



Análisis y diseño de nodos IoT energéticamente autónomos

Néstor Enrique Sanquiz Núñez

Tesis presentada al programa de doble titulación entre la Universidad de La Coruña y el Instituto Politécnico de Bragança como requisito para la obtención de los títulos de Maestría en Eficiencia y Aprovechamiento Energético y Maestría en Energías Renovables y Eficiencia Energética, respectivamente.

Orientador (Instituto Politécnico de Bragança): Prof. Paulo Leitão

Co-orientador (Universidade Da Coruña): Prof. José Luis Calvo Rolle

Bragança

2020/2021

Agradecimientos

Durante mi estancia en Bragança me han acompañado muchas personas que han contribuido de diferentes formas a que este proyecto se desarrollase y pudiese culminarse con éxito. Por todo esto quiero mostrarles mi más sincero agradecimiento.

Agradezco el apoyo prestado por los Profesores Paulo Leitão y José Luis Calvo Rolle, así como también a Jonas Queiroz por su amistad y colaboración.

Finalmente quiero agradecer a mis amigos y familiares por el apoyo incondicional que siempre me brindan.

Abstract

With the advancement of IoT technologies, the use of smart devices to monitor the conditions of certain environments is becoming more and more common, since there are isolated scenarios with activities that require review and recording of data to which the national power grid does not have access. This project addresses the study of IoT technologies and devices for the design and implementation of energy efficient and autonomous nodes that use batteries and photovoltaic panels as a power source to monitor weather conditions, incorporating low-cost devices and components to ensure feasibility of implementation of a large number of nodes, as well as their adoption by the community. To this end, a prototype based on IoT technology was designed and developed and installed outside the Research and Innovation Laboratory of the Research Centre in Digitalization and Intelligent Robotics of the School of Technology and Management of the Polytechnic Institute of Bragança, which is able to measure weather conditions and transmit data through the MQTT protocol in a Wireless Local Area Network over WiFi communication technology to a MQTT Broker (Mosquito Software) that manages messages and publishes them to the flow editor (Node-RED Software) to be collected and sent to a database (InfluxDB Software) where they are stored and secured, and finally the visualization system (Grafana Software) displays the measurements to be analyzed over time.

The prototype is powered by a photovoltaic system composed of a solar panel with a power of 1 Wp and a storage capacity of 3400 mAh, has a consumption of 11 mAh, which is reduced as the battery voltage decreases prolonging the autonomy and preventing it from discharging. In the absence of generation, the panel guarantees an autonomy of over 278 hours and remains in operation during the charge and discharge cycles of the battery.

Keywords: IoT technologies, smart devices, monitoring, autonomy, batteries, photovoltaic panels.

Resumen

Con el avance de las tecnologías de IoT, el uso de dispositivos inteligentes para monitorear las condiciones de ciertos entornos es cada vez más común, ya que existen escenarios aislados donde se llevan a cabo actividades que requieren de la revisión y registro de datos y a los cuales la red eléctrica nacional no tiene acceso, por esta razón el presente proyecto aborda el estudio de tecnologías y dispositivos IoT para el diseño e implementación de nodos energéticamente eficientes y autónomos que utilizan baterías y paneles fotovoltaicos como fuente de alimentación para monitorear condiciones meteorológicas, incorporando dispositivos y componentes de bajo costo para garantizar la viabilidad de la implementación de un gran número de nodos, así como su adopción por parte de la comunidad, bajo este fin se diseñó y desarrolló un prototipo basado en la tecnología IoT instalado en el exterior del Laboratorio de Investigación e Innovación del Research Centre in Digitalization and Intelligent Robotics de la Escola Superior de Tecnologia e Gestão del Instituto Politécnico de Bragança, el cual es capaz de medir las condiciones meteorológicas y transmitir la data mediante el protocolo MQTT en una red de área local inalámbrica bajo tecnología de comunicación WiFi a un Broker MQTT (Software Mosquitto) que gestiona el mensaje y lo publica para que el editor de flujo (Software Node-RED) lo recolecte y envíe a una base de datos (Software InfluxDB) donde son almacenadas y resguardadas, y por último el sistema de visualización (software Grafana) muestra las medidas para ser analizadas en el tiempo.

El prototipo es alimentado por un sistema fotovoltaico compuesto de un panel solar con una potencia de 1 Wp y una capacidad de almacenamiento de 3400 mAh, presenta un consumo de 11 mAh, el cual se reduce a medida que la tensión en la batería disminuye

prolongando la autonomía y evitando que se descargue; en ausencia de generación garantiza una autonomía superior a las 278 horas y permanece en funcionamiento durante los ciclos de carga y descarga de la batería.

Descriptores: tecnologías IoT, dispositivos inteligentes, monitorización, autonomía, baterías, paneles fotovoltaicos.

Índice general

Abstract	VII
Resumen	IX
Acrónimos	XXI
1. Introducción	1
1.1. Contextualización	1
1.2. Antecedentes	2
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos específicos	4
1.4. Estructura del Documento	4
2. Estado do Arte	7
2.1. Internet de las Cosas (<i>Internet of Things</i>)	7
2.1.1. Ambiente inteligente (AmI)	8
2.1.2. Elementos que componen el IoT	10
2.1.3. <i>Wireless Sensor Networks</i> (WSN)	11
2.1.4. Monitorización de condiciones ambientales	11
2.2. <i>Smart devices</i> y tecnologías	12
2.2.1. Sensores	12
2.2.2. Controladores	14

2.3.	Comunicación IoT	15
2.4.	Sistemas fotovoltaicos	17
2.4.1.	Bloque generador	18
2.4.2.	Bloque de almacenamiento	21
2.4.3.	Bloque de controlador de carga	22
3.	Diseño y desarrollo de una solución para un nodo IoT energéticamente autónomo	23
3.1.	Planteamiento del problema y solución propuesta	23
3.2.	Descripción del diseño propuesto para el proyecto.	24
3.3.	Estrategias para el desarrollo del diseño propuesto	26
3.3.1.	Estrategia para el desarrollo e implementación del nodo IoT	26
3.3.2.	Estrategia para el desarrollo e implementación del sistema fotovoltaico con batería	32
3.3.3.	Estrategia para el monitoreo de los parámetros meteorológicos	35
3.4.	Criterios para la selección de los componentes y estructura del sistema planteado	37
3.5.	Cálculos y medidas requeridas para el diseño y funcionamiento del proyecto	38
3.5.1.	Capacidad de generación diaria del sistema fotovoltaico	38
3.5.2.	Cálculo autonomía de la batería	39
4.	Implementación	41
4.1.	Descripción del emplazamiento	41
4.2.	Implementación del nodo IoT energéticamente autónomo	42
4.2.1.	Circuito eléctrico del nodo IoT	42
4.2.2.	Circuito eléctrico programador	44
4.2.3.	Circuito eléctrico de medición	45
4.2.4.	Circuito eléctrico del sistema fotovoltaico	45
4.2.5.	Conectividad de la plataforma IoT con la red de área local Inalámbrica	46
4.2.6.	Lectura de los parámetros medidos	46

4.2.7.	Implementación del protocolo MQTT en el nodo IoT	47
4.2.8.	Implementación del funcionamiento normal del nodo IoT	48
4.2.9.	Prototipo del nodo IoT energéticamente autónomo	49
4.2.10.	Implementación del funcionamiento optimizado del nodo IoT ener- géticamente autónomo	49
4.3.	Implementación de una caja de montaje para la protección del prototipo .	51
4.4.	Implementación del sistema de monitoreo	52
4.4.1.	Implementación de la topología de red de área local inalámbrica . .	53
4.4.2.	Implementación de la base de datos	53
4.4.3.	Implementación del Broker MQTT	54
4.4.4.	Implementación Node-RED (Editor de Flujo)	54
4.4.5.	Implementación del sistema de visualización	55
5.	Pruebas al sistema	57
5.1.	Prototipo del nodo IoT energéticamente autónomo	57
5.1.1.	Pruebas al sistema fotovoltaico	58
5.1.2.	Pruebas de conectividad del nodo IoT con la red	61
5.1.3.	Pruebas de los sensores	62
5.1.4.	Pruebas del protocolo de comunicación MQTT y funcionamiento del Broker MQTT	62
5.1.5.	Pruebas de consumo de corriente del prototipo	64
5.1.6.	Pruebas de autonomía del nodo IoT	68
5.1.7.	Pruebas de carga y descarga	70
5.1.8.	Modelo físico de la caja de montaje para la protección del prototipo	71
5.2.	Prototipo del sistema de monitoreo	72
5.2.1.	Pruebas del editor de flujo	72
5.2.2.	Pruebas de la base de datos	74
5.2.3.	Pruebas al sistema de visualización	75
6.	Conclusiones y Recomendaciones	77

A. Propuesta del nodo IoT optimizado	83
B. Propuesta del nodo IoT normal	89

Índice de tablas

2.1. Ángulo de inclinación de los módulos solares en función de la latitud geográfica del emplazamiento [9].	21
3.1. Cálculo de la Horas Solar Pico de la ciudad de Bragança.	39
5.1. Promedio de corriente consumida por el prototipo de acuerdo al funcionamiento y frecuencia de muestreo.	68
5.2. Porcentaje de descarga de la batería en el tiempo.	69
5.3. Autonomía del prototipo de acuerdo a la frecuencia de muestreo con funcionamiento optimizado.	71

Índice de figuras

2.1. Diagrama de bloque de un sistema IoT simplificado [3].	10
2.2. Sistema Fotovoltaico con banco de energía.	17
2.3. Curva característica Tensión versus Corriente de los paneles solares [9]. . .	19
2.4. Curva característica Tensión versus Potencia de los paneles solares [9]. . . .	19
2.5. Comportamiento de la curva Tensión versus Corriente de los paneles solares en función del nivel de radiación solar [9].	20
2.6. Comportamiento de la curva Tensión versus Corriente de los paneles solares en función de la temperatura ambiente [9].	20
3.1. Arquitectura propuesta para proyecto.	25
3.2. Diseño del Nodo IoT propuesto.	27
3.3. Módulo WiFi ESP-12e [10].	27
3.4. Regulador de tensión S09 V8.0 [11].	29
3.5. Sensor digital de Temperatura y Humedad modelo SHT20.	29
3.6. Sensor digital Radiación Ultravioleta modelo VEML6075 [12].	30
3.7. Sensor Detección de LLuvia modelo YL-38 y resistencia YL-83 [13].	31
3.8. Sensor de corriente eléctrica DC [14].	31
3.9. Diseño del sistema fotovoltaico planteado.	33
3.10. Panel solar de silicio policristalino modelo CNC60x110-6.	34
3.11. Batería recargable tipo Lithium-ion modelo NCR18650B.	34
3.12. Cargador lineal Solar Charger V1.0 [16].	35
3.13. Proceso de monitorización de los parámetros meteorológicos.	36

4.1. Campus Escola Superior de Tecnología e Gestión [22].	42
4.2. Conexión del circuito eléctrico del Nodo IoT.	43
4.3. Topología del bus I ² C entre el módulo ESP-12e y los sensores SHT20, VEMML6075 e INA219.	44
4.4. Circuito eléctrico empleado para programar el nodo IoT.	45
4.5. Circuito eléctrico empleado para las pruebas de medición de tensión, co- rriente y potencia.	45
4.6. Esquema del circuito eléctrico del sistema fotovoltaico.	46
4.7. Diagrama de flujo de conexión entre el ESP-12E y el enrutador inalámbrico.	47
4.8. Diagrama de flujo de la lectura de los parámetros medidos.	48
4.9. Diagrama de publicación de las medidas censadas mediante el protocolo MQTT.	48
4.10. Diagrama de flujo del funcionamiento normal del nodo IoT.	49
4.11. Conexión del circuito eléctrico del Nodo IoT energeticamente autónomo.	50
4.12. Diseño de la placa de circuito impreso del Nodo IoT energeticamente au- tónomo.	50
4.13. Diagrama de flujo del funcionamiento optimizado del nodo IoT.	51
4.14. Diseño de la caja para la impresión 3D que alberga y protege los compo- nentes del nodo IoT.	52
4.15. Topología de Red de Área Local Inalámbrica.	53
4.16. Nombre de la base de datos que almacena los parámetros meteorológicos censados.	54
4.17. Espacio de trabajo “estación meteorológica” para la interconexión de los mensajes MQTT con la base de datos.	55
4.18. Interfaz gráfica para visualizar los parámetros meteorológicos.	56
5.1. Prototipo del nodo IoT energéticamente autónomo.	57
5.2. Esquema de medición empleados para medir corriente, voltaje y potencia del panel solar y la batería.	58

5.3. Gráfico de la tensión y corriente en el panel sola en un periodo de 12 h.	59
5.4. Gráfico de la tensión en el panel sola y la corriente en la batería en un periodo de 12 h.	59
5.5. Gráfico de la tensión en el panel sola y la tensión en la batería en un periodo de 12 h.	60
5.6. Gráfico de la potencia en el panel sola y en la batería en un periodo de 12 h.	60
5.7. Uso de la herramienta de diagnóstico “ping.exe” para verificar la conectividad del nodo IoT.	61
5.8. Resultado de la prueba de funcionamiento de los sensores SHT20, VEML6075, INA219 y FC-38.	63
5.9. Resultado de la implementación del protocolo MQTT en el nodo IoT.	63
5.10. Resultados de la subscripción al tópico “esp/estación”.	64
5.11. Esquema de medición para censar la corriente consumida por el nodo IoT.	65
5.12. Gráfico de la corriente extraída de la batería con funcionamiento normal del nodo IoT con frecuencia de muestreo de 60 s en un periodo de 3 min.	65
5.13. Gráfico de la corriente extraída de la batería con funcionamiento optimizado del nodo IoT con frecuencia de muestreo de 60 s en un periodo de 2 min.	66
5.14. Gráfico de la corriente extraída de la batería con funcionamiento optimizado del nodo IoT con frecuencia de muestreo de 240 s en un periodo de 10 min.	67
5.15. Gráfico de la corriente extraída de la batería con funcionamiento optimizado del nodo IoT con frecuencia de muestreo de 480 s en un periodo de 16 min.	67
5.16. Gráfico de voltaje en la batería en el tiempo con funcionamiento normal del nodo IoT.	69
5.17. Gráfico de voltaje en la batería en el tiempo con funcionamiento optimizado del nodo IoT en un periodo de 92 h.	70

5.18. Gráfico de voltaje en la batería en el tiempo con funcionamiento optimizado del nodo IoT en un periodo de 42 h.	71
5.19. Resultado de la caja de montaje que protege al prototipo.	72
5.20. Prototipo dispuesto en el interior de la caja de montaje.	73
5.21. Prueba de funcionamiento y la correcta implementación del editor de flujo.	73
5.22. Resultados obtenidos del debug.	74
5.23. Prueba de funcionamiento de la base de datos.	75
5.24. Interfaz gráfica desarrollada para el monitoreo de las condiciones meteorológicas.	75

Acrónimos

AmI Ambientes Inteligentes.

ARM *Advanced RISC Machine.*

CeDRI Research Centre in Digitalization and Intelligent Robotics.

CPU Unidad Central de Procesamiento.

EAGLE *Easily Applicable Graphical Layout Editor.*

ESTiG Escola Superior de Tecnologia e Gestão.

HSP Horas Pico Solar.

I²C *Inter integrated circuits.*

IoT Internet de las cosas.

IP Protocolo de Internet.

IPB Instituto Politécnico de Bragança.

ISP Proveedor de Servicio Internet.

KNN k vecinos más próximos.

L2I Laboratorio de Investigación e Innovación.

M2M machine to machine.

SCL serial reloj.

SDA serial de datos.

SPI *Serial Peripheral Interface.*

TCP Protocolo de Control de Transmisión.

UART Transmisor-Receptor Asíncrono Universal.

UDP Protocolo de Datagramas de Usuario.

WAN Red de Área Ampliada.

WSN Red de Sensores.

Capítulo 1

Introducción

1.1. Contextualización

Con el avance cada vez mayor de las tecnologías de IoT, el uso de dispositivos inteligentes para monitorear las condiciones de los entornos es cada vez más común. Además, el valor que se puede extraer de los datos recopilados por estos dispositivos para tomar decisiones ha contribuido a su adopción en diferentes escenarios, desde residencial, industrial o urbano.

Sin embargo, en algunos escenarios aislados, es decir, sin acceso a fuentes de energía, por ejemplo, el monitoreo de entornos agrícolas e infraestructuras de carreteras, estos dispositivos necesitan usar baterías y operar de manera energéticamente eficiente (es decir, equilibrar el procesamiento local y la transmisión de datos para optimizar el uso de la batería). En estos escenarios el uso de fuentes renovables como la solar puede ayudar a incrementar la autonomía de estos dispositivos.

En el contexto de IoT, aún es necesario considerar el costo de dispositivos y componentes, que deben ser bajos para garantizar la viabilidad de la implantación de un gran número de nodos, así como la adopción por parte de la comunidad.

1.2. Antecedentes

Un primer trabajo corresponde a Csernáth, Csenteri y Tuross (2018), quienes realizaron la propuesta de “*Power Management In IoT Weather Station*” en el cual presentan una solución de administración de energía para un prototipo de estación meteorológica operada por batería, basada en Internet de las cosas, donde se desarrolló un completo software de simulación para validar el sistema con datos meteorológicos reales. Mediante este software y los datos obtenidos de las mediciones de consumo de energía realizadas al prototipo, se calculó e implementó un tiempo de ejecución optimizado del sistema. Un segundo trabajo realizado por Martínez (2017), denominado “monitorización de un ambiente inteligente de bajo coste” trata acerca de la realización de un estudio y despliegue de una red de sensores inalámbricos de bajo coste que monitoricen un ambiente inteligente a través de un dispositivo que reciba la información de los sensores de manera inalámbrica y la almacene en una base de datos mientras se reduce la carga computacional de reconocimiento de actividades basados en k vecinos más próximos (KNN) en una configuración de sensores a través de la generación de prototipos implementando un algoritmo de reconocimiento de actividades KNN con los prototipos generados para el apartamento de inteligencia ambiental de la Universidad de Jaén desde una aplicación móvil completa que permita monitorizar el ambiente inteligente y las actividades que se realizan en él.

Un tercer trabajo de Ceja, Rentería, Ruelas y Ochoa (2017), que tiene por título “Módulo ESP8266 y sus aplicaciones en el internet de las cosas” con el uso de internet cada vez mayor, se hace necesaria una interfase entre los dispositivos y el mundo del internet. Por lo que esta investigación se basa en un módulo Wifi ESP8266 de bajo costo, ideal para aplicaciones de Internet de las cosas (IoT) y saber cómo elegir el más adecuado para nuestros futuros proyectos, enfocados en el internet de las cosas, tomando en cuenta factores importantes para la elección de estos dispositivos, como lo son la normativa de comunicación en cada país, la cantidad de memoria y sobre todo el costo beneficio, dependiendo del hardware con que se cuenta en el proyecto.

Un cuarto trabajo de Guerrero (2017), titulada “SGreenH-IoT: Plataforma IoT para

Agricultura de Precisión” en este trabajo se presenta SGreenH-IoT una plataforma IoT de bajo costo y consumo energético para la monitorización de campos de cultivo e invernaderos. La plataforma la conforman una arquitectura de 4 capas, un protocolo de comunicación, el diseño de un nodo de bajo costo y consumo de energía y una aplicación web para la visualización de los datos. Los resultados obtenidos mediante experimentación muestran el funcionamiento efectivo de la plataforma para la recolección de datos y la integración con la toma de decisiones, con porcentaje de pérdidas de datos del cero por ciento. SGreenH-IoT es una solución completa que incluye el diseño de nodos de bajo costo y consumo energético, protocolos de comunicación y una aplicación de visualización de datos y control del sistema. La plataforma realiza el monitoreo y transmite toda la información a la nube para su almacenamiento, análisis y activación de alertas que mejoren la producción agrícola. Los resultados obtenidos en la experimentación mostraron el funcionamiento eficiente del sistema, el porcentaje de pérdida de datos en la transmisión fue nula. Los beneficios para los agricultores obtenidos por el sistema son el acceso en tiempo real de las condiciones del área de cultivo, la reducción en el uso de recursos a través de la aplicación exacta en los tiempos adecuados, contribuyendo así a una producción más sustentable.

Un quinto trabajo de Moras y Rosas (2019) “Diseño, desarrollo e implementación de una Red de Sensores (WSN) para el control, monitoreo y toma de decisiones aplicado en la agricultura de precisión basado en Internet de las cosas. – caso de estudio cultivo de frijol”, en este trabajo de investigación se propuso diseñar, desarrollar e implementar una red de sensores inalámbricos basado en el internet de las cosas aplicado en el cultivo de Frijol con el objetivo de establecer el monitoreo, control y toma de decisiones en la agricultura de precisión, mediante la creación de un dispositivo electrónico capaz de establecer una red de sensores inalámbricos, para ello se utilizó el MCU ESP8266 en su módulo ESP-12E aplicando el IoT con la capacidad de conectarse a internet. Los sensores evaluaron los 5 factores ambientales que proporcionaron información que fue enviado por internet y fue almacenado en una base de datos en la nube. Las informaciones de las variables ambientales pudieron ser visualizadas en tiempo real mediante una interfaz

gráfica sencilla. La interfaz fue fácilmente entendida por el usuario.

1.3. Objetivos

1.3.1. Objetivo General

Estudiar tecnologías y dispositivos IoT para el diseño e implementación de nodos energéticamente eficientes y autónomos que utilizan baterías y paneles fotovoltaicos como fuente de energía para monitorear condiciones meteorológicas, incorporando dispositivos y componentes de bajo costo para garantizar la viabilidad de la implementación de un gran número de nodos.

1.3.2. Objetivos específicos

- Implementación del nodo IoT en el exterior del Laboratorio de Investigación e Innovación (L2I), para el monitoreo de variables meteorológicas.
- Desarrollo de un sistema fotovoltaico con batería.
- Elaboración de un sistema de recolección, visualización y monitoreo de los parámetros medidos por el nodo IoT.

1.4. Estructura del Documento

El documento se encuentra dividido en 6 capítulos:

- En el **capítulo 1**, introducción al tema de estudio y presentación de los objetivos del Trabajo de Fin de Máster.
- El **capítulo 2**, presenta el Estado del Arte, definiéndose conceptos asociados a los nodos de Internet de las Cosas, Ambientes Inteligentes, Redes Inalámbricas de Sensores y sistemas solares fotovoltaicos.

- El **capítulo 3**, presenta la metodología usada para la elaboración del Trabajo de Fin de Máster, así como planteamiento del problema y la solución propuesta.
- El **capítulo 4**, describe la implementación realizada de acuerdo a la solución propuesta.
- El **capítulo 5**, discute los resultados obtenidos de acuerdo a la implementación realizada en el capítulo 4.
- El **capítulo 6**, presenta las conclusiones del Trabajo de Fin de Máster.

Capítulo 2

Estado do Arte

En este capítulo son presentados los fundamentos teóricos y tecnológicos referentes a los nodos de Internet de las Cosas (IoT) para monitorear variables físicas de interés, de forma que el nodo sea energéticamente autónomo, mediante un sistema solar con batería.

2.1. Internet de las Cosas (*Internet of Things*)

Es un concepto que hace referencia a los objetos y dispositivos que son capaces de interactuar entre sí y con el ser humano, analizando, recibiendo y transmitiendo información o datos en un medio de comunicación hacia o desde internet. Estos objetos y dispositivos interactúan mediante *smart devices* (dispositivos inteligentes) que deben estar compuestos de los siguientes elementos: controlador (componente computacional que permite procesar los datos y la información), sensores (permiten medir las variables físicas de nuestro entorno y transformarlas en datos que pueden ser procesados de forma digital) y actuadores (elementos electrónicos que modifican o generan cambios sobre la física del entorno). Este conjunto de elementos componen un nodo IoT, en la actualidad los costes de estos dispositivos electrónicos han disminuido considerablemente al mismo tiempo que se han presentado avances cada vez mayores de estas tecnologías y sumado a que la presencia de internet en el mundo es considerable, nos abre la posibilidad de conectar cualquier objeto y dispositivo a internet de forma que actúe como una fuente

de información de las condiciones de nuestro entorno, permitiendo que se puedan tomar decisiones que contribuyan al desarrollo de la humanidad.

Cómo lo describen los autores Raj, Singh, Vishvakarma y Singh “IoT es una red interconectada de dispositivos de procesamiento, computadoras, artículos inteligentes, teléfonos móviles y otros dispositivos electrónicos que están equipados con inteligencia y protocolos de comunicación” [1]. IoT potencia la capacidad de recopilar e intercambiar información entre la administración del sistema de todos los objetos físicos conectados. El empoderamiento se realiza a través de Internet con la ayuda de hardware integrado, programación y redes de sensores en un entorno denominado *Intelligent Ambient* (Ambiente Inteligente). La Iniciativa de Estándares Globales sobre Internet de las Cosas (GSI-IoT) ha identificado a IoT como un marco mundial para el intercambio y procesamiento de información. En este sentido, el dominio físico se conoce como dispositivos físicos; y el dominio de la información es denominado cosas virtuales que están interrelacionadas e interconectadas a la luz de las TIC existentes y en desarrollo.

IoT permite descubrir, acceder y gestionar los objetos en lugares remotos con el uso de computadoras personales y teléfonos inteligentes de forma competente y económica sin que sea necesario interactuar con el usuario [1].

Los objetos y dispositivos inteligentes conectados a internet cuentan con identidad única para fines de descubrimiento y acceso, es decir, cada objeto inteligente se percibe a través de su marco de procesamiento instalado [1].

2.1.1. Ambiente inteligente (AmI)

Un ambiente inteligente es un elemento de la informática generalizada que aporta inteligencia a los entornos de vida y de negocios para hacerlos más sensibles, adaptables, autónomos y personalizados a las necesidades humanas. Se refiere a interfaces inteligentes que reconocen la presencia humana junto a sus preferencias de forma que ajustan los entornos inteligentes para adaptarse a sus necesidades y requisitos inmediatos. El factor clave es la presencia de inteligencia y capacidades de toma de decisiones en los entornos

IoT. Las aplicaciones y escenarios de AmI incluyen hogares inteligentes, vehículos autónomos, sistemas de salud, carreteras inteligentes, sector industrial, gestión de instalaciones inteligentes, sector educativo, servicios de emergencia y muchos más [2].

El proceso de un ambiente inteligente comprende tres etapas principales: Percepción, Razonamiento y Actuación. La percepción se basa en datos del mundo real recopilados por sensores inteligentes interconectados presentes en el entorno del IoT. Durante el proceso de Razonamiento, los datos son procesados por algoritmos inteligentes integrados dentro de los dispositivos en red que perciben el entorno y aplican enfoques razonados basados en AmI. Esto, a su vez, proporciona capacidad de respuesta y adaptabilidad, para la etapa de actuación del proceso, para reaccionar a las condiciones ambientales y modificar el entorno según los requisitos humanos que ya están construidos dentro de los sistemas ambientales [2].

Ambientes Inteligentes (AmI) es un concepto tecnológico en evolución que agrega inteligencia al entorno informático distribuido existente (por ejemplo, IoT) para hacerlo más sensible a las condiciones ambientales, adaptable en caso de condiciones ambientales cambiantes y receptivo a la presencia humana y sus requisitos. Combina tecnologías avanzadas como inteligencia artificial, aprendizaje automático, redes neuronales, computación generalizada y redes basadas en sensores. El AmI permite la toma automática de decisiones razonadas de acuerdo con la naturaleza del entorno circundante. La información necesaria para la toma de decisiones se recopila mediante el uso de una red basada en sensores que consta de actuadores, termostatos, dispositivos de medición, etc. Así, los algoritmos de AmI dependen en gran medida de los datos obtenidos de dichas redes basadas en sensores. El mecanismo AmI tiene las siguientes características [1].

- **Detección:** esto es necesario para que el sistema pueda detectar varias características y cuantificar variables como temperatura, presión, velocidad, posición, luz y radiación [1].
- **Razonamiento:** esto es necesario para que el sistema sea útil, adaptable y sensible [1].

- **Toma de decisiones:** es necesaria para realizar cambios en el medio ambiente. Depende de los resultados del razonamiento obtenidos después de reconocer el comportamiento o percibir la actividad en el entorno [1].
- **Actividad:** Proporciona la capacidad de actuar de acuerdo con la decisión obtenida de acuerdo al punto anterior [1].

2.1.2. Elementos que componen el IoT

IoT a menudo se conoce como la digitalización del mundo físico. Es una confluencia de diferentes tecnologías a costos suficientemente bajos. Si bien existen diferentes definiciones de IoT, el autor Siu, realiza la siguiente descripción que nos permite presentar un sistema IoT simplificado como: un dispositivo integrado con sensores, procesamiento y / o actuador, conectado a Internet, que comparte su información con otros dispositivos, con el potencial de actuar sobre esta información basándose en algunas reglas e inteligencia.

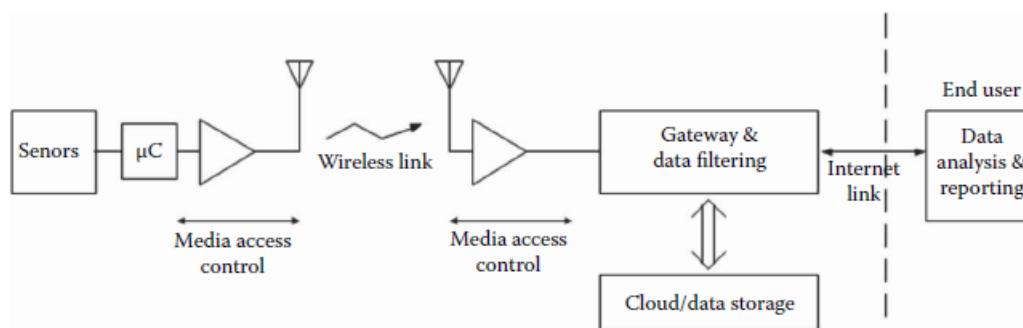


Figura 2.1: Diagrama de bloque de un sistema IoT simplificado [3].

En el diagrama de bloques simplificado que se muestra en la figura 2.1, los sensores de un nodo final recopilan datos a intervalos específicos. El microcontrolador enmarca los datos en paquetes, que también contienen partes de la pila de protocolos para realizar el control de acceso a los medios. Los paquetes se modulan y transmiten a través del enlace inalámbrico, que es recibido por una puerta de enlace conectada a Internet. La puerta de enlace puede tener un motor de reglas para reducir la cantidad de datos antes de que se almacenen, luego, los datos del sensor pueden transferirse a un usuario final

para su posterior análisis y generación de informes. Si bien la puerta de enlace puede estar alimentada por la red, los nodos de los sensores se alimentan con baterías y / o captación de energía. Dado que no es factible cambiar la batería con regularidad en una gran cantidad de nodos sensores, existe una gran motivación para reducir el consumo de energía de los nodos finales tanto como sea posible [3].

2.1.3. *Wireless Sensor Networks (WSN)*

La información necesaria para la implementación de un entorno inteligente se recopila mediante redes inalámbricas basadas en WSN. Estos forman parte de la informática generalizada y se utilizan de forma eficaz para crear entornos digitales inteligentes. Para la recopilación de información relevante, el WSN realiza funcionalidades que incluyen detectar las condiciones ambientales, acceder a los datos medibles, realizar análisis y tomar acciones de adaptación. Una red inalámbrica basada en sensores funciona de manera más eficaz si tiene la topología de red adecuada y los protocolos de comunicación relevantes. Debe ser programable, escalable y seguro. Además de esto, es necesario emplear varios métodos para reducir el consumo de energía. También es necesaria la elección del controlador correcto que tenga la capacidad de minimizar la utilización de energía [1].

2.1.4. *Monitorización de condiciones ambientales*

AmI en microcontroladores para la monitorización remota de las condiciones ambientales dentro del entorno de IoT es actualmente una necesidad deseada para diversas áreas de aplicación. El monitoreo de las condiciones ambientales como temperatura, radiación solar, humedad, velocidad del viento, calidad del aire entre otros son imperativos para muchas aplicaciones, en los diferentes entornos de la vida cotidiana. Entre ellos el control de la temperatura se ha convertido probablemente en la necesidad más importante de nuestra vida diaria, esto debido a que muchos de los equipos o dispositivos funcionan de acuerdo a temperaturas específicas y si hay cambios significativos en la temperatura del ambiente, estos dispositivos pueden dejar de funcionar correctamente. A menudo,

debemos vigilar la temperatura en muchos entornos de nuestras vida como salas de almacenamiento de alimentos, salas de servidores, salas de hospitales, invernaderos entre otros, tener el control de la temperatura de estos espacio son de suma importancia, que frecuentemente son medidos por sensores de temperatura normal, donde es necesario la intervención de un ser humano que constante mente verifique y ajuste la lectura del sensor. Mientras que el monitoreo automático de las condiciones ambientales ofrece una diversa variedad de plataformas de microcontroladores para AmI de bajo consumo de energía [1].

2.2. *Smart devices* y tecnologías

Un *Smart devices* o dispositivo inteligente es un sistema electrónico computarizado sensible al entorno, capaz de captar la información del ambiente de forma autónoma, así como de comunicar o intercambiar esa información con otros dispositivos [4]. Estos elementos juegan un papel importante en el internet de las cosas, puesto que tienen la tarea de:

- Medir las variables físicas de nuestro entorno y transformarlas en datos que pueden ser procesados de forma digital.
- Procesar datos e información.
- Generan cambios sobre la física del entorno.

En general cualquier objeto puede convertirse en un dispositivo inteligente, si agregamos un sensor, capacidad computacional y conectividad de red. Y para ello deben estar compuesto de los siguientes elementos: sensor, controlador y/o actuador.

2.2.1. Sensores

Son dispositivos que permiten medir variables físicas. Cómo indican los autores Corona, Abarca y Mares “Los sensores proporcionan una salida útil para ser usada como variable de entrada a un sistema de procesamiento de la información. El sensor no solo cambia el

dominio de la variable física medida, sino que además la salida del sensor será un dato útil para un sistema de medición” [5]. En el ámbito de la electrónica, estos dispositivos entregan una señal eléctrica analógica o digital a su salida.

Clasificación de los sensores de acuerdo al tipo de variable medida:

Los sensores se suelen clasificar según la variable a medir, cabe resaltar que un mismo sensor puede medir diferentes variables físicas [5].

- Posición, velocidad y aceleración.
- Nivel y proximidad.
- Humedad y temperatura.
- Fuerza y deformación.
- Flujo y presión.
- Color, Luz y visión.
- Gas y pH.
- Biométricos.
- Corriente.
- Etc.

Los sensores poseen características que difieren unos de otros, como:

- **Sensibilidad:** Es el valor mínimo de entrada requerido para generar una salida detectable [5].
- **Rango:** Es el espectro de medida de la variable física para el cual está calibrado el sensor [5].
- **Precisión:** Es el nivel de repetitividad de una medida, dando un mismo resultado en diferentes mediciones [5].

- **Exactitud:** Es la diferencia entre el valor de salida del sensor y el valor real de la variable medida [5].
- **Resolución:** es el valor de medida física más pequeña que puede registrar el sensor [5].

2.2.2. Controladores

Son los componentes encargados de procesar la información proporcionada por los sensores, tomando decisiones y comunicando la información a otros elementos.

- **Controlador PIC:** Por sus siglas en inglés *Programmable Integrated Circuited*, es un circuito integrado programable que ejecuta ordenes que se encuentran almacenadas en una memoria en su interior. Está compuesta por: Unidad Central de Procesamiento (CPU), memoria y periféricos, cada uno de estos ejecuta su función específica. Cabe destacar que inicialmente la programación de estos controladores se realizaba mediante kits de desarrollos costosos que hacían engorroso su uso [6].
- **Controlador Arduino:** es una plataforma de desarrollo para sistemas embebidos de bajo costo, hardware abierto y de software libre (Hardware abierto significa que los dispositivos electrónicos, sus diagramas y especificaciones son de acceso libre y el software libre es que los códigos usados en la programación informática son accesibles por el público en general, es posible usarlo y modificarlo). Estos sistemas no se encuentran vinculados a un fabricante específico. Está arquitectura posee un entorno amigable, compuesto de por módulos que permiten adicionar funcionalidad al microcontrolador agregando módulos compuestos de sensores, actuadores y comunicación. Según el autor Oliveira [6], Arduino tuvo la primicia de convertir estos sistemas embebidos en sistemas accesibles a todos (educacionalmente y económicamente).
- **Controlador Raspberry Pi:** El Desarrollo Raspberry Pi al igual que Arduino es un Sistema embebido hardware abierto y de software libre. Utiliza un procesador

de la familia *Advanced RISC Machine* (ARM), con procesamiento muy elevados, bajo sistema operativo Linux, estos sistemas agregan funcionalidad muy similar a la de un computador. Por sus altas prestaciones estos sistemas embebidos es posible conseguirlo en redes de internet de las cosas como elemento central (servidor) y otros elementos de menor costo que este, realicen funciones más concretas en la red, de forma que se aprovechen todos los recursos que del Raspberry Pi y no aumentar mucho el costo de la red [6].

- **Controlador ESP8266:** es un microcontrolador con muchas funcionalidades, prestaciones y muy bajo costo, muy usado en aplicaciones de nodos IoT, con una alta durabilidad, diseños compactos, una arquitectura para el ahorro de energía que agrega módulos similares a los de Arduino realizando de este modo las mismas actividades que realiza un Arduino [6].

2.3. Comunicación IoT

El modelo de comunicación empleado en internet es basado en dos elementos principales: cliente y servidor. El servidor se encarga de gestionar la comunicación entre los clientes, mientras que el cliente acciona al servidor siempre que requiera realizar una comunicación con otro cliente o servidor [6]. En IoT los nodos pueden desempeñar ambas funciones de tal forma que se puedan comunicar entre sí.

Las redes de comunicación inalámbricas se pueden clasificar de acuerdo a la distancia que estas cubren en:

- Redes de área metropolitana inalámbricas **WMAN** (*Wireless Metropolitan Area Network*) diseñadas para cubrir distancias de varios kilómetros [6].
- **WLAN** (*Wireless Local Area Network*) redes de área local inalámbrica, estas redes tienen una cobertura que va entre los metros y centenas de metros [6].
- **WPAN** (*Wireless Personal Area Network*) Red de área personal Inalámbrica, con un alcance en el orden de las decenas de metros [6].

Protocolos de comunicación de red Existen una variedad de protocolos de comunicación desarrollado para para internet de las cosas, como son la familia de protocolos presentes en el estándar IEEE 802.15, desarrollados para redes inalámbrica de área personal conocida como WPAN, entre los protocolos que hacen parte de este estándar tenemos: Bluetooth, ZigBee, 6LOWPAN, WirelessHart e ISA100.11A. Además de estos protocolos, como por ejemplo el protocolo WiFi que ha sido desarrollado para redes WLAN y adoptado en IoT, a pesar de que esté no fueran concebidos para su uso [6].

El perfil de comunicación planteado para IoT requiere de baja demanda de ancho de banda, bajo consumo de energía, permitir un gran número de abonados a la red y de alcance limitado para distancias cortas [6].

La comunicación para internet de las cosas no se encuentra bien definido, a pesar de eso de acuerdo con el autor Aliveira, WiFi y Bluetooth han tenido gran aceptación para aplicaciones IoT, consolidándose en el mercado y ofreciendo productos de bajo costo, permitiendo la viabilidad para la utilización en IoT.

Protocolos de comunicación de datos Los dispositivos IoT requieren de rutinas para la comunicación entre dispositivos bajo el protocolo machine to machine (M2M) que permite la comunicación entre dos o más dispositivos remotos [7]. M2M establece las condiciones que permiten a los dispositivos intercambiar información con una aplicación a través de una red de comunicación de forma que el dispositivo o aplicación pueda funcionar como base para el intercambio de información, para aplicaciones IoT resaltan los siguientes protocolos:

- ***Message Queue Telemetry Transport (MQTT)***, es un protocolo de comunicación de datos basado en mensajería de publicación-suscripción que permite la comunicación entre múltiples dispositivos gestionados por un componente central llamado broker. Este protocolo usa una topología de red en estrella, donde todos los nodos se conectan directamente al bróker. Los mensajes son enviados empleando el Protocolo de Control de Transmisión (TCP) [7].
- ***Constrained Application Protocol (CoAP)***, es un protocolo cliente-servidor

donde el dispositivo cliente puede dirigir a otro dispositivo mediante un mensaje CoAP. El dispositivo servidor interpreta el mensaje y decide qué acción realizar de acuerdo a su programación. En contraste al protocolo MQTT los mensajes CoAP son enviados mediante el Protocolo de Datagramas de Usuario (UDP) [7].

Los protocolos MQTT y CoAP son solo dos ejemplos de protocolos de comunicación de datos.

2.4. Sistemas fotovoltaicos

Los dispositivos inteligentes que conforman el internet de las cosas requieren de consumo energético, normalmente suministrado por la red eléctrica nacional de corriente alterna a una tensión de 230 V, que se convierte a corriente continua con valores de tensión muy bajos entre 3.3 V y 12 V necesarios para el funcionamiento de los nodos IoT. En lugares donde no es posible conectarse a la red eléctrica nacional, es necesario considerar fuentes alternativas de energía tales como energía solar.

Los sistemas fotovoltaicos tradicionales están formados por un bloque generador, un bloque de acondicionamiento de la potencia y por un bloque de almacenamiento [8] como se muestra en la figura 2.2. A continuación, son presentados los puntos más relevantes de los bloques del sistema fotovoltaico:

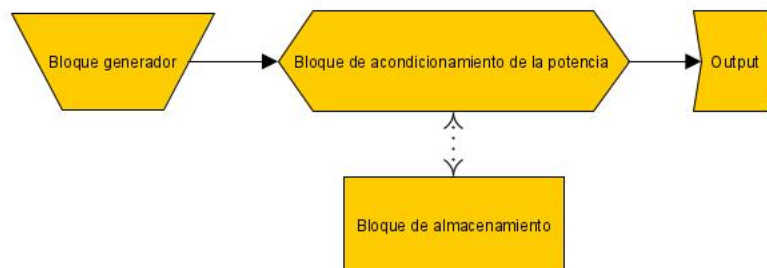


Figura 2.2: Sistema Fotovoltaico con banco de energía.

2.4.1. Bloque generador

El bloque generador está conformado por los paneles solares, estos elementos son los encargados de captar la energía proveniente de la radiación solar y transfórmala en energía eléctrica. En la actualidad existen una variedad de tecnologías para la fabricación de los módulos fotovoltaicos, los más comunes encontradas para el desarrollo de aplicaciones IoT son de silicio monocristalino y de silicio policristalinos, de acuerdo a una comparativa realizada por el autor Gradella, la eficiencia entre ambos módulos comercialmente difiere en un 1 a 3 por ciento, siendo más eficientes los módulos construidos con células fotovoltaicas monocristalinas [9]. Los paneles solares no poseen un comportamiento similar a las fuentes de poder convencionales, los módulos fotovoltaicos no presentan una tensión constante de salida en sus terminales. Los módulos fotovoltaicos poseen unas curvas características similares a las mostradas en las figuras 2.3 y 2.4, donde podemos observar que al no haber tensión en las terminales del panel solar se logra el punto de corriente de corto circuito, que es la máxima corriente que puede ofrecer el panel solar, por otro lado cuando la carga conectada al panel fotovoltaico es nula se obtiene la máxima tensión del panel fotovoltaico, denominadas en las curvas como punto de tensión de circuito abierto, a la hora de realizar instalaciones fotovoltaicas se busca es generar electricidad en el punto de máxima potencia, que es donde se obtiene el mayor rendimiento de los paneles fotovoltaicos, como se puede observar en ambas curva.

Es importante destacar que los paneles solares se ven afectados tanto por la radiación solar como por la temperatura. En la figura 2.5 podemos ver como es el comportamiento de un módulo fotovoltaico de acuerdo a los niveles de radiación del emplazamiento (a mayor radiación solar mayor es la corriente generada). Mientras que en la figura 2.6 se observa el comportamiento del módulo con respecto a la temperatura del lugar (a menor temperatura la tensión de los paneles será mayor).

Para maximizar la producción media diaria de energía y optimizar la producción de energía durante todo el año en la instalación módulos solares, el autor Gradella, recomienda primero ubicar los paneles en dirección al sur para emplazamientos localizados en el



Figura 2.3: Curva característica Tensão versus Corriente de los paneles solares [9].

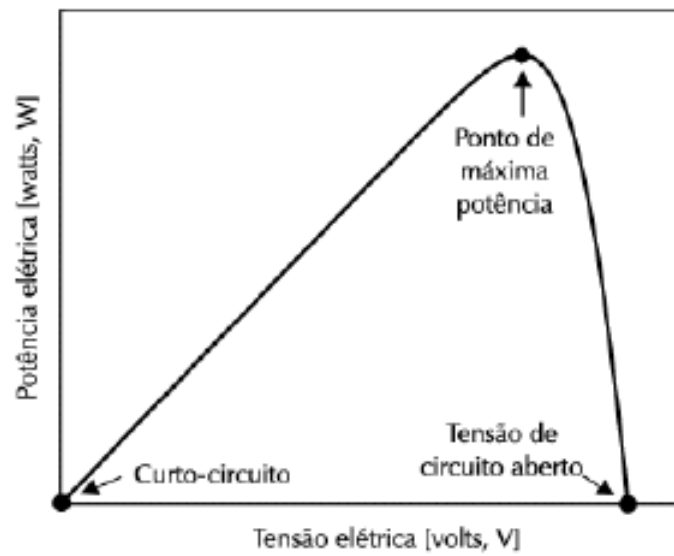


Figura 2.4: Curva característica Tensão versus Potencia de los paneles solares [9].

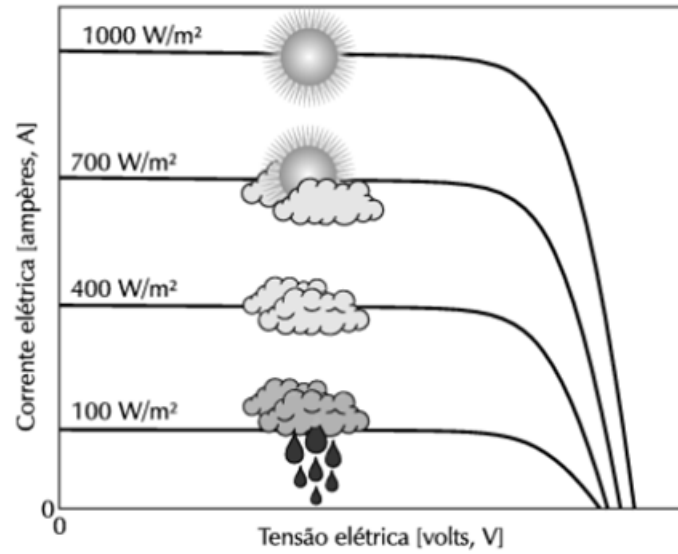


Figura 2.5: Comportamiento de la curva Tensión versus Corriente de los paneles solares en función del nivel de radiación solar [9].

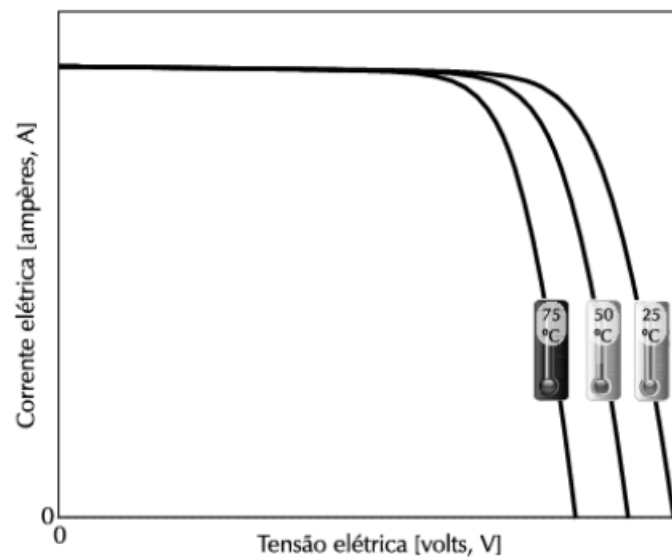


Figura 2.6: Comportamiento de la curva Tensión versus Corriente de los paneles solares en función de la temperatura ambiente [9].

hemisferio norte del planeta siempre que sea posible, luego ajustar el ángulo de inclinación del módulo en relación al suelo, para ello, el ángulo de inclinación debe elegirse de acuerdo con tabla en función del ángulo de latitud geográfica del lugar donde se instala el sistema fotovoltaico [9].

Tabla 2.1: Ángulo de inclinación de los módulos solares en función de la latitud geográfica del emplazamiento [9].

Latitud geográfica del emplazamiento	Ángulo de inclinación (α) recomendado
0° a 10°	$\alpha = 10^\circ$
11° a 20°	$\alpha = \text{latitud}$
21° a 30°	$\alpha = \text{latitud} + 5^\circ$
31° a 40°	$\alpha = \text{latitud} + 10^\circ$
41° o más	$\alpha = \text{latitud} + 15^\circ$

2.4.2. Bloque de almacenamiento

El bloque de almacenamiento está formado por un acumulador eléctrico. Las características intermitente y aleatoria de la radiación solar a lo largo del día, hace necesario el uso de estos elementos, también llamados baterías, la presencia de una batería en estos sistemas es necesaria para proporcionar suministro de energía constante, así como para almacenar el excedente de energía generada. En general el uso de baterías como respaldo a dispositivos de Internet de las cosas, viene en auge, puesto que son fundamentales para garantizar la autonomía de los dispositivos IoT que no pueden ser conectados a la red eléctrica nacional [6].

En el mercado existe una variedad de baterías, que varían según la aplicación y características. Para proyectos IoT podemos encontrar baterías recargables comerciales como NiMH y Li-ion, además de baterías a base Íons de litio como LiPo y LiFePo4. Las principales características que diferencian a cada una de estas baterías son: tensión, corriente y capacidad de almacenamiento, que deben ajustarse a las características técnicas de la aplicación IoT [6].

2.4.3. Bloque de controlador de carga

Finalmente, el bloque de controlador de carga puede estar conformado por convertidores de corriente continua, seguidor de punto de máxima potencia, cargadores lineales y otros dispositivos de protección, supervisión y control. Los sistemas fotovoltaicos con acumuladores de corriente deben poseer un controlador o regulador de carga, puesto que este dispositivo es el encargado de realizar la correcta conexión entre el módulo fotovoltaico y la batería, evitando que ella sufra sobrecargas o descargas excesivas [9]. En el mercado existen gran variedad de controladores de carga que realizan la carga de la batería de acuerdo a las características de las mismas, también existen otros controladores buscadores del punto de máxima potencia de los paneles fotovoltaicos (permite aumentar la eficiencia del sistema fotovoltaico).

Capítulo 3

Diseño y desarrollo de una solución para un nodo IoT energéticamente autónomo

En el presente capítulo es abordado el problema y la solución planteada, así como la metodología empleada para el desarrollo e implementación de un nodo IoT energéticamente eficiente y autónomo por medio de un sistema solar con batería.

3.1. Planteamiento del problema y solución propuesta

Con el avance de las tecnologías de IoT, el uso de dispositivos inteligentes para monitorear las condiciones de los entornos es cada vez más común. Además, el valor que se puede extraer de los datos recopilados por estos dispositivos para tomar decisiones ha contribuido a su adopción en diferentes escenarios, residencial, industrial o urbano. Sin embargo, en algunos escenarios sin acceso a suministros de energía eléctrica o con restricciones que dificultan o imposibilitan la instalación de un sistema alimentación eléctrica, estos dispositivos requieren de baterías para operar de manera eficiente (esto es, equilibrar

el procesamiento local y la transmisión de datos para optimizar el uso de la batería).

La red eléctrica nacional es limitada en cuanto a su alcance y distribución, lo que impide en algunos casos que usuarios potenciales tengan acceso a ella, esto constituye un problema importante debido a que en algunas zonas se llevan a cabo actividades agrícolas, ganaderas, pesqueras, gestión de tráfico, logística para flotas de vehículos, entre otros, donde resulta necesaria la monitorización y recolección de datos para fuentes estadísticas, por lo cual se plantea como solución el diseño y desarrollo de un nodo IoT energéticamente autónomo. En estos escenarios el uso de fuentes renovables como la solar puede ayudar a incrementar la autonomía de estos dispositivos.

En el contexto de IoT, aún es necesario considerar dispositivos y componentes de bajo costo para garantizar la viabilidad de la implementación de un gran número de nodos, así como la adopción por parte de la comunidad.

3.2. Descripción del diseño propuesto para el proyecto.

Para el proyecto se plantea la arquitectura de la figura 3.1, el diseño planteado es una arquitectura MQTT tipo estrella en una red de área local inalámbrica (WLAN) bajo tecnología de comunicación WiFi, donde el elemento llamado “Broker MQTT” (mediante el protocolo comunicación de datos MQTT) gestiona y transmite los mensajes (mediciones de las condiciones meteorológicas del emplazamiento) emitidos por el nodo IoT energéticamente autónomo a la “Base de Datos” donde son almacenados, para la interacción entre ambos elementos se plantea el uso de un “Editor de Flujo” de datos que recopila y reporta o envía los mensajes provenientes del broker a la base de datos, los datos son posteriormente visualizados y presentados de una forma amigable en una interfaz gráfica representada por el elemento denominado “visualización”.

A continuación, se describen las especificaciones y características del diseño:

Especificaciones del diseño:

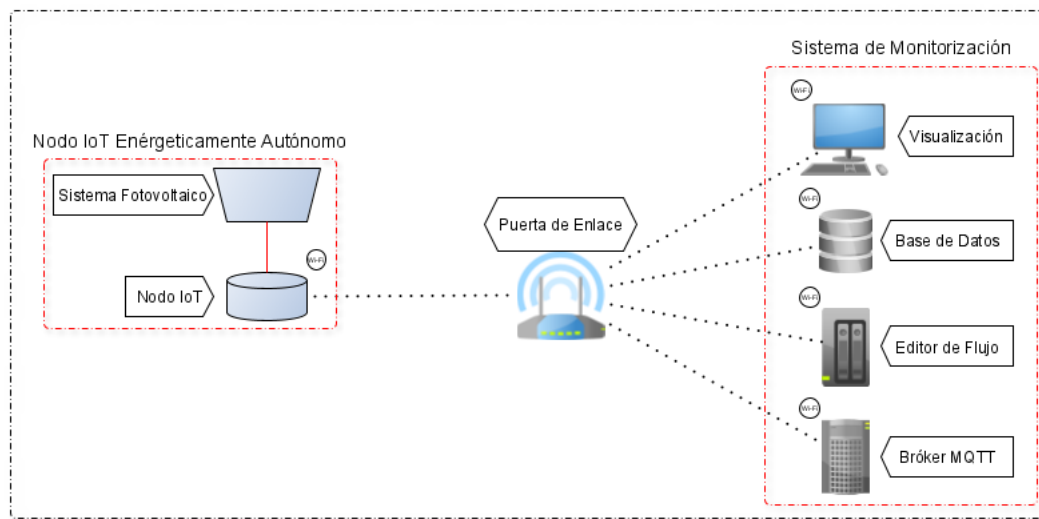


Figura 3.1: Arquitectura propuesta para proyecto.

- El nodo IoT está conformado por una plataforma IoT y sensores para medir temperatura, humedad, radiación UV, detección de lluvia y tensión eléctrica.
- La energía que alimenta al nodo IoT es de fuente renovable, mediante un sistema fotovoltaico con batería.
- La interconexión de los elementos que conforman la topología planteada es realizada mediante la tecnología WiFi en una red de área local inalámbrica (WLAN, de sus siglas en inglés), bajo el estándar IEEE 802.11 b/g/n en la banda 2.4 GHz.
- Todos los elementos deben estar conectados a la WLAN mediante la puerta de enlace o punto de acceso WiFi.
- Las medidas censadas son respaldadas y almacenadas en una base de datos.
- El protocolo de comunicación de datos elegido para gestionar los mensajes que contienen las medidas censadas es el protocolo MQTT.
- Las medidas censadas son gestionadas por el servidor llamado Broker MQTT y direccionadas por el elemento denominado Editor de Flujo que sirve de intermediario entre el broker y la base de datos.

- Las medidas archivadas en la base de datos son visualizadas y analizadas en una interfaz gráfica amigable.

Características del planteamiento:

- Implementación del nodo IoT en el exterior del Laboratorio de Investigación e Innovación (L2I), para el monitoreo de variables meteorológicas: Es fundamental presentar de una forma coordinada y al mismo tiempo ordenada un conjunto de herramientas que permitan monitorear las variables meteorológicas en un área específica.
- Desarrollo de un sistema fotovoltaico con batería: El sistema fotovoltaico garantiza el suministro de energía necesario para el correcto funcionamiento del nodo IoT, ya que existen escenarios donde no es posible tener acceso a la red eléctrica nacional.
- Elaboración de un sistema de recolección, visualización y monitoreo de los parámetros medidos por el nodo IoT: Proporciona versatilidad al sistema, ya que permite al usuario disponer de un historial de los datos recolectados y de una interfaz gráfica amigable que permita visualizar los datos censados en tiempo real.

3.3. Estrategias para el desarrollo del diseño propuesto

En esta sección son abordadas las estrategias para el desarrollo e implementación del nodo IoT, el sistema fotovoltaico con batería y la monitorización de los parámetros meteorológicos de acuerdo a la propuesta de la figura 3.1.

3.3.1. Estrategia para el desarrollo e implementación del nodo IoT

El nodo IoT está conformado por una serie de sensores (encargados de medir las condiciones meteorológicas y de tensión de la batería) conectados a una plataforma IoT como se ilustra en la figura 3.2.

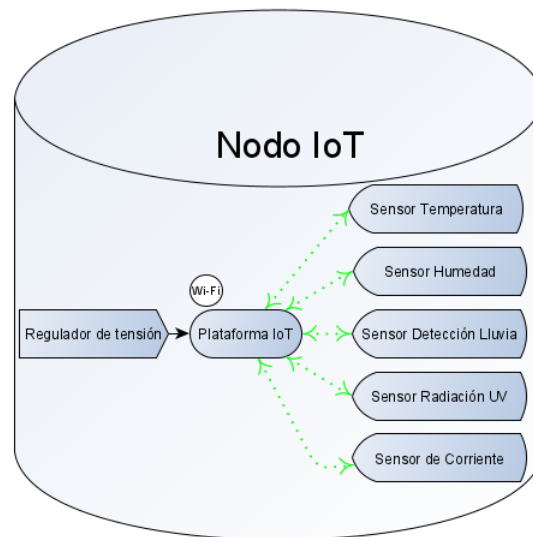


Figura 3.2: Diseño del Nodo IoT propuesto.

Plataforma IoT: para el proyecto se ha elegido una de las plataformas IoT más simple disponible de la casa Espressif System llamado ESP-12e, el cual se muestra en la figura 3.3, es un módulo referente para la creación de soluciones IoT de bajo costo, diseño compacto y una capacidad de procesamiento muy elevado en comparación a las plataformas de otras casas comerciales. Especificaciones técnicas:



Figura 3.3: Módulo WiFi ESP-12e [10].

- Microcontrolador: ESP8266
- Velocidad: 80 MHz
- Memoria RAM 32 kB
- Consumo de corriente en operación: 70 mA

- Tensión de operación: 3.3 V
- Memoria flash externa: 4 MB
- Consumo en modo *deep sleep*: $<10 \mu\text{A}$
- Protocolo de comunicación: IEEE 802.11 b/g/n
- Interfaz de comunicación: *Inter integrated circuits* (I²C), *Serial Peripheral Interface* (SPI), Transmisor-Receptor Asíncrono Universal (UART)
- Dimensiones: 16 x 24 x 3 mm

Regulador: para adaptar la tensión suministrada por el sistema fotovoltaico a la requerida por el nodo IoT, se emplea un regulador de tensión S09 V8.0, como se puede ver en la figura 3.4, cuya características técnicas son:

- Voltaje de entrada: 3 a 15 V
- Voltaje de salida: 3.3 V
- Corriente de salida: 0.6 A
- Eficiencia de conversión: 75 %
- Onda de salida: $<50 \text{ mV}$
- Precisión de salida: $\pm 0.1 \text{ V}$
- Temperatura de trabajo: -20 a 85 °C
- Humedad Relativa: 20 - 90 %
- Tamaño: 19 x 14 x 4 mm

Sensores: para la medición de las condiciones meteorológicas, se han empleado componentes que soportan condiciones adversas, puesto que la ubicación de los mismo es el exterior del laboratorio L2I, a continuación, se muestran los componentes elegidos:

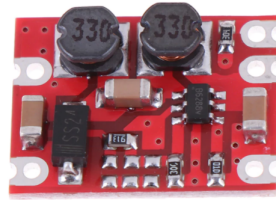


Figura 3.4: Regulador de tensión S09 V8.0 [11].

Para medir la temperatura y humedad relativa del emplazamiento, se hace uso del sensor digital SHT20 que se muestra en la figura 3.5, componente a prueba de agua, lo que lo hace ideal para trabajar en intemperie. Las características se muestran a continuación:

- Tensión de operación: 3.3 V
- Consumo de corriente en operación: 0.3 mA
- Interfaz de comunicación: I²C
- Rh Tiempo de respuesta: 8 s (τ)
- Exactitud: $\pm 3\%$ Rh / $\pm 0.3\text{ }^{\circ}\text{C}$
- Rango de medida: 0 - 100 % Rh / -40 a 125 $^{\circ}\text{C}$
- Dimensiones: 73 x 17 mm



Figura 3.5: Sensor digital de Temperatura y Humedad modelo SHT20.

Para medir la radiación ultravioleta, se ha optado por el uso de un sensor digital VEML6075 como se observa en la figura 3.6, este sensor detecta la radiación UV en las bandas UVA y UVB, sus características técnicas son:

- Tensión de operación: 3.3 V
- Consumo de corriente en operación: 0.48 mA
- Interfaz de comunicación: I²C
- Pico de sensibilidad UVA, UVB: 365, 330 (nm)
- Rango espectral del ancho de banda $\lambda_{0,5}$: ± 10
- Dimensiones: 2.0 x 1.25 x 1.0 mm



Figura 3.6: Sensor digital Radiación Ultravioleta modelo VEML6075 [12].

Para detectar la presencia de lluvia, emplearemos un sensor de lluvia YL-38 mostrado en la figura 3.7, conectado a una resistencia modelo YL-83, sus especificaciones técnicas son:

- Tensión de operación: 3.3 V
- Sensibilidad ajustable por potenciómetro
- Consumo de corriente en corto circuito: 4 mA
- Consumo de corriente abierto: 2 mA

- Interfaz de comunicación: analógica, digital
- Dimensión del sensor: 30 x 15 mm
- Dimensión de la resistencia YL-83: 54 x 39 mm



Figura 3.7: Sensor Detección de LLuvia modelo YL-38 y resistencia YL-83 [13].

El funcionamiento del nodo IoT va a estar determinado de acuerdo al nivel de tensión de batería, para ello usaremos el sensor de corriente eléctrica modelo INA219 como se observa en la figura 3.8, cuyas características técnicas son las siguientes:

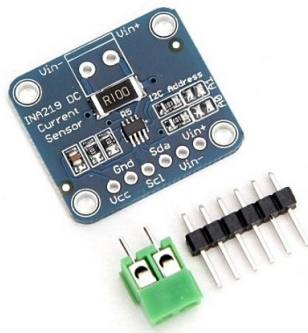


Figura 3.8: Sensor de corriente eléctrica DC [14].

- Tensión de operación: 3 V
- Rango de medición de corriente: ± 3.2 A

- Resolución: ± 0.1 mA (Ajustable)
- Medida de voltaje máximo: 26 V
- Dimensión del sensor: 25.5 x 22.5 mm

El fabricante del ESP-12e provee de una herramienta para la programación del *firmware* llamado *Software Development kit*, pero se optó por la utilización del *Integrated Development Environment* de Arduino [15], empleado librerías de código abierto y libre que facilitan la programación de la plataforma.

Inicialmente las condiciones meteorológicas serán muestreadas y enviadas a la base de datos cada 60 segundos, luego se pretende ejecutar estrategias para mejorar el consumo de la energía del nodo IoT. Esta optimización y mejora de la autonomía del nodo IoT, se llevará acabo considerando algoritmos capaces de adaptar el procesamiento local, específicamente ajuste del tiempo de muestreo de los datos recogidos por los sensores de la frecuencia de muestreo y la frecuencia con que son enviados a la base de datos según el nivel de batería. Adicionalmente será empleado el modo de ahorro de energía disponible en el ESP-12e mediante el modo *deep sleep*, que consiste en poner en reposo el dispositivo mientras este no realiza medición alguna, luego de transcurrido el tiempo de reposo el dispositivo enciende, toma las muestras y las envía a la base de datos, repitiendo este ciclo.

3.3.2. Estrategia para el desarrollo e implementación del sistema fotovoltaico con batería

Para garantizar la autonomía y funcionamiento del nodo IoT en escenarios remotos, con ausencia de energía eléctrica suministrada por la red eléctrica nacional, se hace uso de la energía solar como fuente de energía, para ello se emplea un sistema solar fotovoltaico con batería como se muestra en la figura 3.9, constituido por un cargador, una batería y paneles solares.

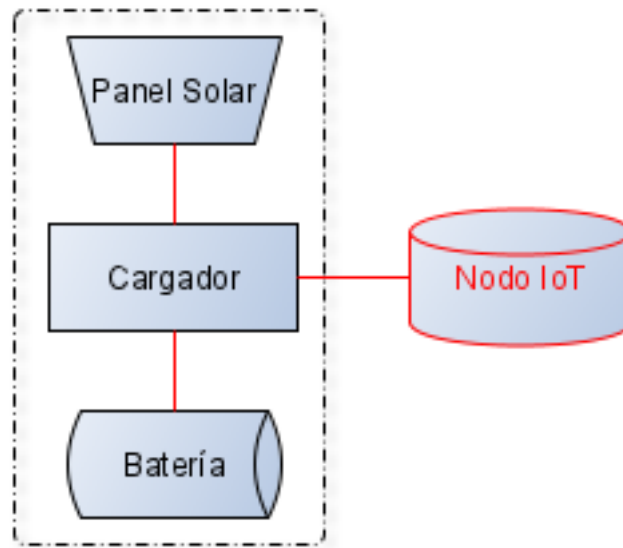


Figura 3.9: Diseño del sistema fotovoltaico planteado.

Para captar la energía solar empleará un panel solar, en el mercado podemos conseguir una variedad de ellos que varían según las características técnicas y aplicación. Para el desenvolvimiento del sistema fotovoltaico nos hemos enfocado en paneles solares para microelectrónica, considerando los más comunes monocromáticos y policristalinos. Particularmente se selecciono un panel policristalinos modelo CNC60x110-6 mostrado en la figura 3.10 , de bajo costo, con unas características:

- Voltaje: 6 V
- Corriente pico: 166 mA
- Potencia pico: 1 W
- Dimensiones: 110 x 60 mm

La energía captada por el panel solar es almacenada en una batería recargable de tipo Lithium-ion modelo NCR18650B (ver figura 3.11) de bajo mantenimiento. Características técnicas de la batería:

- Capacidad: 3400 mAh

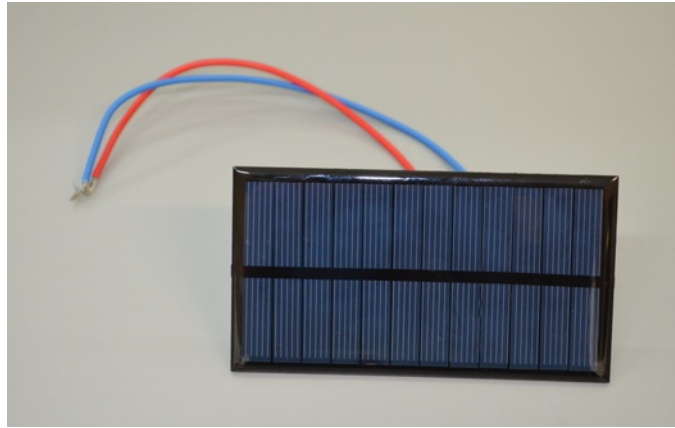


Figura 3.10: Panel solar de silicio policristalino modelo CNC60x110-6.

- Tensión de corte: 2.75 V
- Tensión máxima: 4.20 V
- Tensión nominal: 3.7 V
- Corriente máxima: 6.7 A
- Dimensiones: 18 x 65 mm
- Peso: 45 g



Figura 3.11: Batería recargable tipo Lithium-ion modelo NCR18650B.

Para cargar la batería será empleado un cargador lineal Solar Charger V1.0 modulo basado en el chip CN3065, ver figura 3.12, compatible con baterías Lithium-ion. Las características principales se muestran a continuación:

- Tensión entrada panel solar: 4.4 - 6 V
- Tensión de regulación nominal: 4.2 V
- Tensión de salida: tensión de la batería
- Corriente de carga máxima: 500 mA
- Protección contra corto circuito
- Dimensiones: 40 x 20 x 7 mm



Figura 3.12: Cargador lineal Solar Charger V1.0 [16].

3.3.3. Estrategia para el monitoreo de los parámetros meteorológicos

La estrategia planteada para la monitorización de las condiciones ambientales se describe mediante la figura 3.13, de acuerdo al siguiente proceso:

1. El nodo IoT mide las condiciones ambientales y envía esa información en un mensaje al “Broker MQTT” por medio de la puerta de enlace (medio de comunicación inalámbrica).
2. El broker gestiona los mensajes enviados por el nodo y los publica.

3. Los mensajes publicados por el broker son recolectados por “Editor de Flujo” y enviados a la “Base de Datos”.
4. Por ultimo los mensajes almacenados en la base de datos podrán ser visualizados y analizados en el elemento llamado “visualización”.

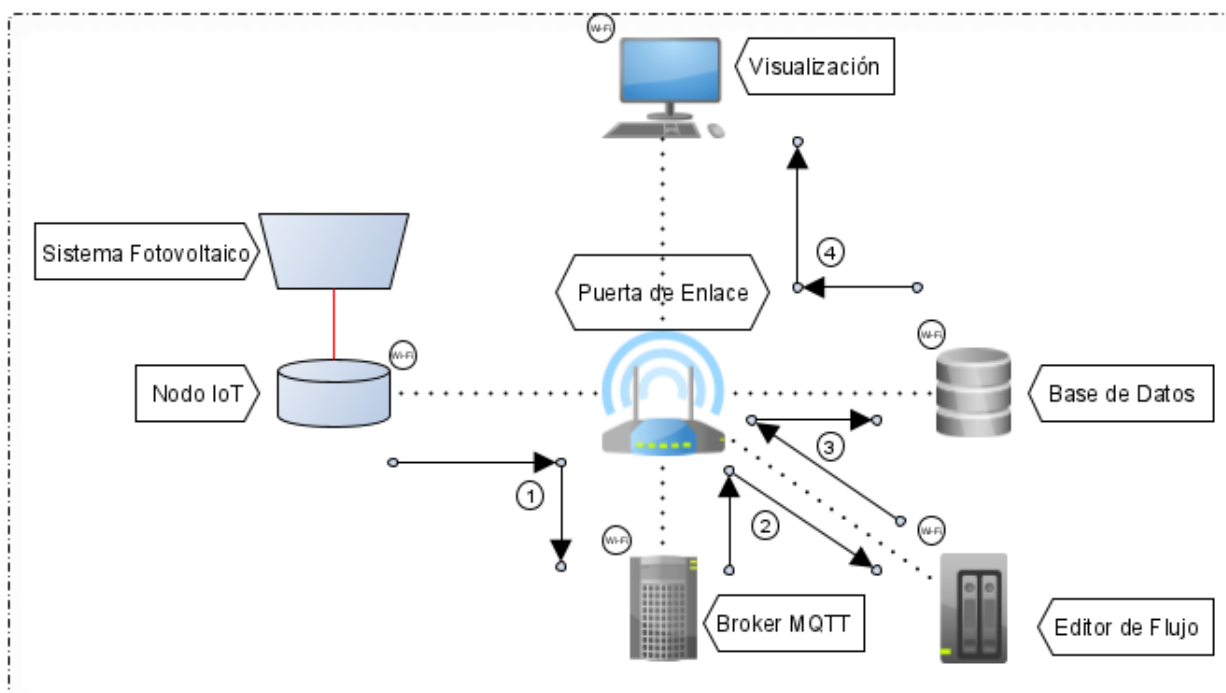


Figura 3.13: Proceso de monitorización de los parámetros meteorológicos.

Específicamente se plantea para cada elementos proceso lo siguiente:

El nodo IoT va a permanecer en el exterior del laboratorio L2I en condiciones ambientales extremas por lo que las medidas de las condiciones meteorológicas censadas requieren ser almacenadas en un servidor externo al dispositivo, esto conlleva a que las medidas sean enviadas a una base de datos, para ello será empleada una base de datos de serie de tiempo, software de código abierto y gratuito llamado InfluxDB [17] donde las medidas serán almacenadas y resguardadas para su posterior utilización.

Para interconectar el nodo IoT con la base de datos se realiza mediante la tecnología de comunicación WiFi, bajo el estándar IEEE 802,11 b/g/n en la banda 2.4 GHz, conformada

3.4. CRITERIOS PARA LA SELECCIÓN DE LOS COMPONENTES Y ESTRUCTURA DEL SISTEMA

por una red de área local inalámbrica, existente en el laboratorio.

El protocolo de comunicación de datos empleado para comunicar las medidas del nodo IoT a la base de datos es el protocolo MQTT, concretamente se ha emplea una arquitectura MQTT tipo estrella que tiene como elemento central el broker, este es implementado mediante una herramienta denominada Mosquito [18] y se encarga de gestionar los mensajes publicados por el nodo IoT. El flujo de mensajes serán recolectados y enviados a la base de datos mediante la herramienta de desarrollo visual llamada Node-RED [19] (software de código abierto y gratuito).

Por último, para visualizar las medidas almacenadas en la base de datos se ha optado por usar un software libre y gratuito denominado Grafana [20].

3.4. Criterios para la selección de los componentes y estructura del sistema planteado

En este apartado se describen los criterios empleados para la selecciones de los componentes y estructura del sistema planteado.

- Bajo costo de los dispositivos y componentes.
- Materiales resistentes a condiciones climáticas extremas.
- Compactos y livianos.
- Poca interacción con el usuario.
- Interfaz gráfica amigable.
- Fácil reposición a futuro.
- Bajo consumo energético.

3.5. Cálculos y medidas requeridas para el diseño y funcionamiento del proyecto

3.5.1. Capacidad de generación diaria del sistema fotovoltaico

La capacidad de generación del sistema fotovoltaica es determinada de acuerdo a la ecuación 3.1, considerando para ello los meses del año donde la irradiación solar útil es la más desfavorable.

$$Capacidaddegeneración_{diaria} = Potencia_{PanelFotovoltaico} \cdot HSP \quad (3.1)$$

Donde:

La potencia del panel fotovoltaico de acuerdo a la ficha técnica es de 1 W y las Horas Pico Solar (HSP) del emplazamiento es determinado mediante la herramienta en-línea denominada *Photovoltaic Geographical Information System* (PVGIS) [21]; para ello se partió de los datos mensuales de irradiación solar de la zona, se tomó como base de datos de radiación solar PVGIS-SARAH (entre los años 2005 y 2015) con un ángulo de irradiación global óptima determinado por la herramienta, posteriormente se realizó un promedio de las radiaciones mensuales de cada año y por último se determinó el número de Horas Solar Pico por día de acuerdo a los días de cada mes, como se muestra en la tabla 3.1.

En la tabla 3.1 podemos observar que los meses de enero, noviembre y diciembre son los meses con menos horas equivalentes para obtener la insolación total de un día, para el calculo de la ecuación 3.2 es empleado 3 HSP, por lo tanto la capacidad de generación diaria del sistema fotovoltaico como se observa en la ecuación 3.2 es de 3 Wh.

$$Capacidaddegeneración_{diaria} = 1W \cdot 3h = 3Wh \quad (3.2)$$

3.5. CÁLCULOS Y MEDIDAS REQUERIDAS PARA EL DISEÑO Y FUNCIONAMIENTO DEL PROYECTO

Tabla 3.1: Cálculo de la Horas Solar Pico de la ciudad de Bragança.

Periodo	Mes	Irradiación global _{Prom.mes} (Incl. 35°)	Horas Solar Pico _{día}
2005 - 2015	Enero	90.58	3
2005 - 2015	Febrero	116.78	4
2005 - 2015	Marzo	155.67	5
2005 - 2015	Abril	166.67	6
2005 - 2015	Mayo	194.54	6
2005 - 2015	Junio	204.31	7
2005 - 2015	Julio	230.70	7
2005 - 2015	Agosto	225.17	7
2005 - 2015	Septiembre	190.03	6
2005 - 2015	Octubre	142.16	5
2005 - 2015	Noviembre	101.21	3
2005 - 2015	Diciembre	95.07	3

3.5.2. Cálculo autonomía de la batería

La autonomía de la batería en horas será calculado empleando la ecuación 3.3.

$$Autonomía_{Batería} = Capacidad_{Batería} \cdot Corriente_{Consumida} \quad (3.3)$$

Donde la capacidad de la batería está definida en las características técnicas y la corriente consumida es la corriente media consumida por el nodo IoT y el circuito regulador de tensión cuando el nodo IoT se encuentre en funcionamiento de acuerdo a los dos modos de funcionamiento que se pretenden implementar y serán discutidos en el capítulo 5.

Capítulo 4

Implementación

En el presente capítulo es abordado el proceso de implementación de una solución para un nodo IoT energéticamente autónomo, se describe el lugar donde será puesto en funcionamiento el dispositivo, así como los pasos seguidos para la selección de los componentes que constituyen el prototipo.

4.1. Descripción del emplazamiento

El proyecto es implementado en el exterior del Laboratorio de Investigación e Innovación (L2I) del Research Centre in Digitalization and Intelligent Robotics (CeDRI), de la Escola Superior de Tecnologia e Gestão (ESTiG) del Instituto Politécnico de Bragança (IPB). ESTiG cuenta con una infraestructura para el correcto desenvolvimiento de estos dispositivos, puesto que dispone de medios de comunicación inalámbricos, así como servidores para la gestión y procesado de datos en tiempo real. L2I identificado en la figura 4.1 con el recuadro azul, se encuentra ubicado en la planta baja de la ESTIG, el edificio cuenta con una azotea plana (área demarcada en color rojo) sin obstáculos, además de un campo abierto en los alrededores, esto permite disponer una óptima ubicación del prototipo, puesto que no presentan obstáculos que impidan la captación de la energía solar.



Figura 4.1: Campus Escola Superior de Tecnologia e Gestão [22].

4.2. Implementación del nodo IoT energéticamente autónomo

En este apartado se describe como se realizó la implementación del nodo IoT para la medición de las condiciones meteorológicas y de tensión eléctrica de la batería, así como la implementación e interconexión del sistema fotovoltaico con el nodo para garantizar el suministro eléctrico necesario para el correcto funcionamiento del nodo IoT, al final del proceso es presentado el prototipo del nodo IoT energéticamente autónomo elaborado mediante la herramienta de diseños electrónicos llamada *Easily Applicable Graphical Layout Editor* (EAGLE) [23].

4.2.1. Circuito eléctrico del nodo IoT

La instalación de los sensores para medir humedad, temperatura, radiación UV y tensión eléctrica, se realizó empleando el protocolo I²C, mientras que para el sensor de

4.2. IMPLEMENTACIÓN DEL NODO IOT ENERGÉTICAMENTE AUTÓNOMO 43

detección de lluvia se realizó mediante el pin digital GPIO12 del ESP-12e al pin digital del sensor. A la entrada del ESP-12e se implementó un convertidor reductor-elevador (regula la tensión de la batería a una tensión de 3.3 V requerida por el ESP-12e) con un condensador en la entrada y otro a la salida para garantizar la estabilidad del convertidor, de esa forma garantizar la tensión de salida necesaria requerida por el ESP-12e. En la figura 4.2 se presenta la conexión eléctrica de los sensores con el ESP-12e.

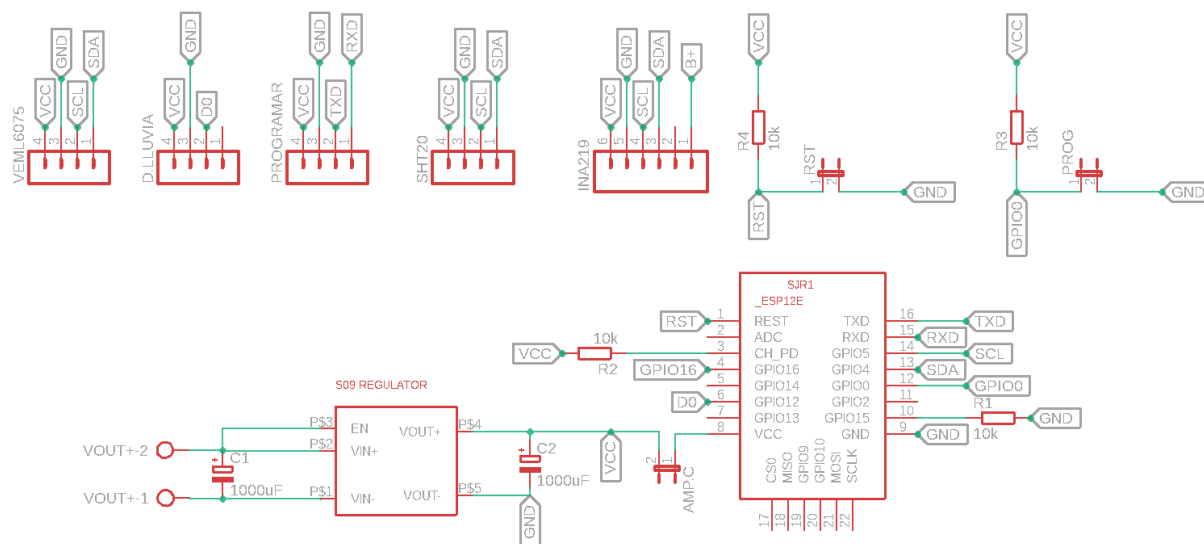


Figura 4.2: Conexión del circuito eléctrico del Nodo IoT.

La comunicación entre el ESP-12e y los sensores SHT20, VEML7065 e INA219 se realiza por medio de las 2 líneas de comunicación (línea serial de datos (SDA) y la línea serial reloj (SCL)) del bus I²C. Para la comunicación de los datos se han empleado los pines GPIO4 como línea serial de datos y el pin GPIO5 como línea serial de reloj del ESP-12e, estas líneas están conectadas a una fuente positiva a través de las resistencias de *pull-up* interna del ESP-12e. El microcontrolador contenido en la plataforma IoT representa en la comunicación la unidad maestra, que contiene las rutinas para las lecturas de las medidas ofrecidas por los sensores. Los sensores digitales SHT20, VEML6075 e INA219 conectados en el bus I²C son unidades esclavas, las cuales están identificadas con una dirección de 7 bits que las identifica para que pueda existir un entendimiento entre los dispositivos, al sensor SHT20 se le asignó la dirección 0x40, al sensor VEML6075 el 0x20 y al sensor

INA219 la dirección 0x41. En la figura 4.3 se muestra la topología del bus I²C para la conexión del ESP-12e con los sensores SHT20, VEML6075 e INA219.

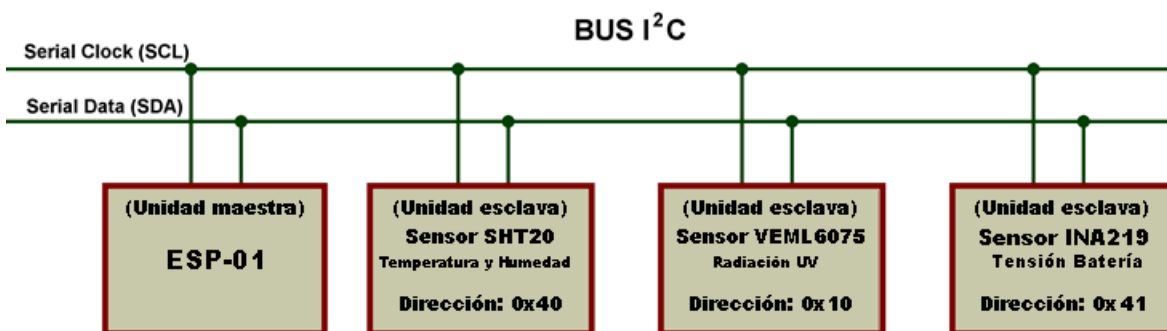


Figura 4.3: Topología del bus I²C entre el módulo ESP-12e y los sensores SHT20, VEML6075 e INA219.

El ESP-12e mediante el pin GPIO12 lee el estado del pin digital del sensor de lluvia que cuenta con un comparador de tensión que genera en el pin GPIO12 una señal alto en presencia de lluvia y una señal bajo en ausencia de lluvia.

4.2.2. Circuito eléctrico programador

La programación del nodo IoT se realizó mediante el conector de 4 pines identificado en la figura 4.2 identificado con el nombre “PROGRAMAR” que nos permite mediante los pines TXD y RXD la programación del mismo, para ello se incluyó una placa de desarrollo NodeMCU V1 [24] compuesto de un microprocesador ESP8266 y un integrado CP2102 que sirve de puente de comunicación entre el computador mediante el bus universal en serie (USB) y el nodo IoT usando el bus de datos serial asíncrono (UART). En la figura 4.4 se muestra el circuito eléctrico empleado para programar el nodo IoT, en el circuito el microcontrolador del NodeMCU se ha deshabilitado mediante la conexión del pin EN a tierra (esto evita que el microcontrolador del NodeMCU inicie y poder tener control del microcontrolador del nodo IoT) y los pines de transmisión y recepción de datos (TX y RX) del NodeMCU se han conectado a sus pares del Nodo IoT.

4.2. IMPLEMENTACIÓN DEL NODO IOT ENERGÉTICAMENTE AUTÓNOMO 45

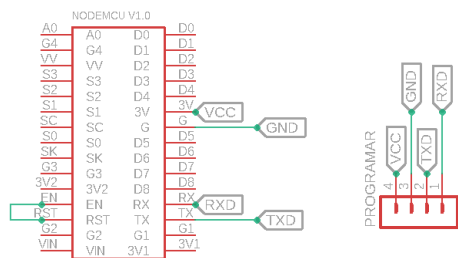


Figura 4.4: Circuito eléctrico empleado para programar el nodo IoT.

4.2.3. Circuito eléctrico de medición

Para determinar el consumo eléctrico del nodo IoT fue empleado un sensor INA219 controlado por un módulo NodeMCU mediante el bus de comunicación I²C. La conexión del circuito eléctrico se muestra en la figura 4.5, el sensor INA219 es configurado con una frecuencia de muestreo de un segundo. Las medición se realiza empleando los pines identificados como “I+” e “I-” conectados en serie al circuito que se desea medir.

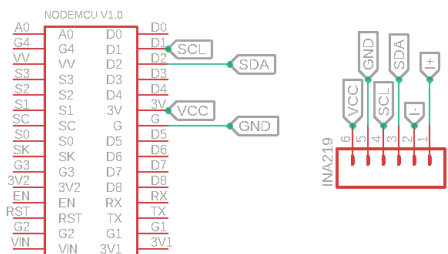


Figura 4.5: Circuito eléctrico empleado para las pruebas de medición de tensión, corriente y potencia.

4.2.4. Circuito eléctrico del sistema fotovoltaico

El sistema fotovoltaico fue ensamblado de acuerdo al esquema de la figura 4.6. La implementación consta de un panel fotovoltaico con una potencia pico de 1 W, para cargar la batería de 3400 mAh.

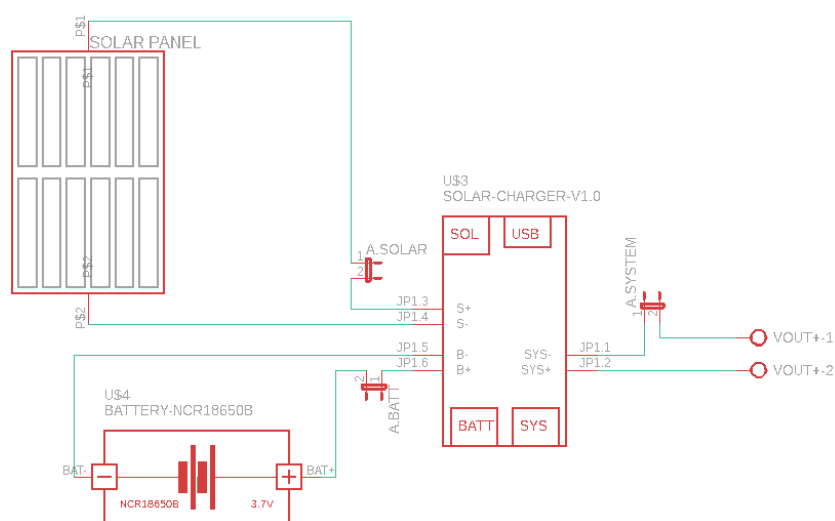


Figura 4.6: Esquema del circuito eléctrico del sistema fotovoltaico.

4.2.5. Conectividad de la plataforma IoT con la red de área local Inalámbrica

La conectividad de la plataforma IoT con la red de área local Inalámbrica se ha realizado haciendo uso de la librería “ESP8266WiFi.h” [25] que contiene las instrucciones específicas del módulo WiFi del ESP-12e, en ella se ejecuta el proceso descrito en el diagrama de la figura 4.7, en las instrucciones se definen los parámetros de red necesario para lograr la conectividad con el enrutador inalámbrico del L2I. El código empleado para conectar el ESP-12E con la red de área local inalámbrica se encuentra en el anexo A.

4.2.6. Lectura de los parámetros medidos

Para poner en funcionamiento y poder obtener las medidas de cada uno de los sensores fueron empleados las siguientes librerías: para el sensor de humedad y temperatura se empleó la librería “DFRobot_SHT20.h” [26], para el sensor de radiación UV se empleó la librería “Adafruit_VEML6075.h” [12] y para el sensor de corriente “Adafruit_INA219.h” [27], las librerías contienen las rutinas necesarias para la lectura mediante el protocolo I²C de las medidas de cada parámetro meteorológico, así como su transformación para nuestro entendimiento. La lectura del sensor de detección de lluvia se ha realizado mediante el puerto

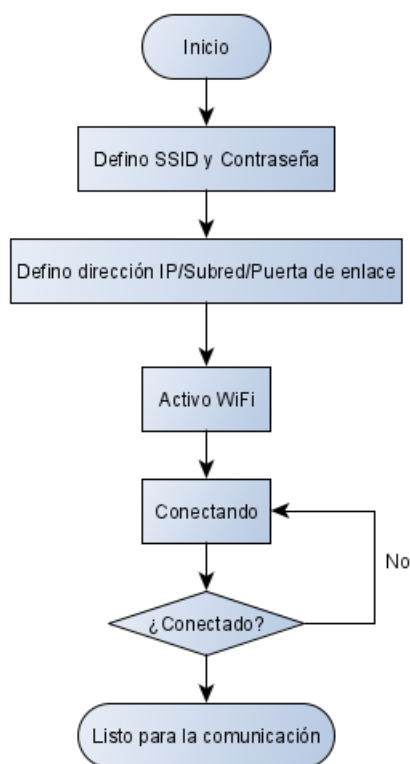


Figura 4.7: Diagrama de flujo de conexión entre el ESP-12E y el enrutador inalámbrico.

digital, para su implementación se ha definido el pin GPIO12 del módulo ESP-12e como pin de entrada. La lectura del sensor es según estado del pin digital, pin en estado lógico bajo “0” significa que está lloviendo y en estado alto “1” que no está lloviendo. En la figura 4.8 se pueden observar las rutinas para la lectura de los parámetros censados.

4.2.7. Implementación del protocolo MQTT en el nodo IoT

Para proveer de la información a la base de datos de los parámetros censados por el nodo IoT se implementó el protocolo MQTT, usando para ello la librería referencia “AsyncMqttClient.h” [28], donde son ejecutadas las acciones necesarias para el funcionamiento del nodo IoT y desenvolver el protocolo MQTT en modo cliente, publicando en la red de área local inalámbrica las medidas censadas. En la figura 4.9 se muestra el funcionamiento implementado.

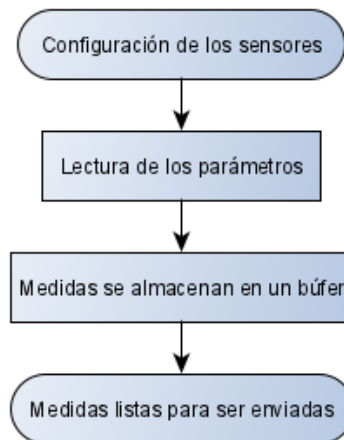


Figura 4.8: Diagrama de flujo de la lectura de los parámetros medidos.

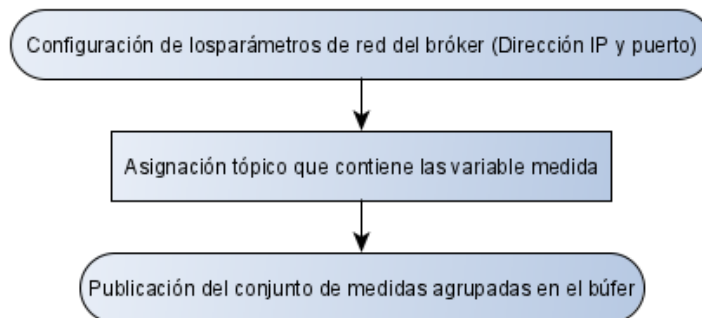


Figura 4.9: Diagrama de publicación de las medidas censadas mediante el protocolo MQTT.

4.2.8. Implementación del funcionamiento normal del nodo IoT

El funcionamiento denominado normal del nodo IoT se ilustra en la figura 4.10 y representa el funcionamiento común de muchos dispositivos de IoT, este funcionamiento es implementado de acuerdo al diagrama de flujo de la figura 4.10 y es empleado como referencia, con la finalidad de lograr mejoras en la eficiencia y el consumo de energía del nodo IoT.

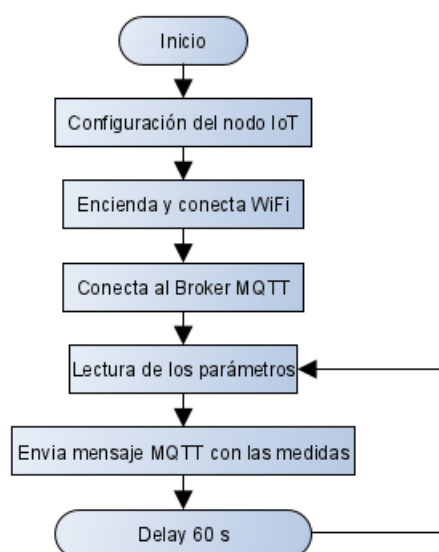


Figura 4.10: Diagrama de flujo del funcionamiento normal del nodo IoT.

4.2.9. Prototipo del nodo IoT energéticamente autónomo

En la figura 4.11 es presentado el circuito del prototipo del nodo IoT energéticamente autónomo y no es más que la interconexión del nodo IoT con el sistema fotovoltaico. Por otra parte en la figura 4.12 es mostrado el diseño de la placa de circuito impreso del prototipo.

4.2.10. Implementación del funcionamiento optimizado del nodo IoT energéticamente autónomo

La optimización del nodo IoT es llevado a cabo modificando la programación que contiene el funcionamiento denominado normal del nodo IoT presentada la subsección 4.2.8, esto es llevado a cabo de acuerdo al diagrama de flujo de la figura 4.13 y pretende optimizar el consumo energético y la autonomía del nodo IoT, de acuerdo a las siguientes medidas:

- Para la optimización del consumo energético se emplea el modo de ahorro de energía *deep sleep*, desconectando el módulo WiFi del ESP-12e al iniciar el dispositivo y

4.3. IMPLEMENTACIÓN DE UNA CAJA DE MONTAJE PARA LA PROTECCIÓN DEL PROTOTIPO

muestreo y de publicación de las mediciones censadas por el nodo IoT, en tres niveles, de acuerdo a la tensión presente en la batería como se observa en la figura 4.11).

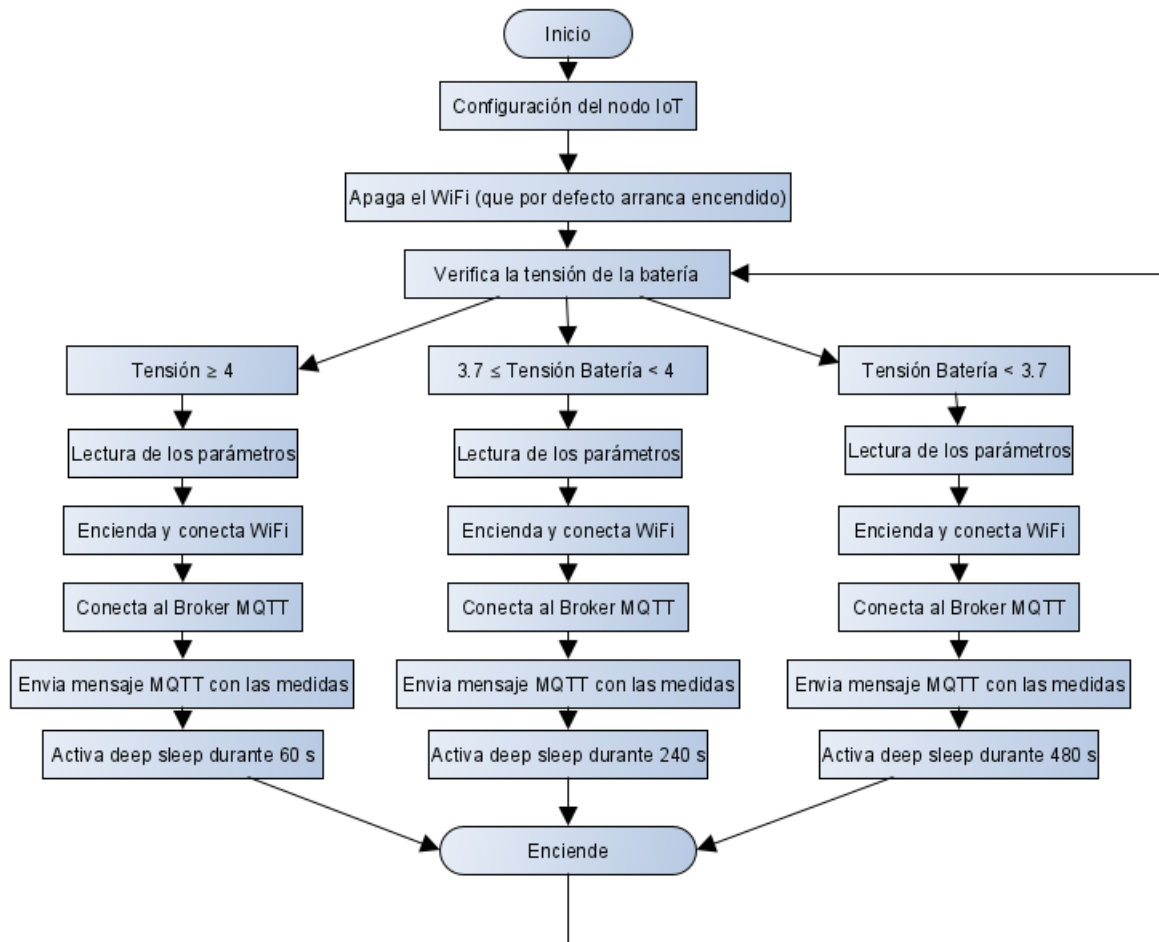


Figura 4.13: Diagrama de flujo del funcionamiento optimizado del nodo IoT.

4.3. Implementación de una caja de montaje para la protección del prototipo

Para proteger la integridad del circuito impreso junto a los componentes que conforman el nodo IoT, se ha diseñado una caja de montaje protectora como se observa en la figura

4.14, para ello se ha empleado el software FreeCAD [29]. La caja está conformada por un espacio interior para almacenar los componentes que no pueden estar expuestos a condiciones ambientales adversas por lo cual se encuentra sellada mediante una tapa que puede ser abierta por el usuario para efectos de mantenimiento; en la parte superior cuenta con una cara con una inclinación de 35 grados respecto al plano horizontal, donde serán colocados el panel solar, la resistencia YV-83 del detector de lluvia y un cristal que protege al sensor de radiación UV, mientras que en la parte inferior se encuentra un orificio donde es ubicado el sensor SHT20.

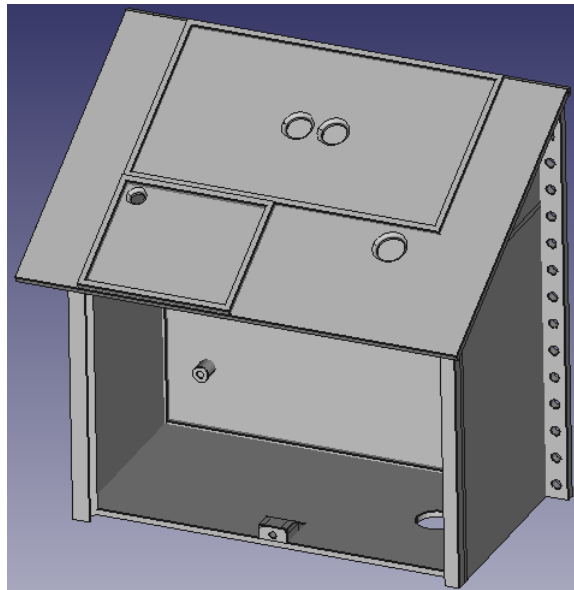


Figura 4.14: Diseño de la caja para la impresión 3D que alberga y protege los componentes del nodo IoT.

4.4. Implementación del sistema de monitoreo

Para el desarrollo del sistema de monitoreo son empleados una serie de softwares que cumplen la función de Base de Datos, Editor de Flujo, Broker MQTT y Visualizador, estos elementos son implementados en un computador (servidor) con sistema operativo Windows que ejecuta los diferentes softwares en paralelo. A continuación, se describe como fue el desarrollo del sistema para el monitoreo de las mediciones meteorológicas:

4.4.1. Implementación de la topología de red de área local inalámbrica

Para la conectividad con la red de área local inalámbrica se realizó de acuerdo al esquema mostrado en la figura 4.15.

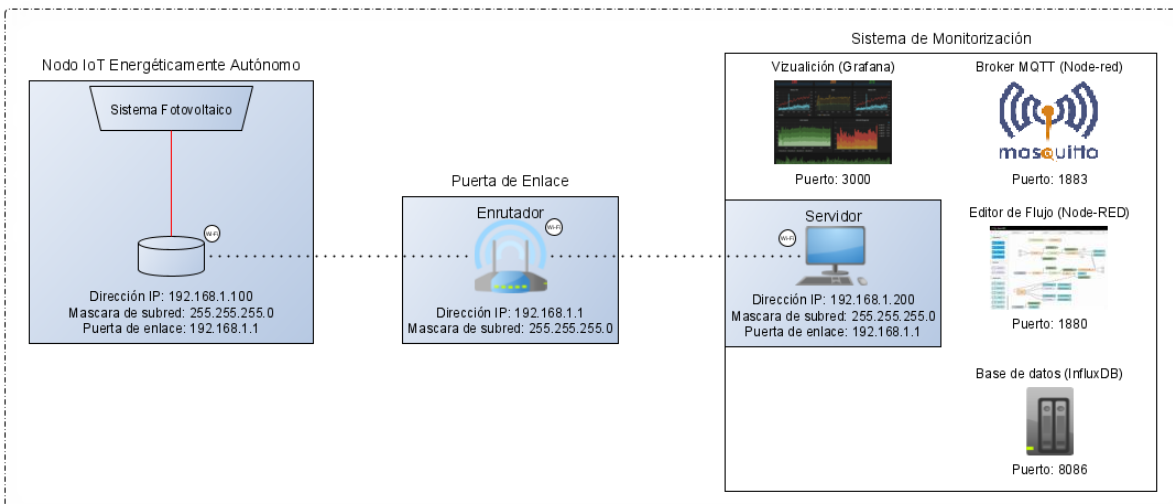


Figura 4.15: Topología de Red de Área Local Inalámbrica.

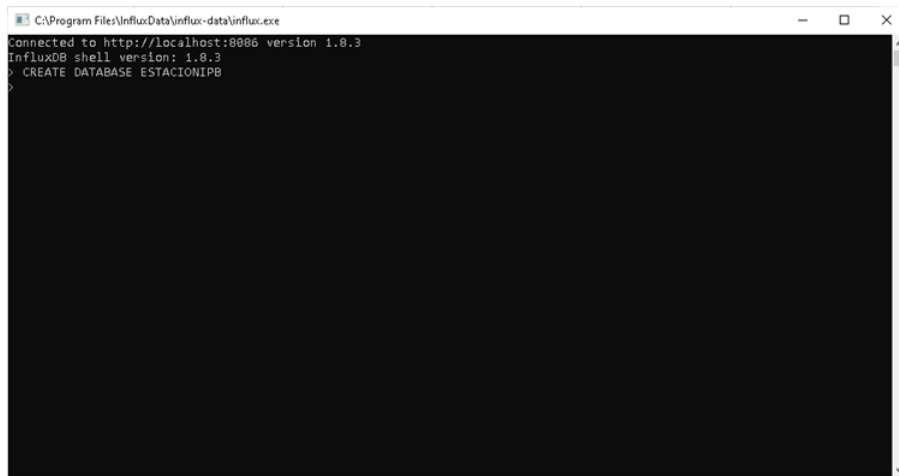
Para proporcionar la conectividad de una red de área local inalámbrica, se recurrió a un enrutador existente en el laboratorio L2I, específicamente un enrutador Cisco System modelo Linksys que proporciona la interconectividad entre los dispositivos de la red, cabe destacar que este dispositivo permite obtener conectividad a una Red de Área Ampliada (WAN) por medio de un Proveedor de Servicio Internet (ISP), pero la implementación está limitada a una red de red local. La red de área local inalámbrica está conformada por direcciones Protocolo de Internet (IP) privadas para la interconectividad. Cada elemento de la topología de red fue configurado según los parámetros de red mostrados en la figura 4.15.

4.4.2. Implementación de la base de datos

La base de datos fue implementada en el servidor, para ello se realizó la instalación de software InfluxDB [30] y puesto en funcionamiento mediante su ejecución, por último,

se creó el nombre de la base de datos como se muestra en la figura nombre, donde las medidas censadas por el nodo IoT de las condiciones meteorológicas son almacenadas, mediante el comando 1.

(Comando 1) **CREATE DATABASE** “nombre”



```
C:\Program Files\InfluxData\influx-data\influx.exe
connected to http://localhost:8086 version 1.8.3
InfluxDB shell version: 1.8.3
> CREATE DATABASE ESTACIONIPB
>
```

Figura 4.16: Nombre de la base de datos que almacena los parámetros meteorológicos censados.

4.4.3. Implementación del Broker MQTT

La implementación se realizó mediante la instalación del software Mosquito [18], una vez instalado, el software es ejecutado automáticamente al iniciar el servidor y contiene el *script* encargado de coleccionar los mensajes con las medidas censadas generadas por el nodo IoT y publicarlos en el tópico definido. La configuración por defecto del puerto del software es el 1883.

4.4.4. Implementación Node-RED (Editor de Flujo)

Esto se ha llevado a cabo para interconectar el Broker MQTT con la base de datos. La implementación de esta herramienta se realizó mediante la instalación del software Node-RED [31]. Luego de ejecutado fue definido el espacio de trabajo en cuatro bloques como se muestra en la figura 4.17, a continuación, es descrito cada bloque:

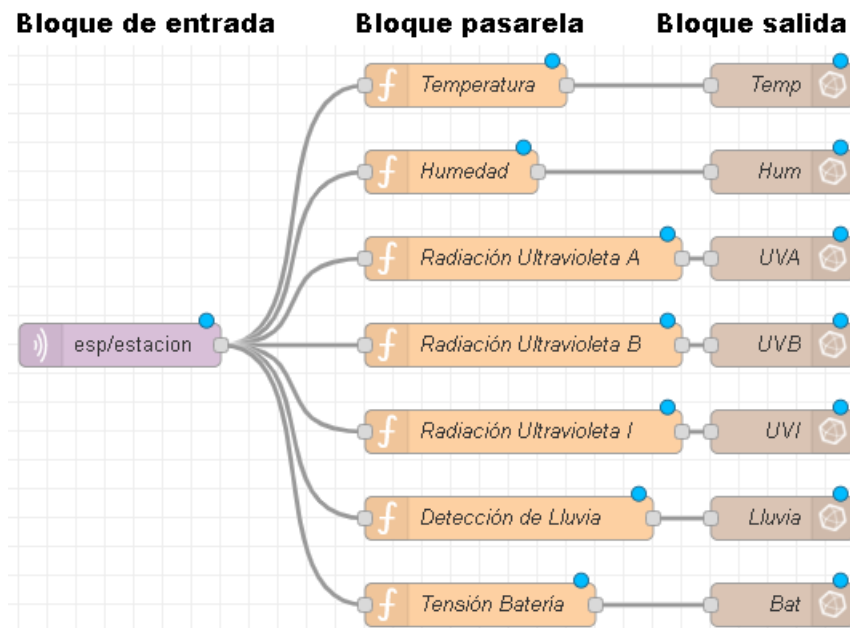


Figura 4.17: Espacio de trabajo “estación meteorológica” para la interconexión de los mensajes MQTT con la base de datos.

- Bloque de entrada: este bloque se encarga de recibir las medidas censadas por el nodo IoT, esto se realizó usando el módulo “mqtt in”, el cual se suscribe al tópico donde son publicadas las medidas censadas.
- Bloque pasarela: es un módulo “function” que contiene una función para extraer y dividir las medidas recibidas por el bloque de entrada correspondiente a cada medida censada.
- Bloque salida: por último, luego de que se obtienen las medidas de cada parámetro, son enviadas a la base de datos mediante el módulo “influxdb out”, para ello solo es definido el nombre de la base de dato y el nombre de la variable donde se desea almacenar las medidas.

4.4.5. Implementación del sistema de visualización

Para visualizar las mediciones obtenidas de las condiciones meteorológica del emplazamiento en tiempo real (almacenadas en la base de datos), se hizo uso de una herramienta

de desarrollo visual denominada grafana (software libre y gratuito), la cual es instalada en el servidor [32]. Por defecto el software es configurado en el puerto 3000, el acceso a grafana se logra con un explorador de Windows por medio de la dirección de red y el puerto, <http://192.168.1.200:3000>. Al acceder a la interfaz gráfica se procedió a configurar la fuente de los datos de la base de datos InfluxDB.

La interfaz creada para el propósito de este proyecto está compuesta por gráficos que varían en el tiempo de acuerdo a cada variable meteorológicas, además de panes que indican el valor actual del parámetro como se observa en la figura 4.18.



Figura 4.18: Interfaz gráfica para visualizar los parámetros meteorológicos.

Capítulo 5

Pruebas al sistema

Las pruebas abarcan la comprobación tanto del hardware como del software, y se realizan al sistema fotovoltaico, al nodo IoT y al sistema de monitoreo.

5.1. Prototipo del nodo IoT energéticamente autónomo

A fin de realizar las pruebas al sistema se diseñó el circuito impreso que se observa en la figura 5.1, el cual está conformado por el sistema fotovoltaico y el nodo IoT.

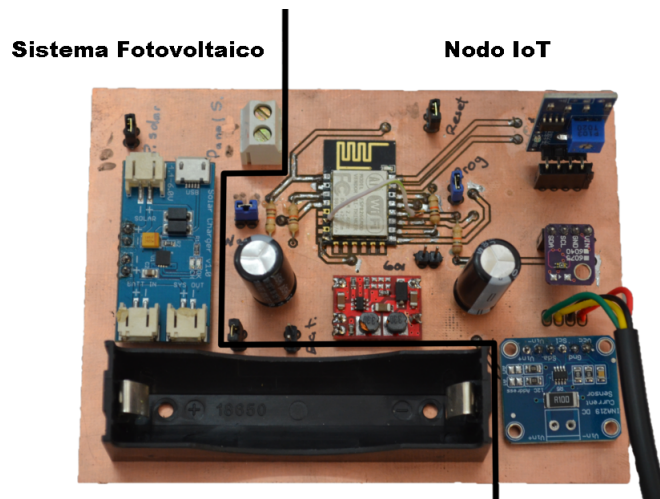


Figura 5.1: Prototipo del nodo IoT energéticamente autónomo.

5.1.1. Pruebas al sistema fotovoltaico

Para comprobar el funcionamiento del sistema fotovoltaico, este fue puesto en funcionamiento, posteriormente se realizó un estudio de la corriente, tensión y potencia al panel solar y la batería, esto se llevó a cabo mediante dos circuitos de medición compuestos por sensores INA219 descritos en el apartado 4.2.3, que fueron ubicados como se muestra en la figura 5.2. La prueba fue realizada en un emplazamiento diferente al planteado, rodeado de obstáculos y con presencia de nubes, a fin de comprobar el funcionamiento de los diferentes componentes, determinar la capacidad de generación y el rendimiento del cargador.

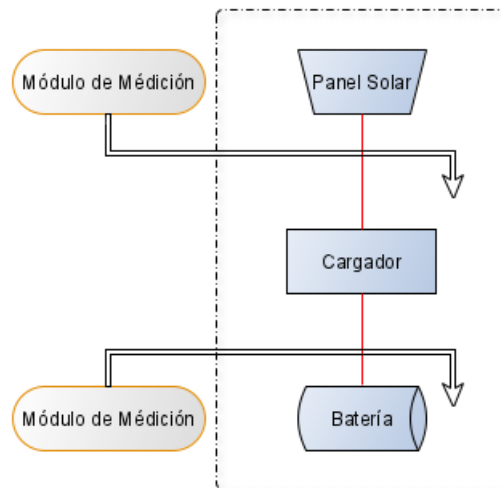


Figura 5.2: Esquema de medición empleados para medir corriente, voltaje y potencia del panel solar y la batería.

Los resultados de las mediciones se muestran en los siguientes gráficos:

En la figura 5.3 podemos apreciar el comportamiento de la tensión con respecto a la corriente en el panel solar, donde para tensiones mayores a 4 V (zona sombreada en verde) el cargador S09 V8 comienza a funcionar y vemos generación de corriente en el panel solar, luego ocurre lo contrario al final del día cuando la tensión baja de los 4 V haciendo que la corriente del panel solar sea cero, a través de la lectura de esta gráfica podemos determinar que efectivamente la entrada del cargador se encuentra funcionando.



Figura 5.3: Gráfico de la tensión y corriente en el panel solar en un periodo de 12 h.



Figura 5.4: Gráfico de la tensión en el panel solar y la corriente en la batería en un periodo de 12 h.

En la figura 5.4 podemos observar el comportamiento de la tensión en el panel solar con respecto a la corriente que se almacena en la batería, se observa que la tensión al superar los 4.6 V comienza a ingresar energía en la batería, lo que demuestra que de igual forma la salida del cargador se encuentra operativa.



Figura 5.5: Gráfico de la tensión en el panel solar y la tensión en la batería en un periodo de 12 h.

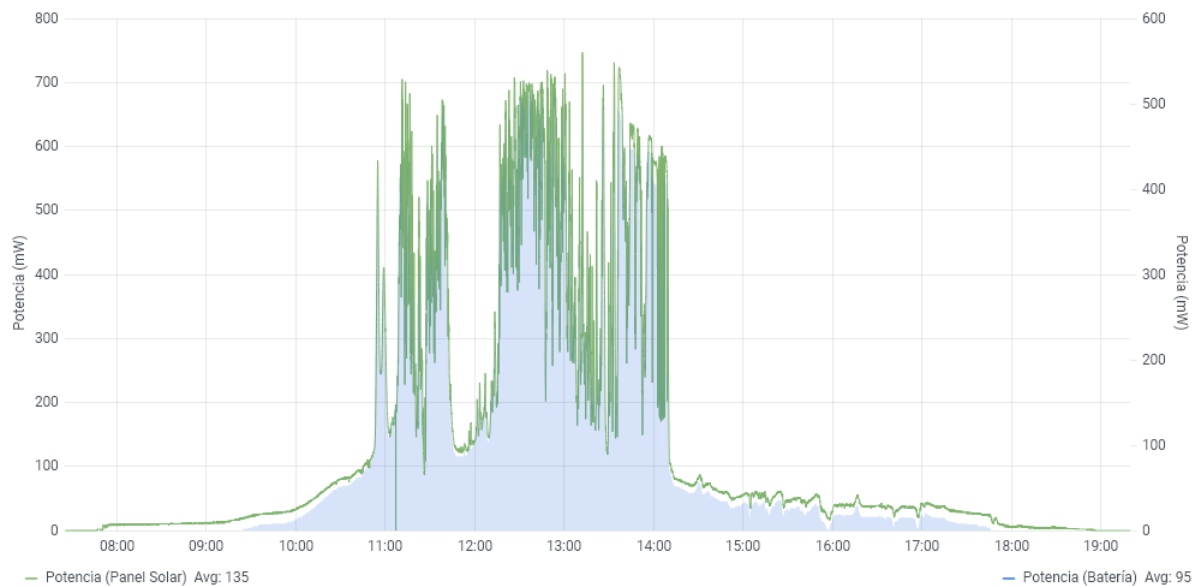


Figura 5.6: Gráfico de la potencia en el panel solar y en la batería en un periodo de 12 h.

En la figura 5.5 se observa como la tensión de la batería se eleva lentamente cuando la radiación solar en el panel es baja y se eleva rápidamente cuando la radiación incidente en el panel es alta. En los gráficos vemos como debido a la presencia de nubes ocurren fluctuaciones evidentes.

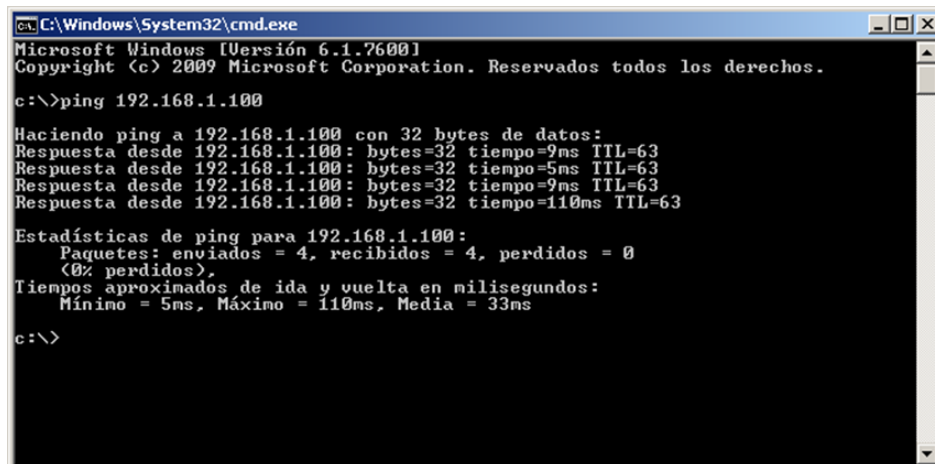
Por último, la figura 5.6 nos permite determinar el rendimiento del cargador, para ello empleamos la ecuación 5.1.

$$Rendimiento_{cargador} = \frac{Potencia_{Batería}}{Potencia_{PanelSolar}} = 0,7037 \Rightarrow 70,37\% \quad (5.1)$$

El rendimiento presente en el cargador es de 70.37%, este resultado puede servir para posibles mejoras o sustitución por un cargador mejor en el prototipo.

5.1.2. Pruebas de conectividad del nodo IoT con la red

En esta prueba se demostró el estado de la conexión del nodo IoT con la red de área local inalámbrica empleando una herramienta de diagnóstico “ping.exe” (disponible en el sistema operativo Windows); esta aplicación realiza la comprobación del estado de los dispositivos conectados a la red con el envío de paquetes *Internet Control Message Protocol* (ICMP), con lo cual se observó que ciertamente el nodo IoT responde a las peticiones (ver figura 5.7).



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

c:\>ping 192.168.1.100

Haciendo ping a 192.168.1.100 con 32 bytes de datos:
Respuesta desde 192.168.1.100: bytes=32 tiempo=9ms TTL=63
Respuesta desde 192.168.1.100: bytes=32 tiempo=5ms TTL=63
Respuesta desde 192.168.1.100: bytes=32 tiempo=9ms TTL=63
Respuesta desde 192.168.1.100: bytes=32 tiempo=110ms TTL=63

Estadísticas de ping para 192.168.1.100:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 5ms, Máximo = 110ms, Media = 33ms

c:\>
```

Figura 5.7: Uso de la herramienta de diagnóstico “ping.exe” para verificar la conectividad del nodo IoT.

5.1.3. Pruebas de los sensores

Luego de comprobada la conectividad con la red, se ratificó el funcionamiento de los sensores, esto fue realizado por medio de la consola monitor serial de la herramienta Arduino IDE y la línea serial del ESP-12e en los pines UTX y URX, permitiendo la comunicación con el nodo IoT, usado con la configuración siguiente:

- Bits por segundo: 115200
- Bits dato: 8
- Paridad: ninguna
- Bits de parada: 1

De acuerdo a la programación B del nodo IoT, el ESP-12e debe imprimir los valores censados en la línea serial.

En la figura 5.8 podemos observar el funcionamiento de los sensores SHT20, VEML6075, INA219 y FC-38, donde satisfactoriamente son mostrados los parámetros medidos, específicamente temperatura, humedad, radiación UV (descompuesta en radiación ultravioleta I, ultravioleta A y ultravioleta B), detección de lluvia y tensión de la batería.

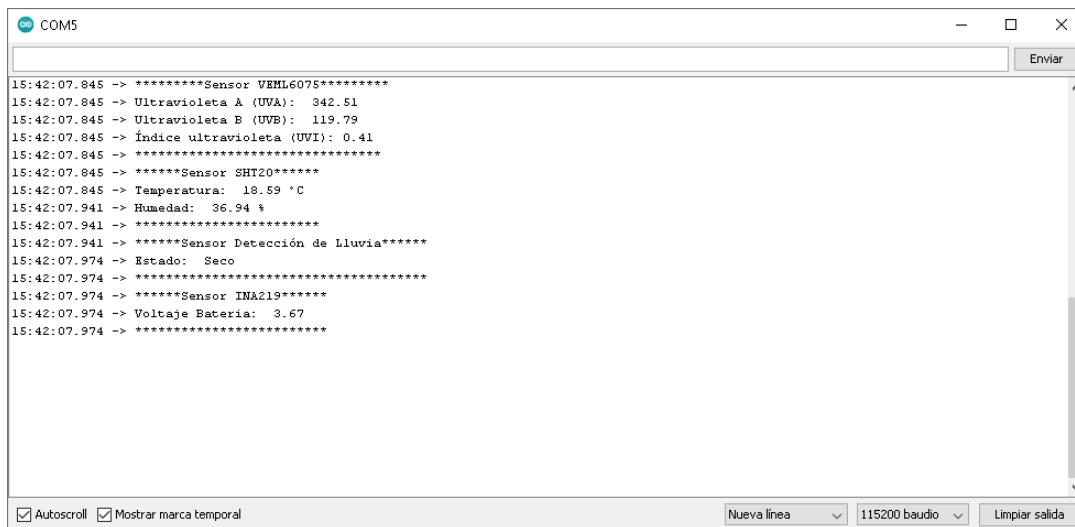
5.1.4. Pruebas del protocolo de comunicación MQTT y funcionamiento del Broker MQTT

Para esta prueba se verificó que el nodo IoT enviara correctamente el mensaje con las medidas censadas y el funcionamiento del Broker MQTT.

La primera prueba fue realizada por medio de la consola monitor serial, donde de acuerdo a la programación B del ESP-12e debe imprimir los datos en el puerto serial.

En la figura 5.9 podemos observar que los mensajes son publicados en el tópico correspondiente y que efectivamente contiene las medidas censadas.

Luego de que se sabe que los mensajes son enviados correctamente se procedió a probar el funcionamiento del Bróker, para esto fue empleado el software MQTTlens [33]

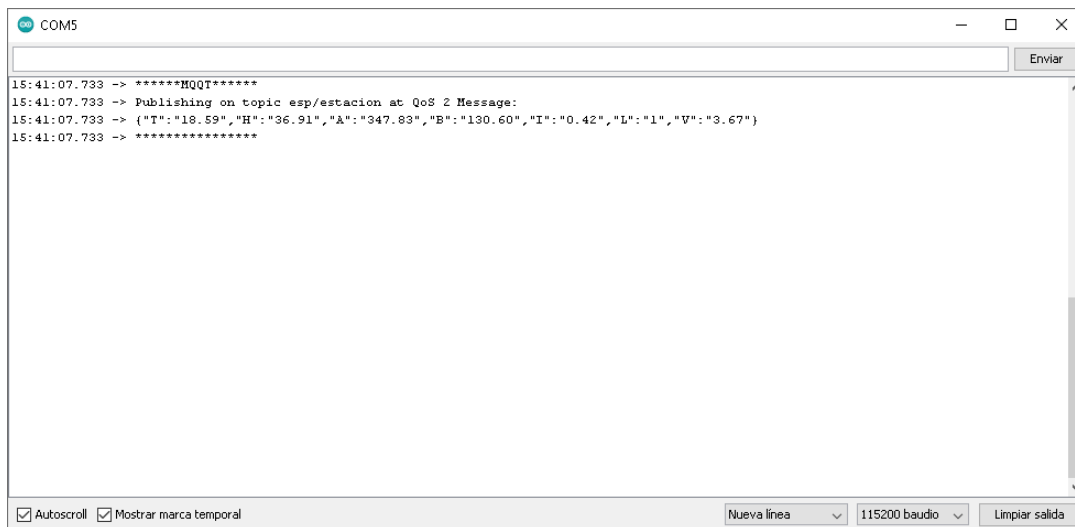


A screenshot of a serial terminal window titled 'COM5'. The window displays a series of sensor readings over time. The data is as follows:

```
15:42:07.845 -> *****Sensor VEML6075*****
15:42:07.845 -> Ultravioleta A (UVA): 342.51
15:42:07.845 -> Ultravioleta B (UVB): 119.79
15:42:07.845 -> Índice ultravioleta (UVI): 0.41
15:42:07.845 -> *****
15:42:07.845 -> *****Sensor SHT20*****
15:42:07.845 -> Temperatura: 18.59 °C
15:42:07.941 -> Humedad: 36.94 %
15:42:07.941 -> *****
15:42:07.941 -> *****Sensor Detección de Lluvia*****
15:42:07.974 -> Estado: Seco
15:42:07.974 -> *****
15:42:07.974 -> *****Sensor INA219*****
15:42:07.974 -> Voltaje Bateria: 3.67
15:42:07.974 -> *****
```

The terminal interface includes a text input field at the top right with an 'Enviar' button. At the bottom, there are checkboxes for 'Autoscroll' and 'Mostrar marca temporal', and a control bar with 'Nueva línea', '115200 baudio', and 'Limpiar salida' options.

Figura 5.8: Resultado de la prueba de funcionamiento de los sensores SHT20, VEML6075, INA219 y FC-38.



A screenshot of a serial terminal window titled 'COM5'. The window displays the output of an MQTT implementation. The data is as follows:

```
15:41:07.733 -> *****MQTT*****
15:41:07.733 -> Publishing on topic esp/estacion at QoS 2 Message:
15:41:07.733 -> {"T": "18.59", "H": "36.91", "A": "347.83", "B": "130.60", "I": "0.42", "L": "1", "V": "3.67"}
15:41:07.733 -> *****
```

The terminal interface includes a text input field at the top right with an 'Enviar' button. At the bottom, there are checkboxes for 'Autoscroll' and 'Mostrar marca temporal', and a control bar with 'Nueva línea', '115200 baudio', and 'Limpiar salida' options.

Figura 5.9: Resultado de la implementación del protocolo MQTT en el nodo IoT.

que permite subscribir al t pico y de esta forma verificar si los mensajes son aceptados y difundidos por el Br ker.

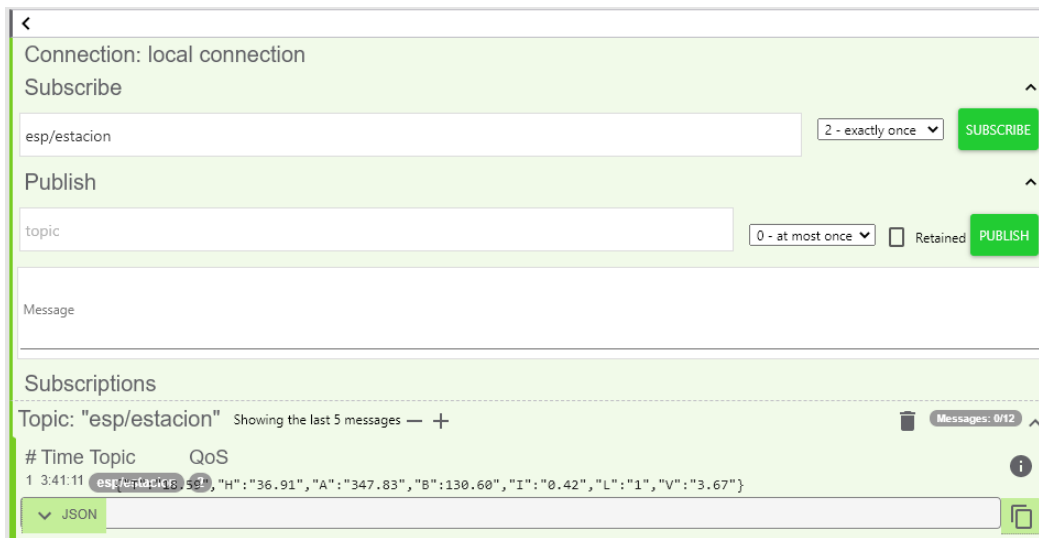


Figura 5.10: Resultados de la subscripci n al t pico “esp/estaci n”.

En la figura 5.10 se muestra como los mensajes son recibidos por la aplicaci n MQTTtens, lo que nos permite inferir en que el Broker se encuentra en funcionamiento.

5.1.5. Pruebas de consumo de corriente del prototipo

Esta prueba es realizada con la finalidad de tener una referencia que permita mejorar el consumo y funcionamiento del nodo IoT.

Una primera prueba consiste en poner en funcionamiento el prototipo usando la programaci n denominada funcionamiento normal (ver anexo B) y mediante un circuito de medici n como el empleado en el apartado 5.1.1 es medido el consumo el ctrico a la entrada del nodo IoT (lo que es lo mismo la corriente extra da de la bater a) como se puede ver en la figura 5.11, la prueba es repetida con la programaci n denominada funcionamiento optimizado (ver anexo A).

Los resultados se muestran a continuaci n:

Funcionamiento normal: En la figura 5.12 podemos observar los picos de corrientes

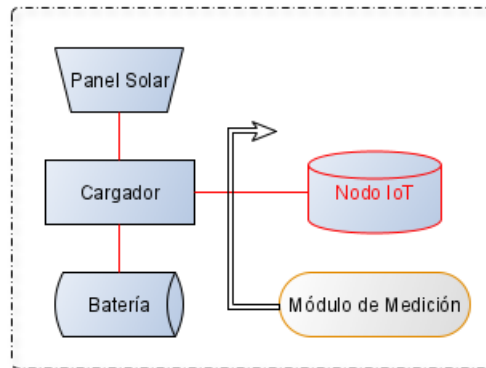


Figura 5.11: Esquema de medición para censar la corriente consumida por el nodo IoT.

enumerados que corresponden a la lectura y envío del mensaje MQTT a la red con frecuencia de 60 segundos aproximadamente, los picos de corriente restantes corresponden a procesos que el prototipo permanece constantemente ejecutando y pueden estar asociados a la verificación que la pila TCP/IP del protocolo de comunicación. Puesto que el prototipo realiza procesos relativamente cíclicos, podemos asumir que el promedio de corriente consumida por el nodo IoT corresponde al medido en esa muestra de tiempo, que es de 31.03 mA.

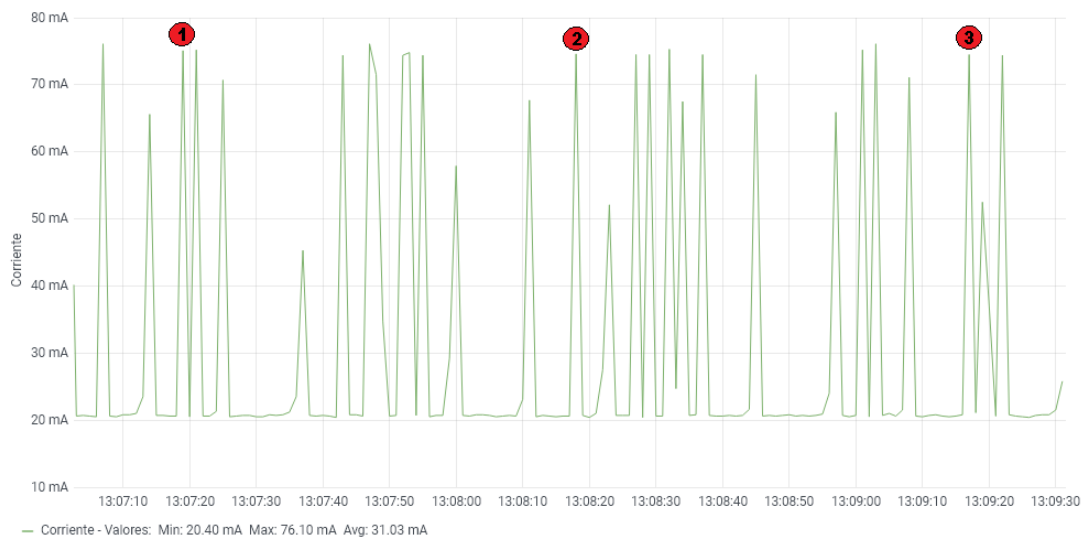


Figura 5.12: Gráfico de la corriente extraída de la batería con funcionamiento normal del nodo IoT con frecuencia de muestreo de 60 s en un periodo de 3 min.

Funcionamiento optimizado: En las figuras 5.13, 5.14 y 5.15 se observan los gráficos para cada frecuencia de muestreo y envío de datos de acuerdo al nivel de tensión de la batería, planteado en el funcionamiento optimizado del nodo IoT.



Figura 5.13: Gráfico de la corriente extraída de la batería con funcionamiento optimizado del nodo IoT con frecuencia de muestreo de 60 s en un periodo de 2 min.

Cómo podemos observar en las figuras 5.13, 5.14 y 5.15 el dispositivo entra correctamente en reposo (fig. 5.13: 60 s, fig. 5.14: 240 s y fig. 5.15: 480 s) y observamos un consumo de corriente que oscila entre los 4 y 6 mA, luego observamos los pulsos donde el dispositivo realiza la lectura de los parámetros censados, activa el módulo WiFi, envía el mensaje MQTT y desactiva el módulo WiFi para nuevamente entrar en reposo. En la tabla 5.1 podemos observar una comparativa del promedio de corriente de acuerdo al funcionamiento y frecuencia de muestreo del nodo IoT.

Estos resultados demuestran que es posible optimizar la programación para hacer que el dispositivo sea eficiente en el uso de la energía, por medio del control de los procesos y haciendo uso de la energía cuando realmente se realiza un proceso. Vemos cómo es posible reducir el consumo con la misma frecuencia de muestreo en un 63.33 %, lo que implica un aumento de la autonomía de la batería.



Figura 5.14: Gráfico de la corriente extraída de la batería con funcionamiento optimizado del nodo IoT con frecuencia de muestreo de 240 s en un periodo de 10 min.

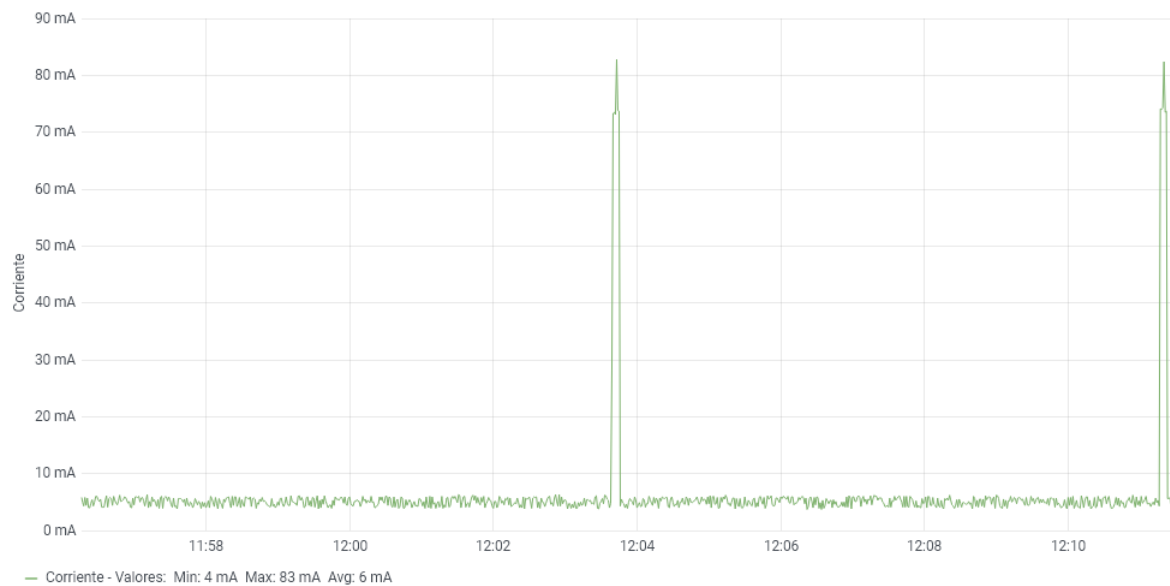


Figura 5.15: Gráfico de la corriente extraída de la batería con funcionamiento optimizado del nodo IoT con frecuencia de muestreo de 480 s en un periodo de 16 min.

Tabla 5.1: Promedio de corriente consumida por el prototipo de acuerdo al funcionamiento y frecuencia de muestreo.

Frecuencia de muestreo (s)	<i>Consumo de corriente Promedio</i>	
	Funcionamiento normal (mA)	Funcionamiento Optimizado (mA)
60	30	11
240	-	9
480	-	6

5.1.6. Pruebas de autonomía del nodo IoT

Se realizaron dos test para determinar la autonomía de la batería de acuerdo a los dos funcionamientos del nodo IoT.

Para determinar la autonomía del nodo IoT fue necesario desconectar el sistema fotovoltaico, y con la batería cargada al máximo se procedió a poner al nodo IoT en marcha hasta que la carga de la batería fuese insuficiente para mantener en funcionamiento el circuito del nodo IoT y por lo tanto este no pueda continuar funcionando, para ello hemos empleado las medidas obtenidas por sensor INA219 del nodo IoT que censa el voltaje de la batería.

Los resultados se muestran a continuación:

Funcionamiento normal:

En la figura 5.16 se muestran las gráficas correspondientes a las medidas de voltaje tomadas por el sensor INA219 del nodo IoT en funcionamiento normal, mediante la observación directa hemos determinado la autonomía del nodo IoT que corresponde a 92 horas, los porcentajes de descarga de la batería se representan en la tabla 5.2, estos valores fueron hallados al dividir el tiempo de autonomía en cuatro partes iguales y cada parte de tiempo interceptado con la curva de voltaje.

Conocidos los valores de autonomía y corriente consumida por el prototipo (tabla 5.1) en funcionamiento normal podemos determinar la capacidad de la batería de la que hacemos uso, esto lo realizamos de acuerdo a la ecuación 5.2.

$$Capacidad_{Prototipo} = \frac{Autonomía_{Batería}}{Corriente_{Func.Normal}} = \frac{92h}{30mA} = 3067mAh \quad (5.2)$$

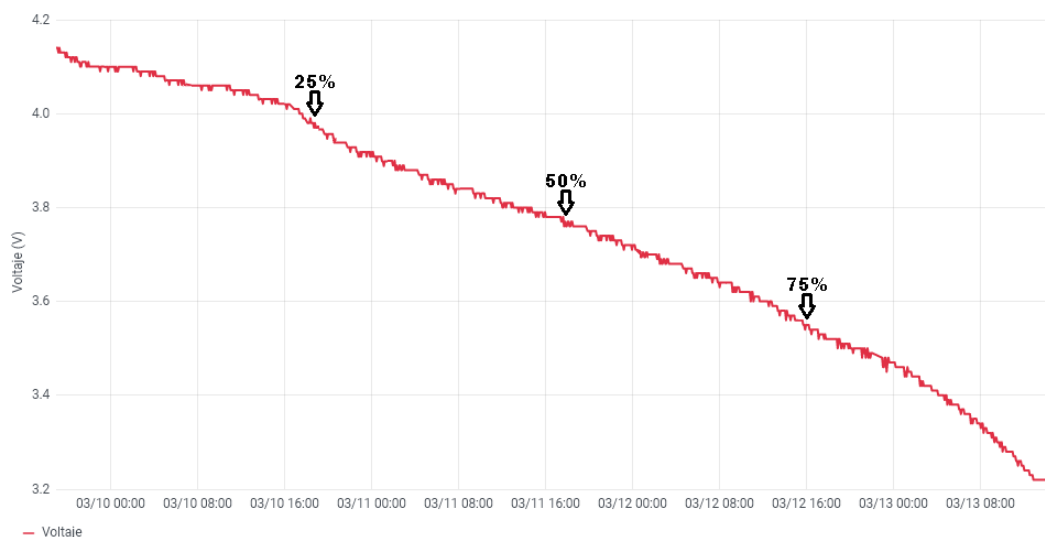


Figura 5.16: Gráfico de voltaje en la batería en el tiempo con funcionamiento normal del nodo IoT.

Tabla 5.2: Porcentaje de descarga de la batería en el tiempo.

Porcentaje de descarga (%)	Tiempo (h)	Voltaje (V)
0	0	4.13
25	23	3.98
50	46	3.77
75	69	3.55
100	92	3.23

La batería tiene una capacidad total de 3400 mAh, de los cuales hacemos uso el 90 % de su capacidad.

Funcionamiento optimizado:

En la figura 5.17 observamos que al cumplir las noventa y dos horas de funcionamiento el nodo IoT continúa funcionando con una tensión en la batería de 3.9 V (de acuerdo a la tabla 5.2 este valor corresponde al 35 %)

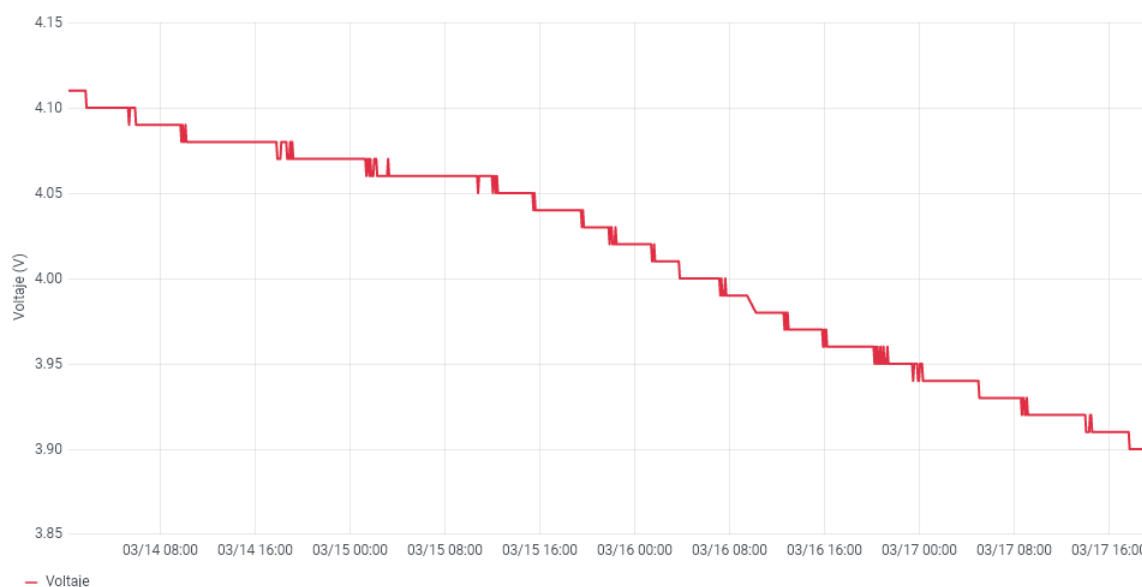


Figura 5.17: Gráfico de voltaje en la batería en el tiempo con funcionamiento optimizado del nodo IoT en un periodo de 92 h.

Mediante la capacidad de la batería del prototipo y la corriente consumida por el prototipo optimizado para un muestreo de 60s, 240s y 480 s es posible determinar el tiempo en el cual el dispositivo puede permanecer funcionando. En la tabla 5.3 se muestran los tiempos en que el dispositivo puede permanecer funcionando de acuerdo al tiempo de muestreo.

5.1.7. Pruebas de carga y descarga

Esta prueba consiste en poner en funcionamiento el prototipo, y mediante el INA219 medimos la tensión de la batería para observar el comportamiento de la tensión cuando

Tabla 5.3: Autonomía del prototipo de acuerdo a la frecuencia de muestreo con funcionamiento optimizado.

Frecuencia de muestreo (s)	Corriente consumida (mA)	Autonomía del prototipo (h)
60	11	278.81
240	9	340.77
480	6	511.16

el sistema se encuentra bajo exposición solar (carga) y cuando se descarga en horas de la noche.

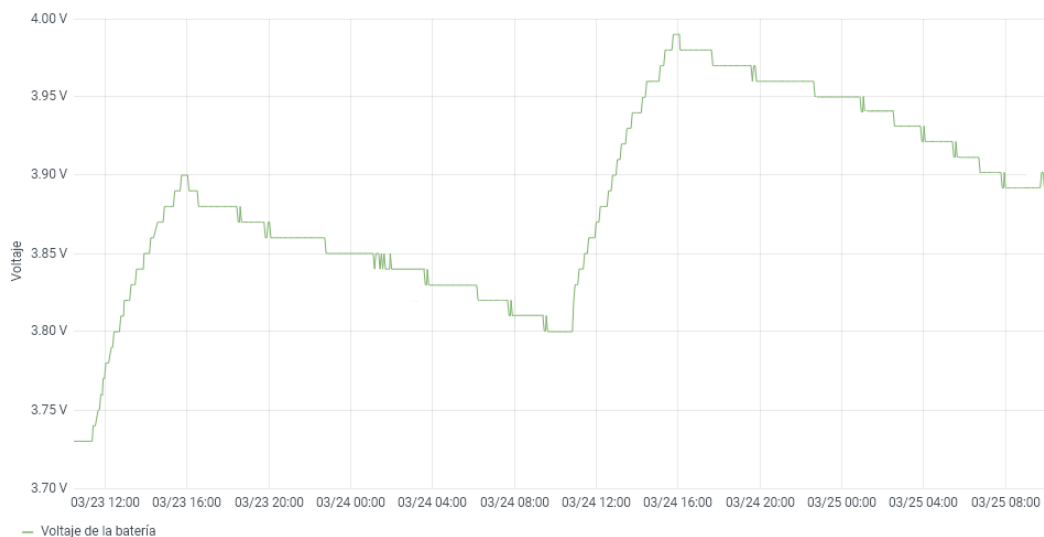


Figura 5.18: Gráfico de voltaje en la batería en el tiempo con funcionamiento optimizado del nodo IoT en un periodo de 42 h.

Como vemos en la figura 5.18 el sistema fotovoltaico permite almacenar un porcentaje de carga superior al requerido por el prototipo.

5.1.8. Modelo físico de la caja de montaje para la protección del prototipo

El modelo logrado de la caja que protege al prototipo se presenta en la figura 5.19. La caja protectora fue elaborada con filamento de poliácido láctico (PLA) con la ayuda de una impresora 3D en color blanco de dimensiones 15.5X9X15cm lo que proporciona

mayor comodidad para su instalación al ser un equipo compacto. En la figura 5.20 se puede observar el prototipo dispuesto en el interior de la caja de montaje, así como la distribución de los diferentes sensores y la ubicación del panel solar. Todos los componentes internamente se encuentran resguardados de agentes externos y de agentes climáticos.

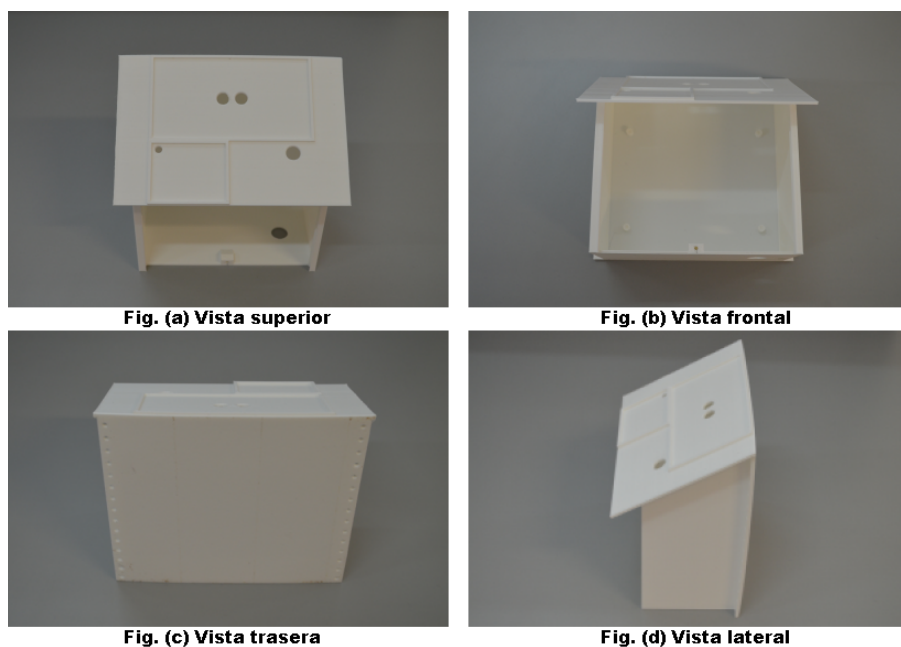


Figura 5.19: Resultado de la caja de montaje que protege al prototipo.

5.2. Prototipo del sistema de monitoreo

En este apartado se discuten las pruebas realizadas al editor de flujo, la base de dato y al sistema de visualización.

5.2.1. Pruebas del editor de flujo

Para verificar el funcionamiento y la correcta implementación del editor de flujo, se utilizó un módulo llamado *debug* disponible en la interfaz Node-RED, como se muestran en la figura 5.21, que permite verificar la información recibida en cada bloque implementado. En esta prueba se ha puesto en funcionamiento el nodo IoT y el Broker.

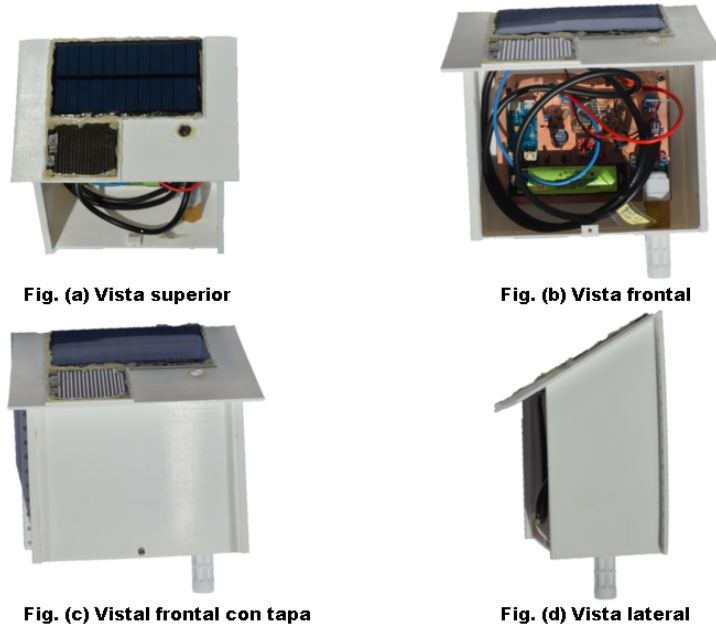


Figura 5.20: Prototipo dispuesto en el interior de la caja de montaje.

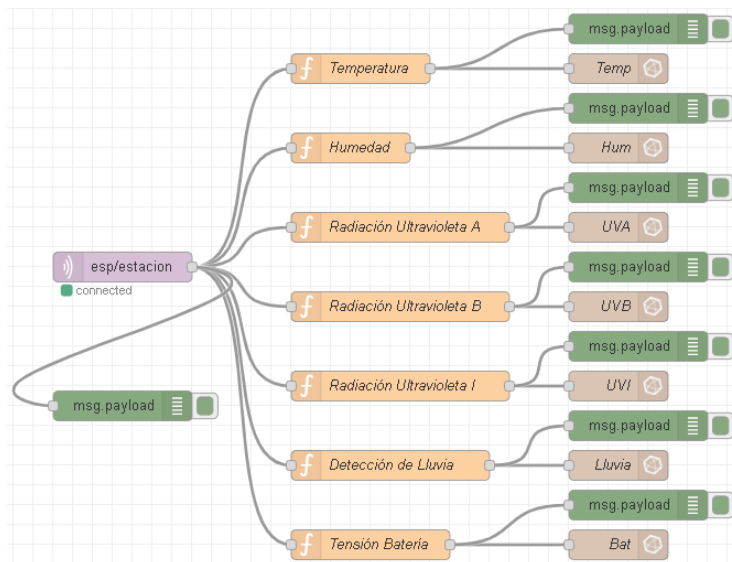


Figura 5.21: Prueba de funcionamiento y la correcta implementación del editor de flujo.

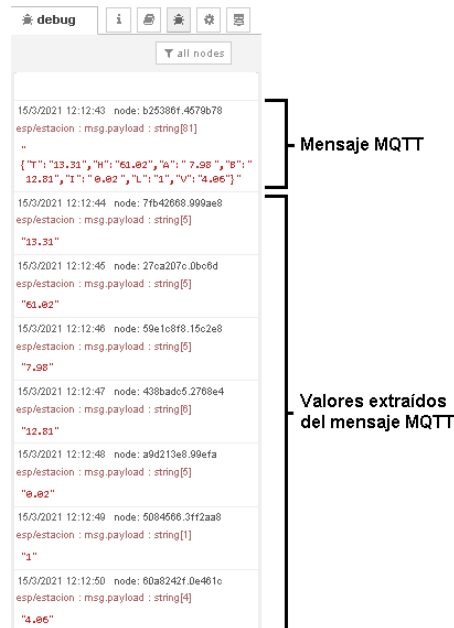


Figura 5.22: Resultados obtenidos del debug.

En la figura 5.22 podemos observar la información que es recibida en cada bloque del espacio de trabajo, el primer valor recibido corresponde al mensaje MQTT (emitido por el nodo IoT) y los valores siguientes corresponden a los valores extraídos del bloque de entrada que corresponden a cada parámetro medido.

5.2.2. Pruebas de la base de datos

La prueba de funcionamiento de la base de datos, consiste en verificar que los datos son recibidos y almacenados en la base de datos, para ello fue invocado el software de la base de datos y ejecutados los comandos mostrados en la figura 5.23, para verificar que las variables de las medidas son creadas satisfactoriamente dentro de la base de datos “ESTACIONIPB” y que dentro de las variables contiene las medidas censadas por los diferentes sensores.

En la figura 5.23, podemos observar que el conjunto de variables que almacenan las medidas correspondientes a los parámetros medidos son almacenados correctamente en la base de datos, así como las medidas de la variable “Bat” de la tensión de la batería.

```

Selecionar C:\Program Files\InfluxData\influx-data\influx.exe
Connected to http://localhost:8086 version 1.8.3
InfluxDB shell version: 1.8.3
> use ESTACIONIPB
Using database ESTACIONIPB
> show measurements
name: measurements
----
Bat
Hum
Temp
Hum
Lluvia
UVA
UVB
UVI
> select * from Bat
name: Bat
----
time                value
-----
1615315764160450000 4.16
1615315765957788100 4.16
1615315771201511500 4.16
1615315827030120100 4.14
1615315833759694300 4.14
1615315840508051800 4.14
>

```

Figura 5.23: Prueba de funcionamiento de la base de datos.

5.2.3. Pruebas al sistema de visualización

En la figura 5.24, se muestra la interfaz gráfica desarrollada, donde podemos observar la correcta implementación. La interfaz gráfica en la parte superior indica el valor actual del parámetro meteorológico medido, mientras que en la parte de abajo se muestran un conjunto de gráficos que contienen el historial de los parámetros en el tiempo.



Figura 5.24: Interfaz gráfica desarrollada para el monitoreo de las condiciones meteorológicas.

Capítulo 6

Conclusiones y Recomendaciones

Al finalizar el proyecto de investigación se puede satisfactoriamente concluir que:

- El uso e implementación de tecnologías basadas en internet de las cosas constituyen una posibilidad muy amplia para proporcionar soluciones alternativas a necesidades particulares en donde la autonomía de los equipos es esencial para el uso de los dispositivos empleados.
- El IoT es un tipo de tecnología amigable y permite trabajar con fuentes de energía alternativas haciendo posible su implementación en lugares muy apartados, en los cuales no existe acceso a la red eléctrica.
- La tecnología IoT resulta muy eficiente en cuanto a la ejecución de tareas en las cuales la interacción con el usuario debe ser reducido
- IoT es una tecnología cuyo mantenimiento es mínimo.
- El equipo diseñado y desarrollado como prototipo es compacto, de fácil instalación y los dispositivos utilizados pueden ser económicos.
- Debido a que el nodo IoT puede ser programado en función de las exigencias requeridas, este puede estar en funcionamiento solo cuando es preciso ejecutar una

acción y puede ser desconectado cuando no es preciso ejecutar una tarea. Debido a esta característica resulta energéticamente eficiente, lo que conlleva a aumentar la autonomía del modelo.

- El prototipo diseñado es capaz de monitorear las variables meteorológicas determinadas para el diseño de manera eficaz.
- La autonomía del aparato está garantizada por un sistema fotovoltaico con batería.
- El prototipo logrado cuenta con un sistema de recolección, visualización y monitoreo de los parámetros medidos por el nodo IoT que funciona de manera óptima. Se recomienda para futuros trabajos:
 - Evaluar con mayor detenimiento los componentes disponibles en el mercado a fin de procurar mejores alternativas en cuanto a calidad, costo y eficiencia en el consumo eléctrico.
 - Sustituir el cargador empleado por uno con una eficiencia superior al 70 %, para lograr un mayor aprovechamiento de la energía generada por el panel solar.
 - El nodo empleado en el proyecto puede ser un precedente importante para el desarrollo de un proyecto futuro para la automatización del laboratorio en el contexto de edificios inteligentes, donde las variables medidas en el exterior junto con las variables del entorno interno (medidas por otros nodos) debe alimentar a uno responsable de monitorear y controlar diversas condiciones del ambiente interno del laboratorio, incluidos los aspectos de ahorro de energía.
 - Desarrollar y probar otras estrategias para determinar las condiciones operativas del nodo que puedan contribuir a una autonomía aún mayor sin afectar los requisitos funcionales de la aplicación.
 - Probar y comparar los pro y contras de otros protocolos diferentes a los empleados en la comunicación (WiFi) y la transmisión de datos (MQTT).

Bibliografía

- [1] B. Raj, J. Singh, S. K. Vishvakarma y S. S. Chouhan, “IoT-based ambient intelligence microcontroller for remote temperature monitoring,” en *Guide to Ambient Intelligence in the IoT Environment*, Springer, 2019, págs. 177-200.
- [2] S. Kim e Y.-I. Yoon, “Ambient intelligence middleware architecture based on awareness-cognition framework,” *Journal Of Ambient Intelligence And Humanized Computing*, vol. 9, n.º 4, págs. 1131-1139, 2018.
- [3] C. Siu y K. Iniewski, “The internet of things—Physical and link layers overview,” *IoT and Low-Power Wireless: Circuits, Architectures, and Techniques*, pág. 1, 2018.
- [4] M. Silverio-Fernández, S. Renukappa y S. Suresh, “What is a smart device?-a conceptualisation within the paradigm of the internet of things,” *Visualization in Engineering*, vol. 6, n.º 1, págs. 1-10, 2018.
- [5] L. G. C. Ramirez, G. S. A. Jiménez y J. M. Carreño, *Sensores y actuadores*. Grupo Editorial Patria, 2014.
- [6] S. de Oliveira, *Internet das coisas com ESP8266, Arduino e Raspberry PI*. Novatec Editora, 2017.
- [7] I. R. Martins y J. L. Zem, “Estudo dos protocolos de comunicação MQTT e COaP para aplicações machine-to-machine e Internet das coisas,” *Revista Tecnológica da Fatec Americana*, vol. 3, n.º 1, 24p-24p, 2015.

- [8] J. T. Pinho y M. A. Galdino, “Manual de engenharia para sistemas fotovoltaicos,” *Rio de Janeiro*, vol. 1, págs. 47-499, 2014.
- [9] M. G. Villalva y J. R. Gazoli, “Energia solar fotovoltaica: conceitos e aplicações,” *São Paulo: Érica*, vol. 2, 2012.
- [10] Módulo Wi-Fi ESP-12e. dirección: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTNN3MzdnjwmvWdk41ZqbNJwdJzEBzZ6EuMW2awpAkWjY7kwuJTGiCERxDBjUsqp=CAC>, Consultado: Noviembre, 2020.
- [11] Regulador de Tensión S09 V8.0. dirección: <https://ae01.alicdn.com/kf/HTB1cytYRCzqK1RjSZPqxq6A4tVXaf/S09-DC-DC-Automatic-Buck-Boost-Power-Module-Step-Up-and-Down-Board-Input-3V-15V.jpg>, Consultado: Noviembre, 2020.
- [12] Sensor digital VEML6075. dirección: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRnw2jGdLKc_k5qtW6tZVGpP7vMpiqI8Yfq7w&usqp=CAU, Consultado: Diciembre, 2020.
- [13] Sensor Detección de Lluvia. dirección: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcToh86xzsDihOG_hcgfZ9PoNz4xZpn4tskQtyMm9V0CwBnfC3HYhCoAUUsqp=CAC, Consultado: Enero, 2021.
- [14] Sensor de Corriente Eléctrica modelo INA219. dirección: https://www.botnroll.com/10454-medium_default/m-dulo-sensor-corrente-ina219-26v-dc-3-2a-max.jpg, Consultado: Enero, 2021.
- [15] Software Arduino Integrated Development Enviroment. dirección: <https://www.arduino.cc/en/guide/environment>, Consultado: Octubre, 2020.
- [16] Solar Charger V1.0. dirección: <https://encrypted-tbn0.gstatic.com/shopping?q=tbn:ANd9GcSaxMR36KY-NzeKm6zVHh8wWlCjn8Uk5fVuUPrgorXlSxMh2uCuxaA9CnUjS4iyw&usqp=CAC>, Consultado: Noviembre, 2020.
- [17] InfluxDB, software base de datos de series tiempos. dirección: <https://www.influxdata.com/products/influxdb/>, Consultado: Diciembre, 2020.

- [18] Mosquito, software servidor de mensajes de código abierto que implementa el protocolo MQTT. dirección: <https://mosquitto.org/download/>, Consultado: Noviembre, 2020.
- [19] Node-RED, software editor de flujo. dirección: <https://nodered.org/>, Consultado: Noviembre, 2020.
- [20] Grafana, software para análisis y supervisión. dirección: <https://grafana.com/get/>, Consultado: Noviembre, 2020.
- [21] Software PVGIS. dirección: https://re.jrc.ec.europa.eu/pvg_tools/es/#MR, Consultado: Enero, 2021.
- [22] "Escola Superior de Tecnologia e Gestão de Bragança -Instituto Politécnico de Bragança". Google Maps. Google. dirección: <https://www.google.com/maps/@41.7963915,-6.7699488,1299m/data=!3m1!1e3>, Consultado: Noviembre, 2020.
- [23] EAGLE, Software para el desarrollo de diagramas de circuitos electricos y PCB. dirección: <https://www.autodesk.com/products/eagle>, Consultado: Noviembre, 2020.
- [24] Kit de desarrollo NodeMCU V1. dirección: <https://github.com/nodemcu/nodemcu-devkit-v1.0>, Consultado: Octubre, 2020.
- [25] Libreria ESP8266WiFi. dirección: <https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi>, Consultado: Octubre, 2020.
- [26] Libreria DFRobot_{SHT20}, *para el control del sensor SHT20*. dirección: https://github.com/DFRobot/DFRobot_SHT20, Consultado: Diciembre, 2020.
- [27] Libreria Adafruit_{INA219}, *para el control del sensor INA219*. dirección: https://github.com/adafruit/Adafruit_INA219/blob/master/Adafruit_INA219.h, Consultado: Enero, 2021.
- [28] Libreria AsyncMqttClient. dirección: <https://platformio.org/lib/show/346/AsyncMqttClient>, Consultado: Noviembre, 2020.

- [29] Software FreeCAD. dirección: <https://www.freecadweb.org/>, Consultado: Febrero, 2021.
- [30] Software InfluxDB, instalación en el sistema operativo Windows. dirección: <https://www.learningmilestone.com/post/steps-to-setup-influxdb-on-windows>, Consultado: Noviembre, 2020.
- [31] Software Node-RED, instalación en el sistema operativo Windows. dirección: <https://nodered.org/docs/getting-started/windows#2-install-node-red>, Consultado: Noviembre, 2020.
- [32] Software Grafana, instalación en el sistema operativo Windows. dirección: <https://grafana.com/docs/grafana/latest/installation/windows/>, Consultado: Noviembre, 2020.
- [33] Software MQTTlens. dirección: <https://chrome.google.com/webstore/detail/mqttlens/hemojaaeigabkbcookmlgmdigohjobjm?hl=pt-BR&gl=US>, Consultado: Enero, 2021.

Apéndice A

Propuesta del nodo IoT optimizado

```
1 #include <ArduinoJson.h>
2 #include "DFRobot_SHT20.h"
3 #include <Adafruit_VEML6075.h>
4 #include <ESP8266WiFi.h>
5 #include <AsyncMqttClient.h>
6 #include <Adafruit_INA219.h>
7
8 char JSONmessageBuffer[100];
9 float hum;
10 float temp;
11 float uva;
12 float uvb;
13 float uvi;
14 int lluvia;
15 float volt;
16
17 #define MQTT_PORT 1883
18
19 #define W_SSID "SSID nombre"
```

```
20 #define W_PASSWORD "WiFi contrasena"
21
22 #define MQTT_HOST IPAddress(192, 168, 1, 200)
23
24 #define MQTT_PUB_BUFFER "esp/estacion"
25
26 AsyncMqttClient mqttClient;
27 Adafruit_VEML6075 uv = Adafruit_VEML6075();
28 DFRobot_SHT20 sht20;
29 Adafruit_INA219 ina219(0x41);
30
31 void connectToWifi() {
32     WiFi.forceSleepWake();
33     delay( 5 );
34     WiFi.persistent( false );
35     WiFi.mode( WIFI_STA );
36     WiFi.begin(W_SSID, W_PASSWORD);
37
38     while(WiFi.status() != WL_CONNECTED){
39         delay(500);
40     }
41 }
42
43 void connectToMqtt() {
44     mqttClient.setServer(MQTT_HOST, MQTT_PORT);
45     mqttClient.connect();
46     while(!mqttClient.connected()){
47         delay(500);
48     }
49 }
50
51 String readTemperature() {
52     temp = sht20.readTemperature();
53     return String(temp);
54 }
```

```
55
56 String readHumidity() {
57     hum = sht20.readHumidity();
58     return String(hum);
59 }
60
61 String readlluvia() {
62     lluvia = digitalRead(12);
63     return String(lluvia);
64 }
65
66 String readUVA() {
67     uva = uv.readUVA();
68     return String(uva);
69 }
70 String readUVB() {
71     uvb = uv.readUVB();
72     return String(uvb);
73 }
74 String readUVI() {
75     uvi = uv.readUVI();
76     return String(uvi);
77 }
78 String readvoltaje(){
79     volt = ina219.getBusVoltage_V();
80     return String(volt);
81 }
82
83 void setup(){
84     WiFi.mode( WIFI_OFF );
85     WiFi.forceSleepBegin();
86     delay( 50 );
87
88     pinMode(12, INPUT);
89     digitalWrite(12, LOW);
```

```
90
91  uv.begin();
92  sht20.initSHT20();
93  ina219.begin();
94  ina219.setCalibration_16V_400mA();
95 }
96
97 void loop(){
98   if (ina219.getBusVoltage_V() >= 4) {
99
100    StaticJsonBuffer<300> JSONbuffer;
101    JsonObject& JSONencoder = JSONbuffer.createObject();
102
103    JSONencoder["T"] = readTemperature();
104    JSONencoder["H"] = readHumidity();
105    JSONencoder["A"] = readUVA();
106    JSONencoder["B"] = readUVB();
107    JSONencoder["I"] = readUVI();
108    JSONencoder["L"] = readlluvia();
109    JSONencoder["V"] = readvoltaje();
110
111    JSONencoder.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer)
112      );
113    //Serial.println(JSONmessageBuffer);
114
115    connectToWifi();
116    delay(50);
117    connectToMqtt();
118    delay(50);
119    mqttClient.publish(MQTT_PUB_BUFFER, 1, true, JSONmessageBuffer);
120    delay(50);
121    mqttClient.disconnect();
122    delay(50);
123    WiFi.disconnect(true);
124    delay(50);
```

```
124 ESP.deepSleep(60000000,WAKE_RF_DEFAULT);
125 delay(100);
126 }
127 else if (ina219.getBusVoltage_V() >= 3.7 & ina219.getBusVoltage_V
    () < 4){
128
129 StaticJsonBuffer<300> JSONbuffer;
130 JsonObject& JSONencoder = JSONbuffer.createObject();
131
132 JSONencoder["T"] = readTemperature();
133 JSONencoder["H"] = readHumidity();
134 JSONencoder["A"] = readUVA();
135 JSONencoder["B"] = readUVB();
136 JSONencoder["I"] = readUVI();
137 JSONencoder["L"] = readlluvia();
138 JSONencoder["V"] = readvoltaje();
139
140 JSONencoder.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer)
    );
141
142 connectToWifi();
143 delay(50);
144 connectToMqtt();
145 delay(50);
146 mqttClient.publish(MQTT_PUB_BUFFER, 2, true, JSONmessageBuffer);
147 delay(50);
148 mqttClient.disconnect();
149 delay(50);
150 WiFi.disconnect(true);
151 delay(50);
152 ESP.deepSleep(240000000,WAKE_RF_DEFAULT);
153 delay(100);
154
155 }
156 else if (ina219.getBusVoltage_V() < 3.7) {
```

```
157
158   StaticJsonBuffer<300> JSONbuffer;
159   JsonObject& JSONencoder = JSONbuffer.createObject();
160
161   JSONencoder["T"] = readTemperature();
162   JSONencoder["H"] = readHumidity();
163   JSONencoder["A"] = readUVA();
164   JSONencoder["B"] = readUVB();
165   JSONencoder["I"] = readUVI();
166   JSONencoder["L"] = readlluvia();
167   JSONencoder["V"] = readvoltaje();
168
169   JSONencoder.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer)
170   );
171   connectToWifi();
172   delay(50);
173   connectToMqtt();
174   delay(50);
175   mqttClient.publish(MQTT_PUB_BUFFER, 2, true, JSONmessageBuffer);
176   delay(50);
177   mqttClient.disconnect();
178   delay(50);
179   WiFi.disconnect(true);
180   delay(50);
181   ESP.deepSleep(480000000, WAKE_RF_DEFAULT);
182   delay(100);
183 }
184 }
```

Listing A.1: Programación del nodo IoT optimizado

Apéndice B

Propuesta del nodo IoT normal

```
1 #include <ArduinoJson.h>
2 #include "DFRobot_SHT20.h"
3 #include <Adafruit_VEML6075.h>
4 #include <ESP8266WiFi.h>
5 #include <AsyncMqttClient.h>
6 #include <Adafruit_INA219.h>
7
8 char JSONmessageBuffer[100];
9 float hum;
10 float temp;
11 float uva;
12 float uvb;
13 float uvi;
14 int lluvia;
15 String lluviaa;
16 float volt;
17
18 #define MQTT_PORT 1883
19
20 #define W_SSID "SSID nombre"
21 #define W_PASSWORD "WiFi contrasena"
22 #define MQTT_HOST IPAddress(192, 168, 1, 200)
```

```
23
24 #define MQTT_PUB_BUFFER "esp/estacion"
25
26
27
28 AsyncMqttClient mqttClient;
29 Adafruit_VEML6075 uv = Adafruit_VEML6075();
30 DFRobot_SHT20 sht20;
31 Adafruit_INA219 ina219(0x41);
32
33 void connectToWifi() {
34     WiFi.mode( WIFI_STA );
35     Serial.println("conectando");
36     WiFi.begin(W_SSID, W_PASSWORD);
37     while(WiFi.status() != WL_CONNECTED){
38         delay(500);
39         Serial.println("C.");
40     }
41 }
42
43 void connectToMqtt() {
44     mqttClient.setServer(MQTT_HOST, MQTT_PORT);
45     mqttClient.connect();
46     Serial.println("MQTT");
47     while(!mqttClient.connected()){
48         delay(500);
49         Serial.println("M.");
50     }
51 }
52
53 String readTemperature() {
54     temp = sht20.readTemperature();
55     return String(temp);
56 }
57
```

```
58 String readHumidity() {
59     hum = sht20.readHumidity();
60     return String(hum);
61 }
62
63 String readlluvia() {
64     lluvia = digitalRead(12);
65     return String(lluvia);
66 }
67
68 String readlluviaa() {
69     lluviaa;
70     lluvia = digitalRead(12);
71     if (lluvia = 1){
72         lluviaa = "Seco";
73     }
74     else {
75         lluviaa = "Llueve";
76     }
77     return String(lluviaa);
78 }
79
80 String readUVA() {
81     uva = uv.readUVA();
82     return String(uva);
83 }
84 String readUVB() {
85     uvb = uv.readUVB();
86     return String(uvb);
87 }
88 String readUVI() {
89     uvi = uv.readUVI();
90     return String(uvi);
91 }
92 String readvoltaje(){
```

```
93  volt = ina219.getBusVoltage_V();
94  return String(volt);
95  }
96
97
98  void setup(){
99      Serial.begin(115200);
100     Serial.println();
101     pinMode(12, INPUT);
102     digitalWrite(12, LOW);
103     connectToWifi();
104     connectToMqtt();
105
106     uv.begin();
107     sht20.initSHT20();
108     ina219.begin();
109     ina219.setCalibration_16V_400mA();
110 }
111
112 void loop(){
113
114     StaticJsonBuffer<300> JSONbuffer;
115     JsonObject& JSONencoder = JSONbuffer.createObject();
116
117     JSONencoder["T"] = readTemperature();
118     JSONencoder["H"] = readHumidity();
119     JSONencoder["A"] = readUVA();
120     JSONencoder["B"] = readUVB();
121     JSONencoder["I"] = readUVI();
122     JSONencoder["L"] = readlluvia();
123     JSONencoder["V"] = readvoltaje();
124
125     JSONencoder.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer)
126     );
```

```

127 mqttClient.publish(MQTT_PUB_BUFFER, 2, true, JSONmessageBuffer);
128
129     Serial.println("*****Sensor VEML6075*****");
130     Serial.print("Ultravioleta A (UVA): "); Serial.println(uv.
readUVA());
131     Serial.print("Ultravioleta B (UVB): "); Serial.println(uv.
readUVB());
132     Serial.print("Indice ultravioleta (UVI): "); Serial.println(
uv.readUVI());
133     Serial.println("*****");
134
135     Serial.println("*****Sensor SHT20*****");
136     Serial.print("Temperatura: "); Serial.print(sht20.
readTemperature()); Serial.println(" C");
137     Serial.print("Humedad: "); Serial.print(sht20.readHumidity
()); Serial.println(" %");
138     Serial.println("*****");
139
140
141     Serial.println("*****Sensor Deteccion de Lluvia*****");
142     Serial.print("Estado: "); Serial.println(readlluviaa());
143     Serial.println("*****");
144
145     Serial.println("*****Sensor INA219*****");
146     Serial.print("Voltaje Bateria: "); Serial.println(
readvoltaje());
147     Serial.println("*****");
148
149     Serial.println("*****MQTT*****");
150     Serial.printf("Publishing on topic %s at QoS 2",
MQTT_PUB_BUFFER);
151     Serial.println("Message: %.2f \n", JSONmessageBuffer);
152     Serial.println("*****");
153
154

```

```
155     delay(60000);  
156 }
```

Listing B.1: Programación del nodo IoT normal