docs.espressif.com/projects/esp8266-rtos-sdk/en/latest/get-started/eclipse-setup.html>.
- [25] W3 Schools. *HTTP Methods GET vs POST*. Visitado el 2020-07-19. 2020. URL: [https://www.w3schools.com/tags/ref\\_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp).
- [26] Microchip. *AT24C32/64*. Visitado el 2020-07-21. 2003. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/doc0336.pdf>.
- [27] Maxim Integrated. *DS3231*. Visitado el 2020-07-21. 2015. URL: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>.
- [28] Github. *sandeepmistry/arduino-LoRa: An Arduino library for sending and receiving data using LoRa radios*. Visitado el 2020-07-27. 2020. URL: <https://github.com/sandeepmistry/arduino-LoRa>.
- [29] Semtech. *SX1278*. Visitado el 2020-07-22. 2020. URL: [https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rc1/QnUuV9TviODKUgt\\_rpBlPz.EZA\\_PNK7RpI8HA5..Sbo](https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rc1/QnUuV9TviODKUgt_rpBlPz.EZA_PNK7RpI8HA5..Sbo).