



INSTITUTO TECNOLÓGICO DE MÉRIDA

TESIS DE MAESTRÍA

Desarrollo de software embebido y de plataforma IoT para equipos de monitoreo de consumos de energía eléctrica

Para obtener el grado de Maestro en Ingeniería

Autor:

Ing. José Alfonso Ureña Pajón

Asesor:

Dr. Jesús Sandoval Gio

Mérida, Yucatán, México

Junio 2019



"2019, Año del Caudillo del Sur, Emiliano Zapata"

DEPENDENCIA: DIV. DE EST. DE POSG. E INV.

No. DE OFICIO: X-135/19

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

Mérida, Yucatán, **20 de mayo de 2019**

C. UREÑA PAJÓN JOSÉ ALFONSO
Pasante de la Maestría en Ingeniería
PRESENTE

De acuerdo al fallo emitido por su asesor el **Dr. Jesús Sandoval Gío, coasesorado por el Dr. Agustín Flores Novelo** y la comisión revisora integrada por el Dr. Carlos Alberto Luján, el Dr. Víctor Sandoval Curmina y el M.C. Marío Renán Moreno Sabido, considerando que cubre los requisitos establecidos en el Reglamento de Titulación de los Institutos Tecnológicos le autorizamos la impresión de su trabajo profesional con la TESIS:

"DESARROLLO DE SOFTWARE EMBEBIDO Y DE PLATAFORMA IOT PARA EQUIPOS DE MONITOREO DE CONSUMOS DE ENERGÍA ELÉCTRICA"

ATENTAMENTE

Excelencia en Educación Tecnológica

M.C. DANIEL ARCÁNGEL LÓPEZ SAURI
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN.



S. E. P.
INSTITUTO TECNOLÓGICO DE MERIDA
DIVISION DE ESTUDIOS DE POSGRADO E INVESTIGACION

C.p. Archivo
DALS/fja



Declaración de Autoría

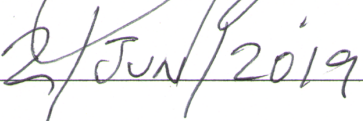
Yo, Ing. José Alfonso Ureña Pajón, declaro que esta tesis titulada, «Desarrollo de software embebido y de plataforma IoT para equipos de monitoreo de consumos de energía eléctrica» y el trabajo presentado en ella son de mi autoría. Confirmando que:

- En el caso de que alguna parte de esta tesis haya sido sometida previamente a una licenciatura u otra titulación en esta Universidad o en cualquier otra institución, esto se ha indicado claramente.
- Donde he citado el trabajo de otros, siempre se da la fuente. Con la excepción de tales citas, esta tesis es enteramente mi propio trabajo.
- He reconocido todas las principales fuentes de ayuda.
- Cuando la tesis se basa en el trabajo realizado por mí mismo junto con otros, he dejado claro exactamente lo que han hecho los demás y lo que he contribuido yo mismo.

Firma:



Fecha:



*«Cuando realizas un descubrimiento por ti mismo...
- Incluso si eres la última persona en la Tierra en ver la luz,
¡Nunca lo olvidarás!»*

Carl Sagan

INSTITUTO TECNOLÓGICO DE MÉRIDA

Resumen

División de Estudios de Posgrado e Investigación

Maestro en Ingeniería

Desarrollo de software embebido y de plataforma IoT para equipos de monitoreo de consumos de energía eléctrica

autoría Ing. José Alfonso Ureña Pajón

El tema de la energía eléctrica es cada vez más importante, su costo se ha convertido en uno de los conceptos de gasto más alto de las empresas y en general de todos los sectores de la sociedad; su precio ha aumentado año con año y no se vislumbra que en un futuro cercano sea posible revertir esta tendencia. Es por esta razón que conocer en tiempo real y de manera histórica cómo, dónde y cuándo se consume la energía ya no es un capricho. Actualmente las empresas lo saben y muchas de ellas cuentan con equipos de medición instalados, sin embargo por cuestiones de precio o por limitaciones tecnológicas, estos medidores no siempre brindan un acceso adecuado a los valores medidos y por tanto no permiten una detección de problemas de manera oportuna.

En plena era del internet de las cosas (IoT), una solución ideal sería adquirir equipos que se conecten a la red y permitan acceso más flexible a la información medida, sin embargo muchas veces remplazar los equipos existentes por otros con capacidades IoT no es viable. Con esta idea en mente se realizó este proyecto, crear una plataforma que permita sin un alto costo de inversión, agregar las capacidades IoT a los medidores de energía existentes en la industria, aprovechando el puerto de comunicación ModBUS con el que casi todos cuentan pero que muy pocas veces se aprovecha.

El proyecto total se divide en dos partes, el nodo medidor, parte de otra tesis complemento a ésta, enfocado a la interfaz con el medidor de energía y el nodo concentrador, objetivo central de esta tesis y cuya tarea principal es recibir los datos medidos en el nodo medidor, almacenarlos y conectarlos a la red.

Abstract

The issue of electric power consumption is increasingly becoming more important as it is now one of the highest costs of companies and in society in general. The price of electric power has increased year by year and it appears that this trend will continue in the near future. Consequently, knowing currently and historically, how, when and where energy is consumed is mandatory. Nowadays companies know this and many of them have installed measurement equipment; however, due to price issues or technological limitations, many of these meters do not have proper access to measured values and therefore don't detect problems promptly.

In the middle of the Internet of Things (IoT) era, a better solution would be to acquire equipment that connects to the Web, allowing more flexible access to measured information. However, replacing existing equipment with those that use IoT capabilities is not always feasible. This project was developed with this idea in mind, to create an affordable system to upgrade existing energy meters in the industry to use IoT capacities, making use of the standard ModBUS communication port to access the measurements.

The project is divided into two parts, the measuring node, explained in a complementary thesis, focuses on the interface with the energy meter, and the hub node, the main subject of this thesis and whose purpose is to receive the measured values from the measuring node, store them and connect them to the network.

Agradecimientos

Antes que nada, quisiera agradecer a mi asesor de tesis, el doctor Jesús Sandoval Gio, por su paciencia, guía y soporte a lo largo del proyecto, es sin duda poseedor de un gran talento para la transferencia del conocimiento y ha sido una fuente de sabiduría para todos los alumnos que tuvimos la suerte de tenerlo como maestro.

A los doctores Ramón Atoche, Agustín Flores y Carlos Luján, que ayudaron amablemente a la realización del proyecto y de la tesis.

A todos los maestros que de alguna manera le han dado forma a mi personalidad y me han interesado en continuar aprendiendo, pero que sería injusto intentar mencionar por que no acabaría y siempre me faltaría alguien.

A mi papá, que deliberadamente y con conocimiento de causa inyectó la semilla de la curiosidad en mí, mucho antes de que pudiera entender los efectos positivos que tendría en mi vida.

A mi mamá y a mi tía Yolanda, "la tía Chata", que sin merecerlo, siempre me han privilegiado con su confianza ciega.

A Zeidy, compañera incansable, incondicional y complemento de mi vida.

A Hyori y Olav, que con su presencia han potenciado mi motivación y han obligando a reinventarme para poder hacer las cosas que siempre quise, pero nunca me creí capaz.

A Cecilia que sin saberlo fue la brújula que le dio la dirección correcta a mi vida.

A mi amigo Marcos Olivera, que en paralelo ha fincado una vida de conocimiento y de superación constante y que entre otras cosas, realizó la parte del nodo medidor, complemento de esta tesis.

A la comunidad que existe actualmente en la red, que generosamente no se guarda nada y que han aportado soluciones e información cuando más se han necesitado.

A la comunidad que elabora software libre de código abierto como Linux, Git, Gimp, LibreOffice, Latex, NodeRED, mySQL etc., literal, sin ellos este proyecto no hubiera sido posible.

Quiero dar un agradecimiento particular a Steve R. Gunn, Sunil Patel y Vel por crear y modificar la plantilla *Masters/Doctoral Thesis* en L^AT_EX, la cual después de algunas modificaciones de lenguaje y formato, fue la base sobre la que se realizó esta tesis.

Gracias.

Índice general

Declaración de Autoría	II
Resumen	IV
Agradecimientos	VI
Índice general	VIII
Índice de figuras	XII
Índice de tablas	XVI
Lista de Abreviaturas	XVII
1. Presentación	1
1.1. Introducción	1
1.2. Planteamiento del problema	3
1.3. Objetivos	8
1.3.1. Objetivo general	8
1.3.2. Objetivos específicos	9
1.4. Justificación	10
1.5. Estado de la técnica	11
1.5.1. Hardware	11
1.5.2. Soluciones integrales de medición de energía eléctrica IoT	12
1.5.3. Servicios de nube	13
1.5.4. Plataformas IoT	14
1.6. Estado del arte	15
2. Marco teórico	17
2.1. Internet de las Cosas	17
2.1.1. Características fundamentales del IoT	19

2.1.2.	Modelo de referencia IoT	20
2.1.3.	Arquitectura IoT	21
2.2.	Redes de Baja Potencia de Área Amplia (LPWANs)	22
2.2.1.	Sigfox	23
2.2.2.	LoRa	23
2.3.	Computadoras de una sola placa	24
2.3.1.	Arduino	24
2.3.2.	BeagleBone	26
2.3.3.	Raspberry Pi	27
2.4.	Lenguajes de programación	29
2.4.1.	Node-RED	29
3.	Metodología	33
3.1.	Descripción del medidor de energía IoT	33
3.2.	Descripción del proyecto de tesis	34
3.2.1.	Comunicación con el módulo LoRa	35
3.2.2.	Almacenamiento en la base de datos	39
3.2.3.	Visualización de datos por medio del “dashboard” de Node-RED	40
	Nodo de entrada <i>Button</i>	42
	Nodo de entrada <i>Dropdown</i>	43
	Nodo de salida <i>Text</i>	43
	Nodo de salida <i>Gauge</i>	44
	Nodo de salida <i>Chart</i>	44
3.2.4.	Otros nodos importantes de Node-RED utilizados	46
	Nodo <i>Debug</i>	46
	Nodo <i>Inject</i>	47
	Nodo <i>Function</i>	49
	Nodo <i>Email</i>	49
3.3.	Diseño y descripción de la lógica del programa	50
3.3.1.	Inicialización del sistema	53
3.3.2.	Servicio al evento de llegada de datos por el puerto serie	55
3.3.3.	Solicitud de consulta a la base de datos	56
	Requisición de actualización de la gráfica del historial de mediciones	56
	Actualización de datos de la tabla de variables estáticas	57
	Solicitud de descarga de registro de mediciones	58

3.3.4.	Respuesta de la base de datos a la consulta realizada	59
3.3.5.	Interfaz de usuario	60
	Historial de mediciones	60
	Consumos en periodo de facturación	61
	Ajustes	62
	Valores instantáneos	62
	Otras mediciones	63
	Facturación y corte	64
3.3.6.	Alertas	64
	Alerta de consumo máximo en periodo de facturación	65
	Alerta de potencia activa instantánea máxima	67
3.3.7.	Visualización del tablero de mediciones	68
4.	Implementación y resultados	71
4.1.	Implementación e instalación	71
4.1.1.	Nodo medidor	71
	Equipo de medición	72
	Lectura de mediciones y formato de trama	73
	Radio LoRa	74
4.1.2.	Concentrador	76
4.1.3.	Equipo de aire acondicionado monitoreado	77
4.1.4.	Distancia entre nodos	78
4.2.	Descripción de las pruebas	79
4.2.1.	Pruebas de mediciones eléctricas	79
4.2.2.	Pruebas de distancia	80
4.3.	Resultados	81
4.3.1.	Resultados de mediciones eléctricas	81
	Mediciones eléctricas instantáneas	81
	Mediciones eléctricas históricas	84
4.3.2.	Resultados pruebas de distancia	89
5.	Conclusiones	93
5.1.	Conclusiones generales	93
5.2.	Cambios	94
5.3.	Recomendaciones	95
5.4.	Trabajo posterior	96

A. Flujos del programa	98
A.1. Actualización de consumos	98
A.2. Recepción de datos	99
A.3. Cambio de variables estáticas	100
A.4. Inicialización del dashboard	101
A.5. Consultas a la base de datos	102
B. Estructura de la interfaz de usuario	103
C. Estructura y análisis de la base de datos	105
C.1. Estructura	105
C.2. Análisis	107
D. Ejemplo de datos capturados	109
D.1. Fragmento de captura del 13 de Febrero del 2019	109
Bibliografía	120

Índice de figuras

1.1. Precio de la energía eléctrica	1
1.2. Consumos netos de energía eléctrica	3
1.3. Proyección IoT	5
1.4. Diagrama a bloques del sistema	8
1.5. Medidor de energía KillaWatt	11
1.6. Medidores de energía Allen-Bradley	11
1.7. Medidores de energía Schneider-Electric	12
1.8. Medidor de energía Smappee	12
1.9. Medidor de energía Curb	13
1.10. Servicios en la nube	14
1.11. Servicios de plataforma IoT	14
2.1. Representación visual del IoT	18
2.2. Dimensiones de la comunicación	19
2.3. Modelo de Referencia IoT	20
2.4. Arquitectura IoT	21
2.5. Tecnologías de comunicación inalámbrica	23
2.6. Placas Arduino	25
2.7. Placas BeagleBone	27
2.8. Algunos modelos de Raspberry Pi	28
2.9. Lenguajes de programación usados en IoT	30
2.10. Node-RED programación para el Internet de las Cosas	30
2.11. Ventana de navegador del editor de Node-RED	31
2.12. Partes del editor de Node-RED	32
2.13. Ejemplo de un flujo de nodos	32
3.1. Medidor de energía IoT	33
3.2. Diagrama a bloques simplificado	34
3.3. Módulo LoRa E32-915T20D	36

3.4. Definición de pines del módulo LoRa E32-915T20D	37
3.5. Puerto GPIO de la Raspberry Pi Zero	37
3.6. Selección y edición del nodo serial en Node-RED	38
3.7. Configuración del puerto serial en Node-RED	38
3.8. Nodo serial del Node-RED conectado	39
3.9. Base de datos	39
3.10. Estructura de la tabla de variables estáticas	39
3.11. Estructura de la tabla de mediciones eléctricas	40
3.12. Conjunto de nodos “storage” y edición del nodo “mysql” en Node-RED	40
3.13. Configuración del nodo de la base de datos	41
3.14. Nodo “mysql” de Node-RED conectado	41
3.15. Nodo del dashboard “Button”	43
3.16. Nodo del dashboard “Dropdown”	44
3.17. Nodo del dashboard “Text”	45
3.18. Nodo del dashboard “Gauge”	46
3.19. Nodo del dashboard “Chart”	47
3.20. Tablero de instrumentos	47
3.21. Nodo de Node-RED “Debug”	48
3.22. Nodo de Node-RED “Inject”	48
3.23. Nodo de Node-RED “Function”	49
3.24. Nodo de Node-RED “Email”	50
3.25. Diagrama de flujo del programa principal	51
3.26. Diagrama de flujo de respuesta a la entrada de datos de medición	52
3.27. Diagrama de flujo de solicitud de consulta a la base de datos	52
3.28. Diagrama de flujo del proceso de respuesta a una consulta	53
3.29. Flujo de recuperación de variables estáticas de la base de datos	54
3.30. Flujo de restauración de variables en el dashboard	54
3.31. Flujo de inicialización del dashboard	54
3.32. Flujo las mediciones en la base de datos	55
3.33. Tabla “Mediciones” con mediciones simuladas	57
3.34. Flujo de requisición de actualización de la gráfica del historial de mediciones	57
3.35. Flujo de actualización de información del usuario	58
3.36. Flujo de requisición de descarga de mediciones	58
3.37. Flujo de respuesta a graficación	60
3.38. Grupos de la interfaz de usuario	61

3.39. Flujo de actualización de consumos	61
3.40. Flujo de actualización de variables de usuario en el dashboard	62
3.41. Flujo de actualización de mediciones instantáneas en el dashboard	63
3.42. Flujo de actualización de los datos de facturación y corte	64
3.43. Proceso de definición de la alerta de consumo máximo	65
3.44. Email ejemplo de alerta de consumo máximo	66
3.45. Proceso de definición de la alerta de potencia activa instantánea máxima excedida	67
3.46. Email ejemplo de alerta de potencia activa instantánea máxima	68
3.47. Dirección IP de la pantalla inicial	68
3.48. Dirección IP con el comando ifconfig	69
3.49. Dirección IP leída directamente del router	69
3.50. Ingreso al dashboard o tablero de mediciones	70
4.1. Diagrama a bloques del nodo medidor	72
4.2. Medidor de energía Eastron modelo SDM120Modbus	72
4.3. Imágenes del nodo medidor	74
4.4. Nodo medidor instalado	75
4.5. Nodo concentrador	76
4.6. Nodo concentrador (continuación)	77
4.7. Equipo de aire acondicionado monitoreado	77
4.8. Distancia entre los nodos	78
4.9. Vista aérea de la zona donde se realizaron las pruebas de distancia	81
4.10. Clima con compresor encendido 2:26 pm	82
4.11. Clima con compresor encendido 20:52 pm	82
4.12. Clima encendido, solo evaporador	83
4.13. Clima apagado	83
4.14. Clima con compresor encendido 17:04 pm	84
4.15. Clima con compresor encendido 22:04 pm	84
4.16. Gráficas de mediciones eléctricas 8 febrero	86
4.17. Gráficas de mediciones eléctricas 13 febrero	87
4.18. Graficas de mediciones eléctricas 14 febrero	88
4.19. Consumo de energía por día	89
4.20. Topología de la zona de pruebas de distancia	90
4.21. Resultados de las pruebas de distancia	91

A.1. Flujo de actualización de consumos	98
A.2. Flujo de recepción de datos	99
A.3. Flujo de cambio de variables estáticas	100
A.4. Flujo de inicialización del dashboard	101
A.5. Flujo de consultas a la base de datos	102
B.1. Estructura jerárquica de la interfaz de usuario	103
B.2. Grupos que conforman la interfaz de usuario	104
C.1. Estructura de la tabla <i>variablesEstaticas</i>	105
C.2. Estructura de la tabla <i>Mediciones</i>	106
C.3. Análisis de la tabla <i>variablesEstaticas</i>	107
C.4. Análisis de la tabla <i>Mediciones</i>	108

Índice de tablas

2.1. Especificaciones de la placa Arduino UNO	26
2.2. Especificaciones de la placa BeagleBone Black	27
2.3. Especificaciones de la Raspberry Pi Zero W	29
3.1. Especificaciones RF del módulo LoRa E32-915T20D	36
3.2. Especificaciones de hardware del módulo LoRa E32-915T20D	36
3.3. Especificaciones diversas del módulo LoRa E32-915T20D	37
3.4. Nodos de dashboard usados en el proyecto	45
4.1. Especificaciones del medidor de energía de una sola fase marca Eastron modelo SDM120Modbus.	73
4.2. Precisión de las mediciones.	73
4.3. Especificaciones del clima utilizado.	78

Lista de Abreviaturas

A/D	Analógico a Digital
API	Application Programming Interface Interfaz de Programación de Aplicaciones
ARM	Advanced RISC Machine
AWS	Amazon Web Services Servicios Web de Amazon
BLE	Bluetooth Low Energy Bluetooth de Baja Energía
BTU	British Thermal Unit Unidad Térmica Británica
CAN bus	Controller Area Network bus
CFE	Comisión Federal de Electricidad
CPU	Central Processing Unit Unidad Central de Procesamiento
CSI	Camera Serial Interface Interfaz Serie para Cámaras
CSV	Comma Separated Values Valores Separados por Comas
DBPSK	Differential Binary Phase Shift Keying Modulación por Desplazamiento de Fase Binaria Diferencial
DIP	Dual Inline Package Encapsulamiento de Doble Hilera
DNS	Domain Name System Sistema de Nombres de Dominio
EEPROM	Electrically Erasable Programmable Read Only Memory Memoria de Solo Lectura Programable y Borrable Eléctricamente
eMMC	embedded MultiMedia Card Tarjeta de Memoria Multimedia Embebida
E/S	Entrada/Salida

GPIO	General Purpose Input Output Entrada Salida de Propósito General
GUI	Graphic User Interface Interfaz Gráfica de Usuario
HAT	Hardware Attached on Top Hardware Montado por la parte Superior
HDMI	High Definition Multimedia Interface Interfaz Multimedia de Alta Definición
HTML	Hypertext Markup Language Lenguaje de Marcas de Hipertexto
I²C	Inter-Integrated Circuit Circuito Inter-Integrado
IBM	International Business Machines
IDE	Integrated Development Environment Entorno de Desarrollo Integrado
IFTTT	If This Then That Si (ocurre) Esto, Entonces (haz) Eso
IoT	Internet of Things Internet de las Cosas
IP	Internet Protocol Protocolo de Internet
ISM bands	Industrial Scientific Medical bands Bandas de radio Industriales, Científicas y Médicas
LAMP	Linux Apache MySQL Php
LAN	Local Area Network Red de Área Local
LCD	Liquid Crystal Display Pantalla de Cristal Líquido
LoRa	Long Range Largo Alcance
LPWAN	Low Power Wide Area Network Redes de Baja Potencia de Área Amplia
MIPS	Million Instructions Per Second Millones de Instrucciones Por Segundo
MQTT	Message Queuing Telemetry Transport

PCB	Printed Circuit Board Placa de Circuito Impreso
PLC	Power Line Communications Comunicaciones mediante Línea de Potencia
PWM	Pulse Width Modulation Modulación por Ancho de Pulsos
RAM	Random Access Memory Memoria de Acceso Aleatorio
RFID	Radio Frequency Identification Identificación por Radiofrecuencia
RISC	Reduced Instruction Set Computer Computador con Conjunto de Instrucciones Reducidas
RPi	Raspberry Pi
SQL	Structured Query Language Lenguaje de Consulta Estructurada
SMA	SubMiniature version A Conector SubMiniatura versión A
SMTP	Simple Mail Transfer Protocol Protocolo para Transferencia Simple de Correo
SPI	Serial Peripheral Interface Interfaz de Periféricos Serie
SSL	Secure Sockets Layer Capa de Puertos Seguros
TCP	Transmission Control Protocol Protocolo de Control de Transmisión
UART	Universal Asynchronous Receiver Transmitter Transmisor Receptor Asíncrono Universal
UI	User Interface Interfaz de Usuario
URL	Uniform Resource Locator Localizador de Recursos Uniforme
USB	Universal Serial Bus Bus Universal en Serie

*Para Zeidy, Hyori, Olav y Cecilia, sin ellos no podría ser
ni estar...*

Capítulo 1

Presentación

1.1. Introducción

El costo de la energía eléctrica se ha convertido en uno de los conceptos de gasto más alto de las empresas y en general en todos los sectores de la sociedad, su precio ha aumentado año con año y no se ve en el futuro cercano que sea posible revertir esta tendencia. La [Figura 1.1](#) demuestra claramente este incremento y como se puede observar los sectores que más han sido más afectados son el comercial y el de servicios ([SENER, Secretaría de Energía 2017](#)).

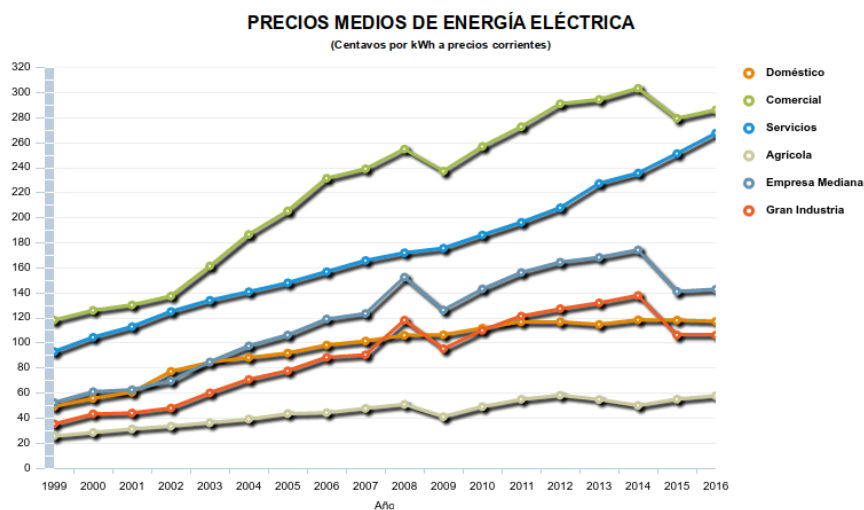


FIGURA 1.1: Precio de la energía eléctrica en México desde 1999

Fuente: Sector Eléctrico Nacional

Por otro lado, no solo el costo de la energía ha aumentado drásticamente con el pasar del tiempo, la mercadotecnia, el consumismo, la presión social y en general nuestra dependencia a la tecnología nos ha provocado una necesidad intrínseca por el aumento de nuestros consumos individuales y en masa. En la [Figura 1.2](#) se

puede ver cómo las cantidades de energía que como sociedad consumimos igual han aumentado. La imagen muestra los consumos de energía en Giga-Watts-hora netos divididos por sector, desde el año 1999 al 2016, y como se indica en la gráfica, los sectores que han resentido más los cambios son el doméstico y las empresas medianas ([SENER, Secretaría de Energía 2017](#)).

Juntos el aumento de la cantidad de energía consumida y del precio por unidad provocan una preocupación por el cuidado de los patrones de consumo, especialmente en los sectores más propensos a sentir los cambios por falta de subsidio del gobierno, como lo son los comercios y la industria, en ellos el uso eficiente de la energía es una necesidad y un factor clave para la reducción de gastos de producción y de operación, lo cual a su vez les permite ser competitivas y así asegurar su supervivencia en el mercado.

Por otro lado, con el desarrollo industrial, las emisiones de contaminantes por uso de la energía se han convertido en una preocupación común para los Gobiernos y el sector privado. La sustentabilidad del desarrollo se ha puesto en peligro debido a la falta de estrategias de uso eficiente de la energía. El gobierno por su parte ha aumentado las sanciones y los impuestos relacionados con el mal uso de la energía y maltrato al medio ambiente, lo cual decrementa aún más las utilidades netas de las empresas.

Del lado de la sociedad y el sector privado se ha puesto de moda la preocupación por nuestro planeta y en general mostrar una imagen ecológica es un factor positivo en la interacción social y comercial. Las empresas inteligentemente se han interesado por reducir las emisiones de carbono para poder ser percibidas como empresas sustentables, sabiendo que más que un gasto, estas estrategias son una inversión tanto por el ahorro inmediato en consumo, como en las consecuencias mercadológicas que estas conllevan.

Por último, pero no menos importante, existe un factor de riesgo relacionado directamente con la cantidad de consumo de energía, en cuanto mayor sea las cantidades netas de consumo, mayor será el riesgo de que los aumentos de precios de la energía o la escasez de suministro puedan afectar seriamente la rentabilidad o incluso la continuidad de un negocio en el mercado.

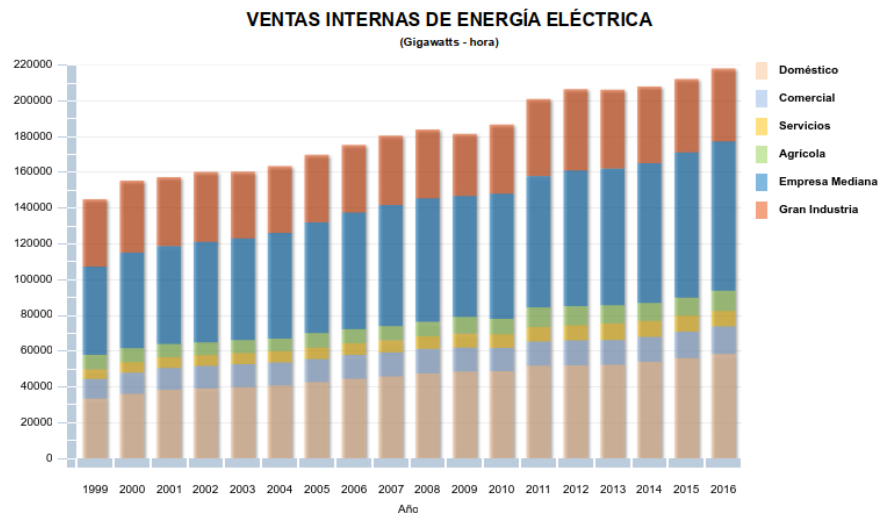


FIGURA 1.2: Consumos netos de energía eléctrica en México desde 1999

Fuente: Sector Eléctrico Nacional

1.2. Planteamiento del problema

El tema de la energía es cada vez más importante en todos los ámbitos de la sociedad y como se mostró en las [Figura 1.1](#) y [Figura 1.2](#), las empresas y la industria son dos sectores que han resentido de una manera más severa los embates del aumento de costo y dependencia de la energía eléctrica; esto provoca que uno de los deseos de las empresas sea encontrar la manera de conocer en tiempo real y de manera histórica cómo, dónde y cuándo se consume la energía, con el objetivo de recopilar información, analizarla y poder implementar mejoras que permitan aumentar los niveles de eficiencia, esto es conocido como *gestión energética*.

La parte más importante de la gestión energética es la medición, sólo las variables que se miden son las que se pueden controlar y por ende mejorar; la medición es el primer paso para saber dónde se encuentra las áreas de oportunidad.

La necesidad de medir energía lleva ya varias décadas de existencia, con el tiempo muchas empresas especializadas han dedicado mucho esfuerzo y dinero a investigación y diseño, de los cuales se han creado diversos equipos con distintas capacidades, sin embargo, los equipos que se encuentran actualmente instalados siempre dejan algo que desear, ya sea en su desempeño, capacidad, interacción con el usuario o precio.

Las situaciones más comunes que se encuentran actualmente en el mercado en lo que respecta a los equipos de medición eléctrica, son las siguientes:

1. Equipos muy caros de marcas importantes, pero con capacidades limitadas en conectividad y en su manera de interactuar con el usuario; en estos equipos las funcionalidades particulares se pueden adquirir al gusto del usuario, pero bajo la adquisición de un equipo adicional o software de la misma marca y de alto precio.
2. Equipos que cumplen con las necesidades de medición de los usuarios, con una interfaz de usuario amigable, pero con acceso a los datos limitado o nulo; muchos equipos comerciales que cumplen con los requerimientos de las empresas solo muestran la información en una pantalla, por lo que los datos históricos los tiene que recabar una persona que anota esta información en una bitácora o que en el mejor de los casos introduce manualmente la información en una base de datos.
3. Equipos con capacidades de medición que cumplen con las necesidades y que permiten acceso a los datos pero de una forma rudimentaria; algunos traen algún puerto de conexión para memorias externas para almacenar o “loggear” los datos históricos, estos aunque representan una mejora con respecto a los que solo tienen una pantalla, evitando los “errores de dedo”, siguen siendo propensos a errores por duplicación, corrupción o inclusive olvido de recolección de datos. Otros equipos traen alguna conexión a computadora tales como puerto serial, usb o Ethernet, y muy pocos con una conexión wifi, sin embargo usualmente estas conexiones necesitan de la compra de licencias de software o de utensilios extra que sólo trabajan con estos equipos o que tienen alguna especie de bloqueo, para forzar a los usuarios a comprar más productos de la misma marca.

En general en los tres casos los equipos se pueden subdividir en dos tipos:

- A. Los de gama baja, enfocados a monitoreo residencial o a instalaciones comerciales pequeñas, normalmente costo accesible, pero con muchas limitaciones.
- B. Los de gama mediana o alta enfocados a instalaciones más grandes y que independientemente de las limitaciones mencionadas, cuentan con un puerto de comunicación ModBus por medio del cual expone todos sus registros; este puerto muy pocas veces se aprovecha ya que normalmente necesita de la adquisición de instrumentos adicionales.

Por otro lado, en los últimos años ha surgido con mucha fuerza una tecnología llamada Internet de las Cosas (IoT por sus siglas en inglés), que consiste en darle

capacidad de conexión a utensilios de nuestra vida diaria y en general a cualquier dispositivo que se encuentre en contacto con información útil. Estos dispositivos llamados genéricamente “cosas”, ahora recolectan datos relacionados con su actividad y con su entorno, los cuales son enviados periódicamente a un servidor o nube, donde son procesados, almacenados y distribuidos a otras “cosas” para ser desplegados o usados en otros procesos.

Con esta tecnología y con las grandes cantidades de información que se generan continuamente, ha nacido el “Big Data”, que se enfoca en analizar esta información como un todo para encontrar tendencias y patrones, dando paso a una manera totalmente distinta a la detección de problemas.

El internet de las cosas va en incremento día a día y se prospecta que para el 2025 habrá 75,000 millones de objetos físicos y virtuales conectados a la red ([Statista 2018](#)) ([Figura 1.3](#)).

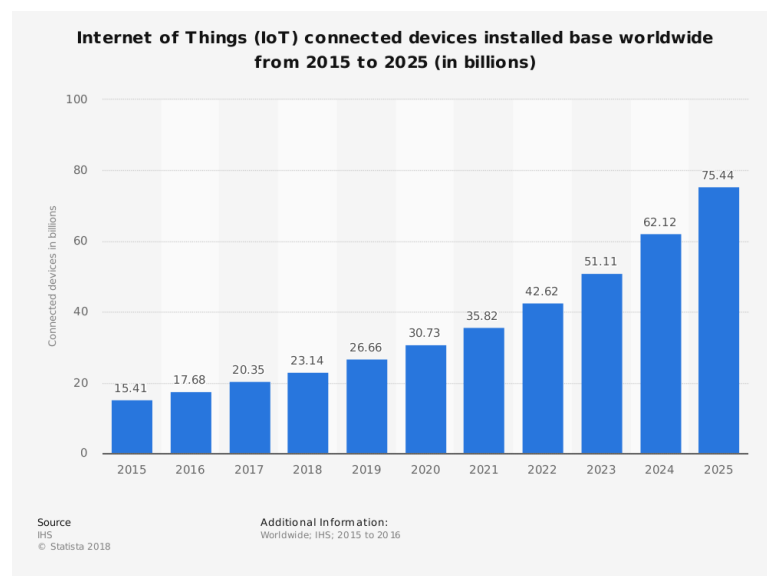


FIGURA 1.3: Proyección de dispositivos conectados a la red.

Fuente: *statista*

Las arquitecturas de hardware IoT por lo general se dividen en dos tipos:

- a.- Arquitectura con concentrador, consiste en el uso de una o más “cosas” a lo largo del predio o planta, las cuales se comunican a un solo concentrador o Gateway, cuya función es recolectar la información, almacenarla y transmitirla a través de la red a donde se requiera. Normalmente esta configuración es de bajo costo debido a que las “cosas” no necesitan altas capacidades de procesamiento, memoria o conectividad y solo realizan prácticamente dos funciones, recolectar

datos a través de sus sensores y mandarlos al concentrador, el cual es el único que posee capacidad de procesamiento y conexión considerables; su principal desventaja es que al fallar el concentrador, falla todo el sistema. Por otro lado, este tipo de conexión proporciona un beneficio extra, la flexibilidad de usar el concentrador solo como Gateway (punto de acceso a otra red), como elemento de preprocesamiento (conocido como computación en la niebla), o la posibilidad de prescindir del servidor externo/nube, y usar el concentrador como servidor local.

- b.- La arquitectura LAN tradicional, en este tipo de sistema cada equipo de medición tiene una dirección IP por sí mismo, lo que permite conectarse directamente a la red. Cuenta con la gran ventaja que, si un equipo falla, sólo se pierde la información del equipo averiado, sin embargo, tiende a ser un sistema más costoso comparado a la arquitectura con concentrador, pues cada equipo necesita tener capacidades de procesamiento, memoria y conectividad más altas que cuando existe un solo concentrador. En esta configuración no existen equipos de gama baja.

Es fácil de ver que un equipo de medición de energía eléctrica con capacidades IoT sería la solución perfecta a toda la problemática descrita, ya que eliminaría las dificultades de conexión, acceso a los datos e interfaz de usuario. Estos equipos ya están empezando a ofrecerse comercialmente, sin embargo, están más enfocados al mercado residencial, con capacidades de medición de cargas o redes eléctricas pequeñas; muchos de estos equipos todavía presentan la gran desventaja de contar con bloqueos o candados para trabajar con productos y software de la misma marca y normalmente solo se conectan a un servidor o nube propios de la marca que fabrica los equipos de medición.

En cuanto al almacenamiento, existen los servicios de nube con todo tipo de soluciones y funcionalidades extra, no obstante implementar una solución para dispositivos de medición de energía representa varios retos, ya que estas plataformas son muy complejas e implican la contratación de personal calificado o de empresas externas, además que no solucionan el problema completo ya que requieren de la parte de hardware.

También existen plataformas IoT genéricas que dan el servicio de procesamiento, almacenamiento, análisis y acceso de datos en la nube, sin embargo, en su afán de ser genéricas a todo tipo de mediciones, presentan varias restricciones e implican el pago de una renta mensual o una tarifa por consumo de datos. Al igual que en los

servicios de nube convencionales, estas plataformas forman parte de la mitad del problema ya que igual carecen de la parte de hardware, pero con la diferencia que en estas plataformas se tiene bien definidos los protocolos, plataformas, procesadores, sistemas operativos etc., que pueden conectarse, por lo que es relativamente fácil agregar a la red un hardware que cumpla con estas características. El IoT es una tecnología poderosa y disruptiva, que si bien soluciona de una manera espectacular los problemas del mercado, todavía no ha terminado de madurar y actualmente se encuentra en el periodo en el que no hay reglas claras y cada quien ofrece su manera de solucionar las cosas, lo cual confunde a los potenciales clientes y abre paso a muchas oportunidades de mercado.

1.3. Objetivos

El proyecto de medición de energía eléctrica IoT se divide en dos partes, la parte correspondiente al hardware medidor de energía (nodo), más enfocado a la etapa física y señales eléctricas, y la segunda parte que es el diseño del concentrador (servidor), con más carga de trabajo relacionada con software.

Este proyecto de tesis cubre la segunda parte del proyecto total, y por tanto consta única y exclusivamente del diseño del concentrador y de la comunicación con el nodo medidor y con los dispositivos de visualización por medio de una aplicación web; esta parte del trabajo se encuentra más orientada al software y manejo de datos (en verde en [Figura 1.4](#)). Por lo tanto, no obstante que el proyecto es de medición de energía eléctrica, mucho de lo que se menciona a lo largo de la tesis se plantea desde la perspectiva del manejo de datos más que de señales eléctricas.

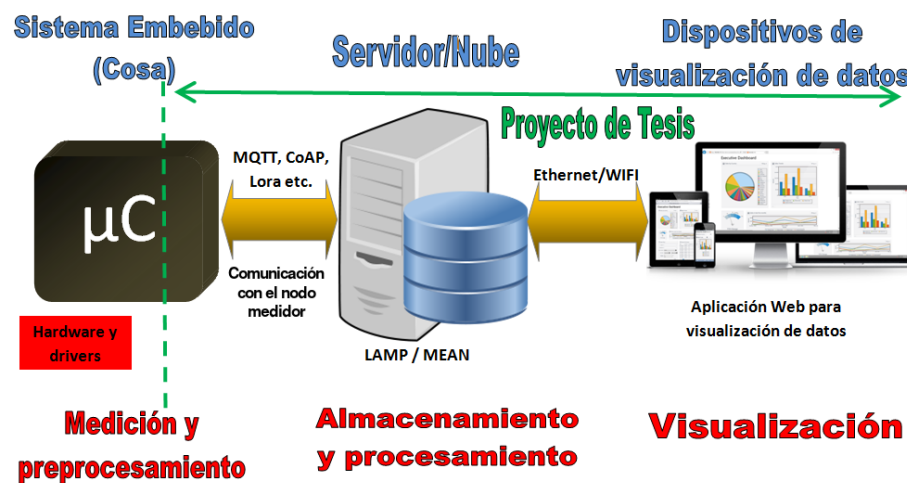


FIGURA 1.4: Diagrama general a bloques del sistema.

1.3.1. Objetivo general

Diseñar un sistema que conforme las diversas etapas del Internet de las cosas (IoT), teniendo como arquitectura una configuración servidor-concentrador sin conexión a la nube, con la finalidad de integrar equipos de monitoreo de energía eléctrica industrial y/o residencial a los servicios IoT.

1.3.2. Objetivos específicos

1. Desarrollar el software del concentrador para que pueda realizar las tareas necesarias, tales como recepción de información proveniente del elemento IoT, almacenamiento de los mismos en la base de datos, y gestión de las consultas provenientes de los dispositivos externos.
2. Diseñar la aplicación web que conforma la pantalla de presentación de los datos, de una manera gráfica y accesible para los usuarios.
3. Diseñar el software necesario para poder habilitar y asegurar una comunicación confiable entre el nodo Medidor de Energía y el dispositivo concentrador Gateway.

1.4. Justificación

Este proyecto no pretende la elaboración de un medidor de energía como tal, ya que actualmente hay muchos en el mercado de todos tipos y capacidades; el proyecto propone algo un poco más flexible, aprovechando el puerto Modbus que ya traen incluido los equipos de gama media y alta, lo que se propone es elaborar una solución que brinde acceso a los datos de las mediciones que se exponen en el puerto del medidor de energía y que por medio de una configuración IoT con concentrador, permita el ingreso de estos datos a la red. De esta manera se estaría atacando al mismo tiempo dos de los problemas más importantes que se encuentran actualmente en el mercado:

1. Se le podría dar acceso a todas las bondades del IoT a los dispositivos medidores con puerto ModBUS que se encuentran actualmente instalados y que en general no tienen la necesidad de ser reemplazados.
2. Para los clientes que no cuentan actualmente con un dispositivo de monitoreo de energía, sería de costo muy accesible usar esta interfaz junto un dispositivo medidor sencillo que cuente con conexión Modbus. Actualmente equipos como estos son muy baratos y se pueden adaptar a las necesidades particulares de cada cliente.

En su primera etapa no se pretende realizar la solución total con servicio de nube, ya que sería muy ambicioso; en este proyecto se va a manejar el concentrador con capacidad de servidor local, lo cual ya es una solución robusta que le permite a los usuarios acceso total a sus datos de una manera local y remota por medio de algún otro servicio como email.

Productos como éste existen, pero caerían en [el caso número uno mencionado en el planteamiento del problema](#) ya que son muy caros y los fabrican las marcas reconocidas como Eaton, Allen-Bradley, Schneider etc., por mencionar algunas, y están hechos para comunicarse solo con los medidores de la misma marca. Este producto le permitiría el acceso a las funcionalidades importantes de monitoreo de energía a un segmento amplio del mercado actual, el cual sólo se encuentra ante las posibilidades de adquirir un equipo de alto precio o continuar con sus limitaciones de medición.

1.5. Estado de la técnica

1.5.1. Hardware

Uno de los medidores más básicos que se pueden encontrar en el mercado es el famoso killawatt, es un medidor muy sencillo que cuenta con las mediciones eléctricas básicas presentadas por medio de una pantalla, solo sirve para monitorear una toma de corriente, [Figura 1.5](#).



FIGURA 1.5: Medidor de energía KillaWatt.

Empresas grandes tales como Allen-Bradley, Schneider, fabrican medidores desde los más básicos que solo miden energía y sacan los resultados por algún puerto, hasta los más complejos que miden la calidad de la energía y tienen servicios de Gateway o Servidores web integrados, [Figura 1.6](#) y [Figura 1.7](#) respectivamente ([Allen-Bradley 2017](#)), ([Schneider 2017](#)).

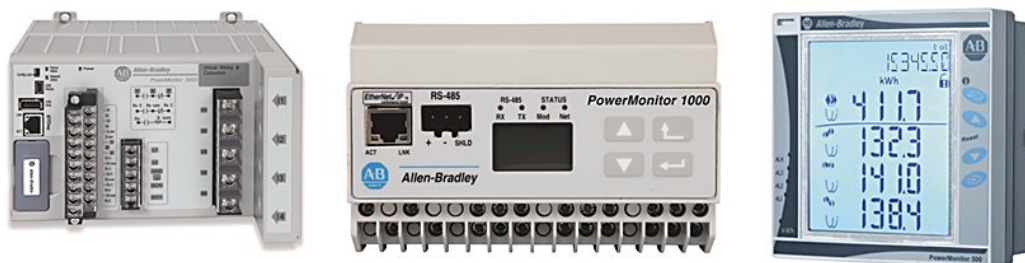


FIGURA 1.6: Medidores de energía Allen-Bradley.



FIGURA 1.7: Medidores de energía Schneider-Electric.

1.5.2. Soluciones integrales de medición de energía eléctrica IoT

Existen muchas empresas que ofrecen toda la solución de medición de energía IoT, venden el hardware, explican como conectar el equipo a la red y dan el servicio de nube en un servidor propio. Estos productos han salido mayormente por medio de financiamientos de la comunidad mejor conocidos como "crowd funding", y por lo mismo están enfocados a servicios residenciales o redes eléctricas pequeñas, un ejemplo de este tipo es Smappee ([Smappee 2018](#)) (Figura 1.8), que conecta un gateway al centro de carga y por medio de perfiles de detección "intuye" qué aparatos son los que están realizando el consumo y luego por medio de una aplicación reporta los resultados históricos.



FIGURA 1.8: Medidor de energía Smappee.

Otro ejemplo de este tipo de equipos de medición es Curb ([Curb 2018](#)) ([Figura 1.9](#)), esta es una solución muy completa que incluye un gateway y varios sensores de corriente, este equipo da por hecho que el voltaje es constante y por medio las mediciones de corriente saca los resultados de los consumos de energía, igualmente utiliza un servidor propio para dar acceso a los usuarios a sus datos.



FIGURA 1.9: Medidor de energía Curb.

1.5.3. Servicios de nube

Hoy en día existen varias empresas que dan servicio de Nube, sin embargo, tres sobresalen de entre todas las demás, tanto por sus servicios y el tiempo que llevan en el mercado, como por las empresas que las respaldan ([Figura 1.10](#)). En primer lugar Amazon Web Services (AWS) ([Amazon Web Services 2018](#)); Amazon es el líder en lo que a computación en la nube se refiere, es el pionero dando inicio desde el año 2006, cuenta con más de 70 servicios, tales como almacenamiento, alojamiento de sitios web, copias de seguridad y recuperación, marketing digital, Big Data, Machine Learning entre otros. Recientemente presento AWS IoT para Internet de las cosas y actualmente cuenta con el 35 % del mercado.



FIGURA 1.10: Servicios en la nube.

Por otro lado se encuentra Azure ([Microsoft Azure 2018](#)), que es el servicio de nube de Microsoft, comenzó a dar servicios en el año 2010 cuenta con menos servicios que AWS, sin embargo, asegura tener mayor cobertura, también ha empezado a dar servicio especializado IoT y cuenta con el 12 % del mercado. Por último, Google Cloud ([Google 2018](#)), empezaron a dar servicio en el 2011, inició con el uso de servidores para dar servicio a sus propios productos, tales como Google y YouTube, habiendo empezado muy tarde en el mercado, se encuentra rezagado con solo el 6 % del mercado, pero amenaza con hacer en los próximos años una fuerte inversión en infraestructura lo cual podría cambiar las circunstancias actuales.

1.5.4. Plataformas IoT

Al igual que con los servicios de nube convencionales, en estos momentos se encuentra un auge muy grande por los servicios de plataformas IoT ([Figura 1.11](#)), los cuales dan servicios enfocados al internet de las cosas, con conexiones ligeras de tipo suscripción-publicación, servicios de nube, almacenamiento de datos y servicio de páginas web para desplegar datos (dashboards), ejemplos de empresas q dan estos servicios se encuentran Ubidots ([Ubidots IoT Platform 2018](#)) y ThingSpeak ([ThingSpeak IoT 2018](#)).



FIGURA 1.11: Servicios de plataforma IoT.

1.6. Estado del arte

De entre el material que se investigó para indagar el estado del arte del proyecto, se encontraron los siguientes trabajos.

En cuestión de IoT, Afef Mdhaffar, Tarak Chaari, Kaouthar Larbi, Mohamed Jmaiel y Bernd Freisleben presentan en su artículo *IoT-based health monitoring via LoRaWAN*, un sistema de monitoreo de salud IoT, con las características de bajo costo y bajo consumo de energía. Este sistema se enfoca en monitorear presión sanguínea, glucosa y temperatura. Para comunicarse con el concentrador, los nodos utilizan una red LoRaWAN. Esta tecnología, debido a que maneja distancias largas de comunicación, permite a gente de áreas rurales no tener que viajar constantemente a los hospitales de las ciudades más próximas para realizarse evaluaciones médicas (Mdhaffar y col. 2017).

Por otro lado, Dumitru-Cristian Trancă, Alexandru Viorel Pălăcean, Andrei Cristian Mihiu y Daniel Rosner, presentan en el artículo *ZigBee based wireless modbus aggregator for intelligent industrial facilities*, una forma de recolectar datos via ZigBee, desde múltiples esclavos Modbus en instalaciones industriales inteligentes. Este sistema aprovecha el protocolo ModBus usado ampliamente en interfaces cableadas como RS485, RS232 o inclusive ethernet. La manera inalámbrica de trabajar de ZigBee, permite poder recopilar en el maestro, toda la información agregada desde el elemento más extremo de la red por medio de una instalación muy sencilla y rápida (Trancă y col. 2017).

En el artículo *Open source IoT meter devices for smart and energy-efficient school buildings*, los autores Lidia Pocero, Dimitrios Amaxilatasa, Georgios Mylonasa y Ioannis Chatzigiannakis describen la elaboración de un sistema domótico para uso exclusivo en escuelas, que incluye automatización y medición. Los sensores utilizan la plataforma Arduino y se conectan a un Gateway por medio de la tecnología XBee. El gateway se conecta a la red por medio de Ethernet (Pocero y col. 2017).

El trabajo denominado *IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid*, elaborado por Bibek Kanti Barman, Shiv Nath Yadav, Shivam Kumar y Sadhan Gope, propone la elaboración de un medidor de energía utilizando un módulo ESP8266, que se conecta directamente a la red y que utiliza servicios de nube (Barman y col. 2018).

En el artículo de revista *IOT Based Smart Energy Tracking System*, R. Jayanthi y S.T. Rama describen la interconexión de varios sensores en PCB tipo arduino, para elaborar la medición de energía. La conexión a la red y visualización de datos es

por medio de una Raspberry Pi 3 con el uso del lenguaje de programación Python (Jayanthi, R, Rama y Maduravoyal, Chennai 2017).

M. P. Shopov de la Universidad Técnica de Sofia Bulgaria, en el artículo *IoT gateway for smart metering in electrical power systems - software architecture*, presenta la implementación de un gateway IoT para medir de manera inteligente sistemas eléctricos de potencia. El gateway está basado en el sistema operativo Ubuntu Core y conecta medidores de energía con servicios de nube externos (Shopov 2017).

Un proyecto más completo, parte de la tesis de maestría de Hans Hansen, Mikkel Skovsmose Tewes y Nicolai Foldager, denominado *A Home Energy Management System with Focus on Energy Optimization*, expone la elaboración de un concentrador basado en Raspberry Pi. Este proyecto utiliza el concentrador para comunicarse con dispositivos caseros inteligentes que hablen el protocolo Z-wave, tales como termómetros, medidores de energía, tomas de corriente, lámparas etc., y se conecta a la nube de Microsoft Azure. El desarrollo incluye la elaboración de una aplicación en Android para la interfaz de usuario. Este trabajo está enfocado a optimización, por lo que incluye minería de datos y algoritmos de optimización. Para la programación local de la Raspberry utilizaron Home Assistant (Hansen, Tewes y Foldager 2018).

Capítulo 2

Marco teórico

2.1. Internet de las Cosas

La primera vez que se usó el término [Internet de las Cosas](#) (IoT por sus siglas en inglés), fue como parte del nombre de una presentación en Procter & Gamble (P&G) realizada por el cofundador y director ejecutivo de Auto-ID Center Kevin Ashton (Ashton 2009). Ashton recalca que en aquel tiempo la información accesible en Internet era en su gran mayoría ingresada al sistema por seres humanos, y por tanto esta información llevaba consigo las características inherentes al hombre, tales como, ser de generación y manejo lento, propensa a errores, sujeta a interpretaciones y limitada en calidad y cantidad. Como alternativa proponía la posibilidad de que los sensores y actuadores que generaban las mediciones y controlaban el mundo real, fueran conectados directamente al sistema evitando así la intervención humana.

Según la Unión Internacional de Telecomunicaciones (International Telecommunication Union), *el IoT es una infraestructura global para la sociedad de la información, que permite la prestación de servicios avanzados mediante la interconexión (física y virtual) de elementos basados en las tecnologías de la información y la comunicación interoperables existentes y en evolución* ([Overview of the Internet of Things 2012](#)).

Ellos mismos definen una 'Cosa' como *un objeto del mundo físico (cosas físicas) o del mundo de la información (cosas virtuales), que es capaz de ser identificado e integrado en redes de comunicación* ([Overview of the Internet of Things 2012](#)).

Las cosas físicas existen en el mundo físico y son capaces de ser detectadas, conectadas y controladas, ejemplos de cosas físicas incluyen robots industriales, autos, relojes, computadoras, celulares, aparatos eléctricos, equipos de medición etc. Las cosas virtuales existen en el mundo de la información y son capaces de ser referenciadas, almacenadas, y procesadas, ejemplos de cosas virtuales incluyen contenido multimedia y aplicaciones de software.

Las cosas físicas pueden ser representadas como cosas virtuales vía un mapeo a memoria, sin embargo las cosas virtuales pueden existir sin estar asociadas a una cosa física (Figura 2.1) ([Overview of the Internet of Things 2012](#)).

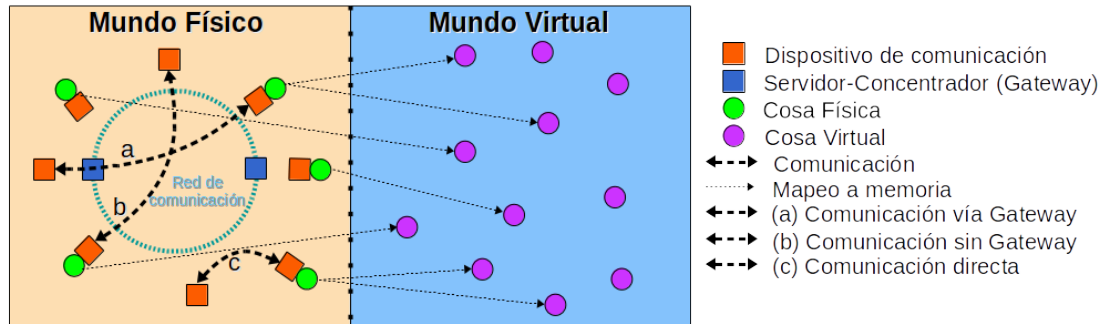


FIGURA 2.1: Representación visual del IoT ([Overview of the Internet of Things 2012](#))

Los dispositivos de comunicación, como su nombre lo dice, deben contar como mínimo con capacidades de comunicación y opcionalmente pueden medir variables, leer información, almacenar datos, controlar variables y procesar datos.

Los dispositivos se comunican con otros dispositivos a través de la red de comunicación por medio de un Gateway (caso a), a través de la red de comunicación sin Gateway (caso b) o directamente, es decir, sin utilizar la red de comunicación (caso c). Además, son posibles las combinaciones de casos a y c, y casos b y c; por ejemplo, los dispositivos pueden comunicarse con otros dispositivos utilizando comunicación a través de una red local con o sin Gateway, este es el caso que aplica para este proyecto de Tesis, comunicación entre dispositivos por medio de un Concentrador en una red local, sin conexión a una Nube.

Es importante aclarar que no obstante la Figura 2.1 sólo indica interacciones entre dispositivos de comunicación en el mundo físico, las interacciones también se pueden dar entre cosas virtuales y entre cosas virtuales y cosas físicas. Las redes IoT pueden hacer uso de redes ya existentes como las redes TCP/IP o de otras tecnologías como BLE, ZigBee, RfId o comunicación mediante la línea de potencia (PLC), por mencionar algunos ejemplos.

Como se puede observar, el IoT agrega la dimensión de “Comunicación de cualquier cosa” a las dimensiones ya existentes de “En cualquier momento” y “En cualquier lugar” (Figura 2.2).

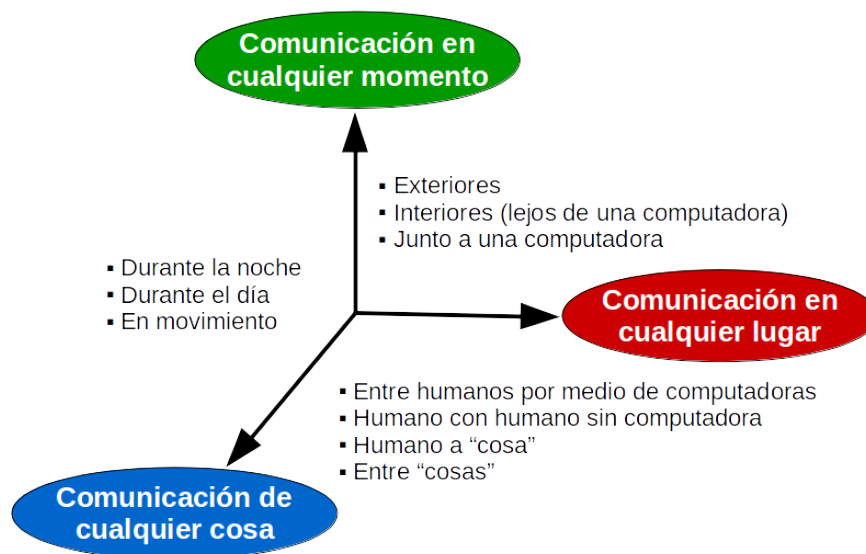


FIGURA 2.2: La nueva dimensión de comunicación introducida con el IoT ([Overview of the Internet of Things 2012](#))

2.1.1. Características fundamentales del IoT

Entre las características más importantes que definen el IoT, se encuentran las siguientes ([Overview of the Internet of Things 2012](#)):

- **Interconectividad:** cualquier cosa puede interconectarse con la infraestructura global de información y comunicación.
- **Servicios enfocados a las cosas:** La red IoT es capaz de brindar dentro de las limitaciones que una cosa representa, servicios especializados como privacidad y consistencia semántica entre la cosa física y la cosa virtual asociada.
- **Heterogeneidad:** los dispositivos en la IoT son heterogéneos basados en distintas plataformas de software y hardware, no obstante pueden interactuar con otros dispositivos o plataformas de servicio a través de diferentes redes.
- **Cambios dinámicos:** el estado de las cosas así como el número de cosas conectadas a la red puede cambiar en cualquier momento.
- **Capacidad de manejo de dispositivos enorme,** se espera que el número de cosas conectadas a la red crezca mínimo una orden de magnitud; las comunicaciones humanas se irán automatizando, siendo suplidas paulatinamente por comunicaciones entre cosas.

2.1.2. Modelo de referencia IoT

El modelo IoT está compuesto de cuatro capas con capacidades de gestión y seguridad asociadas a las cuatro capas ([Overview of the Internet of Things 2012](#)) (Figura 2.3).

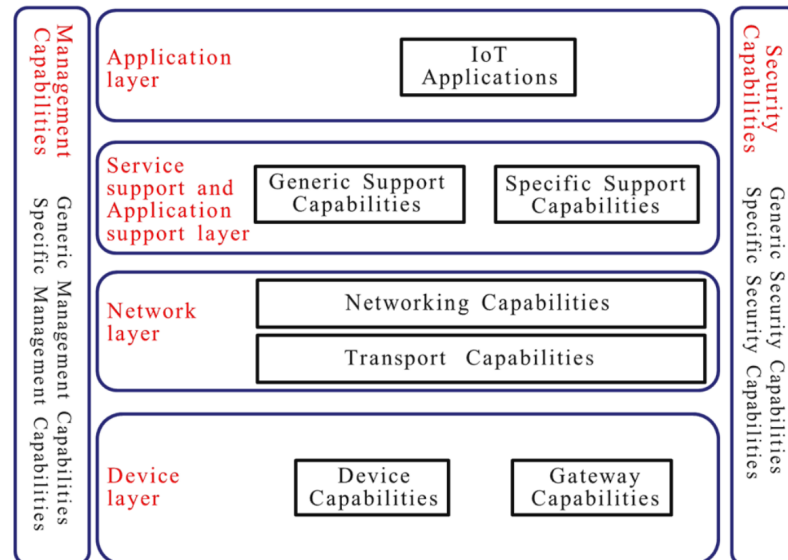


FIGURA 2.3: Modelo de Referencia IoT ([Overview of the Internet of Things 2012](#))

- Capa de aplicación (application layer): esta capa contiene las aplicaciones correspondientes a IoT.
- Capa de soporte para los servicios y las aplicaciones (service support and application support layer): contiene el soporte para servicios comunes compartidos por diferentes aplicaciones IoT tales como procesamiento y almacenamiento de datos, y también contiene soporte para los diversos servicios específicos que distinguen cada una de las aplicaciones IoT.
- Capa de red (network layer): contiene las funcionalidades de red tales como control de acceso, transporte, gestión de movilidad, autenticación etc.
- Capa de dispositivo (device layer): contiene las funcionalidades de los dispositivos y de los gateways tales como la configuración y manera de interconectarse a la red, si son dispositivos con ahorro de energía, manejo de protocolos y conversión de los mismos etc.

2.1.3. Arquitectura IoT

La arquitectura del IoT se conforma a grandes rasgos de los *dispositivos IoT o cosas*, *gateway(s)*, *la nube* y *dispositivos de visualización* que desde la percepción de la red, también son catalogados como *cosas*, solo que se diferencian por la clase de función que desempeñan, ya que el *dispositivo IoT* por lo general contiene sensores e ingresan datos a la red, a diferencia de los *dispositivos de visualización* que sirven más para leer y tomar acciones de control. Como se puede observar en la [Figura 2.1](#), esta arquitectura puede tener varias configuraciones en las que se puede prescindir de alguno de estos elementos o contar con múltiples versiones de ellos. La arquitectura general del IoT se muestra en la [Figura 2.4](#).

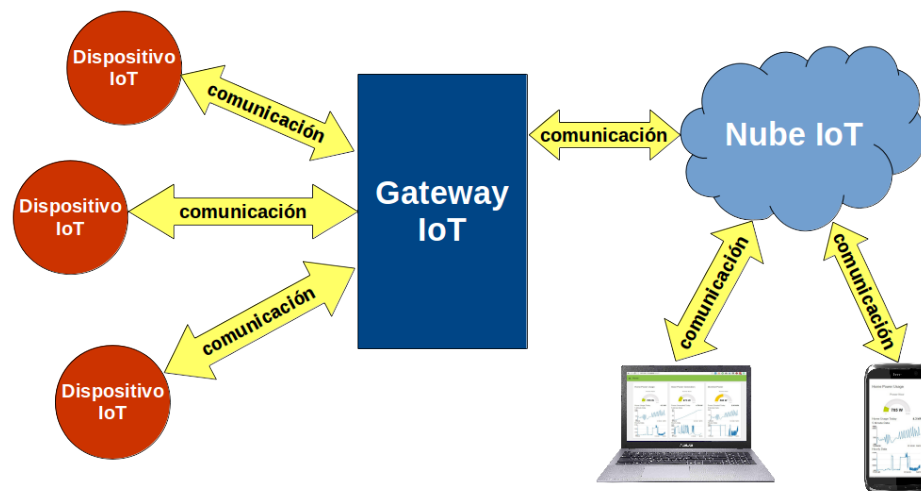


FIGURA 2.4: Arquitectura IoT

Los avances tecnológicos de los últimos años, la miniaturización de los procesadores, el aumento de la eficiencia energética y el aumento de la capacidad de procesamiento han empujado para que el IoT se vuelva una realidad, sin embargo conforme el IoT ha ido creciendo, esta tendencia se ha revertido y ahora muchas tecnologías se crean para satisfacer las necesidades particulares del IoT (K. K. Patel, S. M. Patel y Scholar 2016).

En las siguientes secciones se mencionaran algunas de las tecnologías estrechamente relacionadas con el desarrollo del IoT y los exponentes más representativos de cada tecnología.

2.2. Redes de Baja Potencia de Área Amplia (LPWANs)

Cuando se utiliza la configuración de IoT con gateway/concentrador, se abre un mar de posibilidades para que la comunicación entre el gateway y las cosas sea distinta a Ethernet o Wifi. Un gran porcentaje las cosas conectadas a la red actualmente son elementos de baterías y en movimiento, por lo que el IoT está estrechamente asociado con tecnologías de comunicación inalámbricas; debido a los requerimientos particulares de los dispositivos IoT, desde hace ya algunos años se ha empezado a utilizar *las Redes de Baja Potencia de Área Amplia* o LPWANs (por sus siglas en inglés), cuyas principales características son las siguientes:

Distancias largas Los nodos pueden situarse hasta 10 kilómetros de distancia del gateway dependiendo de la tecnología.

Velocidades de transmisión de datos bajas Proyectos típicamente con velocidades menores a los 5 kbps y con paquetes de entre 20 y 256 bytes.

Proyectos de bajo consumo Usualmente son parte de proyectos de baterías con tiempos de vida de entre 5 y 10 años.

Las redes LPWAN se utilizan cuando otras redes inalámbricas tales como Bluetooth, WiFi y ZigBee no son adecuadas por su corto alcance, y las redes celulares no se ajustan a los requerimientos de consumo y precio. Las redes LPWAN utilizan la topología de estrella, similar a Wifi donde los nodos se conectan a un gateway o punto de acceso. Las tecnologías LPWAN generalmente operan con aproximadamente 140-160 decibelios (dB) de ruta total, lo que puede sumar muchos kilómetros de alcance en las circunstancias adecuadas. Esto se logra principalmente por las altas sensibilidades del receptor. Las sensibilidades del receptor de más de -130 dBm son comunes en las tecnologías LPWAN, en comparación con los -90 a -110 dBm observados en muchas tecnologías inalámbricas tradicionales. Las tecnologías con -130 dBm pueden detectar señales 10,000 veces más débiles que las tecnologías con -90 dBm, parte de donde radica la gran importancia de las LPWAN (LinkLabs 2016).

Las redes LPWAN cumplen exactamente con las características para hacer el IoT viable, tales como requerimientos de bajo consumo, enlaces de comunicación largos y como consecuencia del pequeño tamaño de la información que se genera no son necesarias velocidades de transmisión grandes. Como ejemplos de estas tecnologías se encuentran Sigfox, LoRa, y NB-IoT entre otros. Para la comunicación del nodo

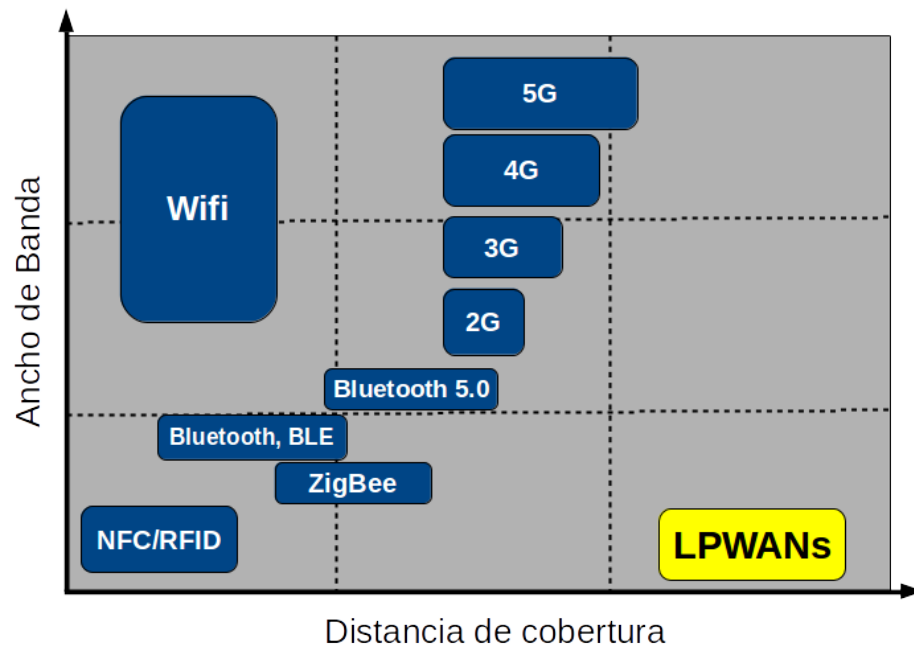


FIGURA 2.5: Distancia de cobertura vs. ancho de banda de diversas tecnologías de comunicación inalámbrica.

medidor con el concentrador se utilizó LoRa, por lo que se va a ahondar un poco más en esta tecnología.

2.2.1. Sigfox

Sigfox emplea la modulación por desplazamiento de fase binaria diferencial (DB-PSK) en una portadora de banda sub-GHz ISM de banda ultra estrecha (100 Hz). Sigfox utiliza bandas ISM sin licencia, por ejemplo, 868 MHz en Europa, 915 MHz en Norteamérica y 433 MHz en Asia. Al emplear la banda ultraestrecha, Sigfox utiliza el ancho de banda de frecuencia de manera eficiente y experimenta niveles de ruido muy bajos, lo que conduce a un consumo de energía muy bajo, una alta sensibilidad del receptor y un diseño de antena de bajo costo a expensas de un rendimiento máximo de sólo 100 bps (Mekki y col. 2018).

2.2.2. LoRa

LoRa es una tecnología de comunicación de datos inalámbrica digital, patentada y desarrollada por la empresa Cycleo en Grenoble Francia en el 2009, luego adquirida por Semtech en el 2012. LoRa utiliza bandas de frecuencia de radio sub-gigahertz sin

licencia como 169 MHz, 433 MHz, 868 MHz (Europa) y 915 MHz (América del Norte). LoRa permite transmisiones de muy largo alcance (más de 10 km en áreas rurales) con bajo consumo de energía y con velocidades típicas de entre 300 bps y 50 kbps. El máximo tamaño de trama es de 243 bytes (Mekki y col. 2018).

La tecnología se presenta en dos partes: LoRa en la capa física y LoRaWAN en las capas superiores¹. La etapa física de LoRa es cerrada y no es posible tener mucha información oficial. En cuanto a la red LoRaWAN, esta más relacionada con la infraestructura de red dedicada a comunicación entre Gateways LoRa, sin embargo no aplica para este proyecto ya que nuestro concentrador no va tener capacidades de Gateway y por tanto no se tiene planeado conectarse a LoRaWAN.

2.3. Computadoras de una sola placa

En todos elementos del IoT se necesita procesamiento, almacenamiento y comunicación. La disminución de los precios de los procesadores y en general de las computadoras han sido la clave del aumento exponencial del número de elementos IoT. En particular en la última década ha habido un aumento del uso de las computadoras de una sola placa, que son ordenadores completos armados en una sola placa de circuito impreso. Estas computadoras contienen microprocesador, memorias, puertos de entrada/salida y otras características extras requeridas por cualquier computadora funcional. Entre ejemplos de este tipo de computadoras se encuentra Arduino, Banana Pi, Beaglebone, Orange Pi y Raspberry Pi, por mencionar algunos. A continuación se describen brevemente algunas de estas computadoras.

2.3.1. Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar. El proyecto **Arduino** comenzó en 2003 como un programa para estudiantes del Instituto de Diseño de Interacción Ivrea en Ivrea Italia, con el objetivo de proporcionar una plataforma sencilla y de bajo costo que le permitiera a principiantes y profesionales crear dispositivos que interactuaran con su entorno utilizando sensores y actuadores. Algunas de las placas Arduino más conocidas se muestran en la [Figura 2.6](#).

¹LoRaWAN es una red de área amplia de largo alcance basada en LoRa y estandarizada por LoRa-Alliance.

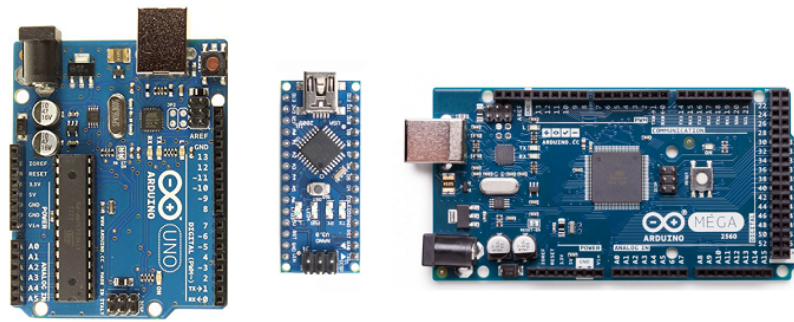


FIGURA 2.6: Diversos modelos de placas Arduino

Entre las principales ventajas que Arduino ofrece son las siguientes:

1. Económico, las placas Arduino se encuentran con costos menores a \$50 dólares.
2. Disponible para varias plataformas, el software de Arduino (IDE) se ejecuta en sistemas operativos Windows, Macintosh OSX y Linux.
3. Entorno de programación simple y claro, el software de Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible como para que los usuarios avanzados también lo aprovechen.
4. Software de código abierto, el software de Arduino se publica como herramientas de código abierto, disponibles para ser modificado por programadores experimentados. El lenguaje puede ser expandido a través de las librerías C++, y la gente que quiera entender los detalles técnicos puede dar el salto desde Arduino al lenguaje de programación AVR C en el que se basa. Del mismo modo, puedes añadir código AVR-C directamente a tus programas de Arduino si lo desea.
5. Hardware de código abierto, los esquemáticos de las placas Arduino se publican bajo una licencia Creative Commons, por lo que los diseñadores de circuitos experimentados pueden hacer su propia versión del módulo, ampliándolo y mejorándolo. Incluso los usuarios relativamente inexpertos pueden construir la versión de protoboard del módulo para entender cómo funciona y ahorrar dinero.

En la [Tabla 2.1](#) se pueden observar las especificaciones de la placa **Arduino UNO**, una de las tarjetas arduino más usadas y documentadas.

Especificación	Valor
Microcontrolador	ATmega328P
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pines digitales de E/S	14 (de los cuales 6 proporcionan salida PWM)
PWM Digital E/S Pins	6
Clavijas de entrada analógica	6
Corriente DC por Pin de E/S	20 mA
Corriente DC para 3.3V Pin	50 mA
Memoria flash	32 KB (ATmega328P) de los cuales 0.5 KB utilizados por el bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
Longitud	68.6 mm
Anchura	53.4 mm
Peso	25 g

TABLA 2.1: Especificaciones de la placa Arduino UNO

2.3.2. BeagleBone

BeagleBone es una computadora de una sola tarjeta de código abierto y de baja potencia producida por Texas Instruments en asociación con Digi-Key y Newark element14. Digi-Key proporcionó los fondos para la producción original, pero ahora la distribución se ha abierto a docenas de distribuidores en todo el mundo. Texas Instruments generosamente le permite a Jason Kridner, administrador de la comunidad, dedicar tiempo para brindar apoyo y desarrollo del proyecto BeagleBoard.org como parte de sus obligaciones en Texas Instruments.

Todos los diseños son completamente de código abierto y los componentes están disponibles para que cualquiera pueda fabricar hardware compatible. Las placas son computadoras de una sola placa de bajo costo y sin ventilador, basadas en procesadores de baja potencia de Texas Instruments que presentan el núcleo de la serie ARM Cortex-A, con toda la capacidad de expansión de las máquinas de escritorio de hoy en día, pero sin el volumen, el costo o el ruido. Algunas de estas placas se pueden ver en la [Figura 2.7](#).

En la [Tabla 2.2](#) se pueden observar las especificaciones de la placa [BeagleBone Black](#).

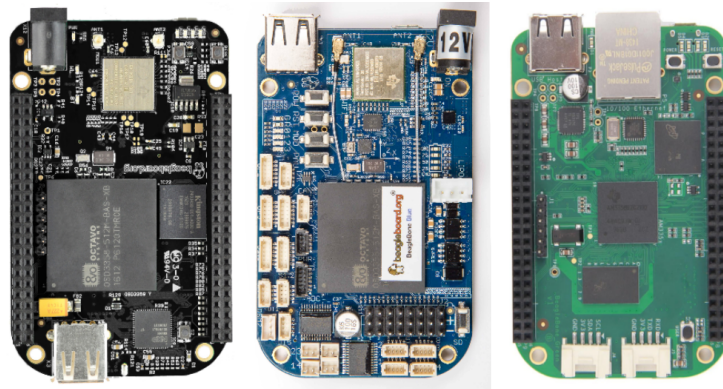


FIGURA 2.7: Diversos modelos de placas BeagleBone

Especificación	Valor
Procesador	AM3358 ARM Cortex-A8
Velocidad máxima del procesador	1GHz
Pins analógicos	7 (1.8V)
Pines digitales	65 (3.3V)
Memoria	512 MB DDR3, 2 GB eMMC ranura para tarjeta microSD
USB	Puerto de cliente miniUSB 2.0 puerto de host USB 2.0
Vídeo	microHDMI
Audio	microHDMI
Interfaces compatibles	4 x UART, 12 x PWM/Temporizadores, LCD, GPMC, MMC1, 2 x SPI, 2 x I2C, convertidor A/D, 2 x CAN Bus
Precio	\$49

TABLA 2.2: Especificaciones de la placa BeagleBone Black

2.3.3. Raspberry Pi

La Raspberry Pi es una pequeña computadora de una sola placa PCB desarrollada en el Reino Unido por la **Fundación Raspberry Pi** para promover la enseñanza de programación en las escuelas y en los países en vías de desarrollo (Figura 2.8).

Algunas características generales de los dispositivos Raspberry Pi incluyen lo siguiente (Molloy 2016):

1. Son de bajo costo, disponibles por tan solo \$10- \$40 dólares.
2. No obstante de su tamaño cuentan con alta capacidad de procesamiento. Por ejemplo, la Raspberry Pi 3 contiene un procesador ARM Cortex-A53 de 1.2 GHz que puede realizar más de 700 millones de instrucciones por segundo (MIPS).

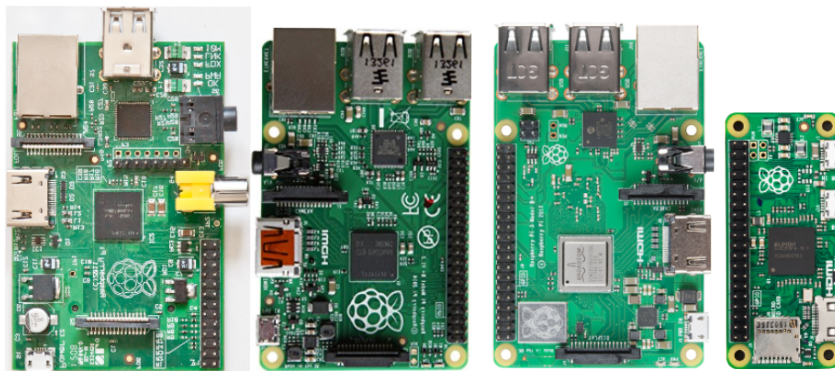


FIGURA 2.8: Algunos modelos de Raspberry Pi, de izquierda a derecha, modelo A, B+, 3B+ y raspberry Pi Zero W respectivamente.

3. Están disponibles en una gama de modelos que son adecuados para diferentes aplicaciones (por ejemplo, el RPi 3 de formato más grande para creación de prototipos y el pequeño formato RPi Zero o Módulo Compute para su implementación).
4. Admiten muchas interfaces estándar para dispositivos electrónicos.
5. Usan poca energía, funcionando entre aproximadamente 0.5 W (RPi zero cuando está inactiva) y aproximadamente 5.5 W (RPi 3 bajo carga).
6. Se pueden expandir mediante el uso de tarjetas secundarias y dispositivos USB conectados a hardware conectado (HAT).
7. Son apoyados por una gran comunidad de innovadores y entusiastas, que generosamente dedican su tiempo a ayudar a la Fundación RPi con su misión educativa.

La plataforma RPi puede ejecutar el sistema operativo Linux, lo que significa que puede usar muchas bibliotecas y aplicaciones de software de código abierto directamente. Es por esta flexibilidad que la RPi es uno de las computadoras de una sola placa más usadas en el IoT (Maksimović y col. 2015).

En la [Tabla 2.3](#) se pueden observar las especificaciones de la [Raspberry Pi Zero W](#) ([Raspberry Pi Zero W 2018](#)).

Especificación	Valor
Procesador	Procesador de un núcleo 1GHz Broadcom BCM2835
RAM	512 MB
Puertos	Mini HDMI y USB On-The-Go
Alimentación	Por Conector Micro USB
Pines	40, compatible con las demás Raspberrys
Video	Compuesto con reinicio de encabezados
Camara	Conector CSI
LAN	inalámbrica 802.11 b/g/n
Bluetooth	4.1, Low Energy (BLE)

TABLA 2.3: Especificaciones de la Raspberry Pi Zero W

2.4. Lenguajes de programación

Los lenguajes de programación que se utilizan en el IoT no difieren mucho de los programas que se utilizan para hacer aplicaciones o paginas web (ver [Figura 2.9](#)), siendo los más comunes C, C++, Java, Python, JavaScript y PHP; sin embargo recientemente se han empezado a utilizar herramientas de programación más abstractas, orientadas a utilizar menos código e interconectar bloques o elementos para formar la lógica del programa, entre ejemplos de este tipo de programación se encuentran IFTTT (If This Then That), Home Assistan, Openhab, Node-RED etc., de entre todos estos ejemplos se distingue "Node-RED", ya que se utiliza por medio de un editor gráfico muy sencillo y a que con el tiempo ha desarrollado una relación muy estrecha con el Internet de las Cosas, en esta herramienta es la que nos vamos a enfocar para el diseño del proyecto de tesis.

2.4.1. Node-RED

Node-RED inició a principios del 2013 como un proyecto de Nick O'Leary y Dave Conway-Jones, empleados de IBM. Originalmente solo era una especie de prueba de concepto para visualizar y manipular mapeos de tópicos MQTT, sin embargo rápidamente se convirtió en una herramienta muy general que podía fácilmente extenderse en cualquier dirección. Desde septiembre del 2013 Node-RED ha sido un proyecto de código abierto y a partir de octubre del 2016 pasó a ser parte de la **Fundación JS**. Node-RED se puede utilizar para solucionar todo tipo de aplicaciones, pero con el tiempo ha evolucionado y se ha relacionado tanto con el IoT, que en

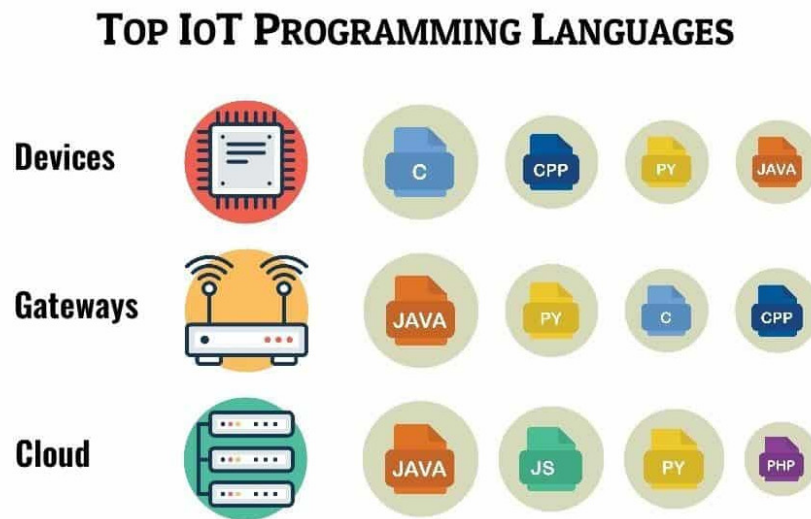


FIGURA 2.9: Lenguajes de programación usados en IoT

Fuente: Eclipse Foundation

la página de Node-RED se anuncian como *Programación para el Internet de las Cosas* (Figura 2.10).

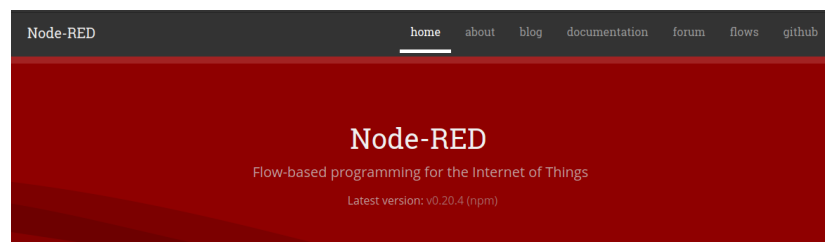


FIGURA 2.10: Node-RED programación para el Internet de las Cosas

Node-RED es un editor visual de programación de código abierto, que permite a los desarrolladores de software de diversos niveles, interconectar todo tipo de entradas y salidas físicas, sistemas basados en la nube, bases de datos y un gran número de APIs (Application Programming Interface) entre otros. Estas características de Node-RED permiten de una manera intuitiva la realización de prototipos mucho más rápido de lo que representaría empezar de cero el desarrollo de un proyecto similar.

Node-RED esta desarrollado sobre **NodeJS**, y por ende es muy ligero, orientado a eventos y bajo el esquema de código sin bloqueos, esto permite que Node-RED se pueda instalar en hardware muy económicos como computadoras de una sola placa o en la nube. Actualmente existen más de 225,000 módulos que permiten extender la funcionalidad de Node-RED.

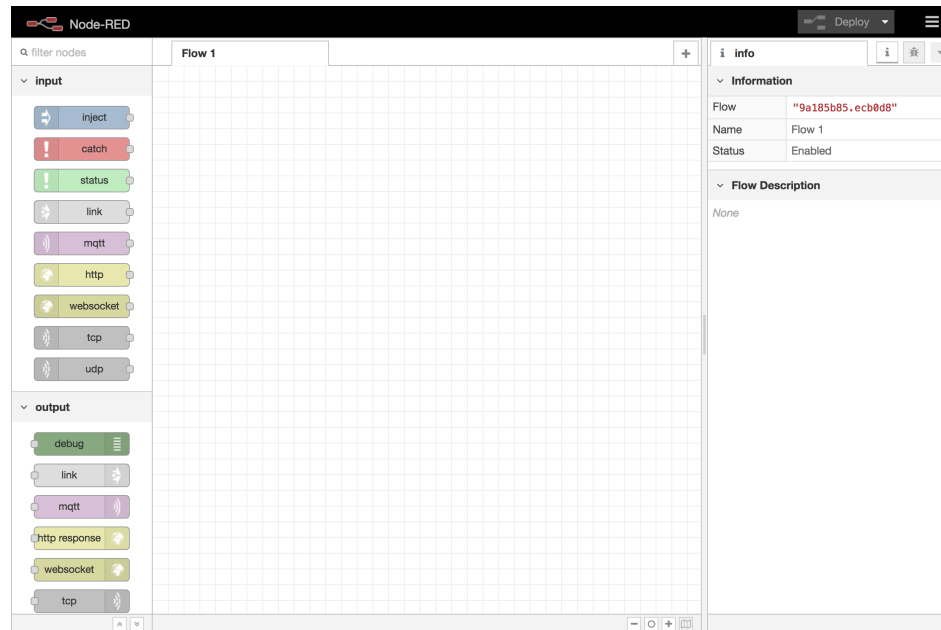


FIGURA 2.11: Ventana de navegador donde se despliega el editor de Node-RED.

El editor de flujos de Node-RED utiliza como interfaz gráfica el navegador de Internet (Figura 2.11). Este editor se conforma de cuatro partes (Figura 2.12):

1. Encabezado (Header), contiene el menu principal y el botón "Deploy", que ejecuta el equivalente a una compilación.
2. La paleta (Palette), que contiene los nodos disponibles para uso.
3. El área de trabajo (Workspace), que es donde se interconectan los nodos para hacer los "programas".
4. La barra lateral (Sidebar), que contiene paneles que proporcionan varias herramientas útiles dentro del editor.

Dentro del editor se crean las aplicaciones arrastrando de la "Paleta" al "Área de trabajo", pequeños módulos llamados "Nodos". Estos "nodos" al conectarse entre sí forman entes más grandes conocidos como "Flows" o "Flujos". Los flujos representan por lo general la realización de una sola tarea y a su vez interactúan entre sí con otros flujos para la realización de soluciones integrales completas, un ejemplo de un flujo se puede observar en la Figura 2.13

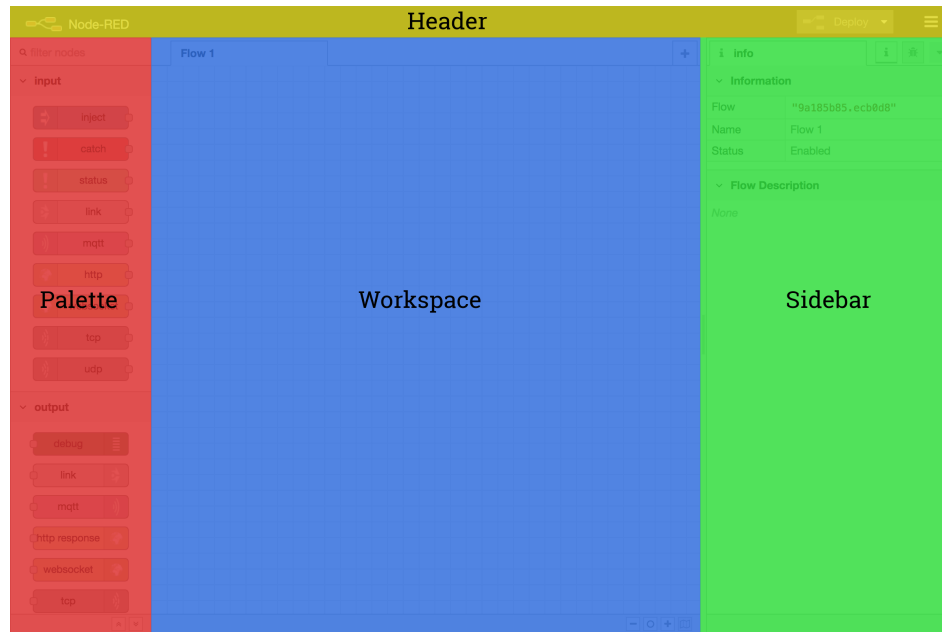


FIGURA 2.12: Principales partes que integran el editor de Node-Red.

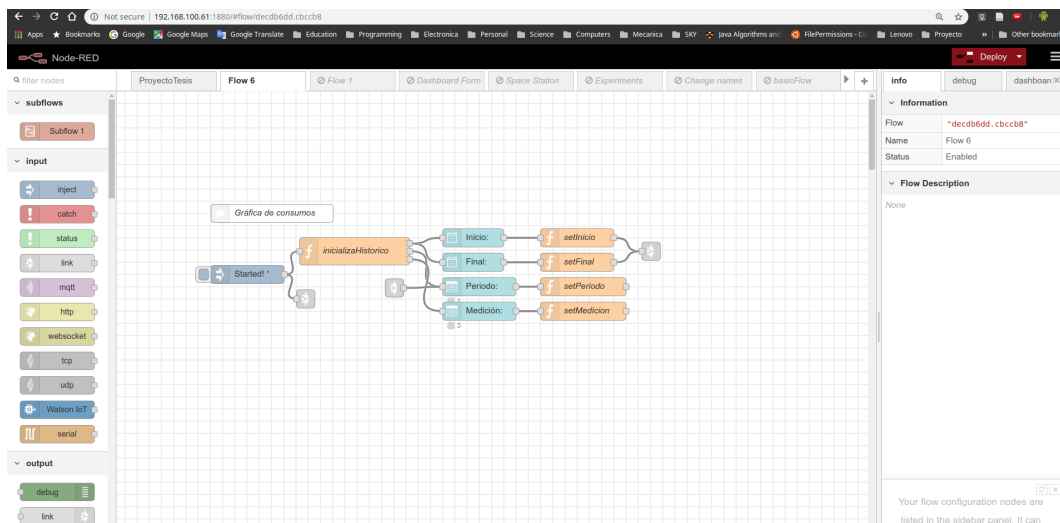


FIGURA 2.13: Ejemplo de la interconexión de un flujo de nodos.

La paleta de nodos puede ser ampliada al instalar nuevos nodos creados por la comunidad.

Capítulo 3

Metodología

3.1. Descripción del medidor de energía IoT

Este proyecto se complementa con el trabajo de la tesis *Desarrollo de hardware para integración de medidores de energía eléctrica a plataforma IoT*, desarrollada por el ingeniero Marcos Ariel Olivera Moreno, correspondiente al nodo medidor de energía. Esta tesis incluye la parte del hardware que se comunica con el medidor de energía eléctrica y la comunicación con el nodo concentrador, juntos el nodo medidor de energía y el nodo concentrador, objeto de esta tesis, conforman un medidor de energía IoT.

Por cuestiones de tiempo, el alcance de este medidor de energía se limitó a una configuración IoT con Servidor-Concentrador (Gateway IoT) y sin servicio de nube (ver [Figura 2.1](#)). Esto implica que el nodo medidor se conecta directamente al gateway y los dispositivos de visualización leen los datos almacenados en el gateway por medio de una red local ([Figura 3.1](#)).

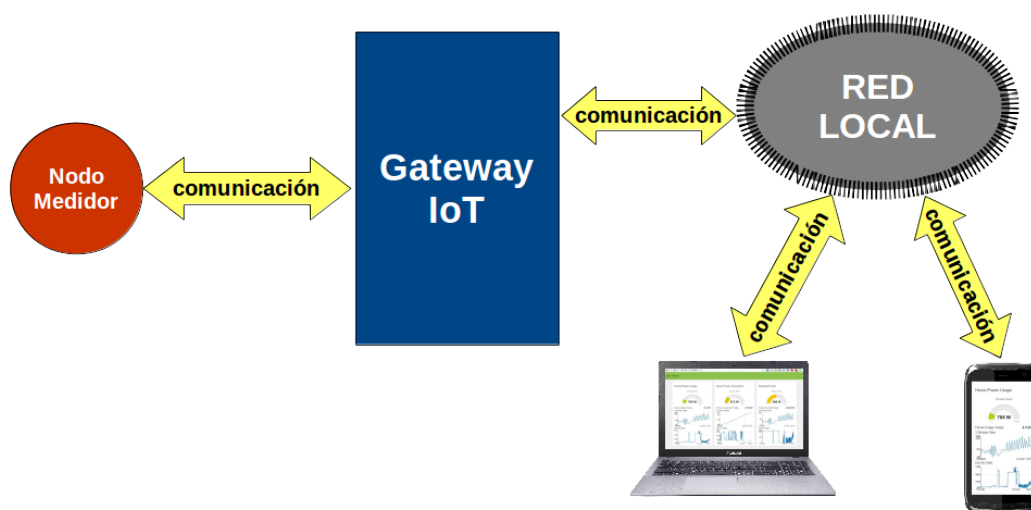


FIGURA 3.1: Medidor de energía IoT.

Es muy importante recalcar que hay diferencias entre las funciones de un gateway, un concentrador y un servidor, ya que el gateway implica un punto de conexión a otra red, el concentrador procesa y almacena de datos recibidos de varias fuentes y un servidor provee servicios a clientes conectados en la red, sin embargo por la configuración tan sencilla del proyecto, el "nodo concentrador" desempeña de una manera básica las tres actividades, por tanto a lo largo de la tesis se estará mencionando indistintamente los términos nodo concentrador, concentrador, gateway, gateway IoT o concentrador-servidor, tratando de darle una predilección particular a la primera opción.

Varias aplicaciones de medidor de energía en la industria requieren el monitoreo de equipos alejados de alguna red de computadoras, por lo que no siempre es factible una conexión wifi cercana, por eso para la comunicación entre el nodo medidor y el nodo concentrador se optó por utilizar la tecnología LoRa.

3.2. Descripción del proyecto de tesis

El proyecto se conforma de tres partes principales (Figura 3.2):

1. La recepción de datos vía LoRa.
2. Almacenamiento y procesamiento de datos en el concentrador.
3. Comunicación y acceso a las variables medidas en los dispositivos de visualización vía un tablero de instrumentos o "dashboard".

El diagrama a bloques de la Figura 3.2 enumera las principales tecnologías utilizadas; en color amarillo y azul se muestran los bloques que forman parte del proyecto, y en gris se muestra el nodo medidor, indicando que no es parte directa de este proyecto.

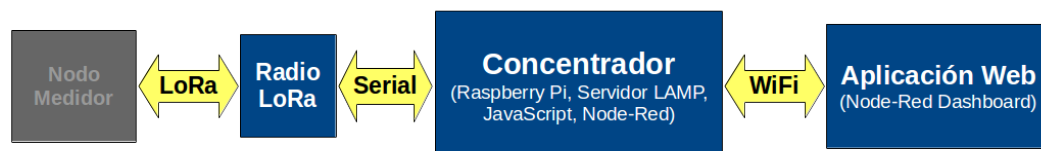


FIGURA 3.2: Diagrama a bloques simplificado.

La comunicación entre el nodo medidor y el concentrador se realiza por medio de LoRa, que es parte de un nuevo tipo de tecnologías para desarrollo de redes de comunicación inalámbrica de baja potencia con cobertura amplia y que por sus características son muy convenientes para su uso en el desarrollo de los nodos o cosas IoT.

Como hardware principal del concentrador se utilizó una Raspberry Pi que es una computadora pequeña de una sola placa PCB, del tamaño de una tarjeta bancaria, basada en los procesadores ARM y a pesar de su tamaño cuenta con grandes capacidades de procesamiento lo que lo hace ideal para ser usada como Concentrador-Servidor.

El servidor LAMP es una plataforma de desarrollo web de código abierto que utiliza Linux como sistema operativo (Raspbian), Apache como el servidor web, MySQL como el sistema de gestión de bases de datos relacionales y PHP como el lenguaje de scripting orientado a objetos.

El entorno de programación utilizado es una herramienta visual de navegador orientada al Internet de las Cosas llamada Node-Red, la cual por medio de funciones escritas en JavaScript y elementos visuales, permite en poco tiempo resolver tareas complejas.

El desarrollo de la aplicación web se desarrolló con una herramienta incluida en Node-Red llamada "Node-RED Dashboard".

El proyecto total se divide en varias partes más simples, que pueden ser abstraídas como actividades preliminares resueltas. Analizar el proyecto desde esta manera ayuda a entender la operación del algoritmo principal, ya que habiendo explicado más o menos a detalle cómo funcionan los procesos individuales, ver el proceso total será más sencillo. Estas partes se enumeran a continuación.

3.2.1. Comunicación con el módulo LoRa

Desde la percepción del sistema, los datos LoRa corresponden al punto de entrada de datos, los cuales se podrían adaptar sin mayor complejidad para ser recibidos con otra tecnología de comunicación, como por ejemplo inalámbrica o totalmente otro medio como cable o infrarrojo.

Para el desarrollo con LoRa se utiliza un módulo fabricado en china de aproximadamente \$10 dólares por pieza, modelo “E32-915T20D”, el cual es muy sencillo de utilizar, ya que la interacción con él es por medio de comunicación serial de interfaz UART (Figura 3.3). En las Tabla 3.1, Tabla 3.2, Tabla 3.3 se muestran las especificaciones más importantes de este módulo (E32-915T20D 2019).

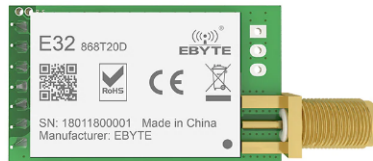


FIGURA 3.3: Imagen del módulo LoRa E32-915T20D utilizado.

Parámetro	Valor	Observación
Frecuencia de trabajo	900~931 MHz	Default: 915MHz
Potencia de transmisión	10~20 dBm	Default: 20dBm, Aprox. 100mW
Sensibilidad de recepción	-146 dBm	velocidad de transmisión: 0.3kbps
Velocidad de transmisión	0.3k~19.2kbps	Default: 2.4kbps
Distancia Máxima	3000 metros	En campo abierto y cielo despejado, potencia máxima, ganancia de antena 5dBi,-altura de 2 m, velocidad de transmisión 2.4kbps

TABLA 3.1: Especificaciones de radio frecuencia del módulo LoRa E32-915T20D.

Parámetro	Valor	Observación
Tamaño	21 * 36 mm	sin incluir SMA
Tipo de antenna	SMA-K	
Tipo de comunicación	UART	Baud rate: 1200~115200, default: 9600
Encapsulado	DIP	
Tamaño del buffer	512 bytes	Se empaquetan 58 bytes automáticamente por cada paquete

TABLA 3.2: Especificaciones de hardware del módulo LoRa E32-915T20D.

La función de cada uno de los pines del módulo se muestra en la Figura 3.4. La conexión se realiza directamente a la UART que expone la Raspberry pi en su conector GPIO que es un estándar para todos sus modelos (Monk 2014).

Esta primera versión del sistema no cuenta con señales de control, por lo que sólo se conectaron las señales de tx del módulo a la de rx de la Raspberry Pi y la respectiva alimentación del módulo.

Parámetro	Min.	Typ.	Max.	Unidades	Condición
Voltaje de alimentación	2.3	5	5.2	V	
Voltaje de comunicación	2.5	3.3	3.6	V	
Corriente durante Tx	110	120	130	mA	20dBm (W)
Corriente durante Rx	13	14	15	mA	
Corriente Dormido	3	4	5	μA	
Temperatura de operación	-40	20	85	$^{\circ}C$	
Nivel de humedad de operación	10	60	90	%	
Temperatura de almacenamiento	-40	20	125	$^{\circ}C$	

TABLA 3.3: Otras especificaciones importantes del módulo LoRa E32-915T20D.

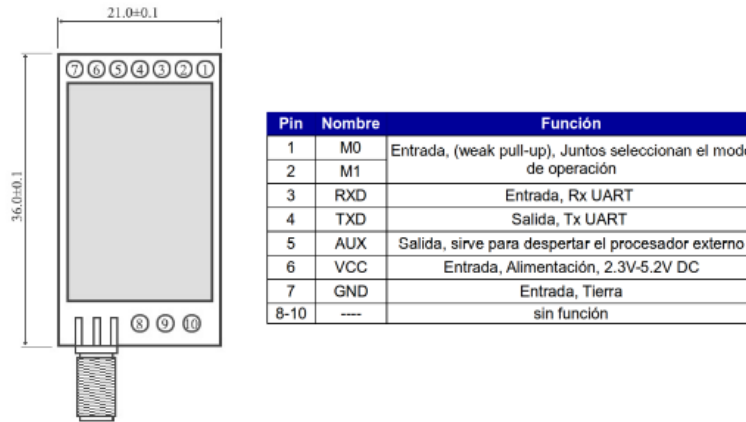


FIGURA 3.4: Definición de pines del módulo LoRa E32-915T20D

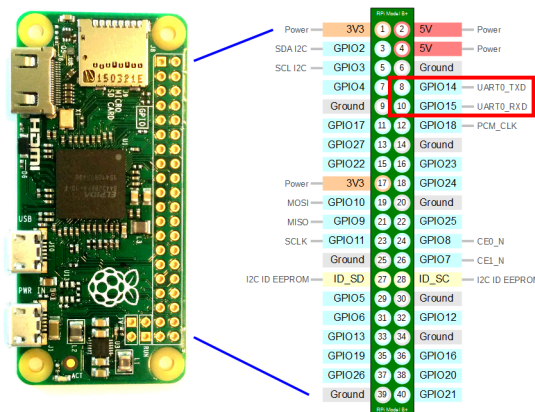


FIGURA 3.5: Puerto GPIO de la Raspberry Pi Zero y acceso a la UART

Una vez realizada la conexión física, es necesario entrar al Node-RED y seleccionar del grupo de nodos de entrada, el nodo serial, arrastrarlo al espacio de trabajo y darle doble-click para configurar el puerto. En esa pantalla es necesario seleccionar el puerto que corresponde al que se expone en el GPIO que normalmente aparece con el nombre

de `/dev/ttyS0`, una vez seleccionado el puerto es necesario darle al lápiz que aparece a la derecha del nombre del puerto para poder editar las configuraciones del mismo, tales como velocidad, paridad, bits de datos etc., esta pantalla de configuración se muestra en la [Figura 3.7](#). Para el proyecto se utilizaron 115200 baudios, con 8 bits de datos, sin paridad y con un bit de paro ([Raspberry Pi Serial \(UART\) Tutorial 2019](#)).

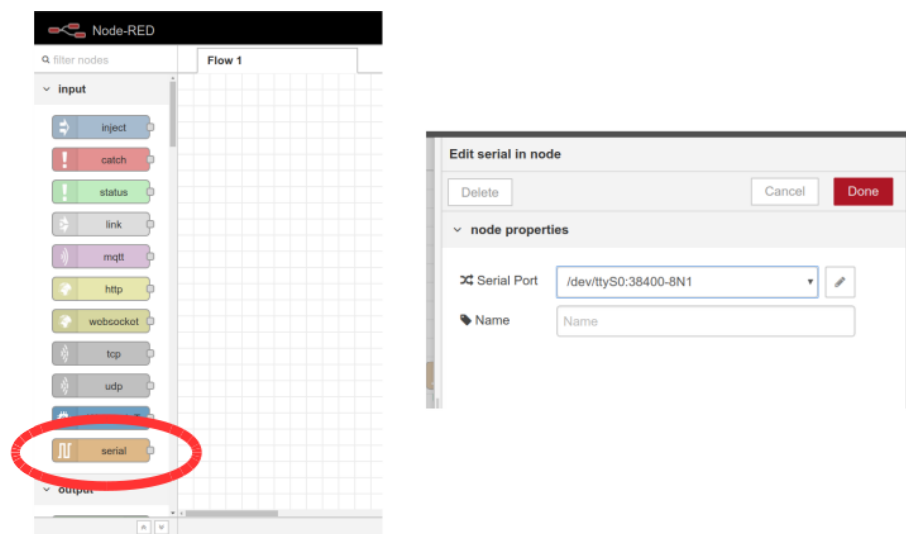


FIGURA 3.6: Selección y edición del nodo serial en Node-RED

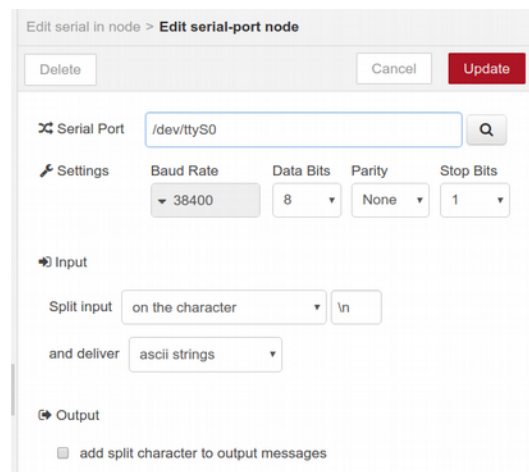


FIGURA 3.7: Configuración del puerto serial en Node-RED

Después de actualizar el proyecto y si todo ocurrió correctamente el módulo debe aparecer como conectado, en este punto los datos que manda el medidor de energía ya deben estar llegando a la salida del nodo serial de Node-RED (Shovic 2016).

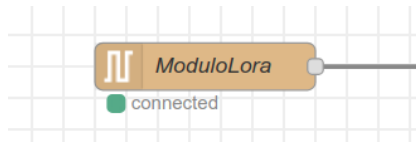


FIGURA 3.8: Nodo serial del Node-RED conectado

3.2.2. Almacenamiento en la base de datos

Para el almacenamiento de datos se creó por medio de la herramienta visual **phpMyAdmin**, una base de datos llamada *MedidorEnergia* (Figura 3.9). Esta base de datos cuenta con dos tablas, una llamada *variablesEstaticas* que se utiliza para guardar información que el usuario actualiza desde el tablero de instrumentos (dashboard) y otra tabla con el nombre *Mediciones* que almacena los valores medidos que llegan del medidor de energía. Para el medidor que se utiliza en este proyecto las variables medidas son voltaje, corriente, potencia activa, potencia aparente, potencia reactiva, factor de potencia, frecuencia, total de la energía activa y total de la energía reactiva. La estructura de cada una de estas tablas aparecen en la Figura 3.10 y la Figura 3.11 respectivamente (contributors 2018).

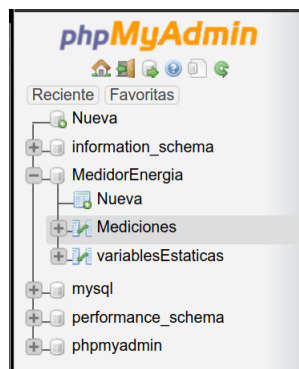


FIGURA 3.9: Base de datos del medidor de energía

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	uniqueKey	tinyint(4)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
2	facturacion	tinyint(4)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
3	fechaCorte	bigint(20)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
4	costoKwh	float			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
5	kwhMax	float			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
6	potMax	float			No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Más
7	email	text	latin1_swedish_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Más

FIGURA 3.10: Estructura de la tabla de variables estáticas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predefinido	Extra	Acción
1	estampaDeTiempo	timestamp	No	CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP		Cambiar Eliminar Primaria Único Más
2	voltaje	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
3	corriente	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
4	potenciaActiva	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
5	potenciaAparente	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
6	potenciaReactiva	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
7	factorDePotencia	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
8	frecuencia	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
9	totalEnergiaActiva	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más
10	totalEnergiaReactiva	float	No	Ninguna	No	Ninguna		Cambiar Eliminar Primaria Único Más

FIGURA 3.11: Estructura de la tabla de mediciones eléctricas

Al igual que como se realizó con el puerto serial para poder tener acceso a la base de datos desde el Node-RED es necesario seleccionar el nodo *mysql* del conjunto de nodos *storage* (Figura 3.12) y arrastrarlo al espacio de trabajo (MySQL Documentation 2018). para configurar el nodo es necesario darle doble-click asignarle un nombre (Figura 3.12) y seleccionar el pequeño lápiz que sale un lado donde se abren los parámetros del nodo, a continuación se llenan el usuario y la clave de acceso al *phpMyAdmin* y los valores adicionales que aparecen en la Figura 3.13, si no hubo ningún error en el proceso, el nodo debe aparecer como se muestra en la Figura 3.14.

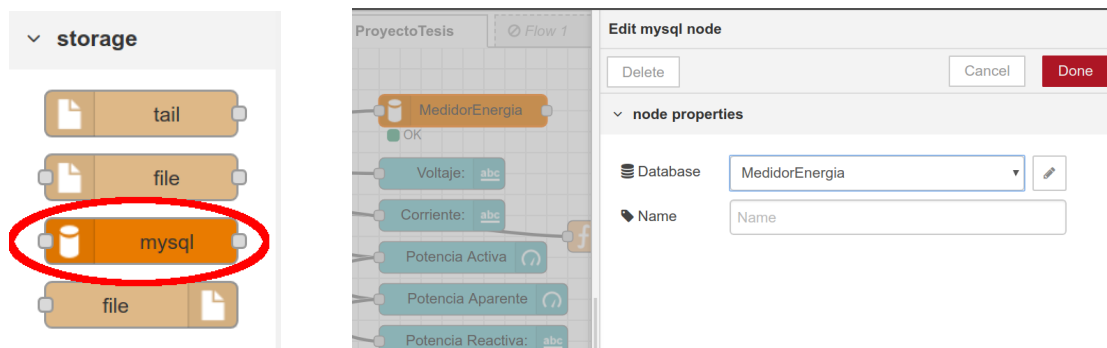


FIGURA 3.12: Conjunto de nodos “storage” y edición del nodo “mysql” en Node-RED

A este punto la base de datos ya se encuentra lista para recibir consultas construidas desde el NodeRED.

3.2.3. Visualización de datos por medio del “dashboard” de Node-RED

Así como Node-RED cuenta con nodos predefinidos para dar solución a problemas comunes en el Internet de las Cosas, tales como entrada de datos, salida de

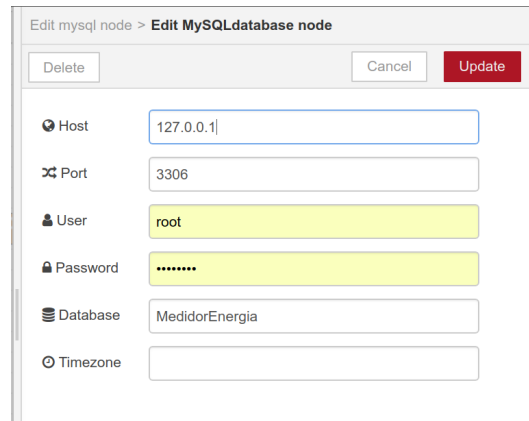


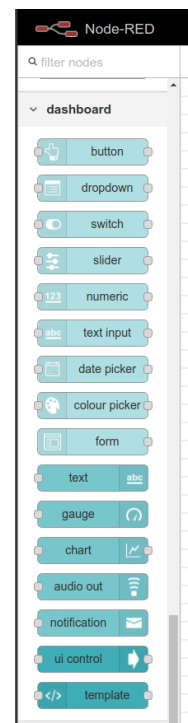
FIGURA 3.13: Configuración de los parámetros de la base de datos en Node-RED



FIGURA 3.14: Nodo “mysql” de Node-RED conectado

datos, almacenamiento etc., también cuenta con una sección dedicada a la interfaz de usuario llamada “*dashboard*”.

El dashboard cuenta con nodos que corresponden a los “instrumentos” más comunes que se usan para poder interactuar con el usuario, para hacer uso de ellos es necesario arrastrar alguno de los elementos del grupo de nodos “dashboard” que se muestra en la figura de un costado al espacio de trabajo y configurar los respectivos parámetros. Estos “instrumentos virtuales” normalmente tratan de emular a los equipos de instrumentación que se pueden encontrar en un cuarto de máquinas y pueden ser de entrada para recibir información por parte del usuario, o de salida para mostrarle el estatus de algún proceso. Cada nodo en el espacio de trabajo se refleja en el *UI dashboard* (Tablero de Instrumentos) como una imagen del instrumento que representa; la palabra *UI* proviene de las palabras “Interfaz de Usuario” por las siglas en ingles de *User Interface*. Para tener acceso a este “Tablero de Instrumentos” es necesario ingresar a la URL - *host:1800/ui*.



Entre el conjunto de nodos se encuentran los de entrada con color azul claro, en un azul un poco más oscuro los nodos de salida y por último con el azul mas oscuro los nodos especiales de *ui control* que sirven para realizar control dinámico del dashboard, y *template* que sirve para cambiar también de manera dinámica la apariencia de los elementos de la interfaz de usuario.

La mayoría de los nodos cuentan con parámetros comunes que sirven para configurar la manera en que se visualizan los instrumentos. Los más comunes son:

- **Group.-** sirve para asociar instrumentos relacionados en un solo grupo para poderlos tratar como una unidad en el tablero de instrumentos; agruparlos ayuda a definir la estructura del tablero y para construir dinámicamente la manera en que el tablero se visualiza en dispositivos con pantallas de distintos tamaños.
- **Size.-** especifica la altura y ancho de visualización del instrumento medido en unidades, (el tablero de instrumentos consta de una cuadrícula que se mide en “unidades”, donde cada unidad mide 48 pixeles por defecto); un valor de “auto” en este parámetro indica que el instrumento se va a ajustar al tamaño del grupo al que pertenece.
- **Label.-** es la etiqueta que se muestra en la parte de arriba del instrumento.
- **Value.-** no todos los instrumentos cuentan con este parámetro, equivale al valor que muestra el instrumento y se le puede dar formato por medio de filtros de *angular*, por ejemplo un valor de `{{value | number:1}}%` en este parámetro indica que se quiere redondear el valor a un decimal y con el símbolo de “%” al final del valor.

Cubrir el funcionamiento de todos los nodos del dashboard y de sus respectivos parámetros escaparía del objetivo de esta tesis, sin embargo a continuación se explica el funcionamiento de algunos de los nodos más importantes que se utilizaron para el desarrollo del proyecto.

Nodo de entrada *Button*

Equivale a un botón común. Permite cambiar los colores del texto y del fondo y permite agregarle un ícono para hacer el botón mas atractivo. Su principal parámetro es “Payload” con el cual se le especifica el contenido del mensaje que se quiere que el botón mande a su salida cuando es presionado (Figura 3.15).

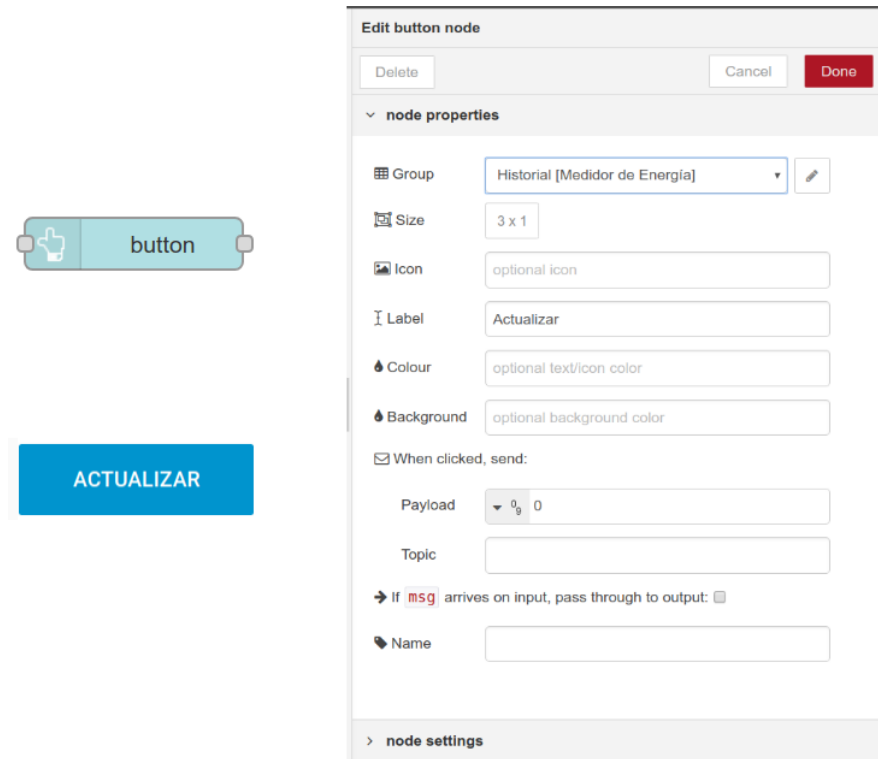


FIGURA 3.15: Nodo, instrumento y ajustes del nodo del dashboard “Button”.

Nodo de entrada *Dropdown*

Agrega una ventana de selección a la interfaz de usuario. Las opciones se especifican a manera de parejas de Valor-Texto donde el “Texto” es el que aparece como opción y el valor es el contenido del mensaje que el instrumento manda en su salida cuando un valor nuevo es seleccionado. Las opciones pueden ser modificadas en tiempo de ejecución (Figura 3.16).

Nodo de salida *Text*

Este es un instrumento que muestra un campo de texto no editable en la interfaz de usuario, cada mensaje que llegue a su entrada actualizará el valor mostrado según el formato que se haya especificado. La estructura de la etiqueta y del valor de texto pueden ser configuradas (Figura 3.17).

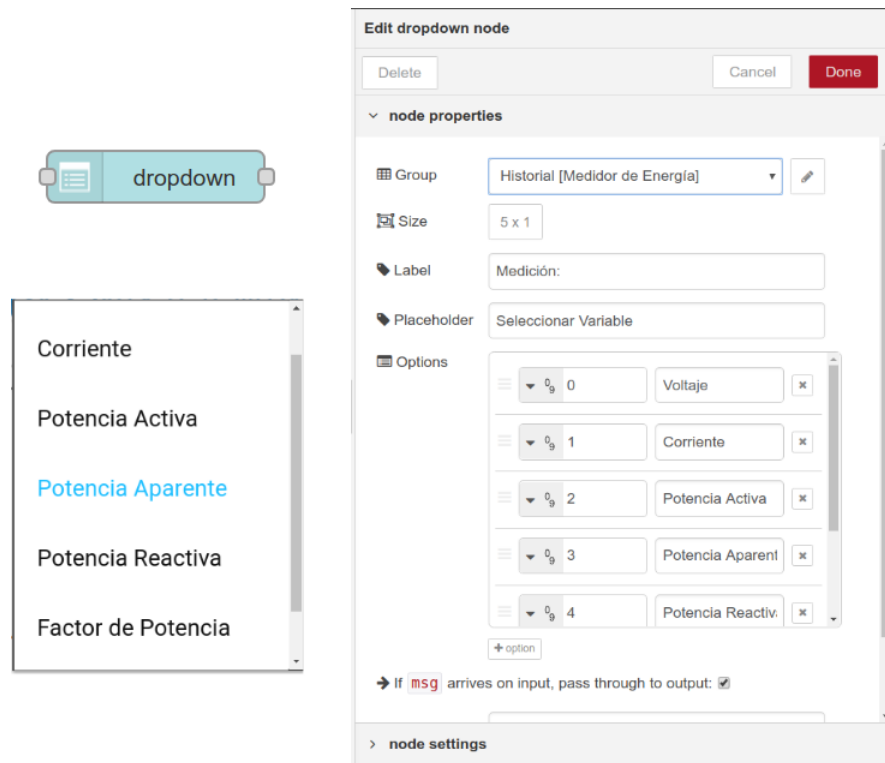


FIGURA 3.16: Nodo, instrumento y ajustes del nodo del dashboard “Dropdown”.

Nodo de salida *Gauge*

Agrega un instrumento tipo manómetro al tablero de instrumentos, cada mensaje que llegue a su entrada actualizará el valor mostrado según el formato que se haya especificado. El “Gauge” puede ser configurado para mostrar uno de cuatro estilos, normal, dona, compás o de onda. Permite editar los colores de algunos de los estilos para que se puede denotar alguna alarma por medio de los colores (Figura 3.18).

Nodo de salida *Chart*

Muestra una gráfica en el tablero de instrumentos, puede ser configurada para mostrar una gráfica normal, de barras o tipo pastel. Las etiquetas de los ejes pueden ser cambiadas en tiempo de ejecución (Figura 3.19).

Para poder facilitar el entendimiento de las secciones siguientes, se hace a manera de resumen un listado del tipo y número de nodos de dashboard que se usan en el proyecto en la Tabla 3.4, así mismo se muestra una pantalla de ejemplo de la interfaz de usuario en la Figura 3.20.

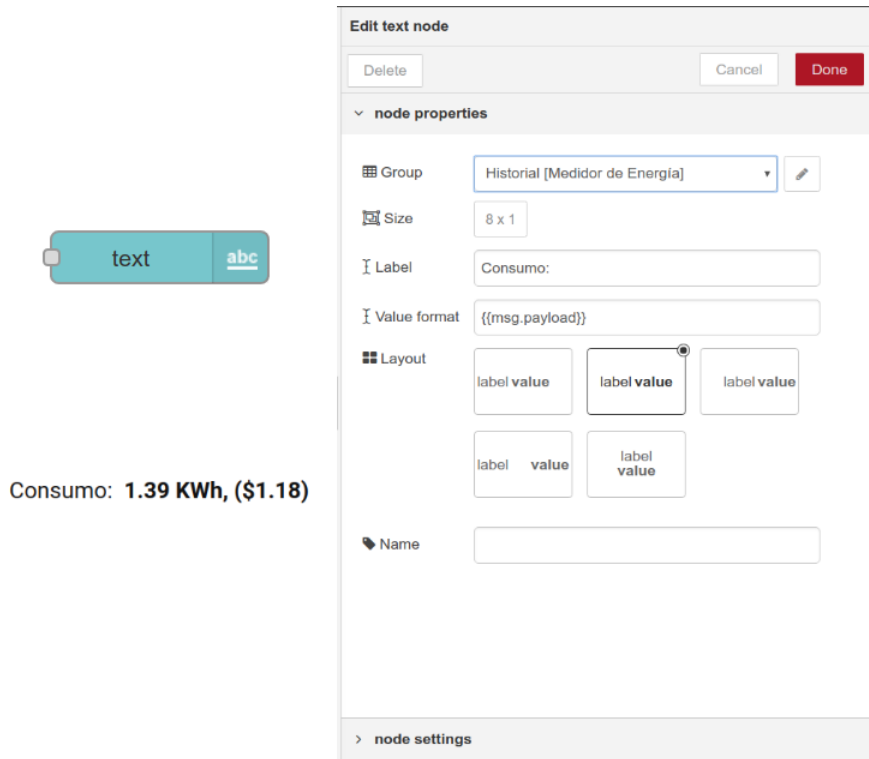


FIGURA 3.17: Nodo, instrumento y ajustes del nodo del dashboard “Text”.

Tipo	#	Etiquetas con las que el Nodo aparece en el dashboard
Button	2	Actualizar y CSV
Dropdown	3	Periodo, Medición y Facturación
Text	7	Voltaje, Corriente, Potencia Reactiva, Frecuencia, Costo KWh, Email, Consumo
Gauge	4	Potencia Activa, Potencia Aparente, Factor de Potencia y Consumo en Periodo de Facturación
Chart	1	chart
DatePicker ¹	2	Inicio y Final
Form ²	1	Form

TABLA 3.4: Nodos del dashboard usados en el proyecto

1 2

Como se podrá observar a lo largo de la tesis se manejarán casi indistintamente los términos *Interfaz de Usuario*, *Tablero de Instrumentos* y *Dashboard*.

¹Nodo del dashboard de Node-RED que despliega un calendario y que sirve para ingresar una fecha, no se menciona entre la explicación de nodos.

²Nodo del dashboard de Node-RED que produce un formulario. Sirve para recopilar información diversa del usuario en una sola intervención, evitando el uso de muchos nodos tipo *botton*, *text input*, *dropdown* etc., no se menciona entre la explicación de nodos.

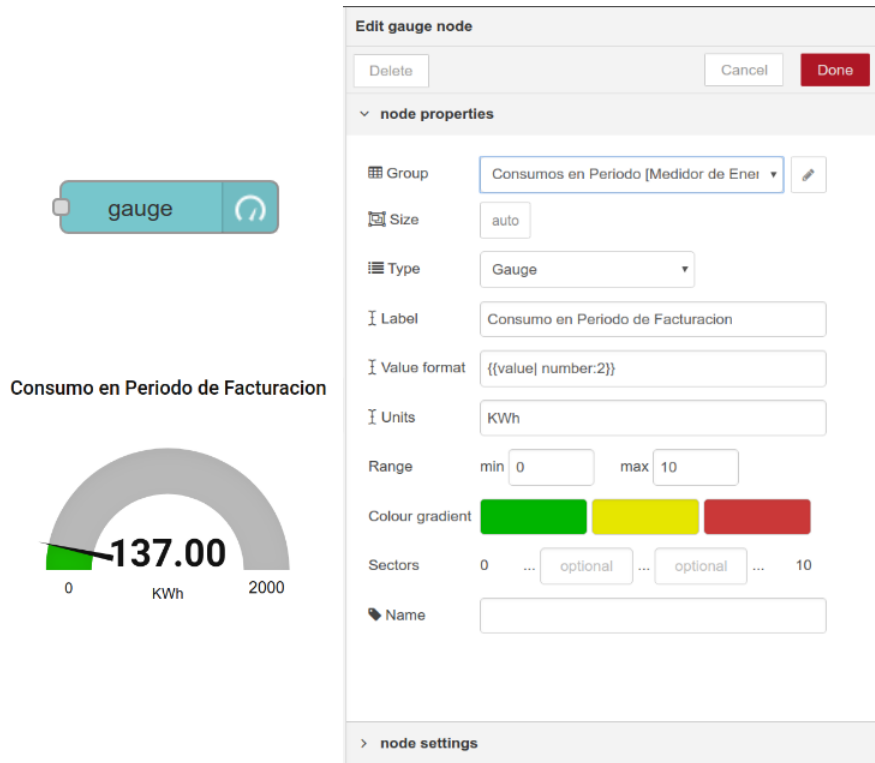


FIGURA 3.18: Nodo, instrumento y ajustes del nodo del dashboard “Gauge”.

3.2.4. Otros nodos importantes de Node-RED utilizados

Adicionales a los nodos ya revisados hay algunos más que sirvieron para el desarrollo del proyecto, a continuación se explica su funcionamiento brevemente.

Nodo *Debug*

El nodo “Debug” sirve para mostrar mensajes . Por defecto muestra *msg.payload* y cuando el mensaje es un objeto de JavaScript, el mensaje se puede contraer y expandir según sea necesario.

Cada mensaje incluye información sobre la hora en que se recibió el mensaje, el nodo que lo envió y el tipo de mensaje. El nodo por defecto manda los mensajes a la pestaña de depuración de la barra lateral, pero también se puede configurar para enviarlos a la consola del sistema o como mensajes cortos (32 caracteres) al texto de estado que se muestra debajo del nodo de depuración. (Figura 3.21).

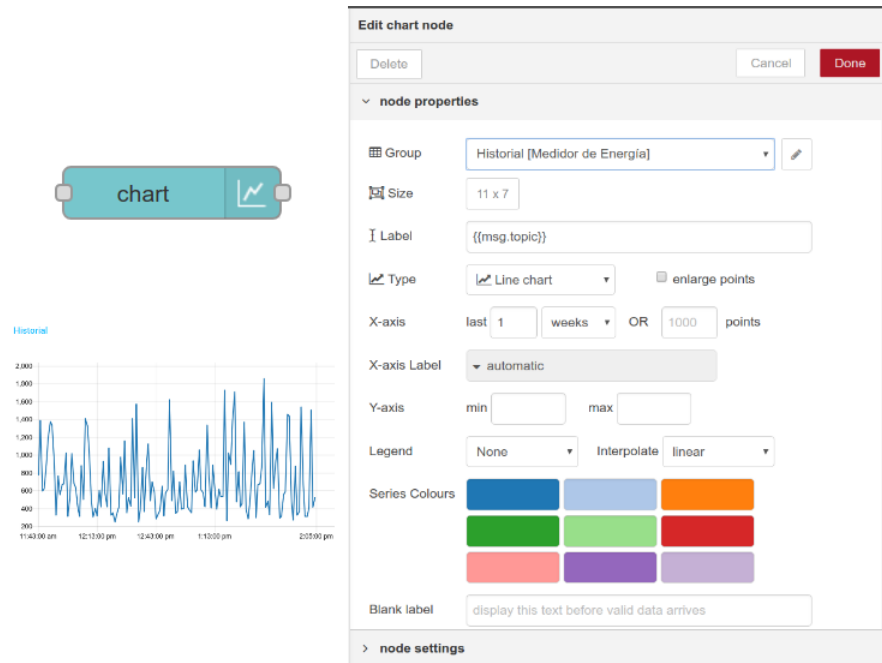


FIGURA 3.19: Nodo, instrumento y ajustes del nodo del dashboard “Chart”.

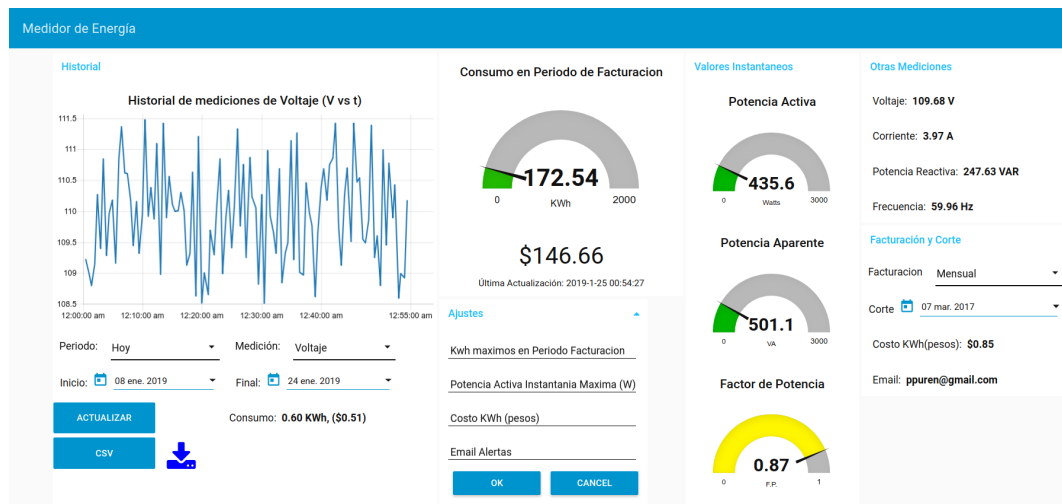


FIGURA 3.20: Interfaz de usuario, también conocida como tablero de instrumentos o dashboard.

Nodo Inject

El nodo “Inject” Inyecta un mensaje en el nodo en el que se encuentra conectados. El contenido del mensaje (payload) puede ser de varios tipos, incluidos cadenas, números, valores booleanos, objetos JavaScript, la hora actual o variables con distintos ámbitos tales como locales, de flujo o globales.

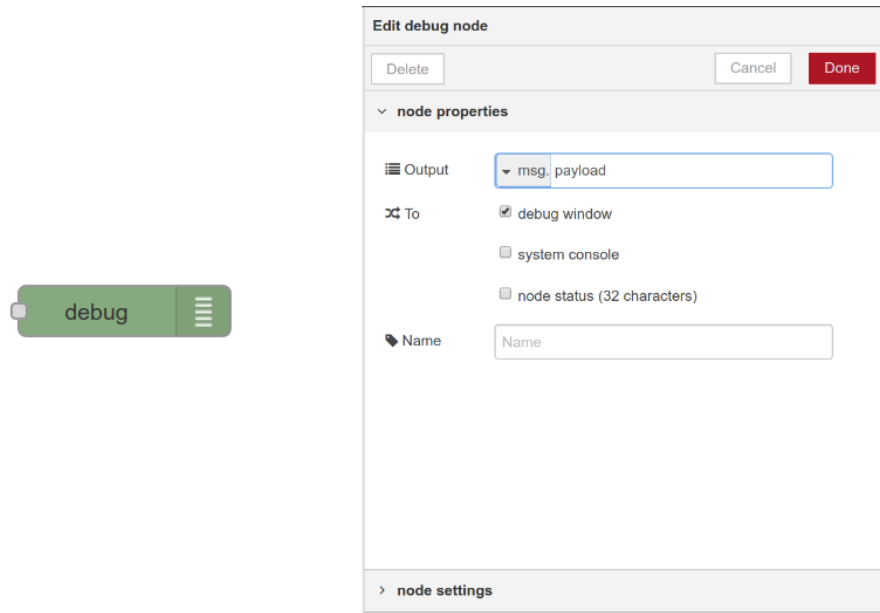


FIGURA 3.21: Nodo y ajustes del nodo de Node-RED “Debug”.

De forma predeterminada, el nodo se activa manualmente haciendo click en su botón dentro del editor. También se puede configurar para inyectar a intervalos regulares o al inicio de la ejecución del programa. El intervalo máximo que se puede especificar es de aproximadamente 596 horas. (Figura 3.22).

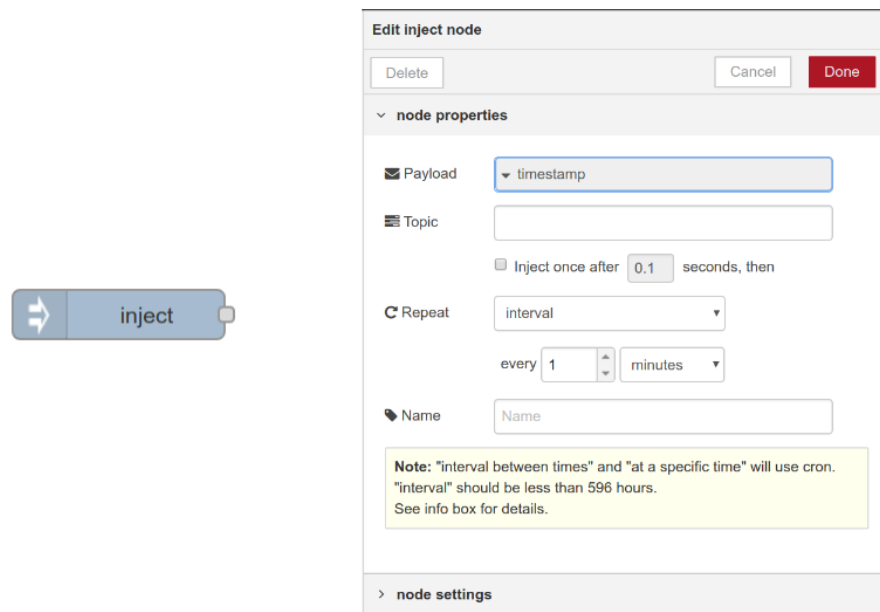


FIGURA 3.22: Nodo y ajustes del nodo de Node-RED “Inject”.

Nodo Function

Éste es uno de los nodos más importantes, ya que contiene un bloque de código en JavaScript que se ejecuta cada vez que se recibe un mensaje de entrada en el nodo. Los mensajes se pasan como un objeto de JavaScript llamado *msg* y por convención, este objeto tiene una propiedad *msg.payload* que contiene el cuerpo del mensaje.

Se espera que la función devuelva un objeto de mensaje (o varios objetos de mensaje a manera de matriz de objetos), pero puede optar por devolver un valor nulo único o como elemento de una matriz, en este caso no se transmite ningún mensaje (Figura 3.23).

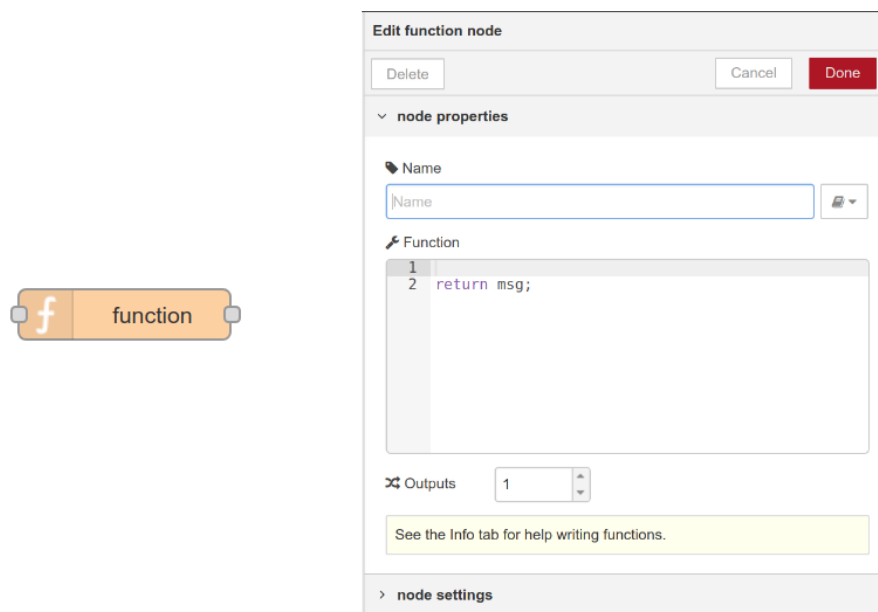


FIGURA 3.23: Nodo y ajustes del nodo de Node-RED “Function”.

Nodo Email

Este nodo sirve para enviar correo electrónicos. El *payload* del mensaje entrante se convierte en el cuerpo del email, teniendo como asunto el *msg.topic*. Puede configurarse en los parámetros del un nodo destinatario predeterminado, pero si se deja en blanco, debe configurarse utilizando la propiedad *msg.to* del mensaje entrante. También se puede especificar las propiedades del email “con copia” *msg.cc* y/o “copia ciega” *msg.bcc*.

Si el *payload* del mensaje es un búfer binario, se convertirá en un archivo adjunto. El nombre del archivo debe establecerse usando *msg.filename*. Por último, para configurar el servicio de email se utiliza SMTP con SSL al puerto 465. (Figura 3.24).

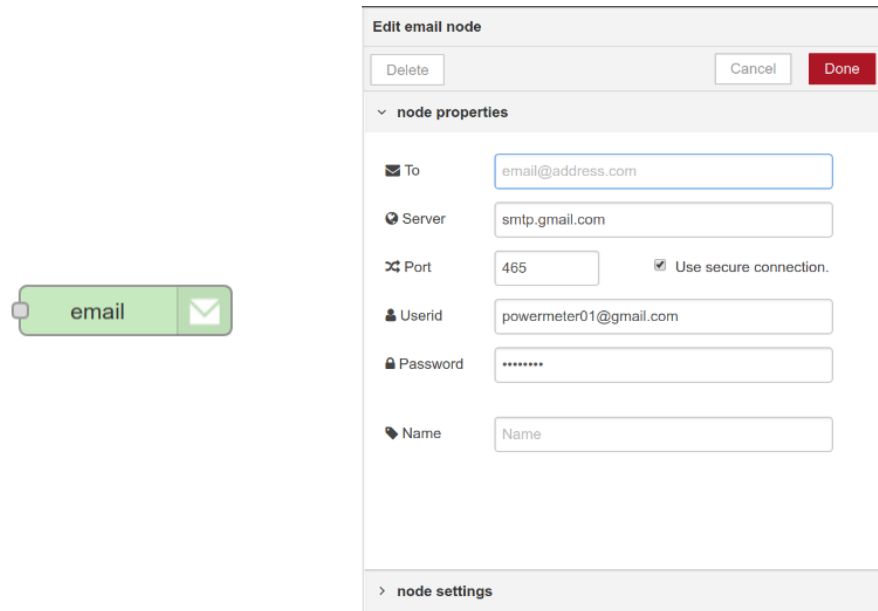


FIGURA 3.24: Nodo y ajustes del nodo de Node-RED “Email”.

3.3. Diseño y descripción de la lógica del programa

En esencia el algoritmo principal consta de dos grandes partes, la inicialización del sistema y un bucle infinito de servicio a interrupciones asíncronas. El diagrama de flujo que define al programa principal se muestra en la [Figura 3.25](#).

Inicialización de sistema es el proceso donde se les da valores iniciales a las variables globales, se crean objetos y nodos y se realizan configuraciones o procesos que se necesitan ejecutar antes de que se entre al estado “ejecución del bucle infinito” .

Bucle infinito de servicio a interrupciones asíncronas es el estado de marcha del concentrador, en el cual se encuentra “escuchando” la generación de los eventos asíncronos del sistema y dando servicio a los mismos según la solicitud.

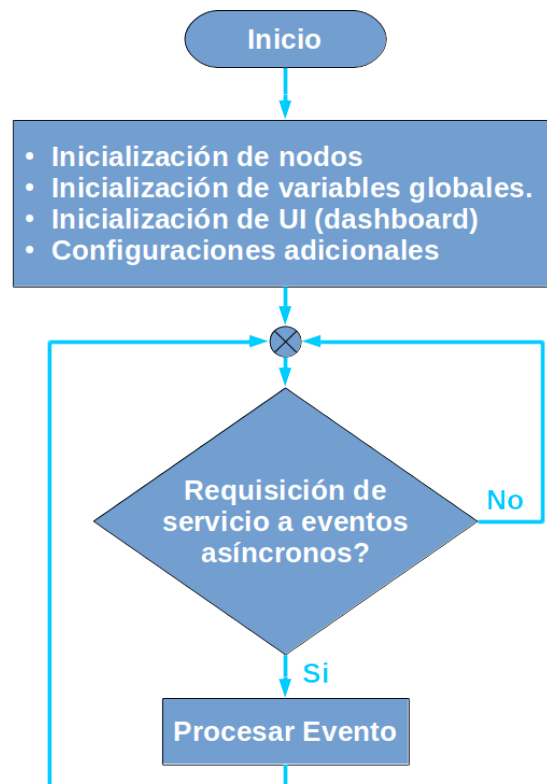


FIGURA 3.25: Diagrama de flujo del programa principal

Entre los eventos asíncronos más importantes se enumeran los siguientes:

1. **Llegada de datos de medición por el puerto serie:** Es el evento que se genera cuando llegan datos al módulo LoRa y son introducidos a la Raspberry pi por medio del puerto serie; al dispararse este evento se realizan la extracción, limpieza de datos y su almacenamiento en la base de datos (Figura 3.26).
2. **Solicitud de consulta a la base de datos:** este evento se genera desde la interfaz de usuario cuando quiere efectuar alguna de las siguientes acciones en el sistema:
 - Requisición de actualización de la gráfica del historial de mediciones.
 - Actualización de datos de la tabla de variables estáticas.
 - Solicitud de descarga de registro de mediciones.

Cuando se dispara este evento el sistema en respuesta construye una consulta SQL y la manda la base datos (Figura 3.27).

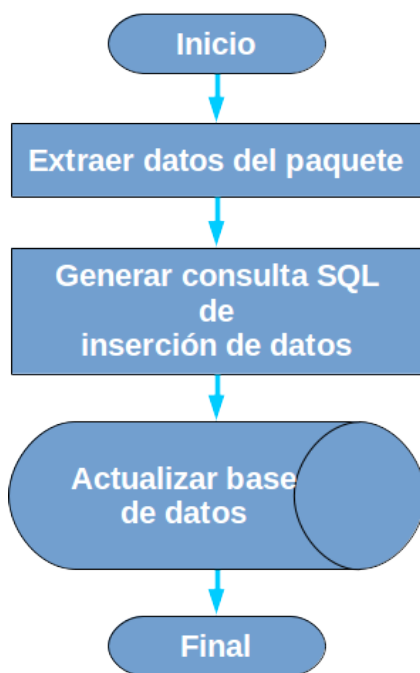


FIGURA 3.26: Diagrama de flujo en respuesta a la llegada de datos de medición del módulo LoRa.

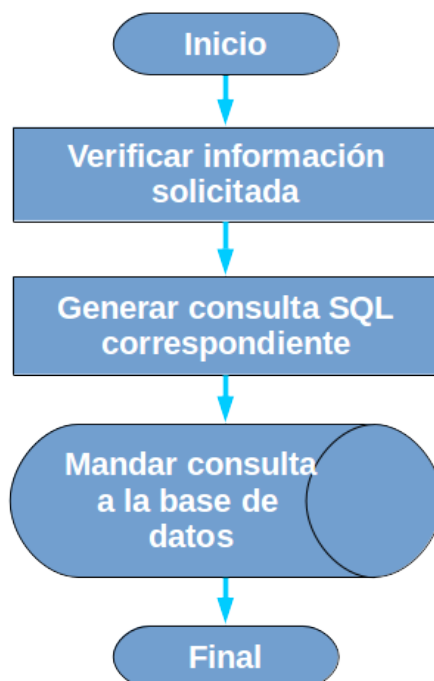


FIGURA 3.27: Diagrama de flujo de solicitud de consulta a la base de datos.

3. **Respuesta de la base de datos a consulta realizada:** La base de datos después de un tiempo proporcional a la complejidad de la consulta y el tamaño de la respuesta, genera un evento para que el sistema le dé término a la acción solicitada por el usuario originalmente (Figura 3.28).

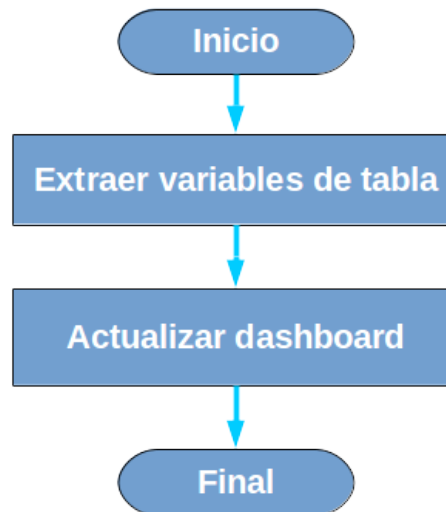


FIGURA 3.28: Diagrama de flujo del proceso que se genera después de una consulta a la base de datos.

A continuación se tocan más a detalle los procesos y servicios a los eventos asíncronos.

3.3.1. Inicialización del sistema

Cuando se energiza el sistema se realizan a grandes rasgos cuatro actividades:

1. **Recuperación de las variables estáticas de la base de datos**, para esta parte se tiene que hacer una consulta a la base de datos para recuperar las variables de la tabla *variablesEstaticas* (Figura 3.10), esta tabla contiene información importante ingresada por el usuario para configurar el medidor de energía tales como *tipo de facturación*, *fecha de corte*, *costo del kilowatt hora*, *email* y *valores máximos permitidos*. La consulta que se hace a la base de datos es la siguiente (MySQL Reference Manual 2018):

```
SELECT * FROM variablesEstaticas
WHERE uniqueKey = 1
```

la conexión de nodos asociada o *Flujo* como se le llama en Node-RED se muestra en la [Figura 3.29](#).

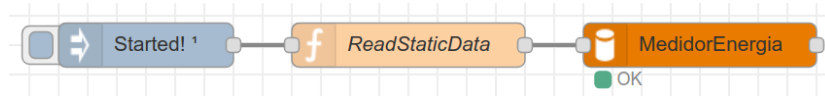


FIGURA 3.29: Flujo de recuperación de variables estáticas de la base de datos.

2. **Inicialización del dashboard con la información de las variables estáticas**, del resultado que se obtiene de la consulta del punto anterior se toma los valores para poder actualizar el dashboard de tal manera que el usuario no tenga que volver a configurar sus datos cada vez que hay una interrupción de energía. la conexión de nodos asociada o *Flow* se muestra en la [Figura 3.30](#).

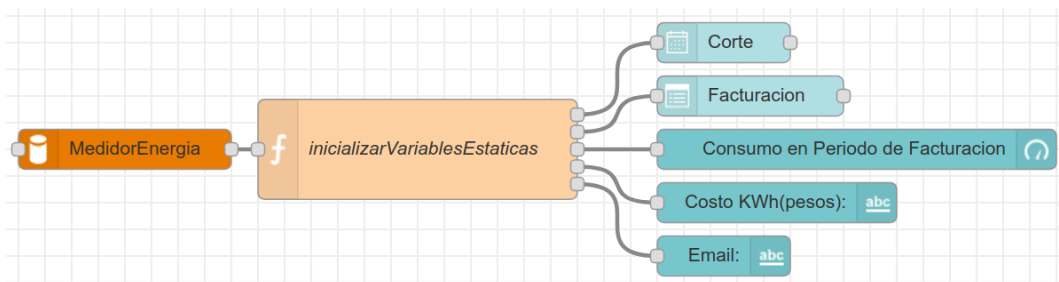


FIGURA 3.30: Flujo de restauración de variables en el dashboard.

3. **Inicialización del dashboard con valores predeterminados**, en el dashboard en la parte que tiene que ver con las consultas al historial de mediciones, hay nodos que al iniciar tienen valores erróneos que podrían causar que una consulta inicial falle. Es por eso que es necesario darles un valor predeterminado a estos nodos como se muestra en la [Figura 3.31](#).

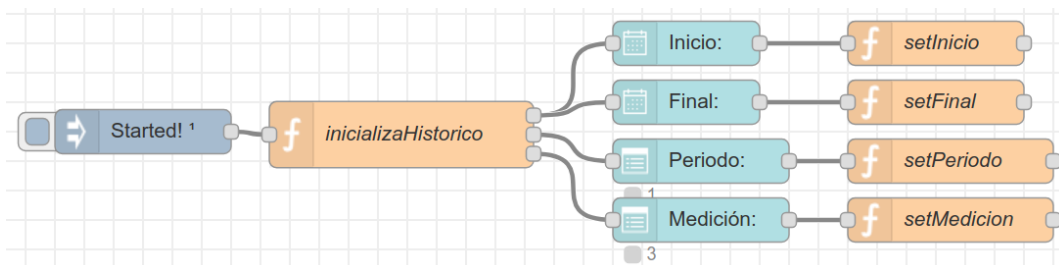


FIGURA 3.31: Flujo de inicialización del dashboard con valores predeterminados.

El código asociado con el nodo *inicializaHistorico* es el siguiente:

```
msg1 = { payload: new Date() };
msg2 = { payload: 1};
msg3 = { payload: 3};

return [ msg1, msg2, msg3 ];
```

4. **Inicialización de las variables globales**, ya por último para no tener que estar consultando las variables de la base de datos cada vez que se necesiten, se crea un conjunto de variables globales espejo de las variables almacenadas en la tabla de variables estáticas, este proceso se puede ver en los nodos de función conectados a la salida de los nodos del dashboard en la [Figura 3.31](#). El código del nodo *setPeriod* para guardar la variable *periodo* es el siguiente:

```
flow.set("periodo", msg.payload);
return msg;
```

3.3.2. Servicio al evento de llegada de datos por el puerto serie

El conjunto o “flujo” de nodos que realiza el proceso que se sirve cuando llegan las mediciones por el puerto serie se muestra en la figura [Figura 3.32](#).



FIGURA 3.32: Flujo que guarda las mediciones en la base de datos.

Los datos llegan al puerto de la Raspberry como una trama ASCII con los valores medidos separados por el símbolo “/”, el primer paso es separar estos valores en distintas variables para poder manejarlas de manera independiente, a continuación se muestra el código en *JavaScript* que realiza en este proceso.

```
var mediciones = msg.payload.split("/");

var voltage = mediciones[0];
var current = mediciones[1];
var activePower = mediciones[2];
var apparentPower = mediciones[3];
var reactivePower = mediciones[4];
```

```
var powerFactor = mediciones[5];  
var frequency = mediciones[6];  
var totalActiveEnergy = mediciones[7];  
var totalReactiveEnergy = mediciones[8];
```

Ya teniendo estas variables en el Node-RED se construye la consulta y se manda a la base de datos *mysql* ([MySQL Reference Manual 2018](#)), la consulta final quedaría de la siguiente manera:

```
INSERT INTO Mediciones  
VALUES (NULL, voltage,current, activePower, apparentPower,reactivePower  
      ,powerFactor, frequency, totalActiveEnergy, totalReactiveEnergy)
```

El valor *NULL* que aparece como el primer dato insertado corresponde a la “Huella de tiempo” y es un valor de “no importa” como se puede constatar en la [Figura 3.11](#), ya que dentro de la estructura de la tabla se definió como valor determinado la estampa de tiempo correspondiente a la llegada del dato.

Es importante recalcar que la variable *mediciones* que aparece en el código no es la misma variable *Mediciones* que aparece en la consulta ya que en esta última se refiere a la tabla de la base de datos ([Figura 3.11](#)).

No obstante que no se cuenta todavía con la parte del medidor se pueden realizar algunas simulaciones de “tramas” llegando al sistema para poder constatar que la base de datos y el proceso de acceso de datos se encuentran trabajando correctamente. En la [Figura 3.33](#) se muestran algunos valores simulados ingresados a manera de prueba en la base de datos.

Este es el único evento asíncrono que no depende de manera directa con la interfaz de usuario, todos los demás o se generan desde el *dashboard* de Node-RED, o la respuesta al evento implica una actualización del mismo.

3.3.3. Solicitud de consulta a la base de datos

Como se comentó, por tres razones se realizan consultas a la base de datos desde el dashboard:

Requisición de actualización de la gráfica del historial de mediciones

Este evento se genera cuando se presiona el botón *Actualizar* del tablero de instrumentos; para poder simplificar este proceso independientemente de la variable que

Mostrando filas 0 - 24 (total de 12204. La consulta tardó 0.0012 segundos.) (estampaDeTiempo: 2019-01-19 19:38:14 - 2019-01-19 19:14:13)

SELECT * FROM Mediciones ORDER BY estampaDeTiempo DESC

estampaDeTiempo	voltaje	corriente	potenciaActiva	potenciaAparente	potenciaReactiva	factorDePotencia	frecuencia	totalEnergiaActiva	totalEnergiaReactiva
2019-01-19 19:38:14	110.71	2.5	278.54	363.5	235.92	0.76	60.09	160.29	160.29
2019-01-19 19:37:14	109.08	2.58	281.56	378.8	253.41	0.74	60	160.28	160.29
2019-01-19 19:36:14	110.06	2.02	222.26	246.39	106.35	0.9	59.97	160.28	160.28
2019-01-19 19:35:14	110.2	3.95	435.19	586.63	408.14	0.73	59.95	160.28	160.28
2019-01-19 19:34:14	111.07	2.47	274.48	319.74	184.01	0.86	59.85	160.27	160.27
2019-01-19 19:33:14	110.54	4.98	550.02	625.91	298.74	0.88	60.1	160.26	160.27
2019-01-19 19:32:14	110.23	9.39	1035.02	1116.72	419.29	0.93	60.12	160.25	160.26
2019-01-19 19:31:14	111.12	3.42	379.73	459.04	257.93	0.83	59.93	160.24	160.24
2019-01-19 19:30:14	108.92	11.64	1268.01	1580.63	943.69	0.8	59.93	160.23	160.25
2019-01-19 19:29:14	110.08	3.43	377.88	416.28	174.65	0.91	59.9	160.21	160.21
2019-01-19 19:28:14	109.55	4.52	494.78	601.83	342.63	0.82	59.82	160.2	160.21
2019-01-19 19:27:14	110.82	2.54	281.42	376.84	250.63	0.75	60.05	160.2	160.2
2019-01-19 19:26:14	111.02	5.92	657.02	678.21	188.23	0.97	60.09	160.19	160.19
2019-01-19 19:25:14	108.96	11.79	1284.85	1823.99	1294.65	0.7	59.85	160.18	160.2
2019-01-19 19:24:13	110.78	3.8	420.96	566.08	378.47	0.74	59.98	160.16	160.16
2019-01-19 19:23:13	109.79	4.65	510.05	588.9	313.89	0.85	59.88	160.15	160.16
2019-01-19 19:22:13	109.22	9.76	1066.55	1416.27	931.82	0.75	60.04	160.14	160.16
2019-01-19 19:21:13	109.03	2.53	275.6	315.15	152.86	0.87	59.95	160.13	160.13
2019-01-19 19:20:13	110.58	7.68	848.71	1149.8	775.71	0.74	59.9	160.12	160.13
2019-01-19 19:19:13	108.99	6.21	676.36	908.27	606.21	0.74	60.13	160.11	160.12
2019-01-19 19:18:13	109.52	3.93	430.22	481.36	215.92	0.89	59.91	160.1	160.1
2019-01-19 19:17:13	111.2	2.26	251.68	254.22	35.84	0.99	60.06	160.09	160.09
2019-01-19 19:16:13	108.82	3.83	416.99	565.56	382.07	0.74	60.11	160.08	160.09
2019-01-19 19:15:13	110.62	3.68	407.4	511.5	309.28	0.8	60.06	160.08	160.08
2019-01-19 19:14:13	108.93	2.14	233.3	284	161.94	0.82	59.8	160.07	160.07

FIGURA 3.33: Tabla “Mediciones” con algunas variables simuladas para probar la base de datos y el código de entrada de datos.

se vaya a desplegar, se elabora una consulta de todas las mediciones en el plazo especificado por el usuario, donde el periodo se especifica desde el tiempo $t_{Inicial}$ hasta el tiempo t_{Final} , el conjunto de nodos o flujo asociado se muestra en la [Figura 3.34](#).



FIGURA 3.34: Flujo de requisición de actualización de la gráfica del historial de mediciones

La consulta SQL final es la siguiente ([MySQL Reference Manual 2018](#)):

```

SELECT * FROM Mediciones
WHERE estampaDeTiempo >= tInicial AND estampaDeTiempo <= tFinal
  
```

Actualización de datos de la tabla de variables estáticas

Este evento se dispara cuando el usuario actualiza el nodo *form* que corresponde al formulario donde el usuario actualiza su información, el conjunto de nodos o flujo asociado se muestra en la [Figura 3.35](#).



FIGURA 3.35: Flujo de actualización de la información del usuario que corresponde a las variables estáticas.

Para realizar los cambios en la información del usuario se actualiza una variable de cadena llamada *cambios* con la pareja de valores del nombre de la variable y el nuevo valor, un fragmento del código que cambia el costo del KWh se muestra de ejemplo.

```
var costoKwh = msg.payload.costoKwh;
var cambios = "";

if (costoKwh){
  flow.set("costoKwh", costoKwh);
  if(cambios) {
    cambios = cambios + ", ";
  }
  cambios = cambios + "costoKwh = " + costoKwh;
}
```

La consulta SQL final correspondiente ([MySQL Reference Manual 2018](#)), quedaría de la siguiente manera:

```
UPDATE variablesEstaticas
SET cambios
WHERE uniqueKey = 1
```

Solicitud de descarga de registro de mediciones

Este evento se genera cuando se presiona el botón CSV del tablero de instrumentos; el periodo se especifica desde el tiempo *tInicial* hasta el tiempo *tFinal*, el conjunto de nodos o flujo asociado se muestra en la [Figura 3.36](#).



FIGURA 3.36: Flujo de requisición de descarga del registro de mediciones.

La consulta *SQL* es muy parecida a la consulta de la gráfica de historial de mediciones ([MySQL Reference Manual 2018](#)), solo que manda a salvar los datos en un archivo, la consulta es la siguiente:

```
SELECT * FROM Mediciones
      WHERE estampaDeTiempo >= tInicial AND estampaDeTiempo <= tFinal
      INTO OUTFILE '/tmp/' + nombreArchivo
      FIELDS TERMINATED BY ','
      LINES TERMINATED BY '\n'
```

3.3.4. Respuesta de la base de datos a la consulta realizada

Una vez que la base de datos termina de procesar la consulta para graficar el historial, responde con una tabla con las mediciones en el periodo especificado, sin embargo la respuesta no se puede mandar directo al nodo *chart*, por lo que entra antes a un nodo *function* para darle formato a los datos y cambiar los encabezados de la gráfica, el flujo correspondiente se muestra en la [Figura 3.37](#), un fragmento de código se muestra a continuación:

```
var variable = flow.get("variable");
switch (variable) {
  case 0:
    unidades = "Volts";
    encabezado = "mediciones de Voltaje (V vs t)"
    break;
  case 1:
    unidades = "Ampers";
    encabezado = "mediciones de Corriente (A vs t)"
    break;
  case 2:
    unidades = "Watts";
    encabezado = "consumo de Potencia Activa (W vs t)"
    break;
  case 3:
    unidades = "VA";
    encabezado = "mediciones de Potencia Aparente (VA's vs t)"
    break;
  case 4:
```



```

    unidades = "VAR";
    encabezado = "mediciones de Potencia Reactiva (VAR's vs t)"
    break;
case 5:
    unidades = "";
    encabezado = "mediciones de Factor de Potencia (Valor vs t)"
    break;
case 6:
    unidades = "Hz";
    encabezado = "mediciones de Freciencia (Hz vs t)"
    break;
}
msg.payload = [{ key: unidades, values : output}];
msg.topic = "Historial de " + encabezado;

return msg;

```



FIGURA 3.37: Flujo de respuesta para graficar historial de mediciones.

3.3.5. Interfaz de usuario

La interfaz de usuario como se muestra en la [Figura 3.38](#) se divide en seis partes. Estos grupos pueden cambiar de orden o lugar dependiendo del tamaño de la pantalla del dispositivo donde se visualice el tablero, pero esto lo hace automáticamente el explorador, siempre tratando a cada grupo como una unidad.

Historial de mediciones

Se conforma de todos los instrumentos destinados a permitirle al usuario ver la gráfica con respecto al tiempo de una de las variables durante un periodo específico, como se vio en la [Sección 3.3.3](#) y la [Subsección 3.3.4](#). Esta parte igual permite exportar las variables medidas en un archivo separado por comas con extensión "csv" ([Sección 3.3.3](#)). Adicionalmente cada vez que se actualiza la gráfica, se muestra el consumo en kilowatts-hora y su respectiva estimación en pesos del periodo solicitado.

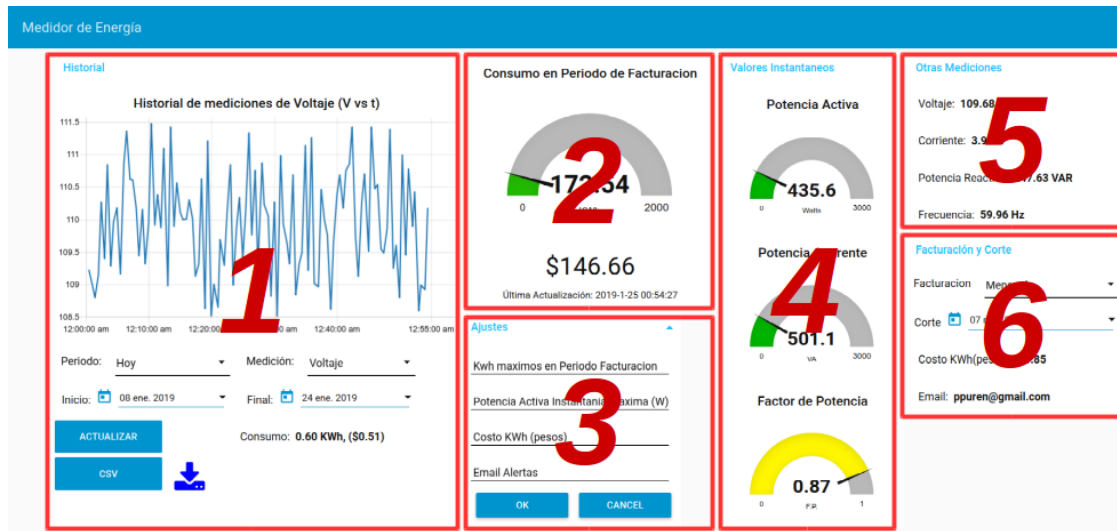


FIGURA 3.38: Grupos de la interfaz de usuario

Consumos en periodo de facturación

Este grupo sirve para informar al usuario a cerca del consumo en kilowatts-hora y su equivalencia en pesos en el periodo de facturación actual. El consumo se refresca cada 10 minutos y para el caso del instrumento tipo manómetro se le puede cambiar el valor máximo permitido del periodo desde el grupo de “Ajustes”. El consumo en pesos mostrado se calcula usando el costo del kilowatt-hora también especificado en el grupo de “Ajustes”; se sabe que CFE maneja brackets, periodos y zonas en los que varia el costo del kilowatt-hora, por lo que la cantidad en pesos reflejada no va a ser exacta, sin embargo sirve para darle al usuario un estimado de sus consumos. El conjunto de nodos asociado se muestra en la [Figura 3.39](#).

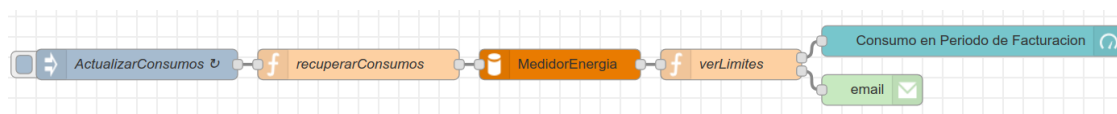


FIGURA 3.39: Flujo que actualiza los consumos periodicamente.

La consulta SQL asociada es la siguiente ([MySQL Reference Manual 2018](#)):

```
SELECT estampaDeTiempo, totalEnergiaActiva
FROM Mediciones
WHERE estampaDeTiempo >= inicio.getFullYear()-(inicio.getMonth()+1)
      -inicio.getDate()
```

Ajustes

Grupo conformado por un solo formulario que sirve para ajustar las variables definidas por el usuario que corresponden al límite máximo de consumo del período, la potencia activa máxima permitida, el costo del kilowatt-hora y el email usado para las alertas. Los cambios que se realizan en este instrumento se reflejan tanto en la base de datos ([Sección 3.3.3](#)) como en el mismo tablero de instrumentos, ya que estas variables se muestra en distintas partes del dashboard. El flujo de nodos se presenta en la [Figura 3.40](#).

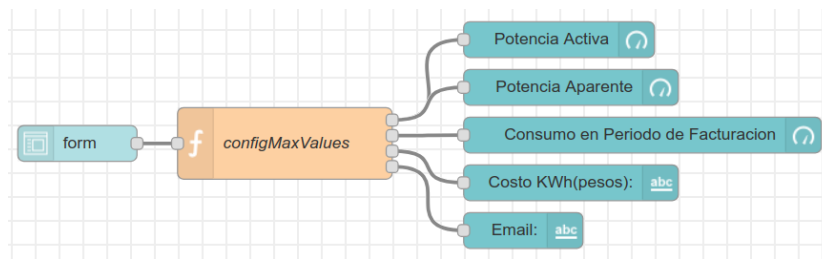


FIGURA 3.40: Flujo de actualización de variables definidas por el usuario.

Valores instantáneos

Consta de tres nodos tipo *Gauge* y sirve para mostrar los valores de la potencia activa, potencia aparente y el factor de potencia instantáneos, los cuales se considera que son los siguientes en orden de importancia después del consumo en el periodo y por eso se muestran en manómetros. Por sencillez al cambiar el límite máximo de la potencia activa, el límite de la potencia reactiva también cambia a este valor. Es importante notar que cuando se registra un valor instantáneo de potencia activa mayor al definido se genera una alarma que se manda vía email al correo especificado en el grupo de ajustes. El flujo de nodos donde se realiza el cambio a estos instrumentos se muestra en la [Figura 3.41](#).

El nodo *actMediciones* de la [Figura 3.41](#) comparte código con el nodo *saveData* que se muestra en la [Figura 3.32](#) de la [Subsección 3.3.2](#), por lo que en la practica en realidad conforman un solo nodo, sin embargo para poder mostrar de una manera más clara los dos procesos se separan en dos nodos.

El código que manda las mediciones que llegan al puerto serie a sus respectivos instrumentos es el siguiente:

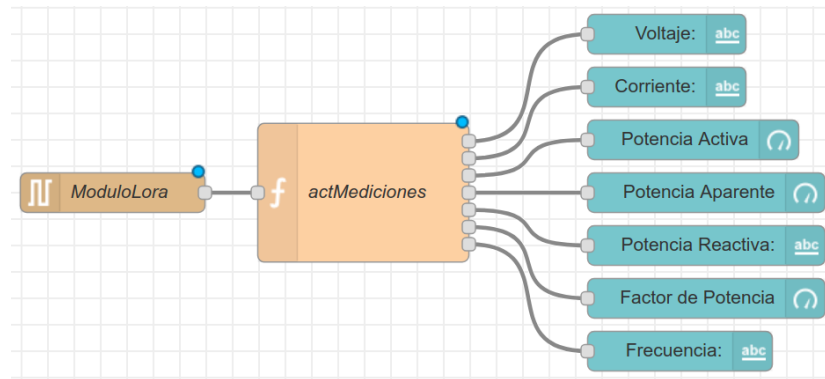


FIGURA 3.41: Flujo de actualización de mediciones instantáneas en el dashboard.

```

var mediciones = msg.payload.split("/");

var voltage = mediciones[0];
var current = mediciones[1];
var activePower = mediciones[2];
var apparentPower = mediciones[3];
var reactivePower = mediciones[4];
var powerFactor = mediciones[5];
var frequency = mediciones[6];
var totalActiveEnergy = mediciones[7];
var totalReactiveEnergy = mediciones[8];

var msg1 = { payload: voltage };
var msg2 = { payload: current };
var msg3 = { payload: activePower };
var msg4 = { payload: apparentPower };
var msg5 = { payload: reactivePower };
var msg6 = { payload: powerFactor };
var msg7 = { payload: frequency};
return [ msg1, msg2, msg3 , msg4 , msg5, msg6 , msg7 ];

```

Otras mediciones

Este grupo se conforma por cuatro nodos de texto que sirven para desplegar los valores instantáneos del resto de las variables que no se han mostrado aún,

voltaje, corriente, potencia reactiva y frecuencia. Varía con respecto a los nodos tipos manómetro del grupo anterior solo en el aspecto del instrumento, ya que el proceso es totalmente el mismo y de hecho comparten todo, el mismo grupo de nodos (Figura 3.32) y el mismo código como se puede ver en las variables *msg1*, *msg2*, *msg5* y *msg7* del fragmento de código anterior.

Facturación y corte

Grupo que sirve para especificar el tipo de facturación (mensual o bimestral) y la fecha de corte de la factura eléctrica para poder realizar los cálculos que se relacionan con el periodo de facturación.

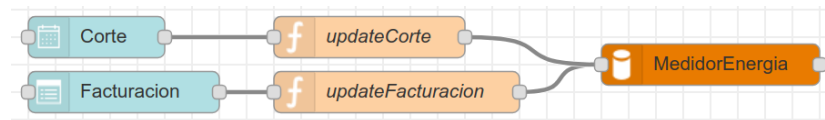


FIGURA 3.42: Flujo de actualización de los datos de facturación y corte.

Las consultas que se le realizan a la base de datos toman el valor que les llega del contenido del mensaje ([MySQL Reference Manual 2018](#)):

```

UPDATE variablesEstaticas
  SET fechaCorte = msg.payload
  WHERE uniqueKey = 1
  
```

```

UPDATE variablesEstaticas
  SET facturacion = msg.payload
  WHERE uniqueKey = 1
  
```

Adicionalmente, este grupo muestra el valor actualizado del costo del kilowatt-hora y el email registrado para mandar las alertas. Ambos se actualizan cada vez que hay un cambio en alguna de estas variables desde el grupo de ajustes (Figura 3.40).

3.3.6. Alertas

El sistema cuenta con dos alertas que se pueden configurar para que el usuario tenga el control de sus consumos y tener la capacidad de detectar cualquier problema con su instalación o con algún equipo eléctrico, estas alertas son enviadas por el concentrador al correo especificado en *Email Alerts* en el grupo *Ajustes* de la Figura 3.38; las alertas habilitadas son la siguientes:

- **Alerta de Consumo Máximo en Periodo de Facturación.-** Se dispara cuando el consumo de KiloWats-hora ha superado el límite definido como *KWh máximos en Periodo de Facturación* en el grupo de *Ajustes* de la [Figura 3.38](#). Esta alerta permite al usuario corregir cualquier tendencia en el periodo que podría pasar el consumo esperado y provocar un incremento en el pago de la factura eléctrica.
- **Alerta de Potencia Activa Instantánea Máxima.-** Se dispara cuando la potencia activa Instantánea pasa el umbral definido como *Potencia Activa Instantánea Máxima* en el grupo de *Ajustes* de la [Figura 3.38](#). Sabiendo el consumo estimado de los equipos eléctricos de la instalación, es posible definir el límite máximo de tal manera que cuando se dispare la alarma refleje una alta probabilidad de que algún equipo se encuentre defectuoso o una posible fuga en la red eléctrica.

A continuación se muestra la manera en que se definieron estas alertas en el programa.

Alerta de consumo máximo en periodo de facturación

Esta alerta forma parte del bloque dedicado a actualizar el consumo del periodo de facturación y que como se mencionó en el inciso de *Consumos en Periodo de Facturación* de la [Subsección 3.3.5](#), es un proceso que se ejecuta cada 10 minutos y es cuando se evalúa si se ha sobrepasado el límite definido; en la [Figura 3.43](#) se muestra los nodos que intervienen para la realización de este proceso.

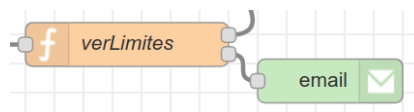


FIGURA 3.43: Sección del proceso de definición de la alerta de consumo máximo.

El código en JavaScript que realiza el proceso de verificación y construcción del cuerpo del mensaje es el siguiente:

```
if (consumo > kwhMax ) {
  node.warn("band1");
  if ((fechaAlarmaEnviadaMs === null)|| (new Date(fechaAlarmaEnviadaMs) < hoy)){
    node.warn("band2");
    var email = flow.get("email");
    var msg2 = {
```

```

    topic : "ALERTA DE CONSUMOS EN PERIODO DE FACTURACION - ENR" +
            (tiempoReg.getTime()).toString(16).toUpperCase(),
    payload : "MAXIMO CONSUMO DE ENERGIA ACTIVA REGISTRADA EN PERIODO"
+ " DE FACTURACION.\n\nConsumo de " + consumo + " Kwh registrado"
+ " durante periodo de facturación, mayor al límite establecido de"
+ " " + kwhMax + " Kwh.\nFecha y hora de registro: "
+ tiempoReg.toLocaleString() +
".\n\nSe recomienda revisar instalación y aparatos eléctricos.",
    to : email
}
fechaAlarmaEnviadaMs = hoy.getTime() + 86400000; // 24hrs sig alerta
context.set("fechaAlarmaEnviadaMs", fechaAlarmaEnviadaMs);
return [msg, msg2]
}
}

```

Un ejemplo del email recibido de alerta de consumo máximo se muestra en la [Figura 3.44](#).

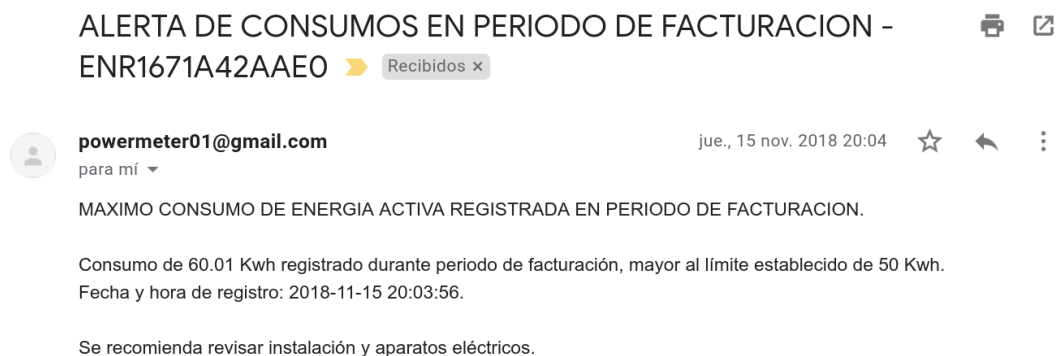


FIGURA 3.44: Email ejemplo de alerta de consumo máximo.

La alerta de consumo máximo cuenta con un periodo de 24 horas después que se disparó la alerta, en el cual se ignoran todas las nuevas alertas generadas, esto con el objetivo de evitar que se llene la bandeja de entrada del usuario y para brindar el tiempo suficiente para redefinir un nuevo límite de consumo. En caso que por alguna razón el usuario no pueda redefinir el límite se respeta esta tendencia, enviando sólo una alerta cada 24 horas.

Alerta de potencia activa instantánea máxima

Esta alerta es un poco más sencilla de generar, ya que sólo se necesita una sentencia 'if' en los datos en el bloque que actualiza la información en el dashboard; en la [Figura 3.43](#) se muestra los nodos que intervienen para la realización de este proceso.



FIGURA 3.45: Sección del proceso de definición de la alerta de potencia activa instantánea máxima excedida.

El código en JavaScript que realiza el proceso de verificación y construcción del cuerpo del mensaje es el siguiente:

```
var potencia = msg.payload;
potMax = flow.get("potMax");
if (potencia > potMax ) {
  var email = flow.get("email");
  var tiempoReg = new Date();
  var msg = {
    topic : "ALERTA DE POTENCIA ACTIVA MAXIMA REGISTRADA - POT" +
      (tiempoReg.getTime()).toString(16).toUpperCase(),
    payload : "LIMITE DE POTENCIA ACTIVA SUPERADO.\n\nMedición de "
      + potencia + " W, mayor al límite establecido de " + potMax +
      " W.\nFecha y hora de registro: " + tiempoReg.toLocaleString() +
      ".\n\nSe recomienda revisar instalación y aparatos eléctricos.",
    to : email
  }
  return msg;
}
```

Un ejemplo del email recibido de alerta de potencia activa Instantánea máxima se muestra en la [Figura 3.44](#).

Al igual que la alerta de consumo máximo, la alerta de potencia activa Instantánea máxima cuenta con un periodo de 24 horas en el cual se ignoran todas las nuevas alertas generadas para evitar que se llene la bandeja de entrada del usuario; en caso que por alguna razón el usuario no pueda redefinir el límite se respeta esta tendencia, enviando sólo una alerta cada 24 horas.

ALERTA DE POTENCIA ACTIVA MAXIMA REGISTRADA - POT168F92FA429 ▷ Recibidos x



powermeter01@gmail.com

sáb., 16 feb. 20:01

para mí ▾

LIMITE DE POTENCIA ACTIVA SUPERADO.

Medición de 1336.81 W, mayor al límite establecido de 1200 W.

Fecha y hora de registro: 2019-2-16 20:01:15.

Se recomienda revisar instalación y aparatos eléctricos.

FIGURA 3.46: Email ejemplo de alerta de potencia activa instantánea máxima.

3.3.7. Visualización del tablero de mediciones

Para poder ver dashboard con las mediciones es necesario seguir los siguientes pasos:

1. Averiguar la dirección IP que el router le asignó a la raspberry pi, para conocerla hay muchas maneras, la más fácil es conectarle un monitor a la Raspberry Pi y reiniciarla, en la pantalla inicial se reporta la IP asignada (Figura 3.47).

```
[ ok ] Setting up console font and keymap...done.
[ ok ] Checking if shift key is held down: No. Switching to
[ ok ] Setting up X socket directories... /tmp/.X11-unix /tmp/.X11-unix
INIT: Entering runlevel: 2
[info] Using makefile-style concurrent boot in runlevel 2.
[ ok ] Network Interface Plugging Daemon...skip eth0...done.
[ ok ] Starting enhanced syslogd: rsyslogd.
[info] Initializing cgroups.
[warn] Kernel lacks cgroups or memory controller not available
Starting aphys-swapfile swapfile setup ...
want /var/swap=100MByte, checking existing: keeping it
done.
[ ok ] Starting periodic command scheduler: cron.
[ ok ] Starting NTP server: ntpd.
[ ok ] Starting system message bus: dbus.
[ ok ] Starting OpenSSH Secure Shell server: sshd.
My IP address is 192.168.0.104
Raspbian GNU/Linux 7 raspberrypi tty1
raspberrypi login: _
```

FIGURA 3.47: Dirección IP leída de la pantalla inicial.

Otra manera, también por medio del uso de un monitor, es desde la línea de comandos usar el comando *ifconfig* (Figura 3.48).

Una última opción ofrecida de todas la múltiples que hay, es entrar directamente a la configuración del router y ver la dirección IP que le asignó (Figura 3.49).

```

pi@raspberrypi:~$ sudo ifconfig
eth0: Link encap:Ethernet HWaddr b8:27:eb:7f:5b:99
      inet addr:192.168.100.61 Bcast:192.168.100.255 Mask:255.255.255.0
      inet6 addr: fe80::b1f3:27c3:8e7e:e544/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:63111 errors:0 dropped:98 overruns:0 frame:0
      TX packets:60050 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:5776168 (5,5 MiB)  TX bytes:12624739 (12,0 MiB)

lo:    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:65536  Metric:1
      RX packets:27047 errors:0 dropped:0 overruns:0 frame:0
      TX packets:27047 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1
      RX bytes:11523685 (10,9 MiB)  TX bytes:11523685 (10,9 MiB)

wlan0: Link encap:Ethernet HWaddr b8:27:eb:2a:0e:cc
      inet6 addr: fe80::b1f3:27c3:8e7e:e544/64 Scope:Link
      UP BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0,0 B)  TX bytes:0 (0,0 B)

pi@raspberrypi:~$
    
```

FIGURA 3.48: Dirección IP con el comando ifconfig.

Host Name	Port ID	Device Type	IP Address	MAC Address	Device Status
raspberrypi	LAN1	dhcpd-6.7.1.Lin.....	192.168.100.61	b8:27:eb:7f:5b:99	Online
ESP_032316	SSID1	--	192.168.100.4	d0:27:00:03:23:16	Online
TL-WA855RE	SSID1	udhcp 1.19.4	192.168.100.5	9e:de:d0:2b:3a:57	Online

FIGURA 3.49: Dirección IP leída directamente del router.

- Entrar al navegador del dispositivo desde el que se quiere leer los datos, este dispositivo tiene que estar conectado a la red local y puede ser cualquier dispositivo digital como una tablet, celular, computadora etc.
- Ingresar en el navegador la dirección³:

<IP> + :1880/ui/#/ + <número de dashboard>

El número de dashboard por lo general es un número incremental y empieza con cero, un ejemplo de la dirección ingresada en el navegador se puede ver en la [Figura 3.50](#).

³Lo ideal sería utilizar un servidor DNS local que permitiera darle un nombre más significativo al medidor como por ejemplo *mi_medidor_energia.iot*, pero hubiera aumentado la complejidad del diseño del proyecto.

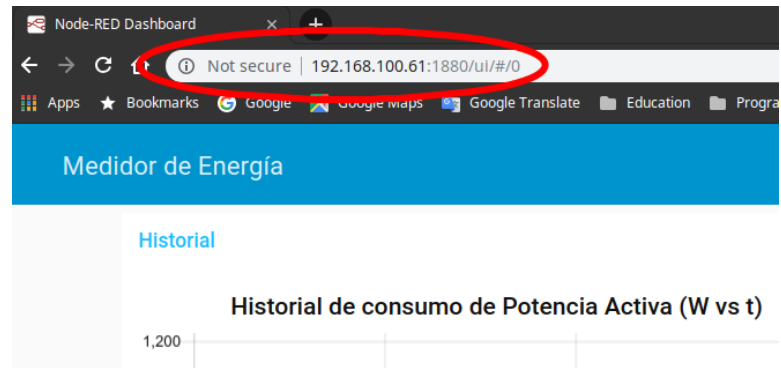


FIGURA 3.50: Ingreso al dashboard o tablero de mediciones.

En esta página web, o como mejor es descrita *aplicación web*, se despliegan las mediciones eléctricas y se tiene acceso a todas las consultas y funcionalidades descritas en este capítulo.

Capítulo 4

Implementación y resultados

4.1. Implementación e instalación

Para probar el proyecto se pidió permiso de instalar el equipo de medición en el edificio de maestría, con el objetivo de medir los consumos eléctricos del aire acondicionado del cubículo asignado.

4.1.1. Nodo medidor

Como se ha mencionado, este proyecto se complementó con el trabajo de la tesis *Desarrollo de hardware para integración de medidores de energía eléctrica a plataforma IoT* desarrollada por el ingeniero Marcos Ariel Olivera Moreno, que corresponde al nodo medidor de energía, enfocado más al hardware y la interfaz de comunicación Modbus con el medidor eléctrico.

Para poder implementar el proyecto de tesis en un ambiente real, conectado a una red eléctrica o aparato y poder medir el consumo, se necesitan ambas partes, el nodo medidor y el nodo concentrador, objeto de esta tesis, por lo tanto:

En esta sección que corresponde al nodo medidor, se explicará a groso modo en qué consiste la parte del nodo medidor de energía y se proporcionará fotos e información que no son intrínsecas a esta tesis, pero que son de suma importancia para poder darle sentido al proceso total de medición.

El diagrama a bloques del nodo medidor se muestra en la [Figura 4.1](#), las partes azules y amarillas forman parte del nodo medidor y la parte gris indica el nodo concentrador descrito en esta tesis.

A continuación se explican muy brevemente cada parte.

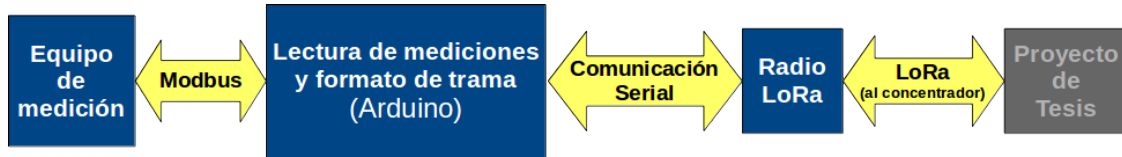


FIGURA 4.1: Diagrama a bloques del nodo medidor.

Equipo de medición

Para la realización de las mediciones eléctricas era necesario usar un equipo de medición que cumpliera como mínimo con las capacidades de medición requeridas por la red eléctrica o el equipo eléctrico a monitorear, y que contase con una interfaz Modbus para poder tener acceso externo de los valores medidos. Para esta parte se utilizó un medidor de energía de una sola fase marca Eastron modelo SDM120Modbus, el cual se consigue en el mercado con un costo de alrededor de \$30 dólares.

El equipo y su diagrama de conexiones se muestra en la [Figura 4.2](#). Las especificaciones más importantes de este medidor se muestran en la [Tabla 4.1](#) y en la [Tabla 4.2](#) se enumeran la precisión con la que mide las diversas variables eléctricas (Eastron 2019).

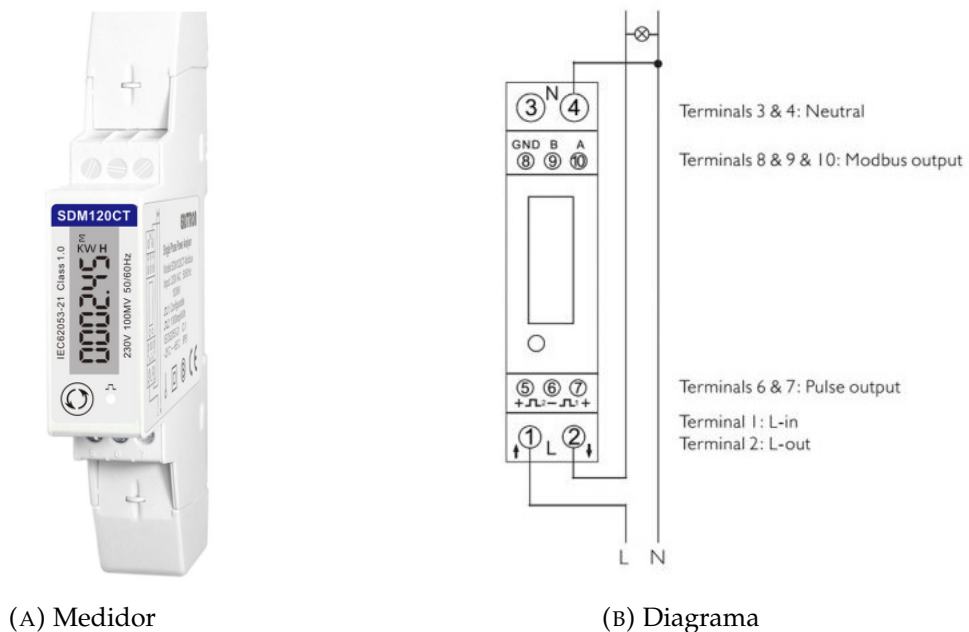


FIGURA 4.2: Medidor de energía Eastron modelo SDM120Modbus.

Especificación	Valor
Voltaje AC (Un)	230V
Rango Voltaje	176~276V AC
Corriente base (Ib)	5A
Max. Corriente (Imax)	45A
Corriente Mínima (Imin)	0.25A
Corriente de inicio	0.4 % of Ib
Consumo de energía	<2W/10VA
Frecuencia	50/60Hz(±10%)
Resistencia a la tensión de CA	4KV por 1 minuto
Tensión de impulso soportada	6KV-1.2uS wavform
Resistencia a sobrecorriente	30Imax for 0.01s
Tasa de salida de pulso	1000imp/kWh (default)
	100/10/1 imp/kWh/kVarh (configurable)
Pantalla	LCD luz de fondo azul
Lectura máxima de energía	99999.9 kWh

TABLA 4.1: Especificaciones del medidor de energía de una sola fase marca Eastron modelo SDM120Modbus.

Medición	Precisión
Voltaje	0.5 % del rango máximo
Corriente	0.5 % del nominal
Frecuencia	0.2 % de la frecuencia media
Factor de potencia	1 % de la unidad
Potencia activa	1 % de rango máximo
Potencia reactiva	1 % de rango máximo
Potencia aparente	1 % de rango máximo
Energía activa	Clase 1 IEC62053-21
	Clase B EN50470-3
Energía reactiva	1 % de rango máximo

TABLA 4.2: Precisión de las mediciones.

Lectura de mediciones y formato de trama

Esta es la parte central del desarrollo del proyecto de tesis complemento a esta tesis, y consta de un sistema mínimo basada en Arduino que realiza a grandes razgos las siguientes actividades:

- Se conecta cada determinado tiempo por medio del protocolo Modbus al equipo de medición para poderle solicitar las variables medidas.
- Después de leer los datos, realiza la limpieza de datos y arma una trama con toda la información recavada con el formato necesario para que el elemento concentrador pueda recuperar los datos.

- Le manda la trama de datos de manera serial por el puerto UART al módulo RF LoRa para que la transmita al receptor que se encuentra en el nodo concentrador.

Todo el hardware necesario se encuentra incluido en una tarjeta PCB como se puede observar en la [Figura 4.3](#).

Radio LoRa

Para la comunicación LoRa se utilizó el mismo módulo “E32-915T20D” utilizado en la parte de recepción del concentrador ([Figura 3.3](#)), este módulo recibe los datos del sistema mínimo de manera serial UART y los envía íntegros como si fuera un cable por radiofrecuencia. Las especificaciones más importantes de este módulo se encuentran en las [Tabla 3.1](#), [Tabla 3.2](#), [Tabla 3.3](#) mencionadas en el capítulo 3.

Fotos del nodo medidor completo instalado se puede ver en la [Figura 4.3](#) y [Figura 4.4](#). De esas imágenes se puede observar que el equipo de medición se conectó directamente en el centro de carga donde se deriva la alimentación a los distintos equipos de aire acondicionado.



FIGURA 4.3: Imágenes del nodo medidor.

Es muy importante señalar que para poder utilizar el máximo de distancia que permite el módulo LoRa, lo ideal era utilizar antenas de alta ganancia externas aéreas, sin embargo, por el precio de las mismas, la complejidad de instalación y por cuestiones de tiempo se decidió usar las antenas pequeñas que se pueden ver en las fotos,

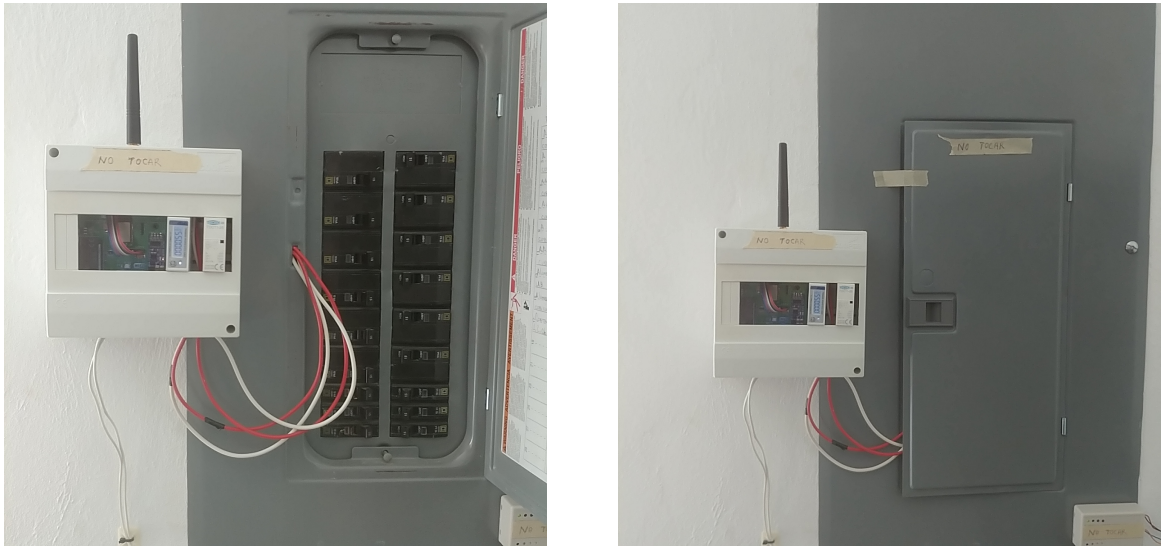


FIGURA 4.4: Nodo medidor instalado.

esto limitó en buena parte el alcance de la comunicación esperada entre los módulos, como se podrá observar en los resultados de las mediciones de distancia realizadas.

A partir de este punto todo lo que se menciona forma parte directa del proyecto asociado con esta tesis.

4.1.2. Concentrador

Como se ha comentado en varias ocasiones, este proyecto está más orientado hacia la parte de software, por tanto el hardware es muy sencillo y se encuentra conformado básicamente por el módulo LoRa “E32-915T20D” (Figura 3.3) y una Raspberry Pi, que por cuestiones de precio y tamaño se escogió una Raspberry Pi Zero W.

Para proteger los módulos y tarjetas electrónicas de la humedad y golpes se utilizó una caja estanca con conectores para permitir el acceso externo de la fuente de alimentación y la antena. La fuente de alimentación que se utilizó fue de 5 Volts a dos Amperios, suficientes para poder alimentar la Raspberry y el módulo LoRa. La interconexión de entre ambas partes se puede consultar en la Sección 3.2. Varias fotos del concentrador armado se puede ver en la Figura 4.5 y Figura 4.6.

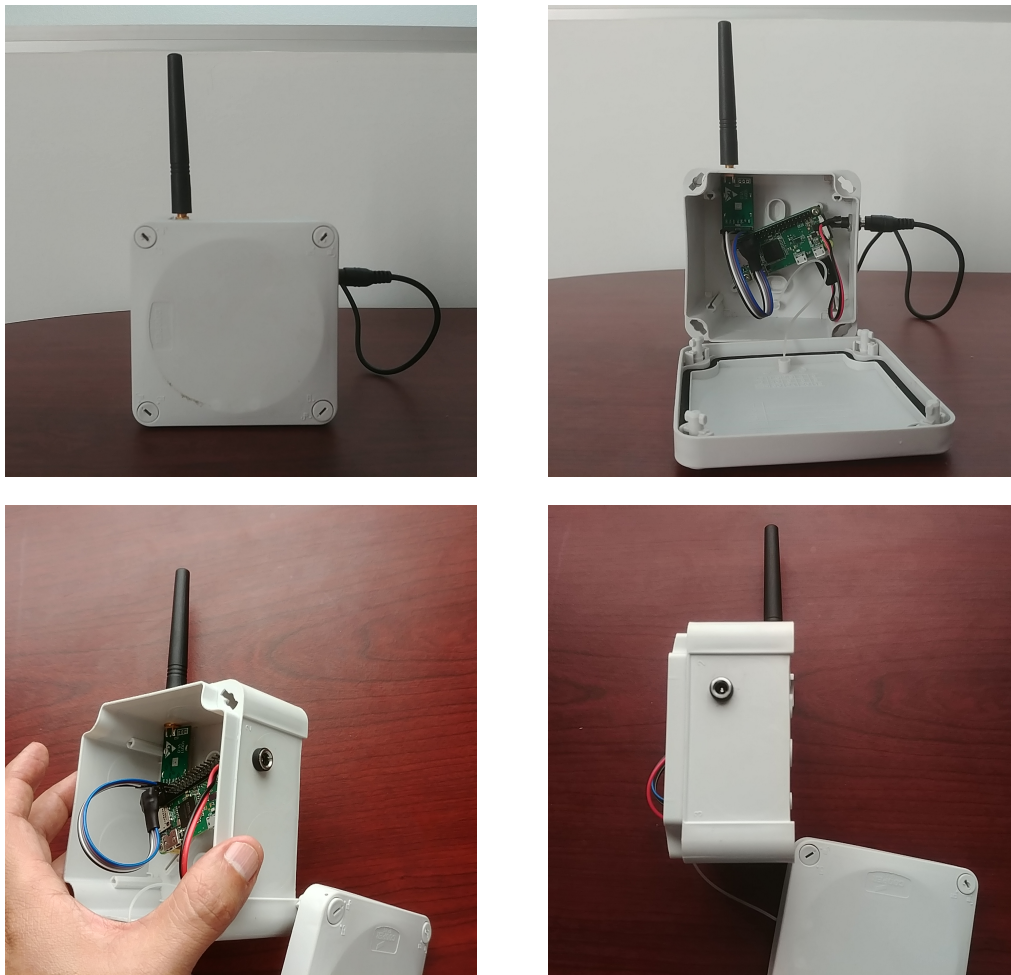


FIGURA 4.5: Nodo concentrador.

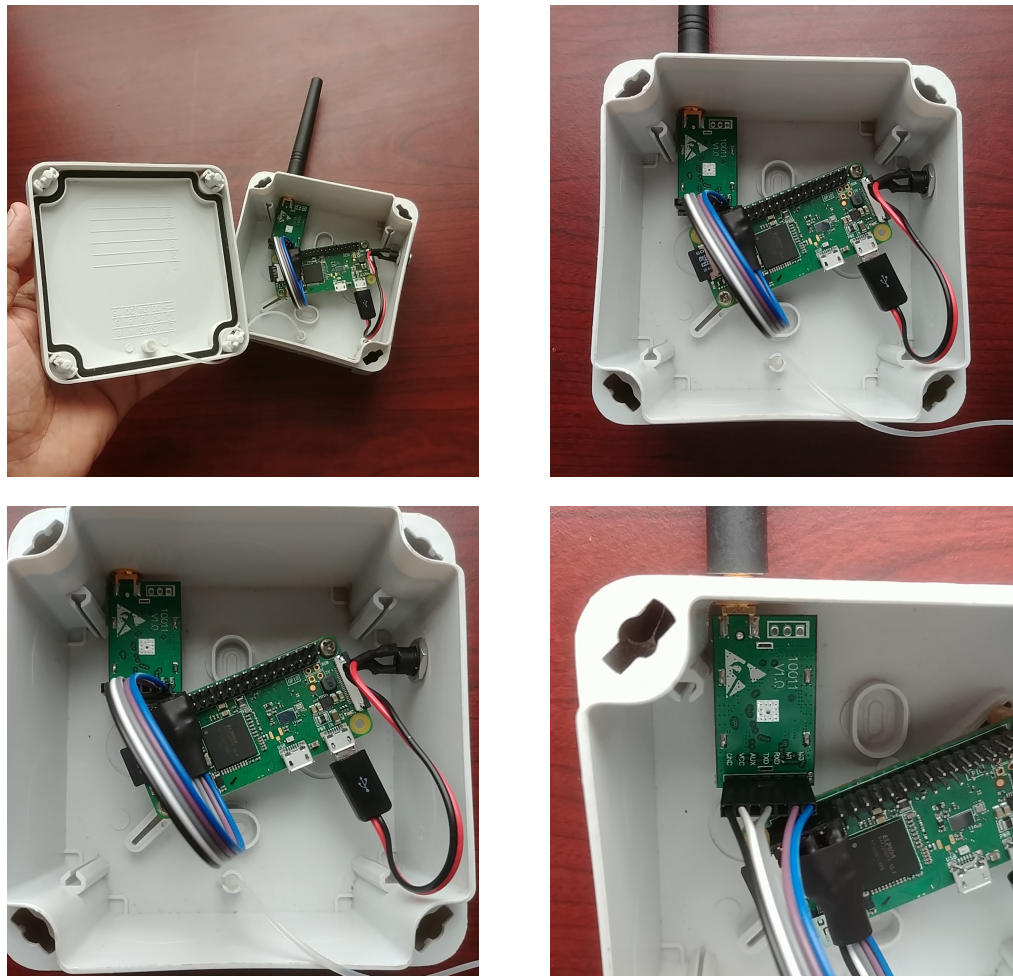


FIGURA 4.6: Nodo concentrador (continuación de fotos)

4.1.3. Equipo de aire acondicionado monitoreado

Para las pruebas de consumo se puso a prueba uno de los equipos de aire acondicionado de los cubículos del edificio de maestría, modelo ICTC-12H2, marca ICECO (Figura 4.7).



FIGURA 4.7: Equipo de aire acondicionado monitoreado.

Las especificaciones más importantes de este clima se muestran en la [Tabla 4.3](#) («ICECO Nueva Linea., Pdf.» 2019).

Especificación	Unidad	Valor
Capacidad	BTU	12000
Alimentación eléctrica	F/V/Hz	1/220/60
Consumo máximo de corriente	A	5.45
Potencia máxima	W	1070
Eficiencia energética EER	W/W	2.94
Separación máxima	m	15 horizontal, 5 vertical
Dimensiones Evaporadora	mm	770x240x180
Dimensiones Condensadora	mm	760x552x256
Peso Evaporadora	Kg	8
Peso Condensadora	Kg	28

TABLA 4.3: Especificaciones del clima utilizado.

4.1.4. Distancia entre nodos

La gran ventaja de usar LoRa es que el equipo monitoreado no tienen que estar cerca de donde se consultan y almacenan los valores medidos, sin embargo para las pruebas que se realizaron, los nodos se encontraron en el mismo edificio a una distancia de cuatro metros, esto debido a la distancia del cubículo asignado con el cuarto donde se encuentra físicamente el centro de carga de los aires acondicionados. Otro punto importante a recalcar es que el nodo concentrador y el aire acondicionado se localizaron en el mismo cuarto, pero esta situación fue totalmente circunstancial, debido a las áreas autorizadas en las que se podían realizar las pruebas. ([Figura 4.8](#)).

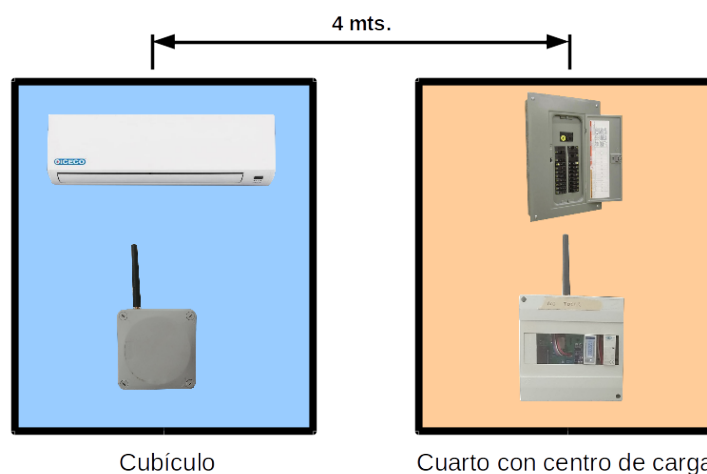


FIGURA 4.8: Distancia entre los nodos

4.2. Descripción de las pruebas

Las pruebas se dividieron en dos grandes partes, la mediciones eléctricas y las pruebas de distancia, a continuación se indican las condiciones bajo las cuales se realizaron cada una de las pruebas.

4.2.1. Pruebas de mediciones eléctricas

La idea original para las pruebas del sistema era dejar el sistema muestreando por algunos días y luego sacar conclusiones de los valores leídos, sin embargo tres situaciones con las que encontramos nos hicieron cambiar esta idea:

1. Debido a que sólo se midió la actividad del clima del cubículo, en las noches y fines de semana no había prácticamente ningún cambio en las mediciones y no había mucha información que analizar.
2. Los primeros días de las mediciones, al parecer por mantenimiento o por gente que tenía acceso al centro de carga, hubo algunos cortes de luz que inicializaban el sistema y ensuciaban la base de datos.
3. Por cuestiones de definición de la señal, se decidió que para estas pruebas el nodo medidor mandaría los datos cada dos segundos, velocidad un poco alta para una aplicación real, tomando en cuenta que el muestreo que capta la información importante se realiza físicamente en el equipo de medición. Para este tipo de aplicación se pronostica que un mejor muestreo sería de entre cada 10 a 20 segundos. La velocidad de muestreo de dos segundos provocaba que la base de datos se llenara de 5 a 10 veces más rápido, y que los periodos sin actividad provocara que la información sin substancia eclipsara los eventos que se habían dado durante el día.

Por esta razón se decidió hacer las pruebas estando en el cubículo durante el día y apagar el sistema en las noches y fines de semana, con esto se pudo conseguir lo siguiente:

- Evitar que se llene la base de datos con información invariante en el tiempo.
- En caso de alguna falla eléctrica estar presente y resolverlo lo antes posible (tomando en cuenta que las pruebas eran para la tesis y se quería que duraran el menor tiempo posible).

- Al realizar las pruebas de una manera presencial se pudo provocar cambios a voluntad en el consumo del aire, tales como apagarlo y prenderlo, hacer cambios de la carga térmica de la habitación con la ocupación y aperturas y cierres de puerta etc.

Las pruebas se realizaron del 4 al 14 de febrero del 2019, teniendo como los días con mayor actividad el 8, 13 y 14 de febrero y que por tanto son los días de donde se tomaron las gráficas y resultados. La temperatura programada del aire acondicionado en todas las pruebas fue de 21 grados.

De los días de prueba reportados es importante destacar la siguiente información:

- El primer día (8 de febrero), se realizaron pruebas de las 10 am a las 6 pm y fue en general un día caluroso, y para variar los consumos se apagó y encendió el aire en varias ocasiones.
- El segundo día (13 de febrero) se realizaron pruebas de las 9 am a casi las 10 pm y fue un día nublado con temperatura fresca, se permaneció todo el tiempo en el interior del cubículo con el aire acondicionado encendido.
- El tercer día reportado (14 de febrero) se realizaron pruebas de las 10 am a casi las 10 pm y fue un día muy caluroso sin nubes, se permaneció todo el tiempo e el interior del cubículo con el aire acondicionado encendido.

4.2.2. Pruebas de distancia

Se realizaron pruebas de comunicación entre el nodo medidor y el nodo concentrador, alejando el concentrador hasta una distancia de 150 metros del nodo medidor, el cual permaneció fijo en el edificio de maestría, el área efectiva donde se realizaron estas pruebas se puede observar de color azul claro en la [Figura 4.9](#).

Las pruebas de distancia fueron más cualitativas que cuantitativas y solo se buscó tener una idea de qué tan lejos el sistema podía seguir recibiendo datos con la antena y las condiciones de prueba elegidas.



FIGURA 4.9: Vista aérea de la zona donde se realizaron las pruebas de distancia

4.3. Resultados

4.3.1. Resultados de mediciones eléctricas

Las mediciones se analizan desde dos puntos de vista, el análisis de las variables instantáneas que se refrescan cada vez que llega una trama de datos con mediciones nuevas, y el análisis de las variables históricas, que corresponden a las gráficas generadas con las consultas a la base de datos desde el dashboard.

Mediciones eléctricas instantáneas

De la [Figura 4.10](#) a la [Figura 4.15](#) se muestran pantallas con mediciones instantáneas capturadas a diversas horas del día. En esta parte el análisis se centró en las mediciones que se muestran en los instrumentos tipo manómetro y texto de los grupos "Valores Instantáneos" y "Otras mediciones". Las gráficas de mediciones históricas se analizarán en la siguiente sección con más detenimiento.

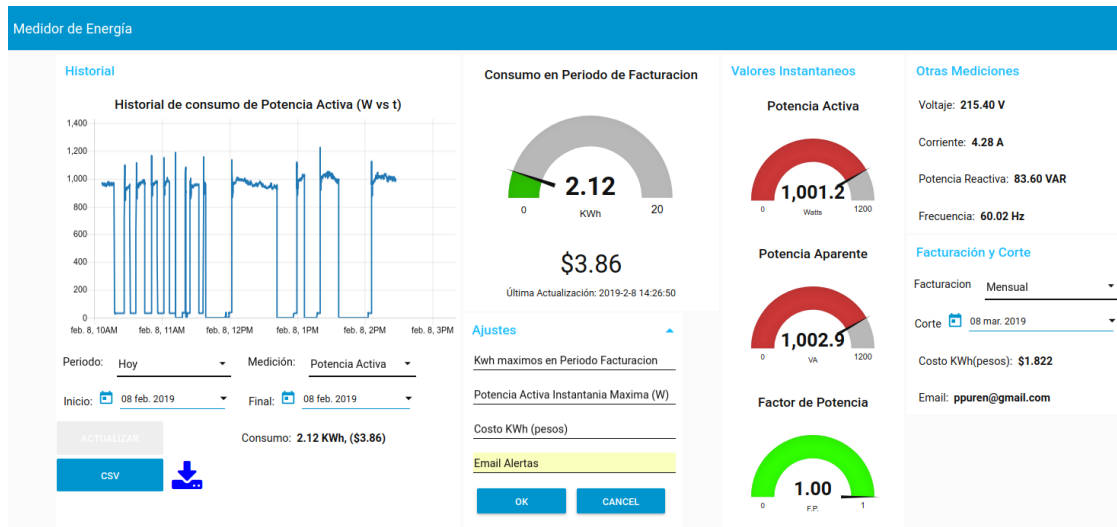


FIGURA 4.10: Clima con compresor encendido 2:26 pm

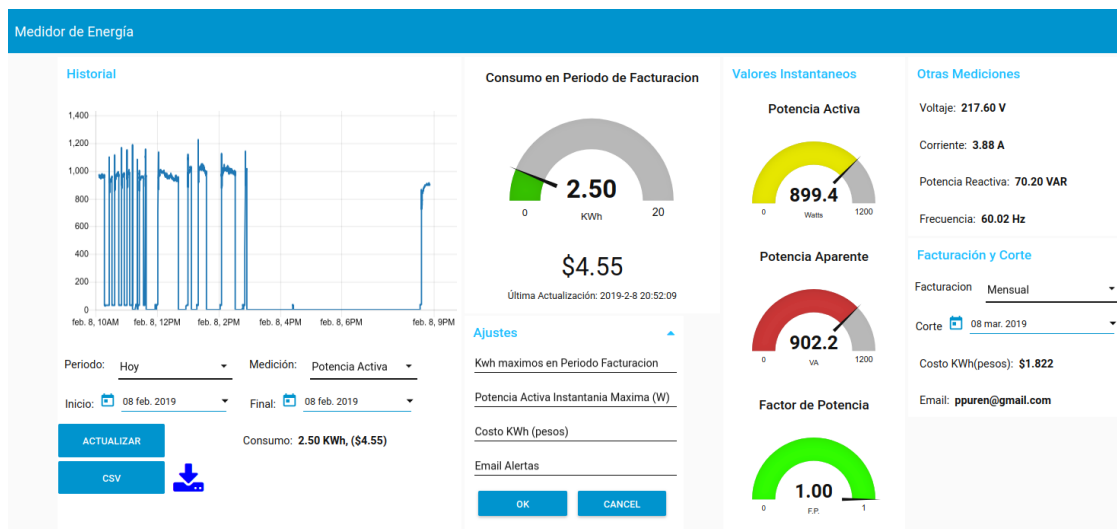


FIGURA 4.11: Clima con compresor encendido 20:52 pm

De las mediciones se desprenden algunas conclusiones:

- El consumo del aire acondicionado reportado en las especificaciones (Tabla 4.3), concuerda con las mediciones realizadas, se puede observar que la corriente máxima leída es de 4.28 amperios por debajo de la corriente máxima especificada de 5.45 amperios, y que el consumo máximo reportado en todos los casos fue de 1000 watts, por debajo de la potencia máxima especificada de 1070 watts.
- El consumo del compresor se ve afectado por la temperatura externa y por tanto, por la hora del día en que se realicen las mediciones, resultando en un

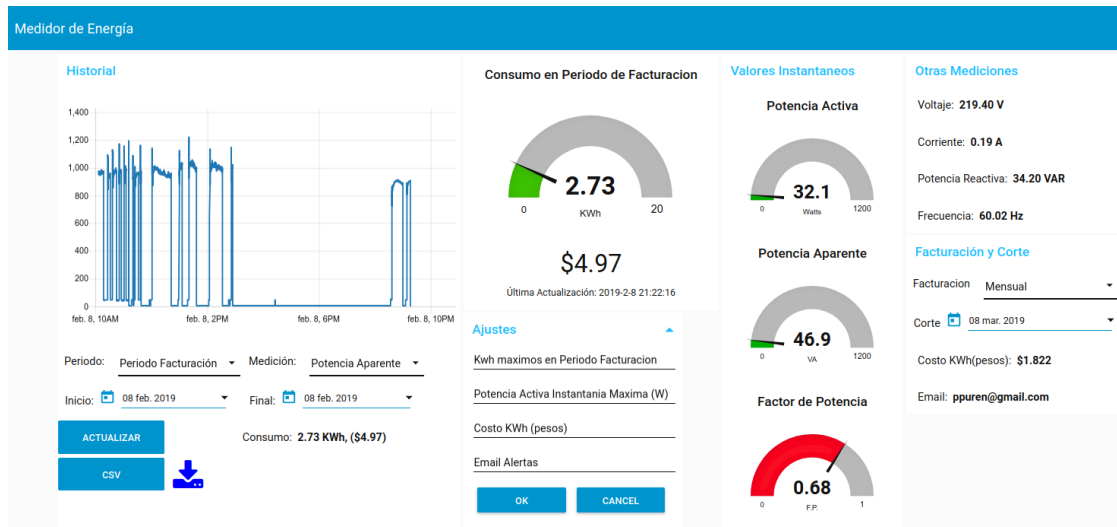


FIGURA 4.12: Clima encendido, solo evaporador

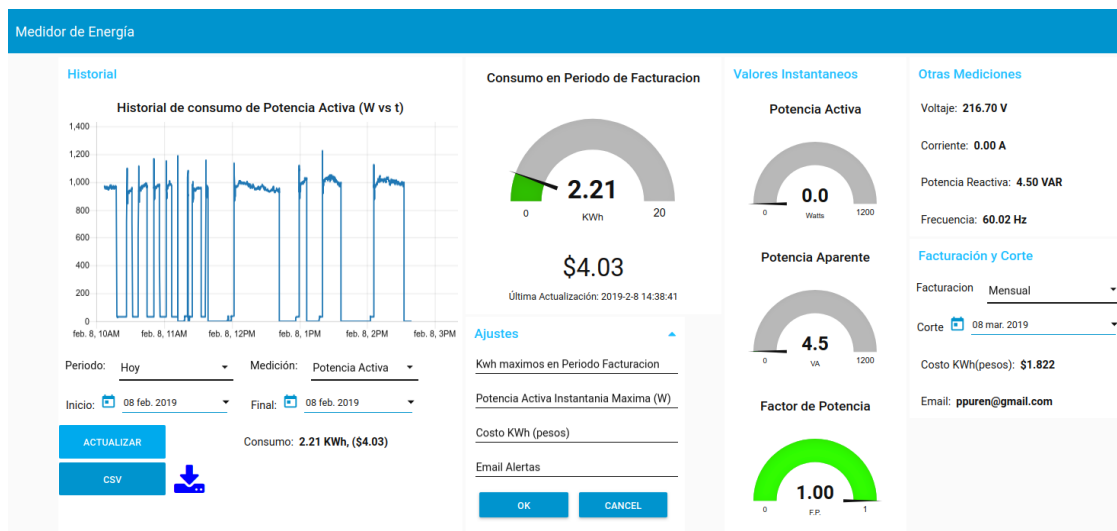


FIGURA 4.13: Clima apagado

trabajo menor cuando la temperatura externa es más fresca y un consumo mayor cercano al consumo máximo cuando la temperatura externa es alta¹.

- Al parecer el aire acondicionado cuenta con algún circuito de compensación, ya que después de un momento de haber arrancado del compresor, la señales de voltaje y corriente entran en fase y el factor de potencia se va a uno.

¹Esto podría parecer obvio, sin embargo debido a que el control del aire es ON/OFF, cuando enciende el compresor trabaja a plena carga y en teoría, el consumo debería ser el mismo independientemente de la temperatura externa. La explicación de este fenómeno escapa de los alcances de este proyecto de tesis, sin embargo se presume que el aumento de consumo se debe a un cambio en la viscosidad del gas de enfriamiento.

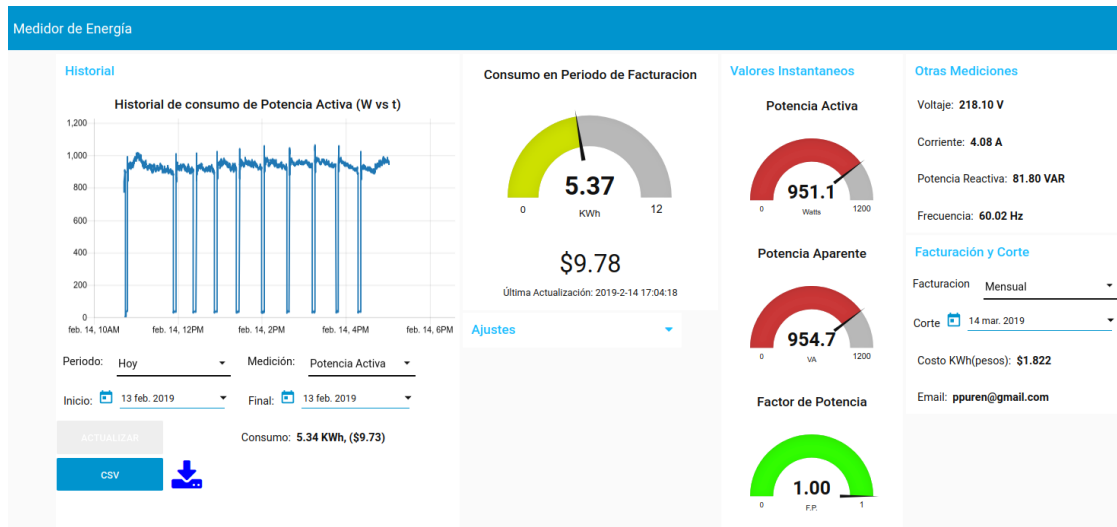


FIGURA 4.14: Clima con compresor encendido 17:04 pm

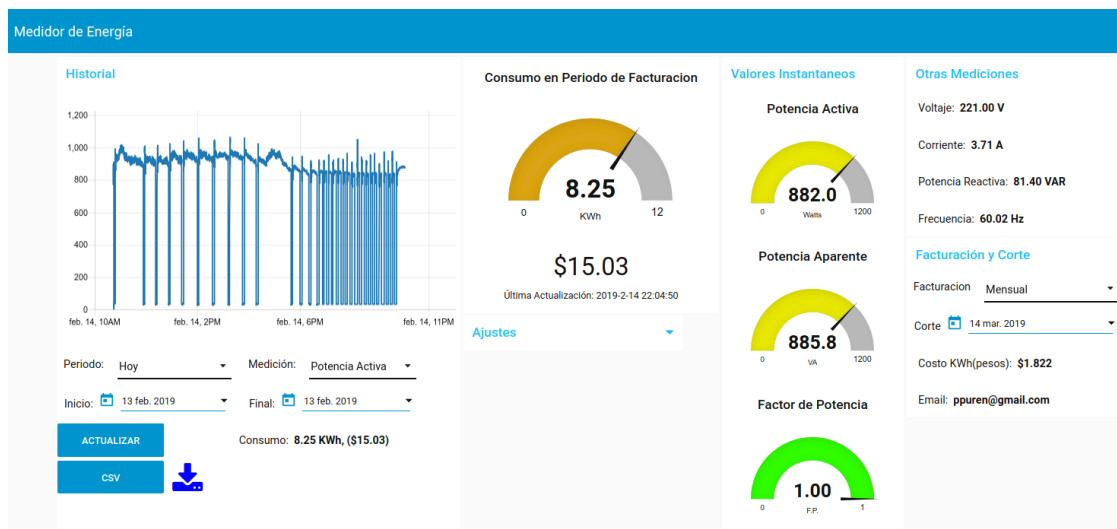


FIGURA 4.15: Clima con compresor encendido 22:04 pm

- Como es de esperarse el consumo del evaporador es casi despreciable comparado con el consumo del compresor.

Mediciones eléctricas históricas

En la [Figura 4.16](#), [Figura 4.17](#) y la [Figura 4.18](#) se muestran gráficas de la información almacenada durante los días con más actividad en el cubículo.

Del análisis de las gráficas se puede observar lo siguiente:

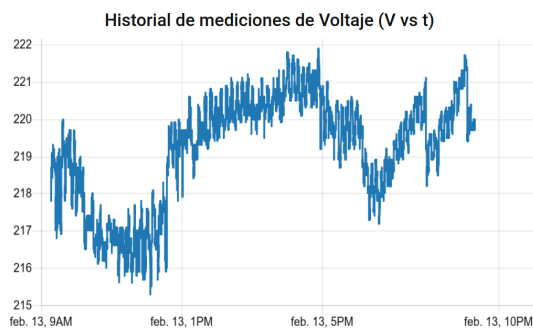
- En cada uno de los días se presentó un rango de variación del voltaje de aproximadamente 6 volts.
- Como es de esperarse al momento del arranque del compresor se presenta un pico de corriente.
- Con respecto al pico de corriente, la razón de que a veces se generen picos de corriente muy altos y en otras ocasiones sean pequeños se debe a la velocidad de muestreo de dos segundos, ya que al ser el pico de corriente muy corto en tiempo, es muy probable que en varias ocasiones éste se dé entre eventos de muestreo y no alcance a ser registrado².
- De las gráficas de potencia activa se puede ver que el compresor necesitaba más energía para trabajar cuando la temperatura externa era mayor³, esto se respetó tanto en los días más calurosos, como en los cambios de temperatura del mismo día.
- También se puede observar que cuando hay más calor en el exterior, al compresor le cuesta más trabajo llegar a la temperatura programada y por tanto permanece más tiempo encendido.
- El factor de potencia permaneció la mayor de parte del tiempo en uno, y únicamente varió en los momentos en que sólo se encontraba encendido el evaporador, o momentáneamente cuando arrancaba el compresor.
- Como se observó de las mediciones instantáneas, el clima cuenta con algún circuito corrección de factor de potencia, lo cual explica que las gráficas de potencia activa y aparente sean prácticamente iguales y la potencia reactiva sea casi despreciable.
- La frecuencia fue la variable medida más estable, permaneciendo casi todo el tiempo en prácticamente 60 Hz, sólo presentando en ocasiones una pequeña variación de apenas ± 1 Hz.

²A manera de justificación la razón de este proyecto no es el análisis de la calidad de la energía sino de los consumos, sin embargo consiguiendo un medidor de energía que contenga mediciones tales como armónicos y transitorios se podrían incluir estas funcionalidades en el dashboard.

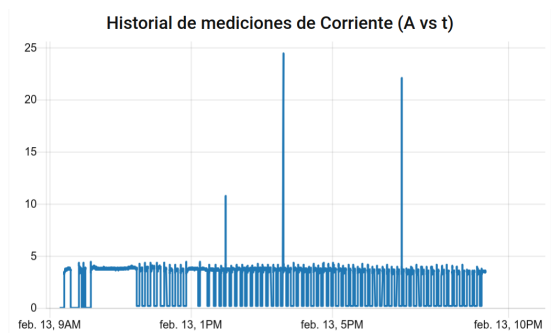
³Conclusión a la que se llegó también derivada los valores instantáneos medidos.



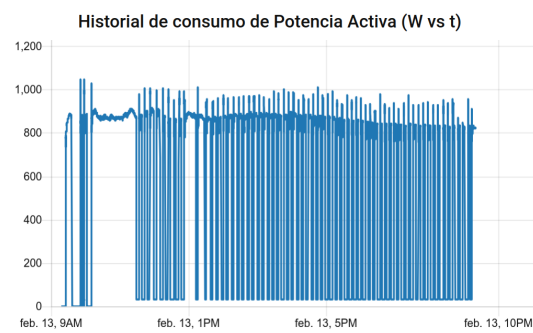
FIGURA 4.16: Gráficas de mediciones eléctricas 8 febrero del 2019



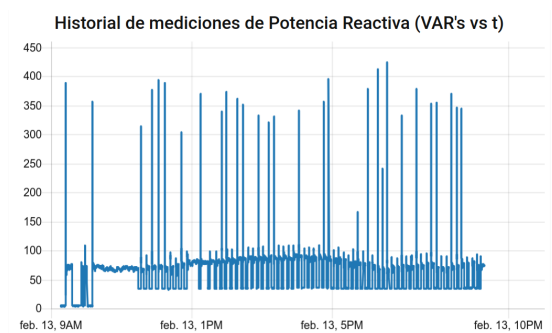
(A) Voltaje



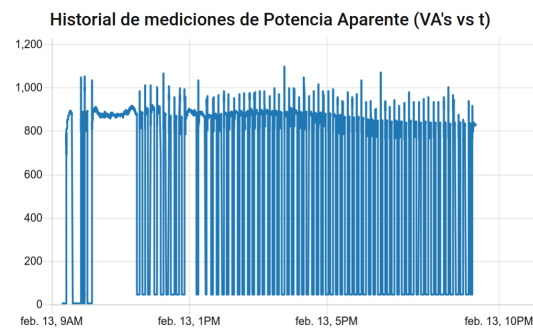
(B) Corriente



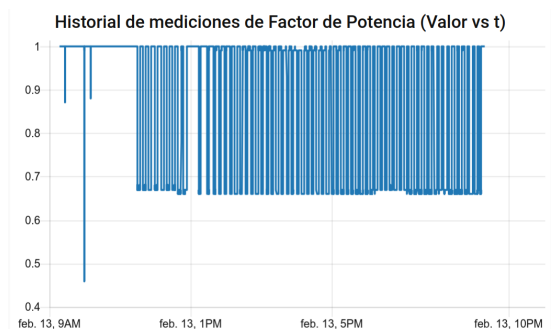
(C) Potencia activa



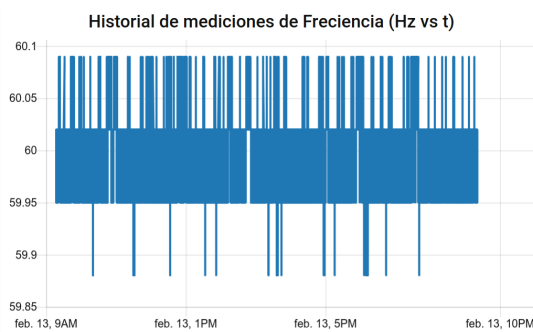
(D) Potencia reactiva



(E) Potencia aparente



(F) Factor de potencia



(G) Frecuencia

FIGURA 4.17: Gráficas de mediciones eléctricas 13 febrero del 2019

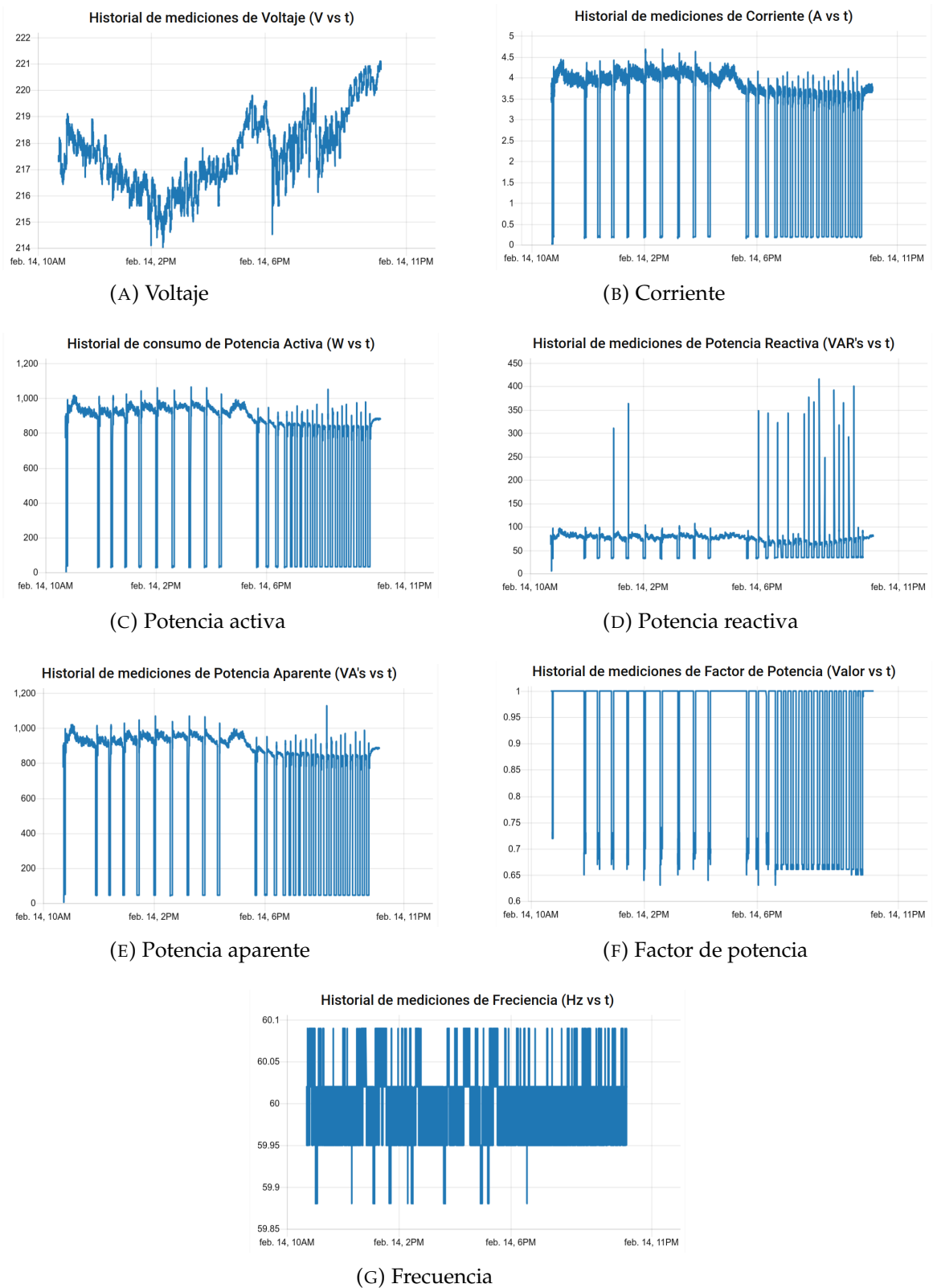


FIGURA 4.18: Graficas de mediciones eléctricas 14 febrero del 2019

Cada vez que se realiza una consulta, el sistema reporta el consumo efectivo realizado en el periodo y una estimación del costo de la energía calculada con el precio definido en el grupo de "Facturación y corte", que para las pruebas se especificó como \$1.822 pesos por kilowatt hora. En la [Figura 4.19](#) se puede observar los consumos que el sistema reportó para cada uno de los tres días de prueba y su respectiva estimación en pesos. Estos resultados revelan lo siguiente:

- El día que menos energía se consumió fue cuando se apagó y encendió el aire y que fue el día que las pruebas acabaron temprano a las 6 pm.
- El 13 y 14 de febrero, días en que la duración de las pruebas fue bastante similar y que en ambos casos el equipo de aire acondicionado permaneció todo el tiempo encendido, se encontró una variación de más de 2 kilowatts hora. La razón fue que el 14 de febrero fue un día más caluroso y por tanto el compresor permaneció más tiempo encendido y se consumió más energía cada vez que el equipo se encendía.

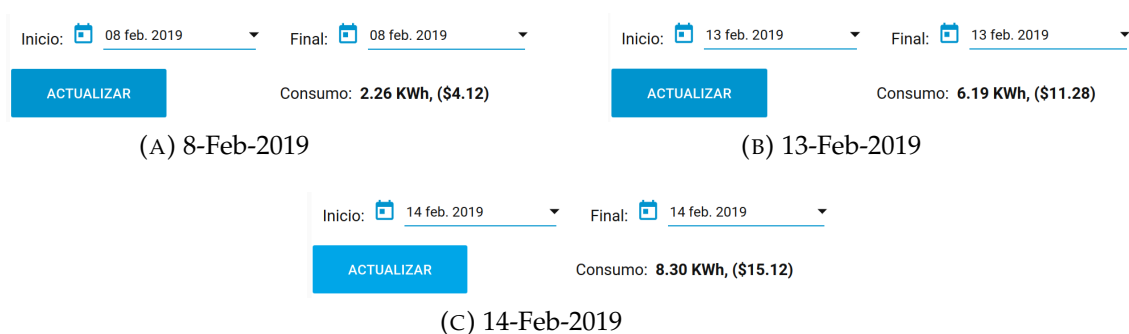


FIGURA 4.19: Consumo de energía por día

4.3.2. Resultados pruebas de distancia

Como se mencionó las pruebas de distancia se llevaron a cabo en una sección circular con un radio de 150 metros ([Figura 4.9](#)).

Para tener una imagen más clara de la topología y de los obstáculos en la zona de prueba, se realiza la siguiente clasificación de los edificios y obstáculos en la zona de prueba ([Figura 4.20](#)):

Edificio de maestría.- Lugar donde se realizaron las pruebas y punto en el que permaneció todo el tiempo el nodo medidor, transmisor de los valores medidos vía LoRa.

Estacionamiento y calle.- Áreas abiertas, que aunque tienen árboles grandes no parecen ser obstáculos tan importantes, y por tanto pueden considerarse línea de vista.

Auditorio de básquetbol y biblioteca.- Edificios muy altos de más de 5 metros con paredes muy gruesas y que representaron un reto para la señal de radiofrecuencia.

Laboratorio de electrónica.- Edificio de un solo piso, con paredes gruesas y con varios cuartos y paredes en su interior.

Como dato último, el suelo en la escuela es plano y no se encuentra ningún cerro ni valle.



FIGURA 4.20: Topología de la zona de de pruebas de distancia

Los módulos LoRa no cuentan con alguna medición o medio de retroalimentación para saber con qué potencia se recibe la señal de radiofrecuencia, por lo que la manera en que se clasificó la calidad de la recepción fue en tres niveles:

1. Recepción de señal fuerte, esta opción correspondió a las zonas donde las tramas llegaban sin problema y sin tener que tratar de buscar la señal moviendo o cambiando la posición del receptor.
2. Recepción de señal débil, este nivel correspondió a las zonas donde era necesario mover el receptor o ponerlo en alto para poder recibir la señal. Por lo general, una vez que se conseguía una recepción exitosa, ya no se volvían a perder más paquetes, por lo que ya no era necesario mover o cambiar de posición el receptor.
3. Falla en recepción de datos, en estas zonas no se conseguía recibir paquetes, inclusive cambiando la posición del receptor, y en las ocasiones que se lograba captar alguna trama, por lo general era aislada y se volvía a perder conexión.

Es importante aclarar que no obstante el módulo LoRa promete 3000 metros en su hoja de datos ([Tabla 3.1](#)), se sabía que los resultados estarían muy por debajo, ya que en la misma hoja de datos se especifica que esa distancia requiere línea de vista, con cielo despejado, a potencia máxima y con una antena aérea a 2 metros del piso.

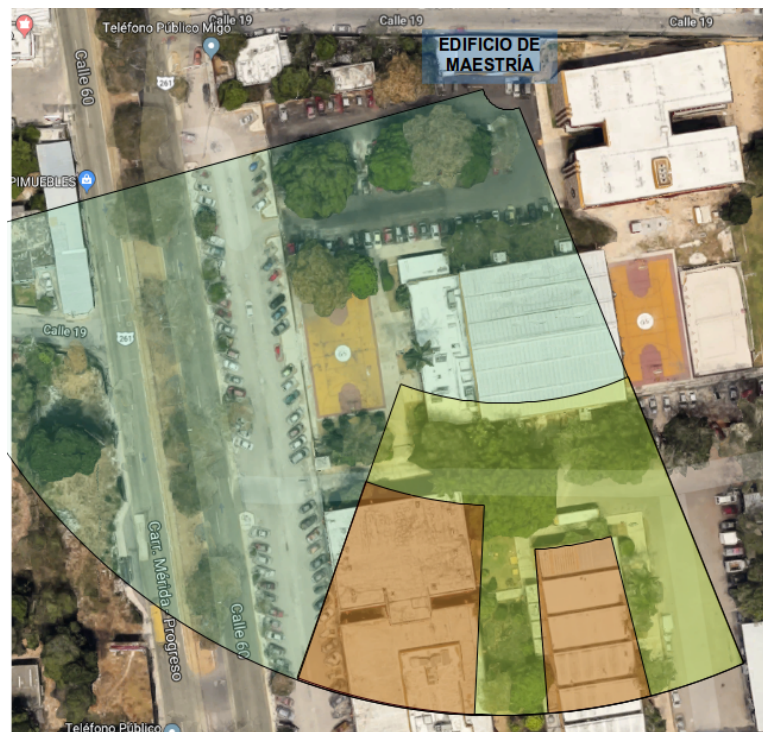


FIGURA 4.21: Resultados de las pruebas de distancia

Los resultados de las pruebas se muestran en la [Figura 4.21](#), con recepción fuerte de señal marcada en verde, la recepción débil en amarillo y la falla en recepción de datos en rojo.

De los resultados se encontró lo siguiente:

- En línea de vista el módulo parece funcionar muy bien, no se pudo verificar más de 150 metros por la limitante física de la topología del lugar.
- Una o dos paredes no parecieron afectar mucho la recepción, por lo menos dentro de los primeros 50 metros.
- Más de dos a tres paredes, especialmente si las paredes eran gruesas, empezaban a provocar pérdidas de paquetes.
- A más de 100 metros, cuando el módulo de recepción se metía a un edificio provocaba la falla de recepción.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

Del [Capítulo 3](#) y [Capítulo 4](#) se puede observar que efectivamente se desarrolló y se cumplió con los objetivos general y específicos propuestos. A lo largo del proceso se llegó a las siguientes conclusiones generales:

- Con un costo relativamente bajo¹, se puede agregar la funcionalidad IoT a cualquier medidor de energía con ModBUS, o como se vio en el [Subsección 4.1.1](#) se puede adquirir un medidor con un precio asequible y agregarle la funcionalidad IoT².
- La tecnología de comunicación inalámbrica LPWAN encaja perfectamente con las necesidades del IoT, ya que con poco consumo de energía se consiguen alcances muy por arriba del de otras tecnologías inalámbricas. Además se confirmó que las bajas velocidades de comunicación tampoco son una limitante, ya que por lo general los paquetes IoT no exceden unas cuantas decenas de bytes que se envían entre periodos relativamente largos³.
- En caso de que la intención del medidor de energía sea tener una mayor fidelidad de la señal, con el fin de capturar cambios efímeros en las variables eléctricas, sería necesario aumentar la frecuencia con la que se almacenan los datos en el concentrador, sin embargo esto contraería un costo de procesamiento

¹Calculando los costos del proyecto total, nodo medidor sin medidor de energía y nodo concentrador, se calcula que el costo no supera los \$130 dólares.

²El medidor utilizado marca Eastron modelo SDM120Modbus tiene un costo de alrededor de \$30 dólares, dando un total de aproximadamente \$160 dólares para el medidor de energía eléctrica IoT.

³Los sensores IoT miden variables que por lo general no cambian abruptamente con respecto al tiempo, como temperatura, humedad, radiación, consumo de energía etc., y que por tanto no necesitan mandar datos constantemente al concentrador, teniendo periodos entre transmisiones en el rango de los segundos o inclusive minutos.

y almacenamiento, por lo que lo ideal sería conseguir medidores que detecten los eventos importantes a reportar⁴, o implementar algoritmos de lectura de eventos en el nodo medidor, de manera que en lugar de aumentar la frecuencia con la que se envían paquetes al concentrador, se pueda detectar estos casos particulares y reportarle al concentrador solo cuando estos sucedan.

- El internet de las cosas es una realidad que vamos a encontrar con más frecuencia en nuestro día a día, la prueba clara de esto es el florecimiento de tecnologías cada vez más especializadas en IoT, que permiten desarrollar en periodos de tiempo más cortos, proyectos más complejos. El uso de estas tecnologías brinda ventajas, pero también aleja un poco al desarrollador del control de lo que realmente esta sucediendo, por tanto es de vital importancia indagar en el funcionamiento de las herramientas y en la manera en qué interactúan entre ellas.

5.2. Cambios

Desde el principio del desarrollo del proyecto se contó con una idea clara de cómo desarrollar el proyecto, sin embargo por cuestiones de tiempo, se presentaron tres grandes cambios en la idea original:

- Se cambió de lenguaje de desarrollo, como se menciona en la [Subsección 2.4.1](#), node red se encuentra desarrollado sobre NodeJS, originalmente se tenía pensado desarrollar el sistema sobre este lenguaje, pero hubiera aumentado mucho la complejidad del proyecto y el tiempo de desarrollo.
- De manera similar, se tenía contemplado realizar de cero la página web, utilizando JavaScript, HTML y CCS, pero se abandonó esta idea durante el desarrollo del proyecto, ya que se observó que no se hubiera podido cumplir los tiempos del cronograma.
- Se pensaba usar comunicación bidireccional con el módulo LoRa para poder definir los registros ModBus del medidor desde el dashboard, pero se optó por dejar de último esta opción si el tiempo lo permitía, lo cual resultó no ser posible.

⁴Como transitorios por ejemplo.

5.3. Recomendaciones

Durante las pruebas, el proyecto tal como se propuso fue funcional, sin embargo se presentaron algunas situaciones que dejaron entrever posibles complicaciones a futuro, a continuación se mencionan:

- (a) Se encontró en algunos casos de procesamiento intenso, que la Raspberry Pi Zero W tardó más de lo esperado en dar respuesta a las consultas, por lo que es altamente recomendable cambiar a otro modelo de Raspberry con mayores capacidades de procesamiento, se recomienda modelos a partir de la Raspberry Pi 3.
- (b) El desempeño de la base de datos fue el esperado, sin embargo se percibió mayor trabajo y tiempo de respuesta al realizar consultas conforme la base de datos aumentaba su tamaño, por tiempo no se pudo migrar o hacer pruebas con otras bases de datos, por tanto se recomienda cambiar la base de datos por otras que se adecuen más al IoT; como ejemplo existen base de datos conocidas como NoSQL, que corresponden a bases de datos no relacionales y que ofrecen algunas ventajas sobre las bases de datos relacionales como por ejemplo:
 - Mejores tiempos de respuesta.
 - Más flexibilidad ante cambios en la estructura de la base de datos.
 - Escalabilidad horizontal⁵.
 - Herramientas de administración, algunas de estas bases de datos contienen funcionalidades que permiten realizar consultas y ejecutar scripts automáticamente, respaldar los datos más antiguos para evitar que la base de datos crezca mas allá de un límite etc.

Algunos ejemplos de bases de datos NoSQL son MongoDB, DynamoDB, Redis, BigTable etc.

- (c) Dependiendo de las necesidades particulares del proyecto, instalar una antena externa área para los módulos LoRa y así poder obtener un canal de comunicación inalámbrica más robusto.

⁵Este término quiere decir que pueden aumentar la carga de trabajo agregando más servidores, en contraste con la escalabilidad vertical que ofrecen las bases de datos relacionales, que implican cambios de las capacidades del ordenador que contiene la base de datos, tales como cambios al CPU por uno con más capacidad de procesamiento, aumento la memoria RAM etc.

5.4. Trabajo posterior

Como se ha podido ver a lo largo de la tesis, el Internet de las Cosas es muy amplio y en constante cambio, por lo que siempre van a existir cosas por agregar o mejorar, especialmente en este proyecto que se enfocó en darle las capacidades mínimas a los medidores eléctricos ya existentes. De entre las capacidades y trabajo que se vislumbra en el futuro cercano se encuentran las siguientes opciones:

- Añadir capacidades de control, la importancia de conocer en tiempo real las variables eléctricas y el consumo de energía es solo el inicio de todo un proceso, el siguiente paso es poder tomar decisiones de acuerdo a los valores medidos o a la velocidad con que éstos cambian. Se tiene la opción de agregarle actuadores virtuales al dashboard para poder hacer cambios manuales o un poco más complejo, permitir cambios automáticos en tiempo real, dependiendo de eventos de disparo predefinidos.
- En caso de no seguir la recomendación de cambiar el tipo de base de datos por alguna que interactúe mejor con el IoT, se podría incluir herramientas de administración de base de datos, tales como respaldo de información cada determinado tiempo o al alcanzar cierto límite de tamaño de la base de datos, también se podría manejar niveles de prioridad de la información por edad, para poder agilizar la velocidad de respuesta de las consultas.
- Incorporar servicios de nube, esto es de vital importancia, especialmente si la meta es un producto comercial, ya que se ampliaría enormemente las capacidades del proyecto, permitiéndole entre otras cosas, acceso a los datos medidos desde cualquier lugar del mundo, mantener acceso a los datos inclusive si el sistema se desconectó o fallo, respaldos de la base de datos, procesamiento en la nube, entre otros.
- Agregar Seguridad, hay maneras de incorporar usuarios y contraseñas al dashboard, de hecho es muy sencillo y lo incluye el Node-RED, sin embargo especialmente si se conecta el medidor a Internet, es de vital importancia encriptar la información y agregar certificados de seguridad, por fortuna estas funcionalidades también las ofertan las compañías que proveen los servicios de nube.
- Incluir un servidor local DNS o contratarlo en caso de agregar los servicios de nube, con el objetivo de que la dirección de acceso al medidor sea un nombre más familiar en vez de la dirección IP.

- Agregar comunicación bidireccional con el módulo LoRa, de tal manera que se pueda mandar información al nodo medidor, permitiendo cambios a la manera que trabaja el nodo medidor desde el dashboard sin necesidad de cambiar su programa. Entre los cambios que se podrían incluir están la reubicación de los registros ModBus según el medidor usado, cambios en en las configuraciones del canal de comunicación o en el formato de la trama, periodo de transmisión de datos del nodo medidor, etc.
- Ampliar las capacidades del nodo concentrador y del dashboard, de manera que el proyecto no solo sea para medir energía, sino que pueda incluir mediciones enfocadas a la calidad de la energía como transitorios y armónicos.

Apéndice A

Flujos del programa

A.1. Actualización de consumos

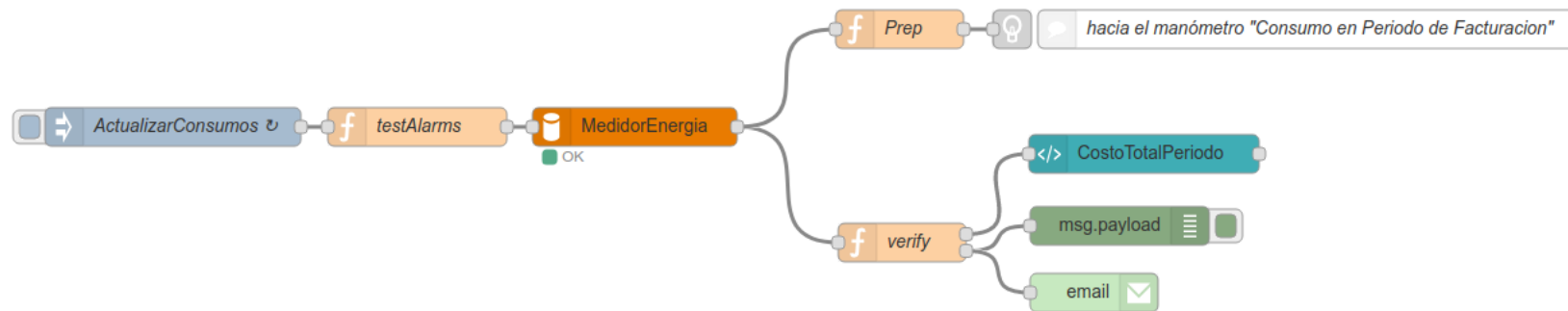


FIGURA A.1: Flujo de actualización de consumos y alerta de consumo excedido

A.2. Recepción de datos

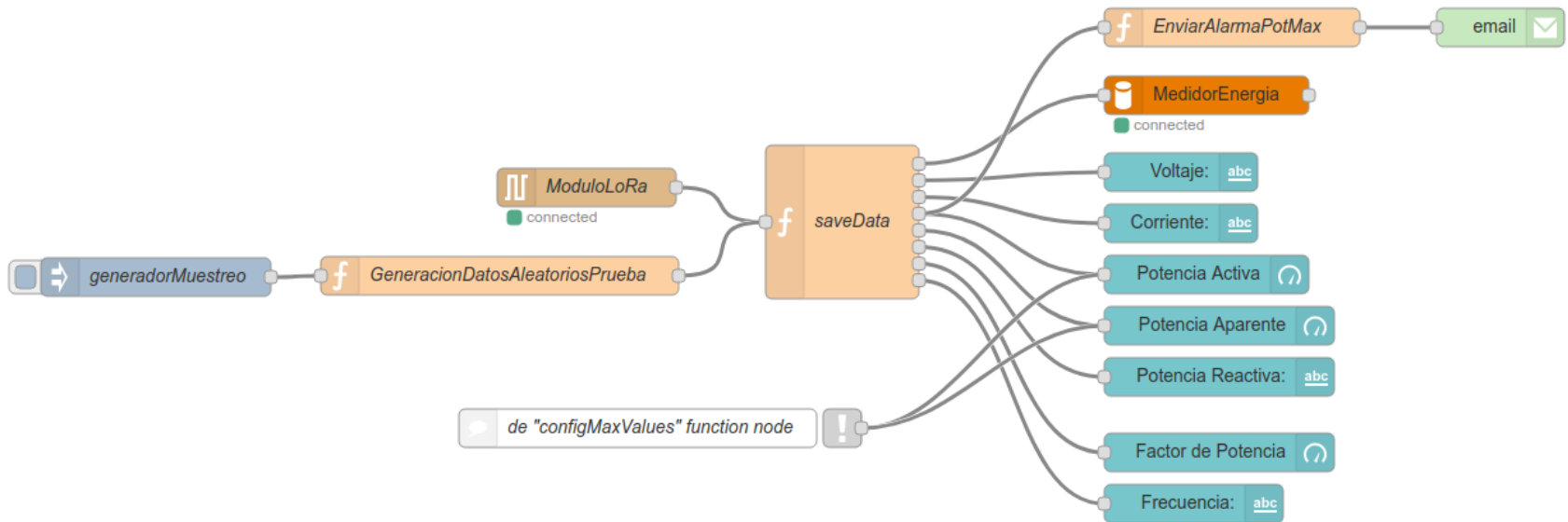


FIGURA A.2: Flujo de recepción de datos y actualización del dashboard con alerta de potencia máxima

A.3. Cambio de variables estáticas

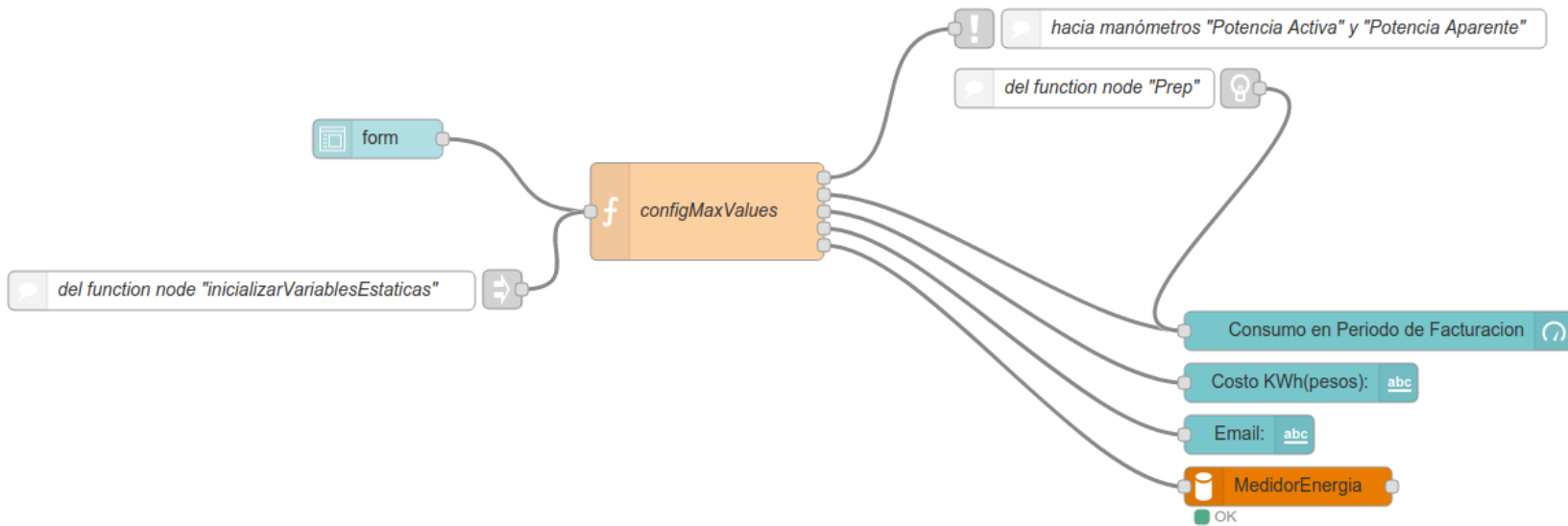


FIGURA A.3: Flujo de cambio de variables estáticas

A.4. Inicialización del dashboard

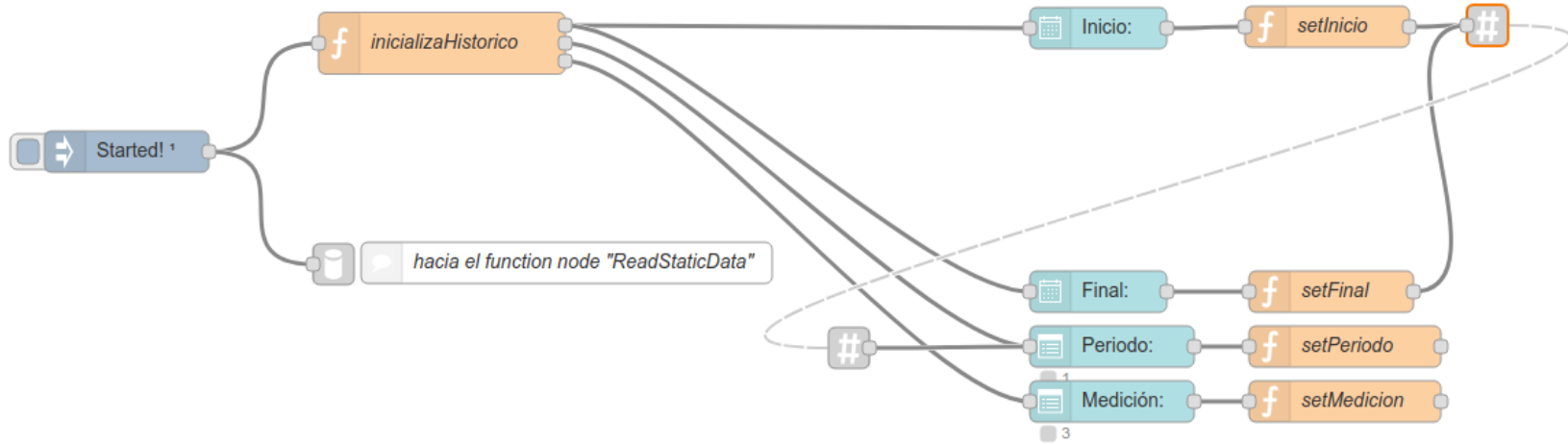


FIGURA A.4: Flujo de inicialización del dashboard

A.5. Consultas a la base de datos

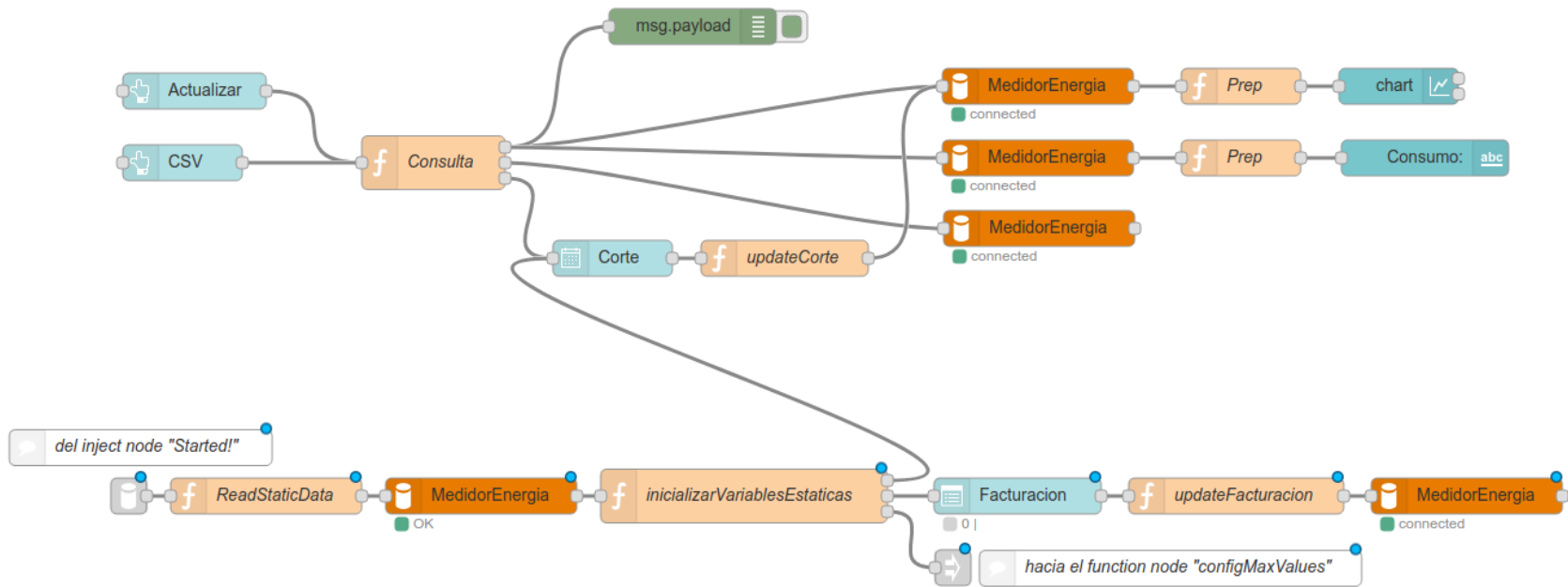


FIGURA A.5: Flujo de consultas a la base de datos e inicialización de variables estáticas

Apéndice B

Estructura de la interfaz de usuario

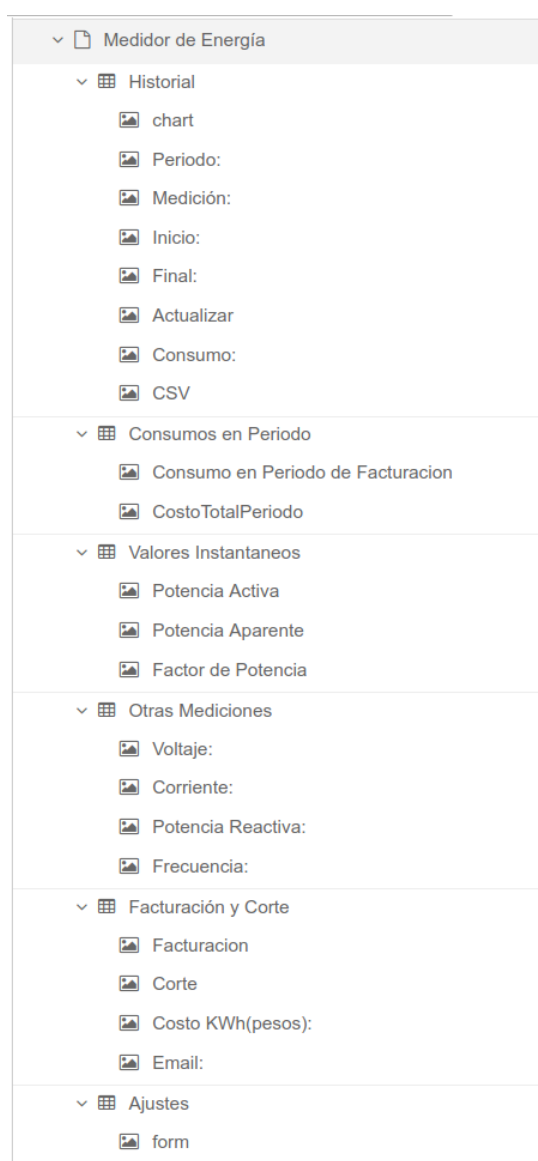


FIGURA B.1: Estructura jerárquica de la interfaz de usuario

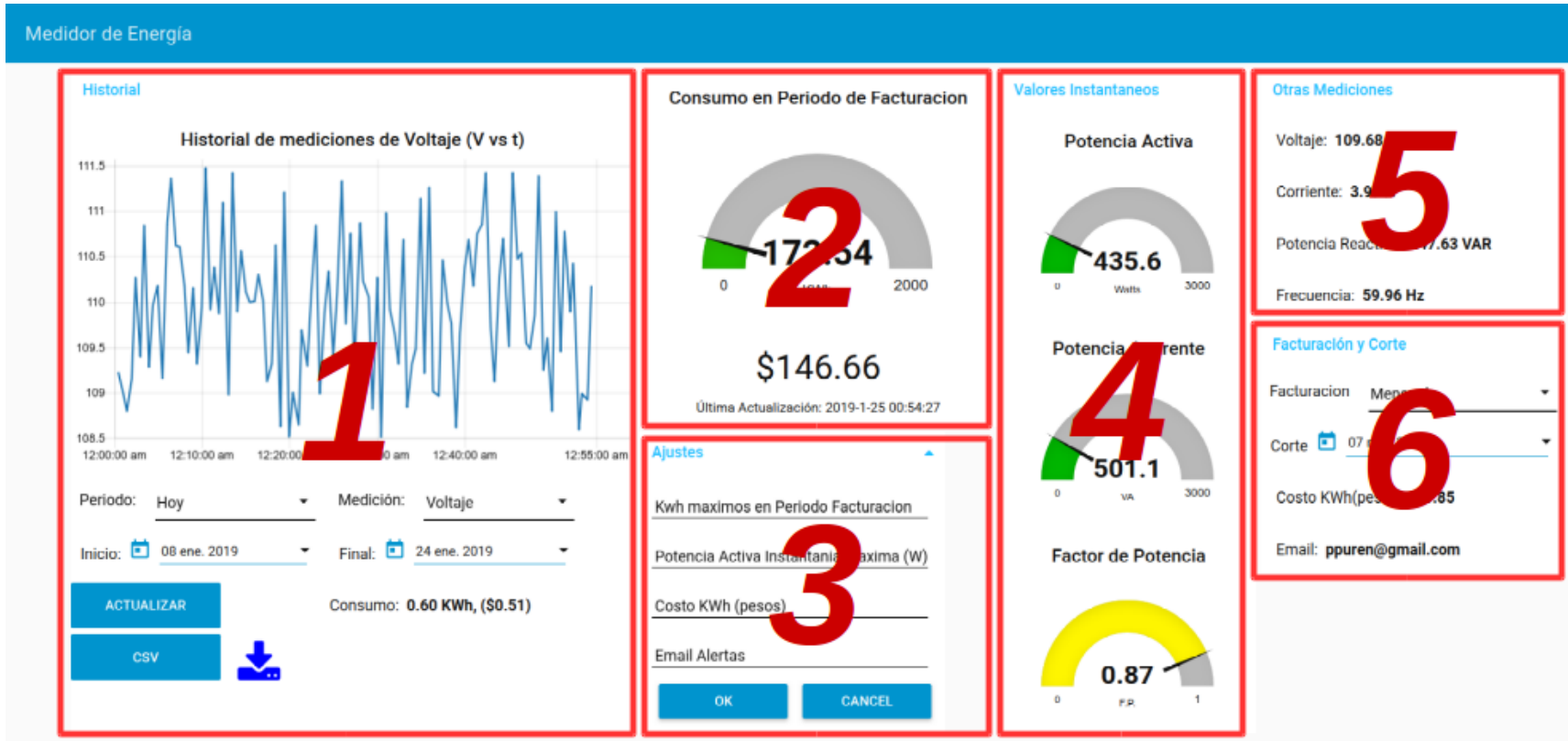


FIGURA B.2: Grupos que conforman la interfaz de usuario o dashboard

Apéndice C

Estructura y análisis de la base de datos

C.1. Estructura

variablesEstaticas

Columna	Tipo	Nulo
uniqueKey	tinyint(4)	No
facturacion	tinyint(4)	No
fechaCorte	bigint(20)	No
costoKwh	float	No
kwhMax	float	No
potMax	float	No
email	text	No

FIGURA C.1: Estructura de la tabla *variablesEstaticas*

Mediciones

Comentarios de la tabla: Tabla de mediciones Eléctricas

Columna	Tipo	Nulo
estampaDeTiempo	timestamp	No
voltaje	float	No
corriente	float	No
potenciaActiva	float	No
potenciaAparente	float	No
potenciaReactiva	float	No
factorDePotencia	float	No
frecuencia	float	No
totalEnergiaActiva	float	No
totalEnergiaReactiva	float	No

FIGURA C.2: Estructura de la tabla *Mediciones*

C.2. Análisis

Resultado SQL

Servidor: localhost

Base de datos: MedidorEnergia

Tiempo de generación: 03-06-2019 a las 19:11:24

Generado por: phpMyAdmin 4.2.12deb2+deb8u3 / MySQL 5.5.60-0+deb8u1

consulta SQL: SELECT * FROM `variablesEstaticas` PROCEDURE ANALYSE();

Filas: 7

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Field_name	Min_value	Max_value	Min_length	Max_length	Empties_or_zeros	Nulls	Avg_value_or_avg_length	Std	Optimal_fieldtype
MedidorEnergia.variablesEstaticas.uniqueKey	1	1	1	1	0	0	1.0000	0.0000	ENUM('1') NOT NULL
MedidorEnergia.variablesEstaticas.facturacion	0	0	1	1	1	0	0.0000	0.0000	ENUM('0') NOT NULL
MedidorEnergia.variablesEstaticas.fechaCorte	1488938349274	1488938349274	13	13	0	0	1488938349274.0000	0.0000	ENUM('1488938349274') NOT NULL
MedidorEnergia.variablesEstaticas.costoKwh	1.8220000267028809	1.8220000267028809	5	5	0	0	1.8220000267028809	0	ENUM('1.8220000267028809') NOT NULL
MedidorEnergia.variablesEstaticas.kwhMax	220	220	3	3	0	0	220	0	ENUM('220') NOT NULL
MedidorEnergia.variablesEstaticas.potMax	1200	1200	4	4	0	0	1200	0	ENUM('1200') NOT NULL
MedidorEnergia.variablesEstaticas.email	ppuren@gmail.com	ppuren@gmail.com	16	16	0	0	16.0000	NULL	ENUM('ppuren@gmail.com') NOT NULL

FIGURA C.3: Análisis de la tabla *variablesEstaticas*

Resultado SQL

Servidor: localhost

Base de datos: MedidorEnergia

Tiempo de generación: 03-06-2019 a las 19:05:02

Generado por: phpMyAdmin 4.2.12deb2+deb8u3 / MySQL 5.5.60-0+deb8u1

consulta SQL: SELECT * FROM `Mediciones` PROCEDURE ANALYSE();

Filas: 10

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Field_name	Min_value	Max_value	Min_length	Max_length	Empties_or_zeros	Nulls	Avg_value_or_avg_length	Std	
MedidorEnergia.Mediciones.estampaDeTiempo	2019-02-04 13:06:45	2019-02-14 22:07:40	19	19	0	0	19.0000	NULL	TIMESTAMP NOT NULL
MedidorEnergia.Mediciones.voltaje	109.12999725341797	221.89999389648438	3	6	0	0	216.83355442814442	7.216257591908498	FLOAT NOT NULL
MedidorEnergia.Mediciones.corriente	0.019999999552965164	24.389999389648438	1	5	7305	0	2.2699076710520205	1.8989959432455197	FLOAT NOT NULL
MedidorEnergia.Mediciones.potenciaActiva	-1080.5	1285.0999755859375	1	7	7580	0	516.290137947702	450.31701721238574	FLOAT NOT NULL
MedidorEnergia.Mediciones.potenciaAparente	4.079999923706055	1280.4100341796875	1	7	201	0	530.7754584517338	437.64054517895465	FLOAT NOT NULL
MedidorEnergia.Mediciones.potenciaReactiva	-61.099998474121094	432.1700134277344	1	6	201	0	52.852780403520256	32.71442754288177	FLOAT NOT NULL
MedidorEnergia.Mediciones.factorDePotencia	-1	1	1	4	0	0	0.9141639496296141	0.18231331360968206	ENUM('-1','0.43999999761: NOT NULL
MedidorEnergia.Mediciones.frecuencia	59.880001068115234	60.20000076293945	2	5	0	0	59.99809883183064	0.03796241081038128	ENUM('59.8800010681152: NOT NULL
MedidorEnergia.Mediciones.totalEnergiaActiva	0.029999999329447746	60.040000915527344	1	5	0	0	16.697989975930557	9.404434459761966	FLOAT NOT NULL
MedidorEnergia.Mediciones.totalEnergiaReactiva	0.009999999776482582	60.040000915527344	1	5	0	0	2.038196260155743	1.3115668039822939	FLOAT NOT NULL

FIGURA C.4: Análisis de la tabla *Mediciones*

Apéndice D

Ejemplo de datos capturados

D.1. Fragmento de captura del 13 de Febrero del 2019

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:00:00	217.6	0.18	30.7	45.78	33.9	0.67	60.02	17.13	2.1
12:00:03	217.4	0.19	30.6	45.69	33.8	0.67	60.02	17.13	2.1
12:00:06	217.5	0.18	30.7	45.69	33.8	0.67	60.02	17.13	2.1
12:00:09	217.5	0.19	30.6	45.69	33.8	0.67	60.02	17.13	2.1
12:00:11	217.4	0.18	30.6	45.63	33.8	0.67	60.02	17.13	2.1
12:00:14	217.4	0.18	30.6	45.76	33.9	0.67	60.02	17.13	2.1
12:00:17	217.4	0.18	30.6	45.64	33.8	0.67	60.02	17.13	2.1
12:00:20	217.4	0.18	30.6	45.57	33.7	0.67	60.02	17.13	2.1
12:00:22	217.5	0.18	30.6	45.76	33.9	0.67	60.02	17.13	2.1
12:00:25	217.6	0.18	30.7	45.83	33.9	0.67	60.02	17.13	2.1
12:00:28	217.5	0.18	30.7	45.77	33.8	0.67	60.02	17.13	2.1
12:00:31	217.5	0.18	30.7	45.89	33.8	0.67	60.02	17.13	2.1
12:00:34	217.6	0.18	30.7	45.67	33.8	0.67	60.02	17.13	2.1
12:00:36	217.5	0.18	30.6	45.81	34	0.67	60.02	17.13	2.1
12:00:39	217.5	0.18	30.6	45.78	34	0.67	60.09	17.13	2.1
12:00:42	217.5	0.18	30.6	45.69	33.8	0.67	60.02	17.13	2.1
12:00:45	217.5	0.18	30.6	45.75	33.9	0.67	59.95	17.13	2.1
12:00:47	217.5	0.18	30.7	45.81	33.8	0.67	60.02	17.13	2.1
12:00:50	217.4	0.18	30.7	45.7	33.9	0.67	60.02	17.13	2.1
12:00:53	217.5	0.19	30.4	45.59	33.8	0.67	60.02	17.13	2.1
12:00:56	217.5	0.19	30.8	45.79	33.8	0.67	60.02	17.13	2.1
12:00:58	217.3	0.18	30.7	45.78	33.9	0.67	60.02	17.13	2.1
12:01:01	217.4	0.18	30.6	45.73	33.9	0.67	60.02	17.13	2.1
12:01:04	217.4	0.19	30.6	45.72	33.8	0.67	60.02	17.13	2.1
12:01:07	217.4	0.18	30.8	45.84	33.9	0.67	60.02	17.13	2.1
12:01:09	217.5	0.18	30.8	45.72	33.8	0.67	60.02	17.13	2.1
12:01:12	217.3	0.19	30.7	45.75	33.9	0.67	60.02	17.13	2.1

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:01:15	217.5	0.18	30.7	45.73	33.8	0.67	60.02	17.13	2.1
12:01:18	217.5	0.18	30.6	45.71	33.8	0.67	60.02	17.13	2.1
12:01:20	217.4	0.18	30.6	45.68	33.8	0.67	60.02	17.13	2.1
12:01:23	217.5	0.18	30.7	45.74	33.8	0.67	60.09	17.13	2.1
12:01:26	217.4	0.18	30.7	45.79	33.9	0.67	60.09	17.13	2.1
12:01:29	217.6	0.19	30.6	45.69	33.9	0.67	60.02	17.13	2.1
12:01:31	217.6	0.18	30.7	45.84	33.9	0.67	60.02	17.13	2.1
12:01:34	217.6	0.18	30.5	45.76	34	0.67	60.02	17.13	2.1
12:01:37	217.4	0.19	30.7	45.74	33.8	0.67	60.02	17.13	2.1
12:01:40	217.5	0.19	30.8	45.85	33.9	0.67	60.02	17.13	2.1
12:01:42	217.4	0.18	30.7	45.76	33.6	0.68	60.02	17.13	2.1
12:01:46	217.1	0.18	30.6	45.62	33.7	0.67	60.02	17.13	2.1
12:01:48	217.1	0.18	30.6	45.58	33.7	0.67	60.02	17.13	2.1
12:01:51	217.2	0.18	30.7	45.69	33.8	0.67	60.02	17.13	2.1
12:01:53	217.2	0.19	30.6	45.58	33.6	0.67	59.95	17.13	2.1
12:01:56	217.2	0.19	30.7	45.63	33.7	0.67	59.95	17.13	2.1
12:01:59	217.2	0.18	30.6	45.61	33.7	0.67	59.95	17.13	2.1
12:02:02	217.3	0.18	30.7	45.67	33.7	0.67	59.95	17.13	2.1
12:02:04	217.3	0.18	30.7	45.61	33.7	0.67	59.95	17.13	2.1
12:02:07	217.3	0.19	30.6	45.68	33.8	0.67	59.95	17.13	2.1
12:02:10	217.4	0.18	30.7	45.78	33.9	0.67	59.95	17.13	2.1
12:02:13	217.4	0.18	30.6	45.68	33.8	0.67	59.95	17.13	2.1
12:02:16	217.4	0.19	30.7	45.75	33.8	0.67	60.02	17.13	2.1
12:02:18	217.4	0.18	30.6	45.62	33.6	0.68	59.95	17.13	2.1
12:02:21	217.3	0.18	30.7	45.78	33.9	0.67	59.95	17.13	2.1
12:02:24	217.4	0.18	30.5	45.63	33.8	0.67	59.95	17.13	2.1
12:02:27	217.3	0.18	30.8	45.76	33.8	0.67	59.95	17.13	2.1
12:02:29	217.3	0.18	30.6	45.64	33.8	0.67	60.02	17.13	2.1
12:02:32	217.3	0.19	30.7	45.76	33.8	0.67	60.02	17.13	2.1
12:02:35	217.4	0.18	30.6	45.61	33.9	0.67	60.02	17.13	2.1
12:02:38	217.4	0.18	30.7	45.78	33.8	0.67	60.02	17.13	2.1
12:02:40	217.4	0.18	30.6	45.68	33.8	0.67	60.02	17.13	2.1
12:02:43	217.4	0.18	30.6	45.73	33.9	0.67	59.95	17.13	2.1
12:02:47	217.5	0.18	30.8	45.82	33.8	0.67	59.95	17.13	2.1
12:02:49	217.4	0.18	30.8	45.82	33.8	0.67	60.02	17.13	2.1
12:02:51	217.5	0.18	30.6	45.65	33.8	0.67	59.95	17.13	2.1
12:02:54	217.5	0.19	30.6	45.68	33.9	0.67	60.02	17.13	2.1
12:02:57	217.5	0.18	30.6	45.79	33.9	0.67	60.02	17.13	2.1
12:03:00	217.5	0.18	30.8	45.7	33.7	0.67	59.95	17.13	2.1
12:03:02	217.5	0.19	30.8	45.83	33.9	0.67	60.02	17.13	2.1
12:03:05	217.5	0.18	30.7	45.69	33.8	0.67	60.02	17.13	2.1

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:03:08	217.5	0.18	30.8	45.95	34	0.67	60.02	17.13	2.1
12:03:11	217.5	0.18	30.7	45.83	33.7	0.67	60.02	17.13	2.1
12:03:14	217.5	2.42	30.7	777.9	393.2	0.86	60.02	17.13	2.1
12:03:16	216.9	4.24	903.7	906.98	76.7	1	60.02	17.13	2.1
12:03:19	216.7	4.24	996.9	1001.33	93.5	1	60.02	17.13	2.1
12:03:22	216.7	4.33	966	970.05	88.4	1	60.02	17.13	2.1
12:03:24	216.6	4.23	990.7	995.13	92.9	1	60.02	17.13	2.1
12:03:27	216.6	3.81	911.3	914.81	79.4	1	60.02	17.13	2.1
12:03:30	216.8	3.87	898.5	901.9	76.6	1	60.02	17.13	2.1
12:03:33	216.8	3.8	891	892.26	75.9	1	60.02	17.13	2.1
12:03:36	216.7	3.87	888.2	891.47	75.6	1	60.02	17.13	2.1
12:03:38	216.7	3.83	886.4	889.68	75.1	1	60.02	17.13	2.1
12:03:41	216.8	3.76	881.1	884.27	74.6	1	60.02	17.14	2.1
12:03:44	216.8	3.64	863.1	866.14	71.9	1	59.95	17.14	2.1
12:03:47	216.8	3.6	848.4	851.22	69.1	1	59.95	17.14	2.1
12:03:49	216.8	3.53	836.8	831.67	65	1	60.09	17.14	2.1
12:03:52	216.7	3.56	819.8	822.33	63.3	1	60.02	17.14	2.1
12:03:55	216.5	3.49	821.6	824.06	62.9	1	60.02	17.14	2.1
12:03:58	216.7	3.6	824.2	826.73	63.4	1	60.02	17.14	2.1
12:04:00	216.6	3.59	827.5	830.02	64	1	60.02	17.14	2.1
12:04:03	216.6	3.65	839.7	842.36	65.8	1	60.02	17.14	2.1
12:04:06	216.6	3.6	850.1	852.88	67.6	1	60.02	17.14	2.1
12:04:09	216.6	3.62	855.4	861.14	68.8	1	60.02	17.14	2.1
12:04:11	216.6	3.68	862.9	865.82	69.7	1	59.95	17.14	2.1
12:04:14	216.7	3.78	865.4	868.28	70	1	60.02	17.14	2.1
12:04:17	216.7	3.81	869.1	872.07	70.8	1	60.02	17.14	2.1
12:04:20	216.7	3.82	873.3	876.29	71.6	1	60.02	17.14	2.1
12:04:22	216.7	3.86	875.2	878.16	71.3	1	60.02	17.14	2.1
12:04:25	216.7	3.7	876.6	879.52	71.1	1	60.02	17.15	2.1
12:04:28	216.7	3.77	877.5	881.18	71.1	1	60.02	17.15	2.1
12:04:31	216.7	3.77	879.4	882.38	71.2	1	60.02	17.15	2.1
12:04:33	216.6	3.71	879.2	882.2	71.5	1	60.02	17.15	2.1
12:04:36	216.6	3.76	878.5	881.53	71.9	1	59.95	17.15	2.1
12:04:39	216.6	3.87	879.3	882.38	72.6	1	59.95	17.15	2.1
12:04:42	216.6	3.86	882.3	885.38	73.3	1	60.02	17.15	2.1
12:04:44	216.7	3.83	883.4	886.29	73.1	1	59.95	17.15	2.1
12:04:47	216.7	3.77	882.9	886.03	73.2	1	59.95	17.15	2.1
12:04:50	216.7	3.85	877.9	880.9	72.4	1	59.95	17.15	2.1
12:04:53	216.7	3.78	876.8	879.81	72	1	60.02	17.15	2.1
12:04:55	216.7	3.77	876.7	879.7	72	1	59.95	17.15	2.1
12:04:58	216.7	3.8	876.5	879.5	71.7	1	60.02	17.15	2.1

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:05:01	216.7	3.77	875.7	878.61	70.6	1	60.02	17.15	2.1
12:05:04	216.6	3.76	875.5	878.84	70.4	1	59.95	17.16	2.1
12:05:06	216.6	3.8	878.9	881.82	70.9	1	60.02	17.16	2.1
12:05:09	216.6	3.73	883	885.95	71.8	1	60.02	17.16	2.1
12:05:12	216.7	3.92	887	890.05	72.9	1	59.95	17.16	2.1
12:05:15	216.7	3.95	891	894.07	73.1	1	59.95	17.16	2.1
12:05:17	216.7	3.83	890.7	893.8	73.4	1	59.95	17.16	2.1
12:05:20	216.8	3.85	894	897.3	74	1	60.02	17.16	2.1
12:05:23	216.8	3.96	894.4	897.48	73.9	1	60.02	17.16	2.1
12:05:26	216.8	3.96	893	896.13	73.8	1	60.02	17.16	2.1
12:05:28	216.9	3.92	893.6	896.74	73.9	1	60.02	17.16	2.1
12:05:31	216.8	3.8	894	897.02	73.1	1	60.02	17.16	2.1
12:05:34	216.8	3.86	894.2	897.26	73.1	1	60.02	17.16	2.1
12:05:37	216.8	3.85	893.1	896.15	73.3	1	60.02	17.16	2.1
12:05:40	216.8	3.87	894.4	898.03	73.6	1	59.95	17.16	2.1
12:05:42	216.8	3.96	893.6	896.53	71.4	1	59.95	17.16	2.1
12:05:45	215.9	3.98	894.4	897.34	71.3	1	59.95	17.17	2.1
12:05:48	215.8	3.85	893.5	896.31	70.6	1	60.02	17.17	2.1
12:05:51	215.8	3.85	894.2	897.04	70.6	1	59.95	17.17	2.1
12:05:53	215.8	3.92	894.3	897.14	70.2	1	59.95	17.17	2.1
12:05:56	215.9	3.82	893.6	896.51	69.5	1	60.02	17.17	2.1
12:05:59	216	3.8	894.8	897.62	70.3	1	59.95	17.17	2.1
12:06:02	215.9	3.88	893.9	896.73	69.9	1	59.95	17.17	2.1
12:06:04	216	3.88	892.8	895.64	69.9	1	60.02	17.17	2.1
12:06:07	216	3.85	894.2	896.99	70.6	1	59.95	17.17	2.1
12:06:10	215.6	3.83	894.3	896.91	68.1	1	59.95	17.17	2.1
12:06:13	215.3	3.91	895.5	898.21	68.2	1	60.02	17.17	2.1
12:06:15	215.4	3.87	894.9	897.56	68.4	1	59.95	17.17	2.1
12:06:18	215.4	3.95	894.5	897.09	67.5	1	59.95	17.17	2.1
12:06:21	215.4	3.91	895.3	897.98	68.4	1	60.02	17.17	2.1
12:06:24	215.5	3.82	897.2	899.85	68.3	1	60.02	17.17	2.1
12:06:26	215.5	3.86	897.1	899.83	68.9	1	59.95	17.17	2.1
12:06:29	215.5	3.89	896.1	898.84	69.5	1	60.02	17.18	2.1
12:06:32	215.5	3.86	895.9	898.39	69.1	1	60.02	17.18	2.1
12:06:35	215.5	3.91	894.9	897.68	69.3	1	60.02	17.18	2.1
12:06:37	215.5	3.83	898.1	900.89	69.6	1	59.95	17.18	2.1
12:06:40	215.6	3.84	897.1	899.89	69.9	1	60.02	17.18	2.1
12:06:43	215.6	3.91	897.2	900.02	70.3	1	60.02	17.18	2.1
12:06:47	215.7	3.97	899.7	902.52	70.3	1	59.95	17.18	2.1
12:06:48	215.7	3.9	900.9	903.71	70.4	1	59.95	17.18	2.1
12:06:51	215.6	3.9	899.2	902.04	70.3	1	60.02	17.18	2.1

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:06:54	215.6	3.9	897	899.76	69.3	1	59.95	17.18	2.1
12:06:57	215.6	3.91	898.9	901.72	70.2	1	60.02	17.18	2.1
12:06:59	215.6	3.94	899.9	902.69	70.2	1	59.95	17.18	2.1
12:07:02	215.6	3.92	900.8	903.64	70.2	1	59.95	17.18	2.1
12:07:05	215.5	3.99	899.1	901.85	70.3	1	59.95	17.18	2.1
12:07:08	215.6	4.03	903	904.1	70.8	1	60.02	17.18	2.1
12:07:11	215.6	3.88	904.1	906.91	70.6	1	60.02	17.19	2.1
12:07:13	215.6	4.05	905.1	907.92	70.9	1	60.02	17.19	2.1
12:07:16	215.6	3.94	906	908.85	71.3	1	59.95	17.19	2.1
12:07:19	215.6	3.99	906	908.83	70.8	1	60.02	17.19	2.1
12:07:22	215.6	3.88	905.9	908.75	70.4	1	60.02	17.19	2.1
12:07:24	215.5	3.99	907.2	911.08	70.9	1	59.95	17.19	2.1
12:07:27	215.6	4.02	907.3	910.14	71.4	1	60.02	17.19	2.1
12:07:30	215.8	4.01	906.4	909.43	73.7	1	60.02	17.19	2.11
12:07:33	216.2	3.95	906	908.95	72.7	1	60.02	17.19	2.11
12:07:35	216.1	3.85	905.8	908.79	73.3	1	60.02	17.19	2.11
12:07:38	216.2	3.91	908.2	911.26	73.7	1	60.02	17.19	2.11
12:07:41	216.2	3.93	910.1	913.2	74.1	1	60.02	17.19	2.11
12:07:44	216.3	3.91	907.3	910.36	73.6	1	59.95	17.19	2.11
12:07:47	216.3	4.02	906.8	909.89	74	1	59.95	17.19	2.11
12:07:49	216.3	3.97	907.3	910.44	74.3	1	59.95	17.2	2.11
12:07:52	215.7	3.88	906.8	909.72	72.2	1	59.95	17.2	2.11
12:07:55	215.9	3.9	906.8	909.78	73.4	1	59.95	17.2	2.11
12:07:57	216	3.85	906.5	909.58	73.9	1	60.02	17.2	2.11
12:08:00	215.9	4.01	906.5	909.39	73.9	1	59.95	17.2	2.11
12:08:03	215.9	4	902.2	905.16	72.4	1	60.02	17.2	2.11
12:08:06	216	3.85	897.5	900.44	72.5	1	59.95	17.2	2.11
12:08:08	216	3.91	894.3	897.16	71.3	1	60.02	17.2	2.11
12:08:11	215.9	3.9	892.7	895.67	71.7	1	59.95	17.2	2.11
12:08:14	215.9	3.91	891.5	894.39	70.9	1	59.95	17.2	2.11
12:08:17	215.9	3.77	889.8	892.67	70.4	1	59.95	17.2	2.11
12:08:19	215.9	3.78	886.1	888.74	69.9	1	60.02	17.2	2.11
12:08:22	215.9	3.86	885.9	888.68	69.9	1	59.95	17.2	2.11
12:08:25	215.9	3.76	885.2	887.91	69.2	1	60.02	17.2	2.11
12:08:28	215.9	3.81	885.4	888.14	69	1	59.95	17.2	2.11
12:08:31	215.8	3.83	886.3	888.98	68.5	1	60.02	17.21	2.11
12:08:33	215.8	3.8	887.3	890.02	68.8	1	60.02	17.21	2.11
12:08:36	215.7	3.8	890.4	895.29	69.4	1	60.02	17.21	2.11
12:08:39	215.6	3.9	893.8	896.59	70.4	1	60.02	17.21	2.11
12:08:42	215.6	3.79	897	899.81	70.9	1	60.02	17.21	2.11
12:08:44	216.7	3.9	900.9	904.05	74.7	1	60.02	17.21	2.11

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:08:47	216.7	3.98	900.6	903.71	73.7	1	60.02	17.21	2.11
12:08:50	216.6	3.88	898.1	901.16	73.1	1	60.02	17.21	2.11
12:08:53	216.8	3.92	896	899.04	72.9	1	60.02	17.21	2.11
12:08:55	216.7	3.95	893.7	895.88	72.9	1	60.02	17.21	2.11
12:08:58	216.8	3.93	891.6	894.52	72.1	1	60.02	17.21	2.11
12:09:01	216.8	3.89	891.7	894.7	72.1	1	60.02	17.21	2.11
12:09:04	216.9	3.76	891.2	894.22	72.2	1	60.02	17.21	2.11
12:09:06	216.9	3.89	889.6	892.58	71.7	1	60.02	17.22	2.11
12:09:09	216.8	3.83	888.2	891.18	71.6	1	60.02	17.22	2.11
12:09:12	217	3.84	887.1	890.21	72.1	1	60.09	17.22	2.11
12:09:15	217.1	3.86	884.8	887.84	72.3	1	60.02	17.22	2.11
12:09:20	217.2	3.9	884.8	887.81	72.5	1	60.02	17.22	2.11
12:09:23	217.2	3.82	883.5	886.5	72.6	1	60.02	17.22	2.11
12:09:26	217.2	3.87	888.4	890.12	73.3	1	60.02	17.22	2.11
12:09:29	217.2	3.84	889.7	892.76	73.6	1	60.02	17.22	2.11
12:09:32	217.2	3.85	890.6	893.67	73.5	1	60.02	17.22	2.11
12:09:34	217.4	0.18	685	687.94	62.4	1	60.02	17.22	2.11
12:09:37	217.8	0.18	31	46.11	34	0.67	59.95	17.22	2.11
12:09:40	217.8	0.18	31	46.13	34	0.67	60.02	17.22	2.11
12:09:43	217.8	0.18	30.9	45.96	33.9	0.67	59.95	17.22	2.11
12:09:47	217.9	0.19	30.7	45.82	33.9	0.67	60.02	17.22	2.11
12:09:48	217.7	0.18	30.6	45.8	33.9	0.67	60.02	17.22	2.11
12:09:51	217.8	0.18	30.6	45.83	34	0.67	60.02	17.22	2.11
12:09:56	217.8	0.18	30.6	45.72	33.9	0.67	60.02	17.22	2.11
12:09:59	217.9	0.18	30.7	45.81	34	0.67	60.02	17.22	2.11
12:10:02	217.9	0.19	30.6	45.79	34	0.67	59.95	17.22	2.11
12:10:05	217.9	0.18	30.5	45.76	34	0.67	60.02	17.22	2.11
12:10:07	217.9	0.19	30.6	45.76	33.9	0.67	60.02	17.22	2.11
12:10:10	217.8	0.18	30.6	45.78	33.9	0.67	59.95	17.22	2.11
12:10:13	217.8	0.18	30.7	45.85	33.9	0.67	60.02	17.22	2.11
12:10:16	217.9	0.19	30.7	45.68	33.7	0.67	60.02	17.22	2.11
12:10:18	217.9	0.18	30.7	45.75	33.9	0.67	60.02	17.22	2.11
12:10:21	217.9	0.18	30.7	45.81	33.9	0.67	60.02	17.22	2.11
12:10:24	218	0.18	30.7	45.82	33.9	0.67	59.95	17.22	2.11
12:10:27	218.1	0.19	30.7	45.94	34	0.67	60.02	17.22	2.11
12:10:29	218	0.19	30.7	45.92	34.1	0.67	60.02	17.22	2.11
12:10:32	217.9	0.18	30.7	45.92	34	0.67	60.02	17.22	2.11
12:10:35	218	0.18	30.6	45.92	34	0.67	59.95	17.22	2.11
12:10:38	217.9	0.18	30.7	45.83	34.1	0.67	60.02	17.22	2.11
12:10:40	217.9	0.19	30.7	45.93	34.1	0.67	60.02	17.22	2.11
12:10:43	217.9	0.19	30.8	45.97	34.1	0.67	60.02	17.22	2.11

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:10:47	217.9	0.19	30.7	45.82	33.9	0.67	59.95	17.22	2.11
12:10:49	217.9	0.19	30.9	46.04	34.1	0.67	60.02	17.22	2.11
12:10:51	217.9	0.18	30.7	45.88	34	0.67	59.95	17.22	2.11
12:10:54	217.9	0.19	30.7	45.86	34	0.67	59.95	17.22	2.11
12:10:57	217.9	0.18	30.7	45.94	34	0.67	59.95	17.22	2.11
12:11:00	218	0.18	30.8	45.99	34.1	0.67	60.02	17.22	2.11
12:11:02	217.9	0.19	30.8	46.01	34.1	0.67	59.95	17.22	2.11
12:11:05	217.9	0.18	30.7	45.8	33.9	0.67	60.02	17.22	2.11
12:11:08	217.9	0.18	30.6	45.85	34	0.67	60.02	17.22	2.11
12:11:11	217.8	0.18	30.7	45.84	33.9	0.67	59.95	17.22	2.11
12:11:14	217.8	0.18	30.6	45.86	33.9	0.67	59.95	17.22	2.11
12:11:16	217.8	0.18	30.6	45.67	33.8	0.67	60.02	17.22	2.11
12:11:19	217.8	0.18	30.7	45.85	34	0.67	59.95	17.22	2.11
12:11:22	217.8	0.18	30.7	45.85	34	0.67	60.02	17.22	2.11
12:11:25	217.8	0.18	30.6	45.78	33.9	0.67	60.02	17.22	2.11
12:11:27	217.8	0.18	30.6	45.71	33.9	0.67	59.95	17.22	2.11
12:11:30	217.8	0.19	30.8	45.93	34	0.67	59.95	17.22	2.11
12:11:33	217.8	0.18	30.7	45.87	34	0.67	59.95	17.22	2.11
12:11:36	217.8	0.18	30.6	45.91	34.1	0.67	60.02	17.22	2.11
12:11:38	217.6	0.18	30.8	45.84	33.8	0.67	60.02	17.22	2.11
12:11:41	217.7	0.18	30.7	45.84	33.9	0.67	59.95	17.22	2.11
12:11:44	217.7	0.18	30.7	45.8	33.9	0.67	60.02	17.22	2.11
12:11:47	217.7	0.18	30.7	45.78	34	0.67	60.02	17.22	2.11
12:11:49	217.7	0.18	30.7	45.78	33.9	0.67	60.02	17.22	2.11
12:11:52	217.7	0.18	30.7	45.76	33.8	0.67	60.02	17.22	2.11
12:11:55	217.6	0.18	30.8	45.92	34	0.67	59.95	17.22	2.11
12:11:58	217.6	0.18	30.8	45.87	33.9	0.67	60.02	17.22	2.11
12:12:00	217.7	0.18	30.9	45.88	33.8	0.67	60.02	17.22	2.11
12:12:03	217.6	0.18	30.7	45.84	33.8	0.67	60.02	17.22	2.11
12:12:06	217.6	0.18	30.7	45.8	33.9	0.67	60.02	17.22	2.11
12:12:09	217.6	0.19	30.8	45.8	33.8	0.67	60.02	17.22	2.11
12:12:11	217.5	0.18	30.7	45.9	34	0.67	59.95	17.22	2.11
12:12:14	217.7	0.18	30.6	45.72	33.9	0.67	59.95	17.22	2.11
12:12:17	217.6	0.19	30.6	45.72	33.8	0.67	60.02	17.22	2.11
12:12:20	217.5	0.18	30.7	45.72	33.8	0.67	60.02	17.22	2.11
12:12:22	217.5	0.18	30.8	45.89	33.9	0.67	60.02	17.22	2.11
12:12:25	217.5	0.18	30.7	45.74	33.8	0.67	59.95	17.22	2.11
12:12:28	217.5	0.18	30.8	45.87	33.9	0.67	59.95	17.22	2.11
12:12:31	217.5	0.19	30.8	45.86	33.9	0.67	59.95	17.22	2.11
12:12:34	217.6	0.18	30.7	45.74	33.8	0.67	59.95	17.22	2.11
12:12:36	217.5	0.18	30.7	45.77	33.8	0.67	59.95	17.22	2.11

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:12:39	217.6	0.19	30.7	45.67	33.7	0.67	59.95	17.22	2.11
12:12:42	217.5	0.19	30.7	45.86	33.9	0.67	59.95	17.22	2.11
12:12:45	217.6	0.19	30.7	45.71	33.7	0.67	60.02	17.22	2.11
12:12:47	217.7	0.18	30.8	45.94	34	0.67	59.95	17.22	2.11
12:12:50	217.7	0.18	30.8	45.9	34	0.67	59.95	17.22	2.11
12:12:53	217.7	0.18	30.7	45.9	34	0.67	59.95	17.22	2.11
12:12:56	217.7	0.19	30.8	45.92	34	0.67	60.02	17.22	2.11
12:12:58	217.7	0.19	30.7	45.77	33.9	0.67	59.95	17.22	2.11
12:13:01	217.7	0.18	30.7	45.77	33.8	0.67	60.02	17.22	2.11
12:13:04	217.6	0.18	30.8	45.78	33.8	0.67	60.02	17.22	2.11
12:13:07	217.6	0.19	30.7	45.84	33.9	0.67	60.02	17.22	2.11
12:13:09	217.5	0.18	30.7	45.83	33.9	0.67	60.02	17.22	2.11
12:13:12	217.4	0.19	30.7	45.82	33.9	0.67	59.95	17.22	2.11
12:13:15	217.4	0.18	30.7	45.77	33.9	0.67	59.95	17.22	2.11
12:13:18	217.4	0.18	30.7	45.69	33.9	0.67	60.02	17.22	2.11
12:13:20	217.4	0.18	30.6	45.76	33.7	0.67	60.02	17.22	2.11
12:13:23	217.5	0.18	30.8	45.82	33.9	0.67	60.02	17.22	2.11
12:13:26	217.4	0.18	30.8	45.84	33.9	0.67	60.02	17.22	2.11
12:13:29	217.5	0.18	30.8	45.76	33.8	0.67	59.95	17.22	2.11
12:13:31	217.4	0.18	30.7	45.85	33.9	0.67	59.95	17.22	2.11
12:13:34	217.4	0.19	30.8	45.85	33.9	0.67	59.95	17.22	2.11
12:13:37	217.4	0.18	30.7	45.8	33.8	0.67	59.95	17.22	2.11
12:13:40	217.5	0.18	30.8	45.81	33.8	0.67	60.02	17.22	2.11
12:13:42	217.6	0.18	30.8	45.89	33.9	0.67	60.02	17.22	2.11
12:13:45	217.5	0.19	30.7	45.78	33.8	0.67	60.02	17.22	2.11
12:13:48	217.5	0.19	30.8	45.85	33.9	0.67	60.02	17.22	2.11
12:13:51	217.5	0.18	30.7	45.81	33.9	0.67	59.95	17.22	2.11
12:13:53	217.1	0.18	30.7	45.68	33.7	0.67	60.02	17.22	2.11
12:13:56	217.4	0.18	30.7	45.65	33.7	0.67	59.95	17.22	2.11
12:13:59	217.4	0.18	30.6	45.7	33.8	0.67	60.02	17.22	2.11
12:14:02	217.4	0.18	30.8	45.81	33.8	0.67	59.95	17.22	2.11
12:14:04	217.4	0.18	30.8	45.8	33.8	0.67	60.02	17.22	2.11
12:14:07	217.3	0.18	30.8	45.78	33.8	0.67	60.02	17.22	2.11
12:14:10	217.4	0.18	30.8	45.85	33.8	0.67	60.02	17.22	2.11
12:14:13	217.3	0.18	30.7	45.77	33.9	0.67	60.02	17.22	2.11
12:14:15	217.3	0.18	30.8	45.87	33.9	0.67	60.02	17.22	2.11
12:14:18	217.3	0.18	30.8	45.77	33.8	0.67	60.02	17.22	2.11
12:14:22	217.3	0.18	30.7	45.75	33.8	0.67	60.02	17.22	2.11
12:14:24	217.4	0.18	30.8	45.86	33.9	0.67	60.02	17.22	2.11
12:14:26	217.4	0.18	30.7	45.77	33.8	0.67	60.02	17.22	2.11
12:14:29	217.4	0.18	30.8	45.79	33.9	0.67	60.02	17.22	2.11

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:14:32	217.5	0.18	30.7	45.8	33.9	0.67	59.95	17.22	2.11
12:14:35	217.4	0.19	30.8	45.83	33.9	0.67	60.02	17.22	2.11
12:14:37	217.5	3.07	990.8	1064.18	388.1	0.93	60.02	17.23	2.11
12:14:40	216.6	4.3	971.6	975.88	90.4	1	59.95	17.23	2.11
12:14:43	216.7	4.11	988	992.56	94.7	1	59.95	17.23	2.11
12:14:47	216.7	4.34	1005.3	1010.18	98.2	1	60.02	17.23	2.11
12:14:48	216.7	3.94	961.7	965.97	82.4	1	59.95	17.23	2.11
12:14:54	216.8	3.76	880.5	883.92	77.1	1	59.95	17.23	2.11
12:14:57	216.7	3.81	879.4	882.85	76.8	1	59.95	17.23	2.11
12:14:59	216.8	3.87	882.4	885.84	77.3	1	59.95	17.23	2.11
12:15:02	216.8	3.78	879.4	882.72	76.3	1	60.02	17.23	2.11
12:15:05	216.8	3.74	863.5	866.75	71.9	1	59.95	17.23	2.11
12:15:08	216.6	3.55	846.6	842.1	69.7	1	59.95	17.23	2.11
12:15:10	216.6	3.57	826.1	828.9	67.1	1	59.95	17.23	2.11
12:15:13	216.6	3.47	810	812.61	64.7	1	60.02	17.23	2.11
12:15:16	216.8	3.49	807.6	810.2	63.9	1	60.02	17.23	2.11
12:15:19	216.7	3.57	808.5	810.98	63	1	60.02	17.24	2.11
12:15:21	216.7	3.55	809.4	811.91	62.6	1	60.02	17.24	2.11
12:15:24	216.7	3.54	813.4	815.8	62.9	1	60.02	17.24	2.11
12:15:27	216.6	3.63	828	838.24	65.8	1	60.02	17.24	2.11
12:15:30	216.6	3.61	840.4	843.04	66.3	1	60.02	17.24	2.11
12:15:32	216.6	3.66	845.2	847.96	67.6	1	60.02	17.24	2.11
12:15:35	216.6	3.66	849.5	852.29	68.4	1	59.95	17.24	2.11
12:15:38	216.5	3.67	855.9	858.71	68.3	1	60.02	17.24	2.11
12:15:41	216.6	3.67	861.3	864.14	69.4	1	60.02	17.24	2.11
12:15:43	216.5	3.7	864.6	867.5	69.7	1	60.02	17.24	2.11
12:15:47	216.6	3.82	866.7	870.27	70.5	1	59.95	17.24	2.11
12:15:49	216.6	3.84	867.7	870.57	70.2	1	60.02	17.24	2.11
12:15:52	216.5	3.85	870	872.93	70.5	1	60.02	17.24	2.11
12:15:54	216.6	3.82	873.7	876.68	71.9	1	59.95	17.24	2.11
12:15:57	216.6	3.79	875.4	878.4	72.1	1	60.02	17.24	2.11
12:16:00	216.6	3.72	876.3	879.33	72.5	1	60.02	17.25	2.11
12:16:03	216.6	3.79	877.2	880.24	71.7	1	59.95	17.25	2.11
12:16:05	216.6	3.79	877.5	880.44	71	1	60.02	17.25	2.11
12:16:08	216.6	3.75	878.2	881.1	70.7	1	60.02	17.25	2.11
12:16:11	216.6	3.71	879.3	882.14	70.4	1	60.02	17.25	2.11
12:16:14	216.7	3.71	875.9	878.68	69.5	1	60.02	17.25	2.11
12:16:16	216.6	3.86	874	876.83	69.6	1	60.02	17.25	2.11
12:16:19	216.5	3.78	873.7	876.57	70.3	1	60.02	17.25	2.11
12:16:22	216.6	3.85	872.1	873.48	70.3	1	60.02	17.25	2.11
12:16:25	216.7	3.69	867.8	870.67	69.7	1	60.02	17.25	2.11

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:16:28	216.7	3.69	865.5	868.28	68.6	1	60.02	17.25	2.11
12:16:30	216.7	3.73	864.1	866.81	68	1	60.02	17.25	2.11
12:16:33	216.6	3.68	864	866.74	67.7	1	60.02	17.25	2.11
12:16:36	216.6	3.7	861.6	864.36	67.9	1	60.02	17.25	2.11
12:16:39	216.6	3.84	863.5	866.24	68.4	1	60.02	17.25	2.11
12:16:41	216.9	3.77	866.2	869.24	69.7	1	60.02	17.25	2.11
12:16:44	216.9	3.84	866.7	869.57	69.3	1	60.02	17.25	2.11
12:16:47	217	3.77	865.9	868.73	69	1	60.02	17.26	2.11
12:16:49	216.9	3.82	865.6	868.41	69	1	60.02	17.26	2.11
12:16:52	216.8	3.66	865.3	868.08	68.9	1	59.95	17.26	2.11
12:16:55	216.7	3.72	865.4	868.2	68.5	1	60.02	17.26	2.11
12:16:58	216.8	3.68	867	869.77	68.2	1	60.02	17.26	2.11
12:17:01	216.8	3.67	867.4	870.04	67.5	1	60.02	17.26	2.11
12:17:03	216.7	3.7	867.3	869.93	67.4	1	60.02	17.26	2.11
12:17:06	216.6	3.76	869.2	871.89	68.1	1	60.02	17.26	2.11
12:17:09	216.6	3.75	871.8	874.57	69.3	1	59.95	17.26	2.11
12:17:12	216.6	3.73	872	874.81	69.8	1	59.95	17.26	2.11
12:17:14	216.7	3.78	870.8	873.69	70	1	59.95	17.26	2.11
12:17:17	216.7	3.75	869.7	872.23	69.7	1	59.95	17.26	2.11
12:17:20	216.7	3.71	869.7	872.55	69.3	1	60.02	17.26	2.11
12:17:23	216.7	3.78	871.1	873.93	69	1	60.02	17.26	2.11
12:17:25	216.8	3.77	874.5	877.24	68.7	1	60.02	17.26	2.11
12:17:28	216.7	3.84	873	875.74	68.1	1	60.02	17.27	2.11
12:17:31	216.7	3.78	874.6	877.31	68.3	1	60.02	17.27	2.11
12:17:34	216.7	3.85	876.4	879.22	69.3	1	60.02	17.27	2.11
12:17:36	216.7	3.88	875.4	877.05	69.4	1	59.95	17.27	2.11
12:17:39	216.7	3.81	873.4	876.19	69.4	1	60.02	17.27	2.11
12:17:42	216.7	3.7	872.2	875.04	69.3	1	59.95	17.27	2.11
12:17:45	216.7	3.75	870.8	873.6	69.6	1	59.95	17.27	2.11
12:17:47	216.7	3.74	870.3	873.1	69.3	1	60.02	17.27	2.11
12:17:50	216.8	3.81	869.7	872.53	68.5	1	60.02	17.27	2.11
12:17:53	216.8	3.74	867.6	870.7	67.9	1	59.95	17.27	2.11
12:17:56	216.7	3.71	870.5	873.28	68.9	1	60.02	17.27	2.11
12:17:58	216.8	3.73	871.8	874.57	69.1	1	59.95	17.27	2.11
12:18:01	216.6	3.77	871.8	874.63	69.8	1	59.95	17.27	2.11
12:18:04	216.8	3.77	871.9	874.85	70.9	1	59.95	17.27	2.11
12:18:07	216.8	3.72	873.5	876.48	71.5	1	59.95	17.27	2.11
12:18:09	216.8	3.77	876.2	879.22	71.4	1	59.95	17.28	2.11
12:18:12	216.9	3.75	875.3	878.82	71.3	1	60.02	17.28	2.11
12:18:15	216.8	3.81	876.6	879.58	71.2	1	59.95	17.28	2.11
12:18:18	216.8	3.72	876.6	879.58	71.1	1	59.95	17.28	2.11

Hora	Volts	Corr	potAct	potAp	potRct	FP	Frec	totEnAct	totEnRct
12:18:20	216	3.75	875	877.71	68.2	1	59.95	17.28	2.11
12:18:23	216	3.75	873.3	875.93	67.4	1	59.95	17.28	2.11
12:18:26	215.7	3.77	872.3	874.91	66.7	1	59.95	17.28	2.11
12:18:29	215.8	3.81	873.6	876.24	67.5	1	59.95	17.28	2.12
12:18:31	215.7	3.76	872.5	875.01	66.9	1	59.95	17.28	2.12
12:18:34	215.7	3.74	871	873.62	67.1	1	59.95	17.28	2.12
12:18:37	215.9	3.88	873.8	876.48	68	1	59.95	17.28	2.12
12:18:40	215.9	3.76	872	874.65	67.5	1	59.95	17.28	2.12
12:18:42	215.9	3.78	870.7	873.36	66.8	1	59.95	17.28	2.12
12:18:45	215.8	3.67	867.6	870.15	65.6	1	60.02	17.28	2.12
12:18:48	215.8	3.73	868.1	870.71	65.8	1	60.02	17.28	2.12
12:18:51	215.8	3.69	864.7	867.31	66	1	59.95	17.29	2.12
12:18:53	215.8	3.78	865.2	867.75	66.2	1	59.95	17.29	2.12
12:18:56	215.8	3.73	865.9	868.46	66.2	1	59.95	17.29	2.12
12:18:59	215.8	3.8	865.2	867.77	66	1	59.95	17.29	2.12
12:19:02	215.9	3.77	865.9	868.53	66.4	1	59.95	17.29	2.12
12:19:04	215.9	3.7	863	865.63	65.9	1	59.95	17.29	2.12
12:19:07	215.9	3.73	865.1	840.08	62.7	1	59.95	17.29	2.12
12:19:10	216.4	0.19	31	45.63	33.4	0.68	59.95	17.29	2.12
12:19:13	216.3	0.18	30.9	45.43	33.2	0.68	60.02	17.29	2.12
12:19:15	215.9	0.19	30.7	45.36	33.3	0.68	59.95	17.29	2.12
12:19:18	216.1	0.18	30.7	45.33	33.3	0.68	59.95	17.29	2.12
12:19:21	216.1	0.18	30.6	45.4	33.4	0.68	59.95	17.29	2.12
12:19:24	216.2	0.18	30.4	45.18	33.3	0.68	59.95	17.29	2.12
12:19:26	216.2	0.18	30.6	45.28	33.4	0.67	59.95	17.29	2.12
12:19:29	216.1	0.18	30.5	45.21	33.3	0.68	59.95	17.29	2.12
12:19:32	216.1	0.18	30.6	45.3	33.3	0.68	59.95	17.29	2.12
12:19:35	216.1	0.19	30.6	45.38	33.5	0.67	59.95	17.29	2.12
12:19:37	216.1	0.18	30.5	45.16	33.2	0.68	59.95	17.29	2.12
12:19:40	216	0.18	30.6	45.2	33.2	0.68	59.95	17.29	2.12
12:19:43	216	0.19	30.6	45.27	33.3	0.68	59.95	17.29	2.12
12:19:47	216.1	0.19	30.7	45.38	33.4	0.68	59.95	17.29	2.12
12:19:48	216	0.18	30.5	45.18	33.2	0.68	59.95	17.29	2.12
12:19:51	216	0.18	30.7	45.29	33.2	0.68	59.95	17.29	2.12
12:19:54	216	0.18	30.5	45.28	33.3	0.68	59.95	17.29	2.12
12:19:57	215.9	0.19	30.5	45.13	33.1	0.68	59.95	17.29	2.12
12:19:59	216	0.18	30.5	45.26	33.1	0.68	59.95	17.29	2.12
12:20:02	216	0.18	30.5	45.3	33.3	0.68	59.95	17.29	2.12

Bibliografía

SENER, Secretaría de Energía (2017).

Estadísticas del Sector Eléctrico.

URL: <http://egob2.energia.gob.mx/portal/electricidad.html> (visitado 22-10-2017)

(Vid. págs. 1, 2).

Statista (2018).

IoT: Number of Connected Devices Worldwide 2012-2025.

URL: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/> (visitado 12-08-2018)

(Vid. pág. 5).

Allen-Bradley (2017).

Monitoreo de Energía.

URL: <https://ab.rockwellautomation.com/es/Energy-Monitoring> (visitado 24-10-2017)

(Vid. pág. 11).

Schneider (2017).

Schneider Electric | Sistema de Monitoreo y Control de la Energía.

URL: <https://www.schneider-electric.com.mx/es/product-category/4100-sistema-de-monitoreo-y-control-de-la-energ%C3%ADa/> (visitado 25-10-2017)

(Vid. pág. 11).

Smappee (2018).

Smappee - Monitor de Énergía - App per iOS y Android.

URL: https://www.smappee.com/eu_es/home (visitado 12-08-2018)

(Vid. pág. 12).

Curb (2018).

Curb-Home Energy Monitoring.

URL: <https://energycurb.com/> (visitado 12-08-2018)

(Vid. pág. 13).

Amazon Web Services (2018).

AWS | Cloud Computing - Servicios de informática en la nube.

URL: <https://aws.amazon.com/es/> (visitado 12-08-2018)

(Vid. pág. 13).

Microsoft Azure (2018).

Microsoft Azure.

URL: <https://azure.microsoft.com/es-mx/free/search/> (visitado 12-08-2018)

(Vid. pág. 14).

Google (2018).

Google Cloud, Computación en la nube, servicios de alojamiento y APIs.

URL: <https://cloud.google.com/gcp/?hl=es> (visitado 12-08-2018)

(Vid. pág. 14).

Ubidots IoT Platform (2018).

Ubidots | IoT Platform | Internet of Things |.

URL: <https://ubidots.com/> (visitado 12-08-2018)

(Vid. pág. 14).

ThingSpeak IoT (2018).

IoT Analytics - ThingSpeak Internet of Things.

URL: <https://thingspeak.com/> (visitado 12-08-2018)

(Vid. pág. 14).

Mdhaffar, A. y col. (2017).

«IoT-Based Health Monitoring via LoRaWAN».

En: IEEE EUROCON 2017 -17th International Conference on Smart Technologies.

ISSN:

Págs. 519-524.

DOI: [10.1109/EUROCON.2017.8011165](https://doi.org/10.1109/EUROCON.2017.8011165)

(Vid. pág. 15).

Trancă, D. y col. (2017).

«ZigBee Based Wireless Modbus Aggregator for Intelligent Industrial Facilities».

En: 2017 25th Telecommunication Forum (TELFOR).

ISSN:

Págs. 1-4.

DOI: [10.1109/TELFOR.2017.8249409](https://doi.org/10.1109/TELFOR.2017.8249409)

(Vid. pág. 15).

Pocero, Lidia y col. (2017).

«Open Source IoT Meter Devices for Smart and Energy-Efficient School Buildings».

En: HardwareX 1, págs. 54-67.

ISSN: 24680672.

DOI: [10.1016/j.ohx.2017.02.002](https://doi.org/10.1016/j.ohx.2017.02.002).

URL: <https://linkinghub.elsevier.com/retrieve/pii/S2468067216300293>
(visitado 09-08-2018)

(Vid. pág. 15).

Barman, B. K. y col. (2018).

«IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid».
En: Towards Smart Technology (ICEPE).

ISSN:

Págs. 1-5.

DOI: [10.1109/EPETSG.2018.8658501](https://doi.org/10.1109/EPETSG.2018.8658501)

(Vid. pág. 15).

Jayanthi, R, S.T. Rama y Maduravoyal, Chennai (2017).

«IOT Based Smart Energy Tracking System».

En: International Journal of MC Square Scientific Research 9.1, págs. 98-108.

ISSN: 09750932.

DOI: [10.20894/IJMSR.117.009.001.012](https://doi.org/10.20894/IJMSR.117.009.001.012).

URL: <http://ijmsr.org/admin/selected/IJMSR2017-91-012IJMSR2017-91-012.pdf> (visitado 03-05-2018)

(Vid. pág. 16).

Shopov, M. P. (2017).

«IoT Gateway for Smart Metering in Electrical Power Systems - Software Architecture».

En: Electronics and Microelectronics (MIPRO).

ISSN:

Págs. 974-978.

DOI: [10.23919/MIPRO.2017.7973565](https://doi.org/10.23919/MIPRO.2017.7973565)

(Vid. pág. 16).

Hansen, Hans, Mikkel Skovsmose Tewes y Nicolai Foldager (2018).

«A Home Energy Management System with Focus on Energy Optimization».

En: pág. 171

(Vid. pág. 16).

Ashton, Kevin (2009).

That 'Internet of Things' Thing.

URL: [http://www.rfidjournal.com/articles/view?4986.%20\[Accessed%20%2005%202013\]](http://www.rfidjournal.com/articles/view?4986.%20[Accessed%20%2005%202013]) (visitado 12-08-2018)

(Vid. pág. 17).

Overview of the Internet of Things (2012).

Recommendation ITU-T Y.2060.

URL: <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060>

(visitado 12-08-2018)

(Vid. págs. 17-20).

Patel, Keyur K, Sunil M Patel y PG Scholar (2016).

«Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges».

En: pág. 10

(Vid. pág. 21).

LinkLabs (2016).

A Comprehensive Look at Low Power, Wide Area Networks.

URL: <https://www.link-labs.com/lpwan>

(Vid. pág. 22).

Mekki, Kais y col. (2018).

«A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment».

En: ICT Express.

ISSN: 2405-9595.

DOI: [10.1016/j.icte.2017.12.005](https://doi.org/10.1016/j.icte.2017.12.005).

URL: <http://www.sciencedirect.com/science/article/pii/S2405959517302953>

(visitado 10-09-2018)

(Vid. págs. 23, 24).

Molloy, Derek (2016).

Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux.

Indianapolis, IN: Wiley.

693 págs.

ISBN: 978-1-119-18868-1

(Vid. pág. 27).

Maksimović, Mirjana y col. (2015).

«Raspberry Pi as Internet of Things Hardware: Performances and Constraints».

En: pág. 7

(Vid. pág. 28).

Raspberry Pi Zero W (2018).

Raspberry Pi Zero W.

URL: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/> (visitado 10-09-2018)

(Vid. pág. 28).

E32-915T20D (2019).

URL: <http://www.ebyte.com/en/product-view-news.aspx?id=131> (visitado 09-06-2019)

(Vid. pág. 36).

Monk, Simon (2014).

Raspberry Pi Cookbook.

Beijing: O'Reilly.

393 págs.

ISBN: 978-1-4493-6522-6

(Vid. pág. 36).

Raspberry Pi Serial (UART) Tutorial (2019).

URL: <https://www.teachmemicro.com/raspberry-pi-serial-uart-tutorial/> (visitado 12-01-2019)

(Vid. pág. 38).

Shovic, John C. (2016).

Raspberry Pi IoT Projects.

Berkeley, CA: Apress.

ISBN: 978-1-4842-1378-0 978-1-4842-1377-3.

DOI: 10.1007/978-1-4842-1377-3.

URL: <http://link.springer.com/10.1007/978-1-4842-1377-3> (visitado 07-06-2019)

(Vid. pág. 38).

contributors, phpMyAdmin (2018).

phpMyAdmin.

URL: <https://www.phpmyadmin.net/> (visitado 23-11-2018)

(Vid. pág. 39).

MySQL Documentation (2018).

URL: <https://dev.mysql.com/doc/> (visitado 19-11-2018)

(Vid. pág. 40).

MySQL Reference Manual (2018).

URL: <https://dev.mysql.com/doc/refman/8.0/en/> (visitado 19-11-2018)

(Vid. págs. 53, 56-59, 61, 64).

Eastron, Jiaying (2019).

Jiaying Eastron Electronic Instruments Co., Ltd.

URL: <http://www.eastrongroup.com/productsview/14.html> (visitado 17-02-2019)

(Vid. pág. 72).

«ICECO Nueva Línea., Pdf.» (2019).

En:

URL: http://www.h2otek.com/tienda/pdf/aire_acondicionado/MANUAL%20AIRE%20CONDICIONADO%20LINEA%20ICECO%20RHEEM.pdf (visitado 04-02-2019)

(Vid. pág. 78).