



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TELECOMUNICACIONES

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA
MONITOREO Y CONTROL DEL CONSUMO ENERGÉTICO EN
REDES DE BAJA TENSIÓN PARA USO RESIDENCIAL”**

Trabajo de Titulación

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR:

DIEGO ALEJANDRO LÓPEZ FREIRE

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TELECOMUNICACIONES

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA
MONITOREO Y CONTROL DEL CONSUMO ENERGÉTICO EN
REDES DE BAJA TENSIÓN PARA USO RESIDENCIAL”**

Trabajo de Titulación

Tipo: Propuesta Tecnológica

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR: DIEGO ALEJANDRO LÓPEZ FREIRE

DIRECTOR: Ing. Diego Fernando Veloz Chérrez MSc.

Riobamba – Ecuador

2023

©2023, Diego Alejandro López Freire

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, DIEGO ALEJANDRO LÓPEZ FREIRE, declaro que el presente trabajo de titulación es de mi autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo

Riobamba, 06 de Abril de 2023

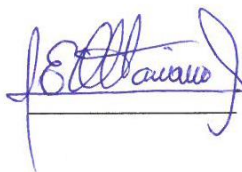
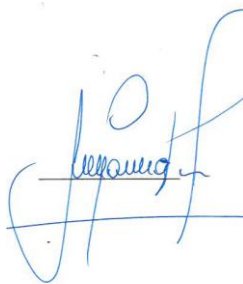
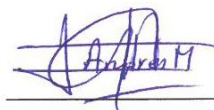
A handwritten signature in blue ink, appearing to read 'Diego Alejandro López Freire', with a large, stylized flourish extending upwards from the end of the signature.

Diego Alejandro López Freire

180381824-2

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA CARRERA
TELECOMUNICACIONES

El Tribunal de Trabajo de Titulación certifica que: El Trabajo de Titulación: Tipo: Propuesta Tecnológica, “**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA MONITOREO Y CONTROL DEL CONSUMO ENERGETICO EN REDES DE BAJA TENSION**”, realizado por el señor **DIEGO ALEJANDRO LÓPEZ FREIRE**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

NOMBRE	FIRMA	FECHA
Ing. Edwin Vinicio Altamirano Santillán Msc. PRESIDENTE DEL TRIBUNAL		2023-04-06
Ing. Diego Fernando Veloz Chérrez Msc. DIRECTOR DEL TRABAJO DE TITULACIÓN		2023-04-06
Ing. Andrés Fernando Morocho Caiza Msc. ASESOR DEL TRABAJO DE TITULACIÓN		2023-04-06

DEDICATORIA

Dedico mi trabajo realizado a mi Dios Jehová, a la memoria de mis madres Laura y Zoila, a mis padres Antonio, Angelita, German y demás seres queridos que han sido parte de mi vida.

A mis amigos que me han acompañado a lo largo de la carrera y saben el esfuerzo y trabajo que se requiere hasta llegar a estas etapas.

Diego L.

AGRADECIMIENTO

Agradezco a Dios por la salud y vida en todo este tiempo, por la sabiduría para enfrentar momentos difíciles y no permitirme darme por vencido hasta cumplir satisfactoriamente la meta planteada. Adicionalmente agradezco a mi familia por inculcarme la meta de estudiar en esta grandiosa y prestigiosa institución ESPOCH.

A todas las personas que se convirtieron en parte de mi familia brindándome su apoyo de manera incondicional a pesar de todas las adversidades en el camino.

Diego L.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS	ix
ÍNDICE DE ILUSTRACIONES	xi
RESUMEN	xv
SUMMARY	xvi
INTRODUCCIÓN	1

CAPITULO I

1. MARCO REFERENCIAL	2
1.1. Antecedentes	2
1.2. Formulación del Problema	3
1.3. Sistematización del Problema.....	3
1.4. Justificación Teórica.....	3
1.5. Justificación Aplicativa	4
1.6. Objetivos.....	4
1.6.1. <i>Objetivo General</i>	4
1.6.2. <i>Objetivos Específicos</i>	4

CAPITULO II

2. MARCO TEÓRICO.....	5
2.1. Internet de las Cosas.....	5
2.1.1. <i>Arquitecturas IoT</i>	5
2.1.1.1. <i>Modelo de 3 capas</i>	6
2.1.1.2. <i>Modelo de 4 capas</i>	8
2.1.1.3. <i>Arquitectura SOA</i>	9
2.1.1.4. <i>Arquitectura API</i>	10
2.1.2. <i>Protocolos de Comunicación</i>	14
2.1.2.1. <i>Protocolos en la capa de percepción</i>	15
2.1.2.2. <i>Protocolos en la capa de red</i>	37
2.1.2.3. <i>Protocolos en la capa de aplicación</i>	42
2.1.3. <i>Procesamiento de datos</i>	55
2.1.3.1. <i>Cloud Computing</i>	55
2.1.3.2. <i>Fog Computing</i>	57
2.2. Energía eléctrica	59

2.2.1.	<i>Corriente eléctrica</i>	59
2.2.2.	<i>Corriente alterna (AC)</i>	59
2.2.3.	<i>Potencia Eléctrica</i>	60
2.2.3.1.	<i>Potencia activa (P)</i>	60
2.2.3.2.	<i>Potencia Reactiva(Q)</i>	60
2.2.3.3.	<i>Potencia Aparente(S)</i>	60
2.2.4.	<i>Energía consumida</i>	61
2.2.4.1.	<i>Dispositivos de uso diario en el hogar</i>	61
2.2.4.2.	<i>Medidas de ahorro energético en el hogar</i>	61

CAPITULO III

3.	MARCO METODOLÓGICO	69
3.1.	Metodología	69
3.1.1.	<i>Metodología Inductiva</i>	69
3.1.2.	<i>Metodología Deductiva</i>	71
3.2.	Estudio del escenario de implementación del sistema	72
3.3.	Requerimientos del diseño del sistema	72
3.3.1.	<i>Análisis comparativo de Arquitecturas IoT</i>	73
3.3.2.	<i>Análisis comparativo de tecnologías de comunicación IoT</i>	73
3.3.3.	<i>Análisis comparativo de protocolos IoT</i>	73
3.3.3.1.	<i>Análisis de protocolos de capa de red</i>	73
3.3.3.2.	<i>Análisis de protocolos de capa de transporte</i>	73
3.3.3.3.	<i>Análisis de protocolos de capa de aplicación</i>	73
3.3.4.	<i>Análisis comparativo de sistemas de computing</i>	74
3.3.5.	<i>Análisis comparativo de sensores para el Smartmeter</i>	74
3.3.6.	<i>Análisis comparativo de tarjetas de desarrollo</i>	74
3.3.7.	<i>Análisis Comparativo de hardware como servidor</i>	75
3.3.8.	<i>Análisis Comparativo de entornos de desarrollo IoT</i>	75
3.3.8.1.	<i>Capa de percepción</i>	75
3.3.8.2.	<i>Capa de Aplicación</i>	76
3.3.9.	<i>Análisis comparativo de componentes de actuadores para la integración de actuadores en el Smartmeter</i>	76
3.3.9.1.	<i>Enchufe Inteligente</i>	76
3.3.9.2.	<i>Dimmer</i>	77
3.4.	Diseño del sistema de monitoreo del consumo energético	78
3.4.1.	<i>Capa de censado</i>	80

3.4.2.	<i>Capa de red</i>	80
3.4.3.	<i>Capa de aplicación</i>	81
3.5.	Simulación del prototipo del sistema de monitoreo energético IoT	82
3.6.	Implementación del prototipo IoT	83
3.6.1.	<i>Conexión física del hardware</i>	83
3.6.2.	<i>Programación de tarjetas de desarrollo</i>	83
3.6.3.	<i>Programación del servidor</i>	88
3.7.	Diseño del sistema de control del consumo energético	102
3.7.1.	<i>Colocación de enchufes inteligentes</i>	103
3.7.2.	<i>Colocación de dimmers en lámparas de uso frecuente</i>	105
3.7.3.	<i>Configuración del sistema de regulación de intensidad de luminarias</i>	106
3.7.4.	<i>Diseño de la base de datos para el registro del consumo eléctrico de electrodomésticos</i>	106
3.7.5.	<i>Configuración del sistema de control de cargas a través de tomacorrientes inteligentes</i>	109

CAPITULO IV

4.	MARCO DE RESULTADOS Y DISCUSIÓN	112
4.1.	Lista de chequeo	112
4.2.	Resultados de la primera fase	113
4.3.	Resultados de la segunda fase	116
4.4.	Resultados del rendimiento del sistema IoT	116
4.4.1.	<i>Latencia</i>	116
4.4.2.	<i>Monitoreo de variables eléctricas</i>	119
4.4.3.	<i>Consumo Eléctrico</i>	119
4.4.4.	<i>Control de intensidad luminaria</i>	120
4.4.5.	<i>Accionamiento de los tomacorrientes inteligentes</i>	122

CONCLUSIONES	1224
--------------	-------	------

RECOMENDACIONES	12225
-----------------	-------	-------

BIBLIOGRAFIA

ÍNDICE DE TABLAS

Tabla 1-2: Descripción de los niveles de la escala de Likert	14
Tabla 2-2: Evaluación de parámetros según la escala de Likert	14
Tabla3-2: Configuraciones MIMO en el estándar 802.11n	19
Tabla4-2: Escala de Likert para evaluación del consumo de potencia de las tecnologías IoT ..	35
Tabla5-2: Comparativa de tecnologías disponibles en la capa de percepción	35
Tabla6-2: Comparativa entre cabeceras IPv4 e IPv6	39
Tabla 7-2: Tipos de mensaje en MQTT	43
Tabla 8-2: Comparativa de protocolos de la capa de aplicación.....	54
Tabla9-2: Comparativa de diferentes técnicas de análisis de datos	58
Tabla10-2: Clasificación de electrodomésticos del hogar	61
Tabla 11-2: Comparativa de sensores para la capa de percepción	63
Tabla 12-2: Comparativa de tarjetas de desarrollo	64
Tabla 13-2: Comparativa de opciones para servidor como hardware	65
Tabla 14-2: Comparativa de entornos de desarrollo IoT	66
Tabla 15-2: Analizadores de paquetes de red	67
Tabla 16-2: Comparativa de bases de datos relacionales y no relacionales	68
Tabla 1-3: Descripción de las secciones del piso 3 de la vivienda	72
Tabla 2-3: Comparativa de tomacorrientes inteligentes.....	76
Tabla 3-3: Precios y especificaciones de dimmers comerciales	77
Tabla 4-3: Lista de materiales del dimmer.....	78
Tabla 5-3: Esquema de direccionamiento IP de los tomacorrientes.	105
Tabla 6-3: Esquema de direccionamiento IP de los tomacorrientes.	105
Tabla 6-4: Comparativa de sensores de temperatura.	110
Tabla 1-4: Checklist de aspectos cualitativos del sistema de monitoreo y control del consumo energético	112
Tabla 2-4: Checklist de aspectos cualitativos del sistema de monitoreo y control del consumo energético	116
Tabla 3-4: Checklist de aspectos cualitativos del sistema de monitoreo y control del consumo energético	116
Tabla 4-4: Resultados de pruebas de latencia hacia el dashboard mediante el uso de herramienta web.....	116
Tabla 5-4: Resultados de pruebas de latencia dashboard-dimmer.	117
Tabla 6-4: Resultados de pruebas de latencia hacia el dashboard mediante el uso de herramienta web.....	118
Tabla 7-4: Medidas realizadas por el sistema IoT implementado respecto a un equipo comercial	119

Tabla 8-4: Evaluación del sistema ante cortes de energía eléctrica.	120
Tabla 9-4: Evaluación de respuesta en cortes eléctricos para los dimmers inteligentes.	122
Tabla 10-4: Evaluación de respuesta en cortes eléctricos para el sistema de control de consumo energético en base a la temperatura.	122

ÍNDICE DE ILUSTRACIONES

Ilustración 1-2: Modelo arquitectónico de 3 capas IoT.....	6
Ilustración 2-2: Modelo arquitectónico de 4 capas IoT.....	8
Ilustración 3-2: Modelo SOA para entornos IoT.....	9
Ilustración 4-2: Esquema de funcionamiento de una API.....	11
Ilustración 5-2: Arquitectura REST.....	11
Ilustración 6-2: Componentes de una red 802.11.....	16
Ilustración 7-2: Arquitectura 802.11 modo infraestructura.....	17
Ilustración 8-2: Topología Ad-hoc en redes 802.11.....	17
Ilustración 9-2: Pila de protocolos en el estándar 802.11.....	18
Ilustración 10-2: Formato de trama MAC del estándar 802.11.....	19
Ilustración 11-2: Arquitectura Zigbee en las capas de red y aplicación.....	23
Ilustración 12-2: Arquitectura de red LoRaWAN.....	24
Ilustración 13-2: Esquema de comunicaciones en redes SigFox.....	26
Ilustración 14-2: Arquitectura en redes NB-IoT.....	27
Ilustración 15-2: Stack de capas en Z-Wave.....	28
Ilustración 16-2: Topología de red malla en Z-Wave.....	29
Ilustración 17-2: Pila de 6LoWPAN.....	30
Ilustración 18-2: Esquema de red LoWPAN.....	30
Ilustración 19-2: Arquitectura 6LoWPAN.....	31
Ilustración 20-2: FDMA con TDMA en GSM.....	32
Ilustración 21-2: Modulación CDMA en 3G.....	33
Ilustración 22-2: Esquema de sistemas de comunicaciones MIMO.....	33
Ilustración 23-2: Espectro OFDM.....	34
Ilustración 24-2: Cabeceras de las direcciones IP. a) Cabecera IPv4. b) Cabecera IPv6.....	39
Ilustración 25-2: Proceso de intercambio de paquetes en una conexión TCP.....	41
Ilustración 26-2: Estructura de un mensaje MQTT.....	42
Ilustración 27-2: Funcionamiento del broker en MQTT.....	44
Ilustración 28-2: Flujo de mensajes en el funcionamiento de MQTT.....	44
Ilustración 29-2: Arquitectura HTTP.....	45
Ilustración 30-2: Esquema de funcionamiento de CoAP.....	47
Ilustración 31-2: Arquitectura CoAP.....	49
Ilustración 32-2: Arquitectura XMPP.....	51
Ilustración 33-2: Estructura mensaje XML.....	52
Ilustración 34-2: Procedimiento de handshaking en WebSocket.....	53

Ilustración 35-2: Esquemas de Websocket.....	53
Ilustración 36-2: Arquitectura de Cloud Computing.....	56
Ilustración 37-2: Arquitectura de Fog Computing	57
Ilustración 38-2: Triangulo de Potencias en corriente alterna.....	60
Ilustración 39-2: Parámetros captados por los sensores	62
Ilustración 1-3: Diagrama de procesos del proyecto en base al método inductivo	70
Ilustración 2-3: Diagrama de procesos del proyecto en base al método deductivo	71
Ilustración 3-3: Modulo PZEM 004Tv3	74
Ilustración 4-3: Arduino Mega 2560 R3	75
Ilustración 5-3: Node MCU ESP8266.....	75
Ilustración 6-3: Tomacorriente Inteligente Kasa EP10	77
Ilustración 7-3: Esquema de conexión del dimmer	78
Ilustración 8-3: Diagrama de procesos del sistema de monitoreo del consumo energético IoT	79
Ilustración 9-3: Esquema de conexión en nivel de censado	80
Ilustración 10-3: Esquema de comunicación serial Arduino-ESP8266.....	81
Ilustración 11-3: Esquema comunicacional del sistema de monitoreo IoT.....	81
Ilustración 12-3: Escenario del sistema de monitoreo IoT en Packet Tracer	82
Ilustración 13-3: Resultado de prueba de ping del sistema de monitoreo IoT	82
Ilustración 14-3: Implementación del prototipo del sistema de monitoreo de consumo energético.	83
Ilustración 15-3: Diagrama de procesos en Arduino Mega 2560.....	84
Ilustración 16-3: Diagrama de procesos involucrados en el ESP8266.....	85
Ilustración 17-3: Información enviada desde ESP8266-NODE-RED vía HTTP POST.	87
Ilustración 18-3: Diagrama de procesos a ejecutar en el servidor local	88
Ilustración 19-3: Sistema operativo Ubuntu 20.0.4 instalado en la computadora HP Pavilion.	88
Ilustración 20-3: Asignación de dirección IP estática en el router de la vivienda del piso 3. ...	89
Ilustración 21-3: Servidor XAMPP instalado en la computadora HP Pavilion.....	89
Ilustración 22-3: Interfaz del panel de servicios de XAMPP.	90
Ilustración 23-3: Creación de nueva base de datos en phpMyAdmin.	90
Ilustración 24-3: Creación las diferentes tablas de la base de datos en phpMyAdmin.	91
Ilustración 25-3: Diagrama entidad relación de la base de datos “medidor_pz”	91
Ilustración 26-3: Registro manual de sensores 1 y 2 en la tabla “estado_actual”	92
Ilustración 27-3: Inicialización del servidor Node-Red	92
Ilustración 28-3: Inicialización del servidor Node-Red	93
Ilustración 29-3: Nodos de recepción y confirmación de recibimiento de datos	93
Ilustración 30-3: Nodos de recepción y confirmación de recibimiento de datos	94
Ilustración 31-3: Nodo de extracción y procesamiento de variables energéticas	94

Ilustración 32-3: Nodo de base de datos SQL.....	95
Ilustración 33-3: Nodos de consulta, escritura de promedios de potencia y energía consumida horaria, diaria y mensual.....	95
Ilustración 34-3: Funciones de consulta horaria de parámetros eléctricos.....	96
Ilustración 35-3: Asignación de parámetros consultados a variables tipo flow.	96
Ilustración 36-3: Inserción de los parámetros horarios hacia la tabla “historial_dia”.....	97
Ilustración 37-3: Funciones de consulta diaria de parámetros eléctricos.....	97
Ilustración 38-3: Asignación de parámetros consultados a variables tipo flow.	97
Ilustración 39-3: Inserción de los parámetros diarios hacia la tabla “historial_mes”.....	98
Ilustración 40-3: Funciones de consulta mensual de parámetros eléctricos.....	98
Ilustración 41-3: Asignación de parámetros consultados a variables tipo flow.	98
Ilustración 42-3: Inserción de los parámetros mensuales hacia la tabla “historial_año”.	99
Ilustración 43-3: Resultados de variables censadas en la tabla “estado_actual”.....	99
Ilustración 44-3: Resultados de variables censadas en la tabla “historial_hora”.	99
Ilustración 45-3: Resultados de variables censadas en la tabla “historial_dia”.....	100
Ilustración 46-3: Resultados de variables censadas en la tabla “historial_mes”.	100
Ilustración 47-3: Resultados de variables censadas en la tabla “historial_año”.....	100
Ilustración 48-3: Esquema de consulta diaria del consumo energético a la base de datos “medidor_pz”.	101
Ilustración 49-3: Esquema de consulta mensual del consumo energético a la base de datos “medidor_pz”.	101
Ilustración 50-3: Configuración del dashboard del sistema de monitoreo del consumo energético.....	102
Ilustración 51-3: Diagrama del sistema de control del consumo energético.....	103
Ilustración 52-3: Colocación de enchufes inteligentes en electrodomésticos de línea blanca y marrón.....	104
Ilustración 53-3: Configuración de los tomacorrientes inteligentes mediante aplicación Kasa.....	104
Ilustración 54-3: Configuración del pool de direcciones IP estáticas en el router del piso 3 de la vivienda.....	105
Ilustración 55-3: Implementación de dimmers en luminarias del cuarto máster.....	106
Ilustración 56-3: Configuración canales Websocket para luminarias del cuarto máster.....	106
Ilustración 57-3: Creación de base de datos para el registro de variables de consumo y temperatura del refrigerador.....	107
Ilustración 58-3: Creación de tabla “estado_actual” en la base de datos “refrigerador”.....	107
Ilustración 59-3: Creación de tabla “historial_hora” en la base de datos “refrigerador”.	108
Ilustración 60-3: Creación de tabla “historial_dia” en la base de datos “refrigerador”.....	108
Ilustración 61-3: Creación de tabla “historial_mes” en la base de datos “refrigerador”.....	108
Ilustración 62-3: Creación de tabla “historial_año” en la base de datos “refrigerador”.....	109

Ilustración 63-3: Diagrama entidad- relación de la base de datos “refrigerador”.	109
Ilustración 64-3: Configuración de procesos de los actuadores en Node-Red.	110
Ilustración 65-3: Implementación del sistema de control del consumo energético en refrigerador.	111
Ilustración 1-4: Pestaña de datos en tiempo real del consumo de “Luminarias”.	113
Ilustración 2-4: Pestaña de datos en tiempo real del consumo de “Tomacorrientes”.	114
Ilustración 3-4: Pestaña de datos en tiempo real “Totales” del consumo en la vivienda.	114
Ilustración 4-4: Pestaña de “Consulta diaria” de los consumos del piso 3 de la vivienda.	115
Ilustración 5-4: Pestaña de “Consulta mensual” de los consumos del piso 3 de la vivienda.	115
Ilustración 6-4: Reporte de latencia en Google PageSpeed Insights desde un móvil-dashboard	117
Ilustración 7-4: Reporte de latencia en Google PageSpeed Insights desde un ordenador-dashboard.	117
Ilustración 8-4: Reporte de latencia en Arduino Serial Monitor, NodeRed-Actuador.	118
Ilustración 9-4: Reporte de estado del sistema IoT ante cortes eléctricos.	119
Ilustración 10-4: Regulación de intensidad luminaria de la lampara 1 al 100% de brillo.	120
Ilustración 11-4: Regulación de intensidad luminaria de la lampara 2 al 100% de brillo.	120
Ilustración 12-4: Regulación de intensidad luminaria de la lampara 2 al 100% de brillo.	121
Ilustración 13-4: Regulación de intensidad luminaria de las lámparas 1 y 2 al 100% de brillo.	121
Ilustración 14-4: Regulación de intensidad luminaria de las lámparas 1 y 2 al 30% de brillo.	121
Ilustración 15-4: Comparativa de resultados obtenidos al realizar la medición del consumo energético.	123

RESUMEN

Hoy en día los contadores eléctricos digitales están reemplazando a los analógicos debido su mejor eficiencia en cuanto al monitoreo y gestión de la red de abonados por parte de la empresa eléctrica. Los sistemas de medición energética actuales no presentan informes detallados sobre consumos realizados de manera mensual ni diaria, así como en tiempo real; y la gestión de dispositivos inteligentes para el ahorro energético en varias plataformas supone un gran problema para los usuarios. Para solucionar la ocurrencia de estos inconvenientes es necesario promover el uso de un sistema basado en Internet de las cosas (IoT) para monitorear y controlar el consumo eléctrico. El presente trabajo describe una metodología inductiva para el diseño e implementación de un prototipo IoT, empleando los datos censados para el accionamiento de diferentes eventos que actuarán en el lugar de estudio. Los principales resultados del estudio indican que los valores medidos por el sistema son similares a los equipos de medición confiables, gran capacidad de reintegración al servicio ante cortes eléctricos, latencia adecuada del servicio y un ahorro considerable del consumo eléctrico en refrigeradores, aplicando un sistema de control de accionamiento de tomacorrientes inteligentes en base a la temperatura interna del electrodoméstico. En consecuencia, el sistema IoT implementado permite al usuario una mejor gestión de la energía consumida en el hogar, así como una integración de varios servicios a través de una interfaz amigable para un manejo fácil y rápido.

Palabras claves: <INTERNET DE LAS COSAS (IOT)>, <CONSUMO ENERGÉTICO>, <DISEÑO DE PROTOTIPO>, <SISTEMA DE CONTROL>, <COMUNICACIÓN INALÁMBRICA>, <PROTOCOLO WEBSOCKET>, <PROTOCOLO HTTP>.



SUMMARY

Nowadays, digital electric meters are replacing analog ones due to their better efficiency in terms of monitoring and management of the subscriber network by the electric company. Current energy measurement systems do not present detailed reports on consumption carried out on a monthly or daily basis, as well as in real time; and the management of smart devices for energy saving on various platforms is a big problem for users. To solve the occurrence of these drawbacks, it is necessary to promote the use of a system based on the Internet of Things (IoT) to monitor and control electricity consumption. The present work describes an inductive methodology for the design and implementation of an IoT prototype, using the census data to activate different events that will act in the study place. The main results of the study indicate that the values measured by the system are similar to reliable measurement equipment, great capacity to reintegrate into service in the event of power cuts, adequate latency of the service and considerable savings in electrical consumption in refrigerators, applying a system of Smart outlet drive control based on the internal temperature of the appliance. Consequently, the implemented IoT system allows the user a better management of the energy consumed in the home, as well as an integration of various services through a friendly interface for easy and fast management.

Keywords: <INTERNET OF THINGS (IOT)>, <ENERGY CONSUMPTION>, <PROTOTYPE DESIGN>, <CONTROL SYSTEM>, <WIRELESS COMUNICATION>, <WEBSOCKET PROTOCOL>, <HTTP PROTOCOL>.



Msc. Wilson G. Rojas
CI: 0602361842

INTRODUCCIÓN

El crecimiento de la población y la aparición de nuevos dispositivos cada día conlleva un aumento incesante del consumo de energía y ha planteado grandes retos en cuanto a la gestión de la energía por parte del consumidor (Saleem et al., 2021: p.1). Actualmente los medidores de energía eléctrica convencionales se limitan a registrar el consumo domiciliario de energía en kWh consumidos y esto a su vez se ve reflejado al consumidor monetariamente, este proceso es realizado de manera mensual sin ningún plan de ahorro energético y económico, además el usuario no puede monitorear su consumo realizado diariamente, lo cual no permite la concientización del uso adecuado de la energía eléctrica. Dado que los usuarios poseen hábitos como no desconectar dispositivos electrónicos (cargadores de laptop, celulares, televisiones, etc) que no están siendo utilizados, se tiene un consumo mínimo al permanecer en stand-by el cual es denominado como desperdicio. El Smart Meter contribuirá al monitoreo y control de las variables eléctricas: voltaje y corriente, mediante un servidor local, los datos de consumo estarán disponibles en la red para permitir el acceso del cliente a su respectiva visualización de valores de consumo diarios o mensual, a través de una interfaz intuitiva al usuario que muestre una comparativa entre el correcto consumo y desperdicio de energía eléctrica, impulsando así el ahorro y reducción del recurso, para ello podrán ingresar desde una página web o aplicación móvil predeterminada.

CAPÍTULO I

1. MARCO REFERENCIAL

En este capítulo se detallan antecedentes, problema y justificación del que está inmerso el trabajo de titulación. Así también se realiza una revisión de publicaciones existentes a nivel Nacional e Internacional sobre temas relacionados que han tenido un impacto en la investigación del mismo.

1.1. Antecedentes

En Bangladesh se desarrolló un contador de energía eléctrica basado en IoT (Wali-ur et al., 2019: p.4). La medición de variables eléctricas como: voltaje y corriente, se envían a un servidor central de manera inalámbrica, procesando acciones en base a la información recolectada como la suspensión del servicio en caso de alteración de valores por parte del consumidor y recordatorios de pago del servicio.

Lapo (2020, p.17) realizó el diseño e implementación en dos etapas: circuitería y programación, de un sistema de monitoreo de parámetros energía eléctrica, factor de potencia y calidad de energía consumida, mediante dispositivos inteligentes en una vivienda aplicando IoT. Los resultados de los parámetros medidos alcanzaron un nivel de confiabilidad del 94,64% y el porcentaje de error de la corriente se incrementó en $\pm 7,05\%$ al sobrepasar los 30 Amperios, y la recepción de un mensaje SMS cuando existe un consumo inusual.

Chávez y Herrera (2020, p.17) diseñaron e implementaron un medidor inteligente de energía eléctrica basado en IoT. La curva de consumo de energía eléctrica a nivel domiciliario presentó la mayor demanda de consumo desde las 18H00-22H00 horas. La información captada por los sensores de voltaje y corriente se aloja en una base de datos y es visualizada a través de una página web y aplicación móvil al usuario para la optimización del recurso energético consumido.

En India Sheeba et al. (2021: p.1) desarrollaron un contador inteligente de energía eléctrica, adaptando el contador proporcionado por la empresa eléctrica en los domicilios con un circuito adicional y las lecturas de mediciones son almacenadas en una base de datos alojada en la nube para automatizar la facturación y gestionar los datos recogidos. Comunicando la compañía eléctrica con los consumidores para la visualización del consumo realizado, facturación y pago del servicio de manera remota en tiempo real a través de la aplicación móvil o página web.

En la Isla Mauricio, Murdan y Kullootee (2020, p.1) desarrollaron un sistema de control interactivo basado en Wi-Fi y LoRaWAN para la gestión de la energía doméstica, con el fin de obtener las

mismas prestaciones de un smart meter profesional a un precio accesible. Las lecturas de las variables eléctricas fueron censadas y almacenadas en una base de datos en la nube, visualizándolas en tiempo real, procesándolas y accionando el relé ante niveles de voltaje y corriente bajos, desconectando el circuito y optimizando el consumo energético. Concluyendo que para entornos interiores donde existan altas transmisiones paquetes información, Wi-fi es la tecnología ideal; así como para ambientes exteriores donde la cobertura Wi-fi no es viable y las tasas de transmisión de paquetes información son bajas, LoraWAN es la tecnología indicada.

1.2. Formulación del Problema

¿Cómo diseñar e implementar un prototipo IoT para el monitoreo y control del consumo energético en redes de baja tensión para uso residencial?

1.3. Sistematización del Problema

- ¿Cómo monitorear el consumo energético de una vivienda en tiempo real mediante el uso de IoT?
- ¿Qué técnicas permiten controlar el consumo energético en una vivienda para optimizar el consumo energético?
- ¿Qué elementos y arquitecturas conforman la red IoT para el monitoreo y control del consumo energético en redes de baja tensión para uso residencial?
- ¿Qué software y hardware permiten el desarrollo del prototipo del control del consumo energético en redes de baja tensión para uso residencial a través de una infraestructura IoT?
- ¿Cómo implementar el prototipo IoT de monitoreo y control del consumo energético en redes de baja tensión para uso residencial que cumplan los requerimientos propuestos al inicio de la investigación?
- ¿Qué beneficios obtendrá el propietario de la residencia con la implementación de este sistema del monitoreo y control del consumo energético?

1.4. Justificación Teórica

De los trabajos expuestos anteriormente, se puede deducir que emplean la tecnología IoT (Internet de las cosas), pero no explotándola debidamente, ya que se enfocan únicamente al control de acceso de los datos censados por el medidor, suspendiendo el servicio ante la modificación de estas lecturas, la generación de alertas o recordatorios de pago del servicio y la visualización del consumo energético ya sea en un histórico mensual, diario o en tiempo real a través de páginas web y aplicaciones móviles.

El presente trabajo propone la correcta utilización de la tecnología IoT, enfocándose no solamente en la visualización de parámetros de energía eléctrica y valor de consumo; si no también en el control del consumo energético, buscando minimizar el costo de la planilla del servicio eléctrico en un domicilio a través de procesos automatizados.

1.5. Justificación Aplicativa

El presente proyecto pretende emplear de mejor manera el potencial de la tecnología IoT, mediante el uso de sensores para la percepción de variables eléctricas como voltaje, corriente y potencia en un domicilio de red monofásica de tensión; las diferentes lecturas de los parámetros energéticos se alojarán en un servidor de base de datos local. Toda esta información será enviada a Internet para su posterior procesamiento; mediante el uso de una página web y una aplicación móvil, permitirán la visualización diaria, mensual o en tiempo real de estas variables eléctricas que adicionalmente mostrarán el consumo energético y valor mensual a pagar por el servicio. Accionando los diferentes procesos en base a las diferentes lecturas de variables energéticas, de esta forma optimizando el consumo energético en el hogar.

1.6. Objetivos

1.6.1. Objetivo General

Diseñar e Implementar un prototipo IoT para el Monitoreo y Control del consumo energético en redes de baja tensión para uso residencial.

1.6.2. Objetivos Específicos

- Estudiar la tecnología IoT con sus principales componentes, protocolos y arquitecturas.
- Diseñar el sistema de monitoreo de las variables eléctricas para su visualización y almacenamiento en un servidor local mediante IoT.
- Controlar el consumo energético en base a la regulación de intensidad de luminarias y verificación de las cargas conectadas en los tomacorrientes inteligentes para que muestre su respectivo análisis.
- Evaluar el rendimiento del sistema IoT mediante factores como la latencia de comunicación, monitoreo de las variables eléctricas, consumo eléctrico, regulación de intensidad de luminarias y control del accionamiento de las cargas mediante los tomacorrientes inteligentes.

CAPÍTULO II

2. MARCO TEÓRICO

En el presente capítulo se abordará de manera breve conceptos y características acerca del internet de las cosas, definición del consumo energético residencial, sistemas de monitoreo energético comerciales, medidas de ahorro energético, sensores, placas de desarrollo, tarjetas de desarrollo, dispositivos de comunicación, elementos de visualización y herramientas para el desarrollo de interfaces web.

2.1. Internet de las Cosas

El internet de las cosas (IoT) está comprendido como la red de objetos físicos conectados a través de Internet, los cuales pueden interactuar mediante sistemas embebidos, redes de comunicación, mecanismos de computación de respaldo y aplicaciones típicamente en la nube. Permitiendo la comunicación entre los objetos, acceder a información de internet, capturar, almacenar e interactuar con usuarios humanos, así como con otros sistemas y aplicaciones, creando ambientes mucho más conectados e inteligentes (Quiñonez, 2019, p.13).

Mediante el uso de sensores en dispositivos de uso diario se recopila, procesa y presenta la información a los usuarios finales, para una fácil interpretación. Existiendo de fondo la comunicación entre maquinas o también llamada en inglés (Machine to Machine, M2M), recolección, análisis de datos y despliegue de resultados. Todo este compendio de acciones ayuda a la automatización de procesos para la toma adecuada de decisiones por parte del usuario (Amazon Web Services, 2022).

2.1.1. Arquitecturas IoT

El Internet de las Cosas (IoT) es un ecosistema complejo de elementos que están interactuando como: sensores, gateways, servidores y plataformas de acceso a la información. El crecimiento de esta tecnología a menudo se ve obstaculizado por su fragmentación, ya que es una tarea difícil conseguir que varios componentes de hardware y software distintos se comuniquen entre sí y operen juntos como un todo (HOLOGRAM, 2020).

Se comprende por arquitectura a la infraestructura para la descripción de una red de componentes físicos y su configuración y organización funcional, sus principios y procedimientos

operacionales, y los tipos de datos que se intercambian entre ellos. Es decir, se detalla cómo los componentes físicos de IoT recolectan datos, cómo los procesan e intercambian, y en qué formatos lo hacen (T-Systems, 2018).

A continuación, se detallan a fondo las principales arquitecturas empleadas en IoT, partiendo de la arquitectura básica en Internet de las cosas.

2.1.1.1. Modelo de 3 capas

La arquitectura de tres capas es la estructura más común y generalmente conocida. Se utilizó por primera vez en las fases iniciales de este estudio de IoT. La ilustración 1-2, detalla tres niveles: percepción, red y aplicación.

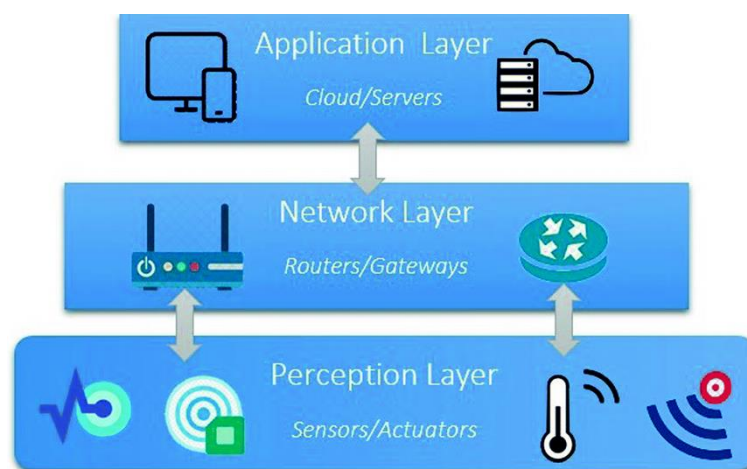


Ilustración 1-2: Modelo arquitectónico de 3 capas IoT.

Fuente: mPython,2017

Capa de percepción

Representa a la capa física de la arquitectura IoT, en ella se utilizan principalmente sensores y sistemas embebidos, los cuales recolectan grandes cantidades de datos en función de las necesidades; incluyendo dispositivos de borde, sensores y actuadores que se comunican con el entorno, esta capa detecta ciertos parámetros espaciales o detecta otras cosas u objetos inteligentes en el entorno como: WSN, dispositivos heterogéneos, sensores tipo objetos del mundo real, humedad y temperatura, etc.; para después transformarlos en una estructura digital que será enviada a la capa de red (Muhammad, 2017, p.5). Los objetos poseen una identidad digital (identificador único universal, UUID), permitiendo monitorear al objeto en el mundo digital.

Capa de red

Es la capa central de IoT, cuya función principal es asegurar la transmisión de datos entre la capa de percepción y aplicación de la arquitectura IoT. Es una estructura que soporta enlaces alámbricos e inalámbricos entre los objetos, permitiendo percibir su entorno, habilitando a compartir datos entre las cosas conectadas, posibilitando la gestión de eventos y el procesamiento inteligente de IoT; esta capa también garantiza capacidades únicas de direccionamiento y enrutamiento para la integración unificada de incontables dispositivos en una única red cooperativa (GeekofGeeks, 2021).

La capa de red es crucial en cualquier enfoque de IoT, considerando funcionalidades de QoS, gestión eficiente de energía en la red y en las cosas, seguridad en los datos, privacidad, entre otras (Sosa et al., 2016: p.4).

Capa de aplicación

Es el nivel superior de esta arquitectura, el cual define todo el software necesario para ofrecer un servicio específico que varía según la necesidad del cliente. En esta capa se almacenan, agregan, filtran y procesan los datos de los niveles anteriores y se utilizan bases de datos, software de análisis, etc. Como resultado de este procesamiento, los datos se ponen a disposición de las aplicaciones reales de IoT. Esto se hace a menudo utilizando cierto software definido como middleware, que tiene la tarea de ocultar la heterogeneidad de las capas subyacentes (GeekofGeeks, 2021). Algunas tecnologías de software que se utilizan actualmente de forma generalizada para gestionar la gran cantidad de datos que proporcionan los dispositivos son:

- **Cloud Computing:** Los servicios como el almacenamiento o el procesamiento de datos se proporcionan a partir de un conjunto de recursos preexistentes, configurables y disponibles remotamente en forma de arquitectura distribuida.
- **Fog Computing:** Los recursos para el almacenamiento y la preparación de los datos abandonan la nube pública o el centro de datos y se distribuyen en una capa intermedia en la red por medio de nodos fog o unidades de preprocesamiento
- **Edge Computing:** El procesamiento de datos se distribuye parcialmente en los nodos periféricos de la red para aumentar el rendimiento de los sistemas IoT.

2.1.1.2. Modelo de 4 capas

La arquitectura de tres capas representa un esquema simplificado del entorno IoT. En vista del continuo desarrollo y despliegue de IoT, este modelo no podría satisfacer todos los requisitos; debido a las múltiples aplicaciones de IoT, múltiples organizaciones han definido modelos de cómo podría organizarse la red de IoT. Uno de los principales modelos presentados por la Unión Internacional de Telecomunicaciones (UIT) presenta una arquitectura de cuatro capas para IoT en 2012 con la recomendación ITU-T Y 2060 (Ebrary, 2018).

El modelo arquitectónico IoT propuesto por la UIT, está conformado por cuatro capas y de capacidades de gestión y seguridad relacionadas a ellas. La ilustración 2-2 muestra las cuatro capas de esta arquitectura, los cuales son: dispositivo, red, apoyo a servicios y aplicaciones y aplicación.

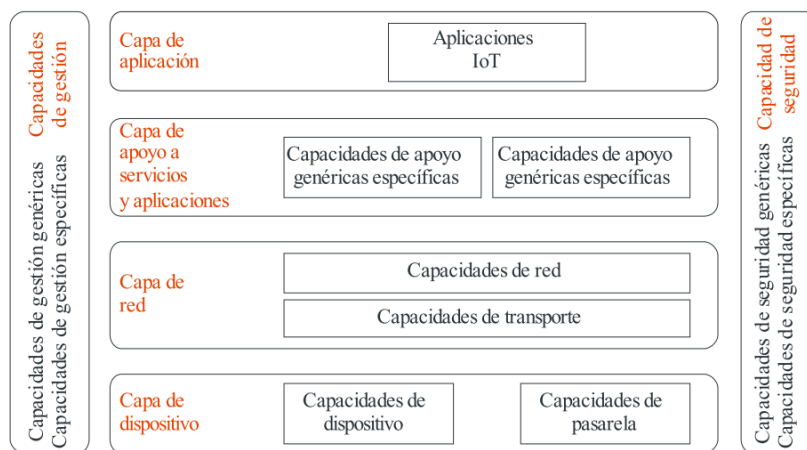


Ilustración 2-2: Modelo arquitectónico de 4 capas IoT.

Fuente: ITU, 2012, p.13

El modelo de 4 capas de UIT posee el mismo principio de funcionamiento para las capas de percepción, red y aplicación; pero agrega una capa adicional llamada “Capa de apoyo a servicios y aplicaciones”, la cual se detalla a continuación.

Capa de soporte de servicios y aplicaciones

Esta capa ofrece la información correcta a la aplicación adecuada en el momento adecuado, transforman los datos brutos en información relevante y lista para ser consumida por las aplicaciones, de modo que se exhiban comportamientos inteligentes (Kalatzis et al., 2019: p.9). Existen 2 tipos de capacidades:

- **Capacidades de soporte genéricas:** Son empleadas diversas aplicaciones IoT, tales como almacenamiento o procesamiento de datos; también pueden emplearse en otras capacidades específicas como crear otras capacidades específicas (ITU, 2012, p.14).
- **Capacidades de soporte específicas:** Atienden las necesidades particulares de diversas aplicaciones, pueden consistir en diversos grupos de capacidades precisas que ofrecen distintas funciones de apoyo a las varias aplicaciones IoT (ITU, 2012, p.14).

2.1.1.3. Arquitectura SOA

También conocida como arquitectura orientada a servicios, define a un sistema complejo como un grupo de objetos bien definidos o subsistemas, los cuales pueden ser reutilizados, manteniendo su forma individual. De esta forma, sus componentes de hardware y software en una arquitectura IoT, implementados mediante SOA, pueden ser reutilizados y actualizados de manera eficiente; su diseño brinda extensibilidad, escalabilidad, modularidad e interoperabilidad entre cosas heterogéneas, así como las capacidades y funcionalidades que se encapsularon en un conjunto de servicios (IBM, 2020). La arquitectura basada en SOA para el desarrollo de sistemas de IoT propuesta se conforma de cuatro capas tal como se muestra en la ilustración 3-2, las cuales se detallan a continuación.



Ilustración 3-2: Modelo SOA para entornos IoT.

Fuente: Sosa et al., 2016: p.3

Nuevamente se excluyen las demás capas a excepción de la de servicio, ya que la concepción de las demás es la misma que en el modelo expuesto de 3 capas.

Capa de Servicio:

En este nivel se establecen y manejan los servicios solicitados por los usuarios o aplicaciones de software. Representa una de las capas de operación crítica de la arquitectura, funciona en modo bidireccional, trabaja como interfaz entre la capa de objetos y aplicación. Permite la gestión de dispositivos, gestión de información, filtrado de datos, agregación de datos, análisis semántico, y descubrimiento de información (Sosa et al., 2016: p.4).

La capa de servicios se conforma de:

- **Descubrimiento de servicios:** Descubre objetos que proporcionan el servicio requerido y la información necesaria, a través del UUID en el registro de servicios o repositorio de servicios.
- **Composición de servicios:** Permite la interacción entre las cosas conectadas a través de la combinación de los servicios disponibles para la ejecución de una tarea determinada, es decir, cuando los servicios están creados y almacenados en el repositorio de servicios, se pueden combinar en servicios de mayor nivel de complejidad a partir de la lógica de negocio (Sosa et al., 2016: p.4).
- **Gestión de servicios:** Gestiona y determina los métodos de confianza para entender las solicitudes de servicios.
- **Interfaces de servicios:** Se emplean para apoyar las interacciones entre todos los servicios proporcionados.

2.1.1.4. Arquitectura API

API

La interfaz de programación de aplicaciones (API) representa un conjunto de rutinas, protocolos y herramientas para crear aplicaciones de software; especificando cómo deben interactuar los componentes de software. Permite interacciones entre sistemas siguiendo un conjunto de estándares y protocolos para compartir características, información y datos, de esta forma permite a los desarrolladores crear, diseñar productos y servicios que se comunicarán con otros productos y servicios (Walker, 2022).

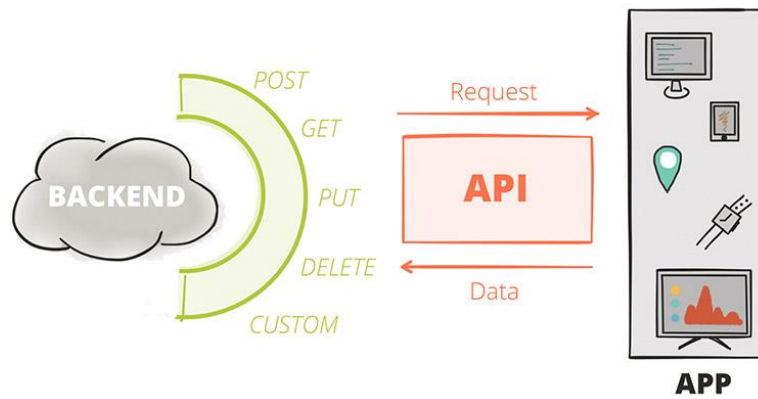


Ilustración 4-2: Esquema de funcionamiento de una API.

Fuente: Altexsoft, 2021

REST

Transferencia de Estado Representacional describe una arquitectura de desarrollo web que se fundamenta en el protocolo HTTP y en los hipervínculos que construyen la web. Definida en el año 2000 por Roy Fielding. La mayoría de las aplicaciones que se desarrollan para servicios profesionales disponen de una API REST para el intercambio de información entre el front y el back. Lo que la hace tan potente es precisamente el aislamiento que proporciona entre la lógica del back-end y cualquier cliente consumidor de éste (T.W.S, 2021).

El protocolo HTTP posee una semántica para cada una de sus operaciones que engloban las operaciones básicas de CRUD (crear, leer, actualizar y eliminar), códigos de estado para el resultado de la operación y direcciones de los recursos; las páginas web retornan HTML, los servicios REST como formato de datos suelen usar JSON (Pizarro, 2021).

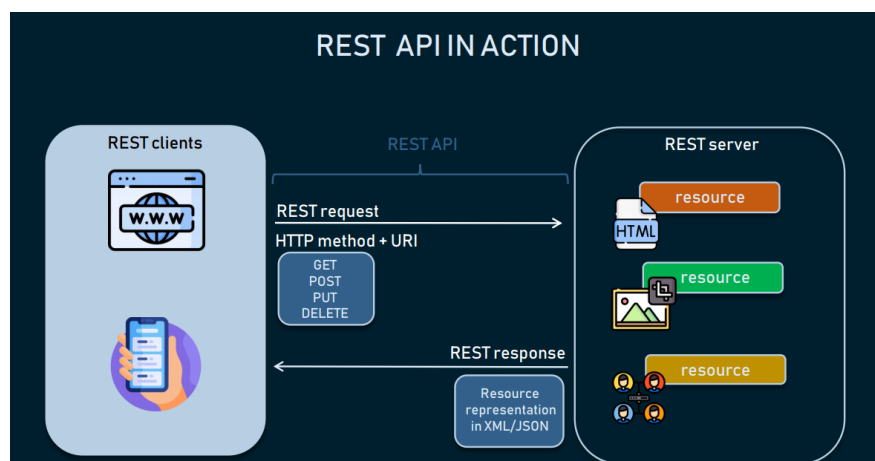


Ilustración 5-2: Arquitectura REST.

Fuente: Altexsoft, 2021

Para la construcción de una aplicación web bien diseñada, se emplean principios REST que brinden apoyo al desarrollar servicios que sean más escalables, confiables y flexibles. Para lograr este objetivo, la arquitectura REST tiene seis pautas y una API que cumpla todas estas condiciones se denominará RESTful (T.W.S, 2021).

A continuación, se detallan las diferentes pautas del sistema REST.

a) Cliente-Servidor

El principio fundamental de la arquitectura web cliente-servidor es la separación de tareas, lo que significa que el cliente que envía la solicitud es completamente independiente del servidor que devuelve la respuesta (Altexsoft, 2021).

b) Sin estado

Toda la información (estado) que se requiere en una petición debe ser enviada por el Cliente. Por lo tanto, el Servidor no debe almacenar ningún dato durante una comunicación Cliente-Servidor, lo que significa que cada solicitud es una solicitud independiente (T.W.S 2021).

c) Capacidad de almacenamiento en memoria caché

Es una estructura de almacenamiento computacional centrada en mantener almacenados los datos a los que se accede con frecuencia, mejorando el rendimiento y la eficiencia de la red. Debido a esta técnica, es posible reducir o incluso eliminar la necesidad de que el cliente envíe peticiones al servidor, el cual debe informar si la petición puede ser almacenada en caché o no (IBM, 2021).

d) Interfaz Uniforme

Significa la forma en que el cliente y el servidor compartirán la información definiendo una interfaz que debe seguirse en cada solicitud. En otras palabras, es un contrato entre el cliente y el servidor que determina las normas para su comunicación (IBM, 2021).

A continuación, se describen las cuatro restricciones adicionales que conforman la Interfaz Uniforme.

- **Identificación de recursos:** Un recurso es una información que puede ser etiquetada, se emplea en una petición para identificar lo que el cliente desee acceder en el servidor (T.W.S, 2021).

- **Manipulación de recursos mediante representación:** El cliente debe estar seguro de que la petición al servidor tiene suficiente información para manipular (crear, recuperar, actualizar, eliminar) el recurso informado, que puede representarse mediante múltiples formatos, como JSON, XML, HTML, etc. En otras palabras, el cliente puede especificar la representación deseada de un recurso en cada solicitud a un servidor (T.W.S, 2021).
 - **Mensajes autodescriptivos:** Un mensaje autodescriptivo asegura una interfaz uniforme en la comunicación al contener toda la información que un cliente o un servidor necesita para entender la petición y la respuesta con sólo comprobar la semántica del mensaje (T.W.S, 2021).
 - **HATEOAS:** El hipertexto como motor del estado de la aplicación, significa que una respuesta enviada desde el servidor debe contener información sobre lo que el cliente puede hacer en posteriores peticiones. Es decir, el servidor indica qué acciones puede realizar el cliente a continuación, los servidores solo deben enviar hipertexto (enlaces) a los clientes (T.W.S, 2021).
- e) **Sistema por capas**
- Se refiere al hecho de que puede haber más componentes y subsistemas entre un cliente y un servidor. En otras palabras, el cliente no puede asumir que se está comunicando directamente con el servidor, y no conoce la complejidad para procesar la petición y devolver la respuesta; por ejemplo, un cliente envía una solicitud a un servidor, pero primero pasa por una capa proxy para comprobar la seguridad (IBM, 2021).
- f) **Código bajo demanda**
- Es la única restricción opcional, implica que el servidor puede enviar un código ejecutable como respuesta al cliente. Es decir, lo que ocurre cuando un navegador recibe una respuesta del servidor con una etiqueta HTML (script) para que, cuando se cargue el documento HTML, se pueda ejecutar el script (IBM, 2021).

A continuación, la tabla 1-2 presenta los criterios cuantizados para la escala de Likert, que serán empleados en la comparativa de arquitecturas expuestas en la tabla 2-2.

Tabla 1-2: Descripción de los niveles de la escala de Likert

Nivel	Descripción
1	No especifica
2	No cumple
3	Parcialmente
4	Cumple

Realizado por: Lopez, Diego; 2022

Tabla 2-2: Evaluación de parámetros según la escala de Likert

Características	Modelo 3 capas	Modelo 4 capas	Modelo API	Modelo SOA
Procesamiento de información de manera remota	4	4	4	4
Portabilidad de servicios	1	3	4	4
Compatibilidad con otros sistemas	1	1	3	4
Escalabilidad	1	3	4	3
Maneja servicios desde una aplicación	1	3	4	3
Maneja diferentes lenguajes informáticos	1	1	4	3
Gestión eficiente del canal de comunicación	1	1	4	1
Implementa medidas de seguridad en sus servicios	1	1	4	1
Puntaje total	11	17	31	23

Realizado por: Lopez, Diego; 2022

2.1.2. *Protocolos de Comunicación*

Partiendo del modelo arquitectónico IoT de tres capas expuestas previamente, se detallará a fondo los diferentes protocolos involucrados en cada una de sus capas.

2.1.2.1. Protocolos en la capa de percepción

Esta capa representa el nivel físico de los objetos e interactúa con el entorno que los rodea recogiendo y procesando información. Los objetos inteligentes, que son los bloques fundamentales en los que se basa IoT, pueden ser objetos de uso común (nevera, un televisor, un coche, etc.) o simples dispositivos equipados con sensores y capacidades informáticas. Los cuales están dotados de las siguientes propiedades esenciales (Lombardi et al., 2021: pp. 3-4).

- **Comunicación:** Los objetos pueden conectarse entre sí y con los recursos de Internet para emplear datos y servicios, actualizar su estado y cooperar para alcanzar objetivos comunes.
- **Identificación:** los objetos deben ser identificados de forma única. Dependiendo de la aplicación específica, también pueden añadirse una o más de las siguientes propiedades
- **Direccionalidad:** los objetos pueden ser directamente accesibles, es decir, dirigidos, para ser interrogados y/o configurados a distancia.
- **Detección y actuación:** los objetos pueden recoger información sobre el mundo circundante y manipularla mediante el uso de sensores y actuadores.
- **Procesamiento de la información integrado:** Los objetos inteligentes están dotados de capacidades de cálculo para procesar los resultados de los sensores y accionar los actuadores.
- **Localización:** Los objetos son conscientes de su ubicación física o pueden ser localizados
- **Interfaz de usuario:** los objetos pueden comunicarse adecuadamente con los usuarios a través de pantallas u otras interfaces.

Las tecnologías implicadas en esta capa se detallan a continuación.

WIFI

Representa varios avances tecnológicos que se han desarrollado a lo largo de varios años, cada avance se define mediante una enmienda al estándar que se identifica con un sufijo de una o dos letras a "802.11". El estándar 802.11 original permitía velocidades de hasta 2 Mbps sólo en la banda de 2,4 GHz (JUNIPER, 2018).

Arquitectura

La ilustración 6-2 muestra que las redes 802.11 constan de cuatro componentes físicos principales, que se detallan a continuación.

- **Estaciones**

Las redes se construyen para transferir datos entre estaciones, las cuales son dispositivos informáticos con interfaces de red inalámbricas. Las estaciones normalmente son: laptops, smartphones, entre otros dispositivos inteligentes (Gast, 2005, p.37).

- **Puntos de acceso**

Las tramas de una red 802.11 deben convertirse en otro tipo de trama para su entrega al resto del mundo. Los dispositivos denominados puntos de acceso (AP) realizan la función de puente de inalámbrico a cable.

- **Medio inalámbrico**

Para trasladar las tramas de una estación a otra, la norma utiliza un medio inalámbrico. Se definen varias capas físicas diferentes; la arquitectura permite desarrollar múltiples capas físicas para soportar la MAC 802.11. Inicialmente, se estandarizaron dos capas físicas de radiofrecuencia (RF) y una capa física de infrarrojos, aunque las capas de RF han resultado ser mucho más populares (Gast, 2005, p.37).

- **Sistema de distribución**

Cuando se conectan varios puntos de acceso para formar una gran área de cobertura, deben comunicarse entre sí para seguir los movimientos de las estaciones móviles. El sistema de distribución es el componente lógico de 802.11 utilizado para reenviar las tramas a su destino. 802.11 no especifica ninguna tecnología concreta para el sistema de distribución. En la mayoría de los productos comerciales, el sistema de distribución se implementa como una combinación de un mecanismo de puente y un medio de sistema de distribución, que es la red troncal utilizada para retransmitir tramas entre los puntos de acceso; a menudo se denomina simplemente red troncal. Actualmente se ha mudado de ethernet a fibra óptica como tecnología de red troncal (Gast, 2005, p.38).

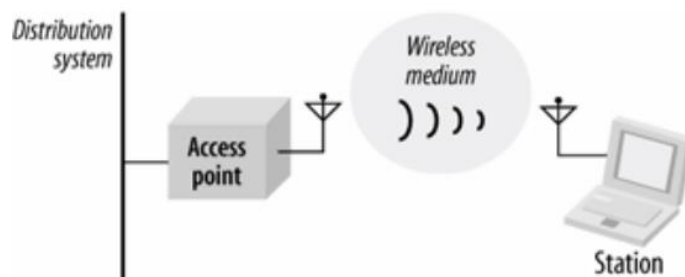


Ilustración 6-2: Componentes de una red 802.11.

Fuente: Gast, 2005, p.38

El bloque básico de una red 802.11 es el conjunto de servicios básicos (BSS), que es simplemente un grupo de estaciones que se comunican entre sí. Las comunicaciones tienen lugar dentro de un

área algo difusa, denominada área de servicio básica, definida por las características de propagación del medio inalámbrico (Gast, 2005, p.39).

Cuando una estación se encuentra en la zona de servicio básica, puede comunicarse con los demás miembros del BSS. Los BSSs vienen en dos versiones, BSS independiente también conocida como ad-hoc y la BSS de infraestructura (Hoffman, 2016).

Modo infraestructura

El modo de infraestructura utiliza un punto de acceso (AP) que está conectado a la red. Los clientes envían y reciben paquetes a través del AP. Se pueden conectar varios AP para formar una red ampliada (Salazar, 2012, p.19).

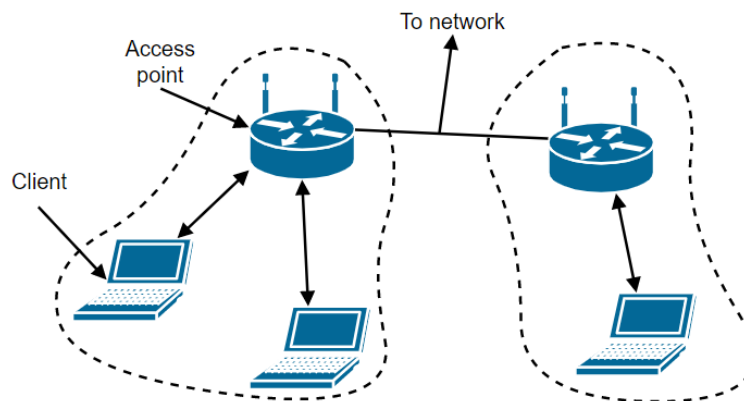


Ilustración 7-2: Arquitectura 802.11 modo infraestructura.

Fuente: CS Notes, 2014

Modo Ad-hoc

Llamado también conjunto de servicios básicos independientes (IBSS), es el grupo de estaciones o dispositivos inteligentes que se asocian para poder enviarse directamente tramas entre sí, sin la necesidad de un access point. Un ejemplo de este modo de operación es WiFi Direct (Salazar, 2012, p.19).

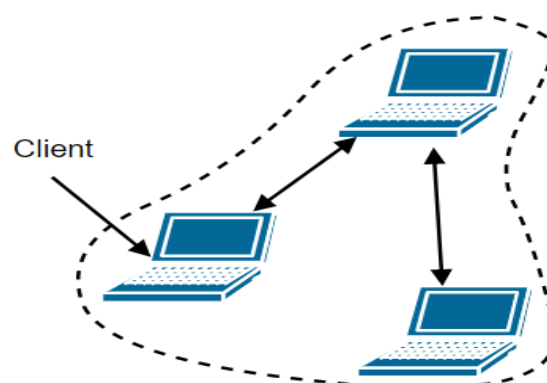


Ilustración 8-2: Topología Ad-hoc en redes 802.11.

Fuente: CS Notes, 2014

Como se muestra en la ilustración 9-2, la IEEE 802 presenta dos capas individuales para la capa de enlace de datos del modelo OSI que son: la subcapa de control de enlace lógico (LLC) y la subcapa de control de acceso al medio (MAC). Las especificaciones para la capa física y capa MAC que se comunica con el nivel de control de enlace lógico están dictaminadas por el estándar IEEE 802.11.

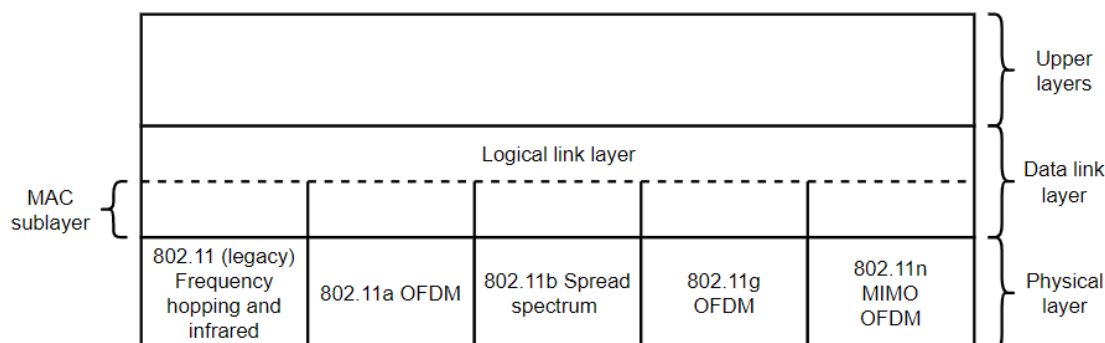


Ilustración 9-2: Pila de protocolos en el estándar 802.11.

Fuente: CS Notes, 2014

Capa Física

Todas las versiones 802.11 establecen una serie de modelos de codificación y transmisión para las comunicaciones inalámbricas como: Espectro Ensanchado por Salto de Frecuencia (FHSS), Espectro Ensanchado por Secuencia Directa (DSSS) y Multiplexación por División de Frecuencias Ortogonales (OFDM); utilizan radios de corto alcance para transmitir señales en las bandas de frecuencia ISM de 2,4 GHz o 5 GHz, estas bandas no tienen licencia, por lo que son compartidas por muchos otros dispositivos. Todos los métodos de transmisión 802.11 definen múltiples tasas, se enfatiza el estándar IEEE 802.11.n, ya que es la versión del estándar con la que trabajan la mayoría de las compañías de servicio de Internet en Ecuador; fue introducido al mercado en 2009 y se diseñó para obtener un mayor rendimiento, utilizando múltiples antenas para aumentar la velocidad de transmisión de datos, que ahora es de hasta 600 Mbps. Operan en dos bandas de frecuencia de 2,4 GHz y 5 GHz, y empleando canales con un ancho de banda de 20 MHz y 40 Mhz; brindando una cobertura superior en interiores y exteriores, duplicando esencialmente el alcance de sus predecesores. Emplea una nueva tecnología *MIMO* (Multiple-input Multiple Output), una técnica para enviar y recibir más de una señal inalámbrica en el mismo canal de radio al mismo tiempo, las diferentes configuraciones para MIMO se muestran en la tabla 3-2, y adicionalmente emplean una modulación 64QAM (WirelessSolution, 2014, p.2).

Tabla3-2: Configuraciones MIMO en el estándar 802.11n

Configuración MIMO	Velocidad máxima de datos según el ancho del canal	
	20 MHz	40 MHz
1x1	65-72 Mbps	150 Mbps
2x2	130 -144 Mbps	300 Mbps
3x3	195- 216 Mbps	450 Mbps
4x4	260- 288 Mbps	600 Mbps

Realizado por: López, Diego; 2022

Se ha seleccionado la configuración MIMO 2x2, ya que es la más común en todos equipos proporcionados por los proveedores de servicio de internet en Ecuador.

Capa MAC

Es la subcapa inferior que define las acciones de acceso a los medios que realiza el dispositivo. Proporciona la dirección de la capa de enlace de datos y la identificación de datos según los requisitos de señalización física del vehículo y el tipo de protocolo de capa de enlace de datos utilizado. La separación de la capa de enlace de datos en subcapas con el tipo de trama especificado por la capa superior permite el acceso a diferentes tipos de medios definidos por la capa inferior. Este es el caso de muchas tecnologías alámbricas e inalámbricas (CS Notes, 2014).

Trama MAC 802.11

La trama MAC 802.11 de la ilustración 10-2 consta del encabezado MAC, el cuerpo de la trama y la secuencia de verificación de trama (FCS). El formato de trama MAC está formado por un grupo de nueve campos con un orden fijo en todas las tramas.

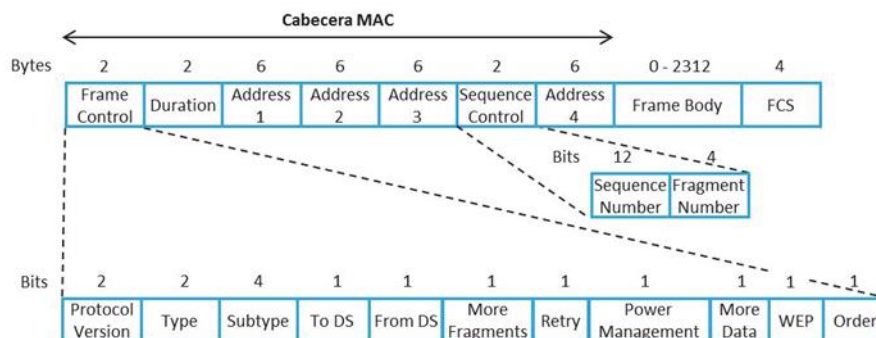


Ilustración 10-2: Formato de trama MAC del estándar 802.11

Fuente: Salazar, 2012, p.25

1. **Control de tramas:** Posee información de control utilizada para determinar el tipo de trama MAC 802.11 y proporciona la información necesaria en los siguientes campos para

comprender cómo se manejan las tramas MAC; cada uno de los subcampos se detallan a continuación:

- **Versión del protocolo:** Proporciona la versión actual del protocolo 802.11 en uso. Las estaciones receptoras usan este valor para determinar si se admite la versión de protocolo de la trama recibida.
- **Campo Tipo y Subtipo:** Definen la funcionalidad de la trama, existen tres diferentes tramas: control, datos y gestión. Hay varias subcategorías para cada tipo de trama. Cada subtipo define la función específica que realiza el tipo de trama asociado.
- **Las tramas hacia DS y desde DS:** Muestran si la trama está dirigida o sale del DS (Sistema de distribución) y se usan únicamente en tramas del tipo de datos de las estaciones vinculadas con un AP.
- **Más segmentos:** Muestran si existen más segmentos de la trama, ya sea de datos o de gestión.
- **Retry:** Indica si se está retransmitiendo la trama, ya sea de datos o de gestión.
- **Manejo de energía:** Muestra si la estación emisora está en modo activo o en modo de ahorro de energía.
- **Más datos:** indica que la STA en modo de ahorro energético tiene muchas tramas para enviar al AP. También lo utiliza el AP para indicar que hay tramas de difusión/multidifusiones adicionales.
- **WEP:** Indica si el cifrado y la autenticación se utilizan dentro de la trama. Se puede configurar para **todos las tramas** de gestión y **datos cuyo** subtipo se establece en autenticación.
- **Order:** Indica que todas las tramas recibidas deben procesarse en orden.

2. Campo de duración/ID

Es empleado para todas las tramas de tipo de control excepto a las del subtipo de sondeo de ahorro de energía (PS) e indica el período restante necesario para recibir la siguiente

transmisión de la trama. Para el subtipo de encuesta PS, el campo contiene la identidad de asociación (AID) de la STA que envía la trama.

3. Campos de dirección

Según el tipo de trama, los cuatro campos de dirección contienen las siguientes combinaciones de tipos de dirección:

- **Identificador BSS (BSSID):** Identifica de forma única cada BSS. Si la trama es de la STA de la infraestructura BSS, el BSSID es la dirección MAC del AP. Si la trama se origina en una STA que pertenece a una IBSS, el BSSID es una dirección MAC administrada localmente generada aleatoriamente por la STA que inició la IBSS.
- **Dirección de destino (DA):** Muestra la dirección MAC del destino final para recibir la trama.
- **Dirección de origen (SA):** Especifica la dirección MAC de la fuente desde la que se creó y envió la trama por primera vez.
- **Dirección del receptor (RA):** Indica la dirección MAC de la siguiente STA para recibir la trama.
- **Dirección del transmisor (TA):** Indica la dirección MAC de la STA que envió la trama.

4. Control de secuencia

Este campo consta de dos subcampos: Número de fragmento y Número de secuencia. Cada uno de los subcampos anteriores se describe a continuación.

- **Número de secuencia:** Especifica el número de secuencia de cada trama, este número de secuencia es el mismo para cada trama enviada en tramas fragmentadas. De lo contrario, el número de secuencia comienza en cero y se incrementa en 1 hasta llegar a 4095.
- **Número de fragmento:** Muestra el número de fragmento de la trama fragmentada, su valor inicial se establece en 0 y se incrementa en 1 por cada fragmento de trama enviado.

5. **Cuerpo de la trama:** Contiene los datos o la información contenida en cada trama de gestión o trama de tipo de datos.

6. **Secuencia de verificación de trama:** También conocida por sus siglas en inglés (FCS), la estación transmisora emplea una verificación de redundancia cíclica (CRC) en todos los campos del encabezado MAC y el cuerpo de la trama para generar valores FCS. La estación receptora emplea el mismo cálculo de CRC para determinar su propio valor de campo FCS y comprobar errores de trama en tránsito.

Subcapa de control de enlace lógico

Conocida por sus siglas en inglés (LLC), brinda a la capa de red servicios de capa de enlace de datos independientes de los medios físicos, también provee servicios de verificación de errores y control de flujo. Esto permite que diferentes protocolos en las capas superiores de la pila de protocolos accedan a diferentes tipos de redes físicas; el LLC coloca en la trama información que identifica el protocolo de capa de red que se usa en la trama. Esta información permite que varios protocolos de la capa 3, tales como IPv4 e IPv6, utilicen la misma interfaz y los mismos medios de red.

ZIGBEE

Opera en la banda de 2.4 GHz, y posee un alcance de 100 metros, se basa en el estándar de protocolo de comunicación IEEE 802.15.4 y se utiliza en redes PAN; originalmente fue creado para entornos residenciales e industriales. IEEE 802.15.4 es una colección de estándares que expone las características de la capa física y MAC para redes de área personal inalámbricas de baja velocidad (LRWPAN), estos estándares proporcionan velocidades de transmisión de 40 Kbps a 250 Kbps y trabajan en frecuencias de 868/915 MHz y 2.4 GHz a velocidades de datos altas y bajas y proveen una comunicación inalámbrica bidireccional. Las capas superiores de la pila de protocolos: red y aplicación también son estandarizadas, permitiendo en el nivel de red el enrutamiento multihop, descubrimiento y mantenimiento de rutas, la seguridad y la conexión/salida de una red, a través de la asignación de direcciones cortas de 16 bits de longitud a los dispositivos recientemente incorporados. Mientras para la capa de aplicación proporcionan un marco para el desarrollo y la comunicación de aplicaciones distribuidas (Cazalla, 2019, p.14).

La ilustración 11-2 presenta el esquema arquitectónico de esta tecnología, definiendo a la capa de aplicación como el nivel más alto de la especificación y está conformada por objetos de aplicación (APO), subcapa de aplicación (APS) y un objeto de dispositivo Zigbee (ZDO). Los servicios de administración de dispositivos y redes son proporcionados por el ZDO, facilitando el descubrimiento entre sí de las APO y se organicen en una aplicación distribuida, definiendo el rol

de cada uno. Los servicios de transferencia de datos y seguridad son proporcionados por la APS a los APO y ZDO. Los APO se encargan de definir el perfil de la aplicación, soportando hasta 240 objetos para la comunicación o aplicaciones diferentes. Además, ZigBee incluye un proveedor de servicios de seguridad (SSP). Sin embargo, cada capa (MAC, red y aplicación) puede protegerse y compartir claves de seguridad. El SSP se inicializa y configura a través del ZDO y requiere la implementación del algoritmo de encriptación AES (Singh, 2015, p.2). Las configuraciones que permite esta tecnología son: estrella, malla y árbol.

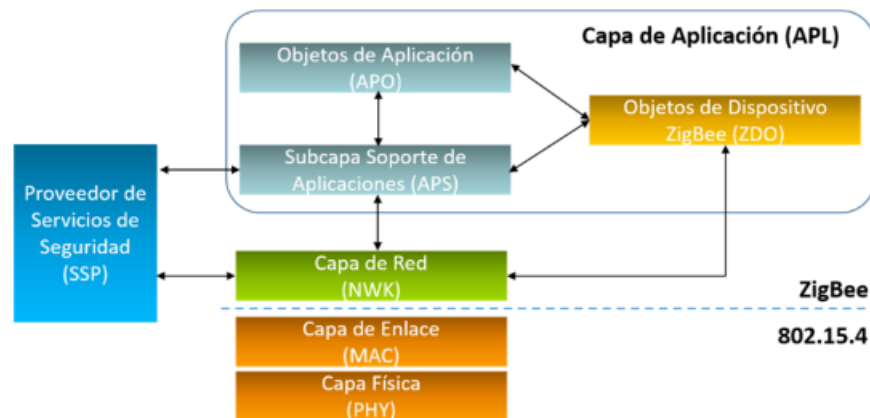


Ilustración 11-2: Arquitectura Zigbee en las capas de red y aplicación.

Fuente: Yacchirema, 2019, p.51

Con la finalidad de facilitar sus estándares interoperables de mercado, la alianza Zigbee en la actualidad ofrece tres especificaciones como base.

- **ZigBee PRO:** Concebido para brindar la base para IoT, ya que está optimizado para soportar grandes redes con miles de dispositivos y minimizar el consumo energético.
- **ZigBee RF4CE:** Diseñado para aplicaciones bidireccionales de control de dispositivo a dispositivo (D2D), que no ameritan las prestaciones de una configuración en malla con todas las funciones que ofrece la especificación ZigBee.
- **ZigBee IP:** Brinda solución de red a través del estándar, ofreciendo una arquitectura escalable, que permite integrar los protocolos IPv6 en la red basada en IEEE 802.15.4 para lograr una conexión sin fisuras de los dispositivos ZigBee a Internet. Brinda compatibilidad con el estándar 6LoWPAN, así como todos los estándares de protocolo establecidos por los grupos IETF, pero aún no se ha logrado establecer en su totalidad en el mercado.

LoRa

Long Range o largo alcance, representa a la capa física o la de modulación usada en la creación de comunicaciones de largo alcance. Se basa en el espectro extendido de chirp (CSS), la cual es

una modulación que así como la modulación FSK optimiza el consumo energético pero extiende aún más su rango de comunicación (Bolaños y Rios, 2013, p.35). Permite velocidades de transmisión de 0,3 kbps hasta 50kbps, optimizando el consumo energético de la batería en los dispositivos en alrededor de 10 años; esta tecnología funciona en entornos urbanos con una cobertura de 2 a 5 Km, suburbanos de hasta 15 Km y rurales de hasta 45 km. Funciona en las bandas ISM de 433, 868 o 915 MHz, según la región en donde se esté trabajando (LoRa Alliance, 2014).

LoRaWAN

Es un protocolo de capa MAC, siendo un estándar abierto desarrollado por la Alianza LoRA, que proporciona un mecanismo de control de acceso medio, que permite que varios dispositivos finales se comuniquen a través una puerta de enlace mediante la modulación LoRa (Bolaños y Rios, 2013, p.36). La ilustración 12-2 representa la arquitectura LoRaWAN, en la que se detallaran sus componentes.

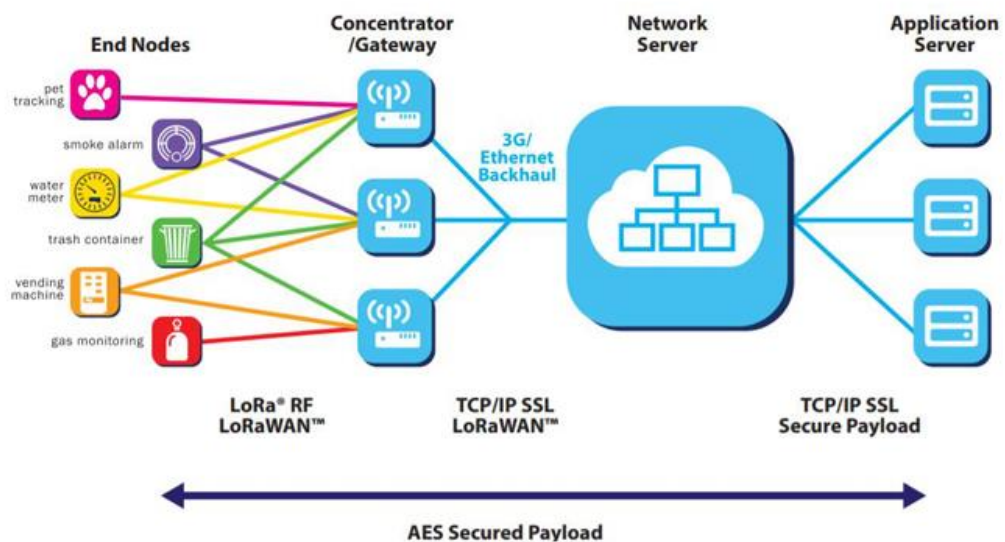


Ilustración 12-2: Arquitectura de red LoRaWAN

Fuente: Semtech, 2020, p.12

- **Nodos finales:** Recopilan y envían los datos censados en *upstream* y reciben datos en *downstream* desde el servidor de aplicaciones y emplean comunicación inalámbrica de un solo salto con una o varias *gateways* (Bolaños y Rios, 2013, p.36).

- **Gateways:** Reenvían mensajes entre los nodos finales y el servidor de red central mediante una interfaz de *backhaul* con un rendimiento más alto, utilizando comúnmente conexiones Ethernet o 3G (Bolaños y Rios, 2013, p.36).
- **Servidor de la red:** Decodifica la información enviada por los nodos finales, conectándose a varias puertas de enlace mediante una conexión TCP/IP segura, usualmente a través de ethernet o 3G; elimina los mensajes duplicados, decidiendo qué compuerta debe responder a un mensaje de nodo final. Gestiona el nodo final, las velocidades de transmisión de datos con una velocidad de datos adaptable (ADR), destinadas a maximiza la capacidad de la red y optimizar el consumo energético del nodo final (Yacchirema, 2019, p.85).
- **Servidor de aplicación:** los datos del servidor de red son recibidos por cada aplicación, para después decodificar los paquetes encriptados y utilizar la información (Yacchirema, 2019, p.85).

SIGFOX

Diseñada para redes LPWAN, fue creada en 2009 y basada en la topología de red celular, proporcionando un bajo rendimiento y ancho de banda. Opera en las bandas ISM sub-GHz sin licencia, en los 915 MHz para Norteamérica y 868MHz en Europa; los dispositivos finales (ED) enlazados a los gateways o estaciones base SigFox utilizan la modulación de desplazamiento de fase binaria diferencial (D-BPSK) para transmitir datos en una banda ultra estrecha de 100Hz, esto permite a SigFox explotar eficientemente la banda de frecuencias y proporciona niveles muy bajos de ruido, obteniendo un bajo consumo energético y alta sensibilidad en el receptor, con una velocidad máxima de transmisión de datos de 100 bps y 600 bps en recepción. Esta tecnología soporta tanto comunicaciones de enlace ascendente (UL) como de descendente (DL) en su versión más reciente, con 140 mensajes al día para las comunicaciones UL y 4 mensajes al día para las comunicaciones DL. Por lo tanto, todos los mensajes UL no tienen confirmación de recibo (ACK) (Lalle et al., 2019).

Para garantizar que la entrega de los mensajes UL a los receptores, se emplea la diversidad de tiempo/frecuencia y la duplicación de la transmisión. La carga útil máxima de cada mensaje DL es de 8 bytes, mientras que esta carga útil es de 12 bytes para las comunicaciones UL. La vida útil de las baterías de los dispositivos finales puede durar hasta 10 años. La red SigFox está dispuesta en una topología de estrella en la que los ED están conectados a la nube SiFox a través de las estaciones base SigFox (BS).

La ilustración 13-2 muestra el funcionamiento de una red Sigfox, donde las BS escuchan todos los canales disponibles para recibir mensajes; para la comunicación UL, un mensaje enviado por un ED es recibido por cualquier BS SigFox en el rango y, en promedio, el número de BS que pueden recibir estos datos es de 3. Luego las BS envían el mismo mensaje a la nube SigFox que es el núcleo de la red SigFox. La nube SigFox se encarga de procesar los mensajes y los envía a la ubicación de los clientes. También contiene herramientas que analizan los datos generados por la red. Para la transmisión DL, se adopta el camino inverso.

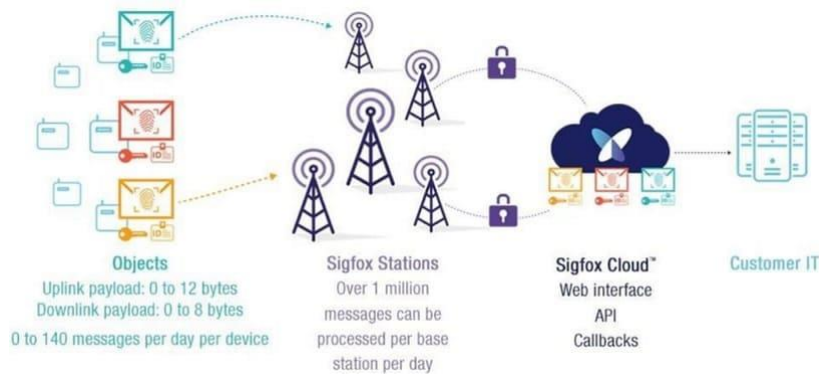


Ilustración 13-2: Esquema de comunicaciones en redes SigFox

Fuente: SigFox, 2019

NB-IoT

La IoT de banda estrecha (NB-IoT), conecta hasta cien mil dispositivos finales por célula empleando operadores de redes celulares existentes. Brinda una duración de batería de 8 a 10 años, gran cobertura, bajo coste y alta seguridad de la red. Posee un ancho de banda de 200 KHz que corresponde a un bloque de recursos físicos en la transmisión GSM y LTE; existen tres posibles modos de funcionamiento (Lalle et al., 2019, p.3).

1. **Funcionamiento autónomo:** Es posible que NB IoT utilice una o más portadoras GSM existentes.
2. **Funcionamiento en banda de guarda:** Utilizando los bloques de recursos no utilizados en la banda de guarda del espectro LTE.
3. **Funcionamiento en banda:** Utilización de bloques de recursos dentro de una portadora LTE.

NB-IoT reutiliza varias funcionalidades de LTE y las adapta según las necesidades de las aplicaciones IoT, como por ejemplo reutilizar el sistema back-end de LTE para emitir mensajes válidos para todos los ED de una célula. La velocidad de datos es de 200 kbps y 20 kbps respectivamente para las comunicaciones UL y DL. La carga útil máxima de cada mensaje es de

1600 bytes; NB-IoT emplea modulación QPSK (Quadrature Phase Shift Keying), para la transmisión de datos UL, utiliza la modulación de acceso múltiple por división de frecuencia de portadora única (SC-FDMA) y FDMA ortogonal para la transmisión de datos DL. El núcleo de la arquitectura NB-IoT se basa en el sistema de paquetes evolucionados (EPS) y en dos procedimientos de optimización para el IoT celular (CIoT). Estos procedimientos son el EPS de CIoT en el plano de usuario y el EPS de CIoT en el plano de control.

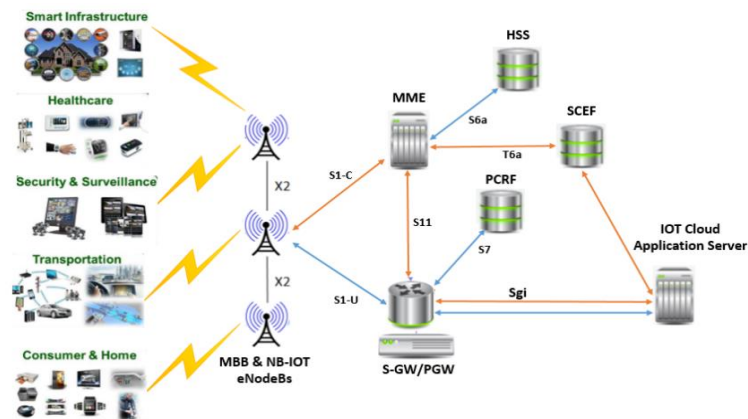


Ilustración 14-2: Arquitectura en redes NB-IoT

Fuente: Avsystem, 2018

Z-Wave

Es un protocolo de red inalámbrica de baja potencia que cumple con el estándar ITU-T G.9959, desarrollado principalmente para automatizaciones de hogares inteligentes. Puesto a que no es compatible con el Protocolo de Internet (IP), no puede interactuar con Internet ni con objetos llamados dispositivos comunes como teléfonos, tabletas, computadoras; sin embargo, mientras gestiona todos los dispositivos con su propio dispositivo de control, puede interactuar con dispositivos comunes e Internet gracias a este dispositivo, que también actúa como una gateway (Shea, 2018). Emplea cifrado AES y el modo operativo CBC-MAC se utilizan para autenticar y garantizar la integridad de la fuente de datos. Opera en la banda de frecuencias de 868 MHz para Europa y 908 MHz para Norteamérica y su velocidad de transmisión de datos varía entre 9,6 y 100 kbps y el rango de cobertura para la comunicación punto a punto varía entre 30 a 35 metros, y su propósito principal es transferir datos a uno o más nodos (Ayşe, 2019, p.3).

Todos los dispositivos se comunican únicamente conectándose entre sí a través de la red, sin necesidad de ningún otro intermediario para la transferencia de datos. Como se muestra en la ilustración 15-2, la pila de protocolos z-wave consta de 5 capas.

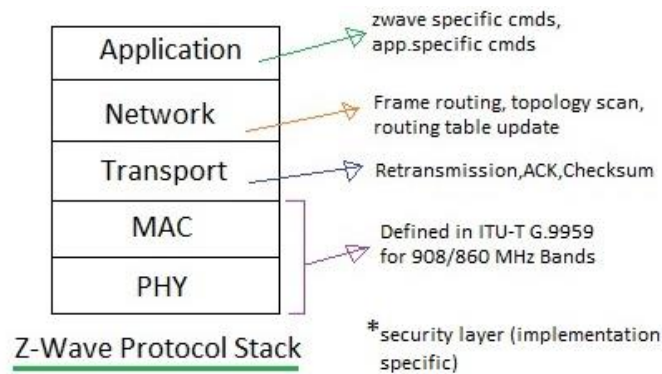


Ilustración 15-2: Stack de capas en Z-Wave

Fuente: RFWW, 2016

1. **Capa Física (PHY):** Se encarga de la modulación y de la asignación del canal de RF, así como de la adición del preámbulo en el transmisor y de la sincronización en el receptor mediante el preámbulo. La entrada para configurar la capa física de z-wave es la velocidad de datos que va desde 9,6 hasta 100 Kbps (Ayşe, 2019, p.2).
2. **Capa MAC:** Se encarga de HomeID y NodeID, controla el medio entre los nodos basándose en el algoritmo de evitación de colisiones y el algoritmo de backoff (RWW, 2016).
3. **Capa de transporte:** Se encarga de la retransmisión, el acuse de recibo de los paquetes, la activación de los nodos de la red de baja potencia y la autenticación del origen de los paquetes; esta capa consta de cuatro tipos de tramas básicas, las cuales se utilizan para transferir comandos en la red (Ayşe, 2019, p.2).
4. **Capa de red:** Enruta las tramas, escanea la topología y actualiza las tablas de enrutamiento. En esta capa, como se muestra en la ilustración 16-2, dos tipos de dispositivos llamados controladores y esclavos se comunican según la topología de malla. Los controladores en esta estructura se dividen en controladores principales y secundarios, el controlador principal crea el controlador secundario e inicia los comandos, mientras que los dispositivos esclavos ejecutan los comandos dados y transmiten los comandos a los otros nodos. La tabla de enrutamiento es construida por el controlador primario basándose en la información recibida de todos los nodos de la red Z-Wave (Ayşe, 2019).

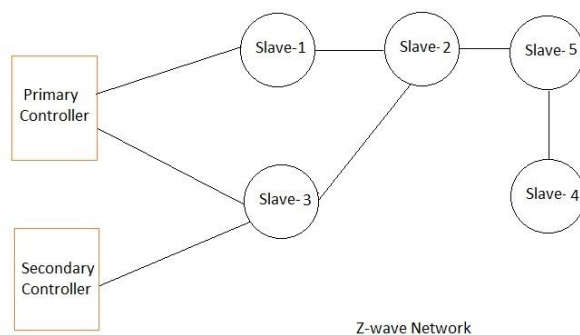


Ilustración 16-2: Topología de red malla en Z-Wave

Fuente: IoTPoint, 2019

5. **Capa de Aplicación:** Esta capa se encarga de la decodificación y ejecución de comandos en una red z-wave (IoTPoint, 2019).

6LoWPAN

El IPv6 sobre redes inalámbricas de baja potencia (6LoWPAN) es una red de área personal de corto alcance y baja potencia; que permite conectarse directamente a otra red IP sin entidades intermedias como las gateways y la conexión entre elementos 6LoWPAN se realiza a través de IEEE 802.15.4; esta tecnología puede soportar diferentes tipos de topologías, como la de malla y la de estrella, conceptos que se encuentran detallados en la RFC 4919 (Kushalnagar et al., 2007).

La tecnología 6LoWPAN, permite construir IoT trabajando conjuntamente con el estándar IEEE802.15.4 e IP; permitiendo el uso eficiente de IPv6 a través de redes inalámbricas de baja potencia y velocidad en dispositivos integrados simples a través de una capa de adaptación y la optimización de los protocolos relacionados (Kyun et al., 2012: p.4).

Puesto que la tecnología IP no está optimizada para PAN, puede adaptarse para hacer un uso perfecto de las WSN con internet gracias al uso de 6LoWPAN; las principales características de esta tecnología son: la compresión y encapsulación eficiente de cabeceras detallada en la RFC 4944, la autoconfiguración de la red mediante el descubrimiento de vecindad, el soporte de multidifusión *unicast* y *broadcast*, la fragmentación, el soporte de enrutamiento IP mediante RPL (Routing Protocol For Low Power Lossy Networks) y el soporte de malla de capa de enlace. La pila de protocolos 6LoWPAN se detalla en la ilustración 17-2.

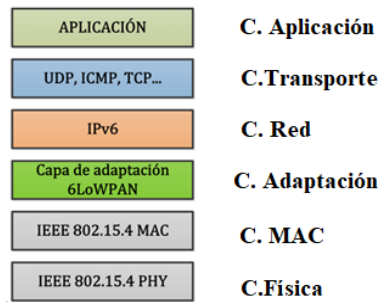


Ilustración 17-2: Pila de 6LoWPAN

Fuente: Menchaca, 2012, p.30

6LoWPAN es una especificación de protocolo que permite utilizar los estándares IPv6 en redes inalámbricas de baja potencia mediante el estándar IEEE802.15.4. En el presente estándar la cabecera de IPv6 se comprime a sólo unos pocos bytes mediante la introducción de una capa de adaptación que reside entre la red y la capa MAC/PHY, manteniendo las principales funcionalidades de IPv6. La transmisión de la Unidad de Transmisión Máxima (MTU) de IPv6 de 1280 bytes sobre IEEE 802.15.4 también es posible gracias a la fragmentación y el reensamblaje proporcionados por esta capa de adaptación, la cual está detallada en el RFC 4944 (Montenegro et al, 2007).

Tipos de dispositivos

La ilustración 18-2 muestra el diagrama de una red LoWPAN, donde el edge router encamina el tráfico dentro y fuera de la red, los routers actúan como intermediarios en la comunicación colaborando en la propagación del prefijo de red y los hosts representan a los dispositivos más simples de la red, ya que no requieren estar siempre activos como los routers por lo que es posible implementar medidas para regular su consumo energético (Menchaca, 2012, p.30).

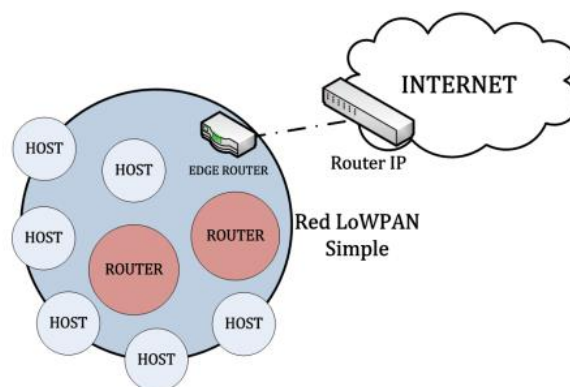


Ilustración 18-2: Esquema de red LoWPAN

Fuente: Menchaca, 2012, p.30

Las direcciones IPv6 de los nodos de la red tienen un tamaño fijo de 128 bits, contienen un bloque de 64 bits con el prefijo de red y otro de 64 bits con el identificador de interfaz (IID: Interface ID) (Menchaca, 2012, p.31).

Arquitectura

Una red LoWPAN es un conjunto de nodos 6LoWPAN que comparten el mismo prefijo de red, la ilustración 19-2 muestra las tres posibles arquitecturas de red; LoWPAN Simple es la red que dispone de un sólo edge router (Menchaca, 2012, p.30). LoWPAN Extendida es la red con varios edge routers, la cual es muy común en aplicaciones de movilidad o con un gran número de nodos, que adicionalmente pueden crear una subred con otros dispositivos IP y finalmente LoWPAN ad-hoc que representa una red que no está conectada a Internet donde uno de los routers se configura para actuar como edge router (Prisantama et al., 2016: p.3).

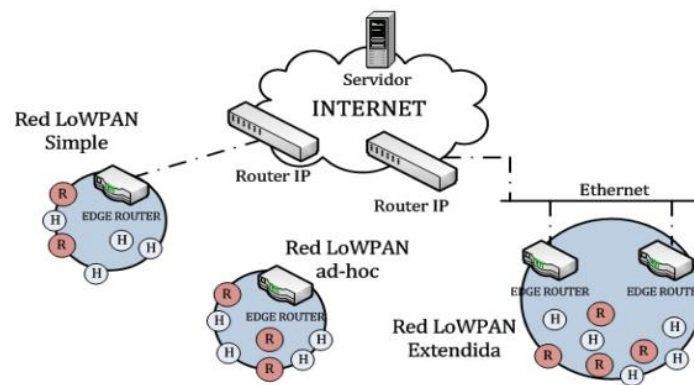


Ilustración 19-2: Arquitectura 6LoWPAN

Fuente: Prisantama et al., 2016: p.3

2G

Está basado en la tecnología GSM (Global System for Mobile Communication). El sistema 2G utiliza una combinación de TDMA (acceso múltiple por división de tiempo) y FDMA (acceso múltiple por división de frecuencia). De este modo se pueden conectar más usuarios a la vez en una banda de frecuencias determinada. Como se muestra en la ilustración 20-2, una ranura de frecuencia específica se divide en ranuras de tiempo, por lo que varios usuarios pueden utilizar una ranura de frecuencia específica. El sistema GSM utiliza dos bandas de 25MHz, 890-915MHz y 935-960MHz para las bandas de transmisión y recepción del sistema móvil respectivamente, obteniendo una velocidad aproximada de unos 14,4 Kbps con una modulación por desplazamiento mínimo gaussiano (GMSK) (Khare et al, 2013: p.1).

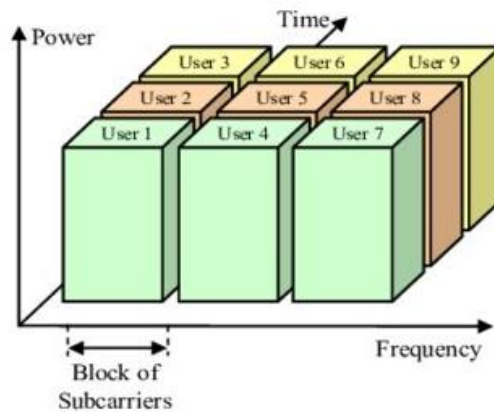


Ilustración 20-2: FDMA con TDMA en GSM

Fuente: Iliev et al., 2009: p.4

La red básica utilizada en 2G es la RTPC (Red Telefónica Pública Conmutada), GSM emplea la conmutación de circuitos. Al aumentar la necesidad de enviar datos a través de la interfaz aérea, el GPRS (General Packet Radio Service) se unió a la red GSM existente. Gracias a ello, se puede alcanzar una velocidad óptima de hasta 150 Kbps. Sin embargo, cuando surgió la necesidad de aumentar la velocidad de datos, se introdujo EDGE (Enhanced Data GSM Environment) diseñado para aplicaciones multimedia como streaming y obteniendo velocidades de hasta 384 kbps, con una modulación por desplazamiento de ocho fases (8PSK) (Gu y Peng, 2010, p.2).

3G

El sistema 3G emplea CDMA (acceso múltiple por división de código) y WCDMA (acceso múltiple por división de código de banda ancha), la cual es una técnica en la que se asigna un código único a cada usuario que utiliza el canal en ese momento. Tras asignar un código único, se utiliza de forma eficiente todo el ancho de banda disponible en él; gracias a ello, un gran número de usuarios puede utilizar el canal al mismo tiempo en comparación con TDMA y FDMA (Khare et al., 2013, p.3).

Como se muestra en la ilustración 21-2, se asigna un código único a cada usuario, por lo que se puede formar un número N de canales a la vez. 3G utiliza un espectro de frecuencias de 15 MHz a 20 MHz y la banda de frecuencias para 3G va de 1800 MHz a 2500 MHz. En el sistema 3G básico se alcanza una velocidad máxima de unos 2 Mbps (ARCOTEL, 2018, p.7). WCDMA utiliza una frecuencia de portadora mucho mayor, por lo que se puede dar cabida a un mayor número de usuarios en comparación con CDMA; la red básica utilizada en los sistemas 3G es una combinación de conmutación de circuitos y conmutación de paquetes (RantCell, 2022).

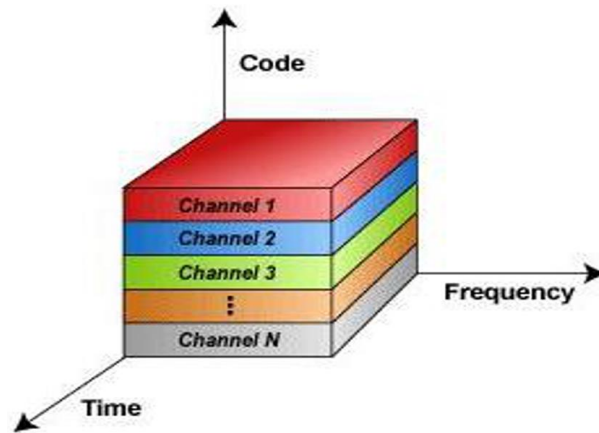


Ilustración 21-2: Modulación CDMA en 3G

Fuente: Soffar, 2017.

Para aumentar aún más la velocidad de los datos, se introdujo HSPA y HSPA+ (High Speed Packet Access), que gracias a este último las redes pueden ampliarse para funcionar a velocidades de banda ancha introduciendo por primera vez el concepto de MIMO (Multiple Input Multiple Output). Gracias a él, las velocidades de transmisión de datos pueden alcanzar los 42 Mbps. HSPA y HSPA+ pueden considerarse 3,5G y 3,75G respectivamente. La técnica de modulación utilizada en HSPA+ es QAM de 64 bits (ARCOTEL, 2018, p.7).

Como se muestra en la ilustración 22-2, MIMO es un método en el que se utiliza el concepto de -propagación multitrayecto para mejorar el enlace de radio. La misma señal se recibe varias veces en el lado del receptor. De este modo, se reduce la probabilidad de error y se mejora el rendimiento general (ELPROCUS, 2021).

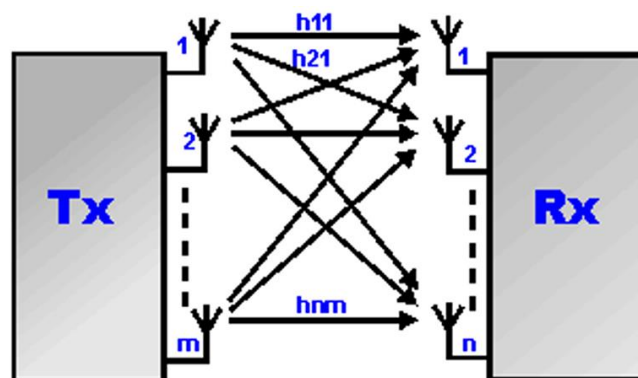


Ilustración 22-2: Esquema de sistemas de comunicaciones MIMO

Fuente: ELPROCUS, 2021

Otra ventaja del sistema 3G es la desconexión. En este caso, el equipo del usuario se conecta a dos torres simultáneamente, por lo que no se produce una caída de la llamada durante el hand-off.

4G

LTE (Long Term Evolution) es un estándar de comunicación móvil 4G basado en las tecnologías GSM/EDGE y UMTS/HSPA. LTE utiliza CDMA multiportadora u OFDM (Orthogonal Frequency Division Multiplexing) (Khare et al., 2013: p.4). En OFDM, el flujo de modulación de alta velocidad de datos se divide y se coloca en muchas subportadoras de banda estrecha lentamente moduladas y espaciadas, tal y como se muestra en la ilustración 23-2.

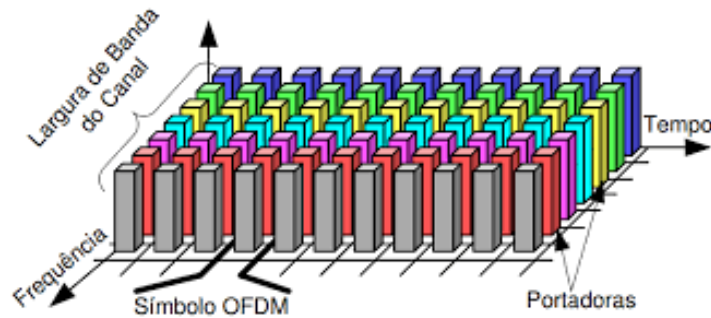


Ilustración 23-2: Espectro OFDM

Fuente: Orosco et al., 2012: p.4

La banda de frecuencias utilizada en 4G va de 2000 MHz a 8000 MHz, con un ancho de banda de entre de 5Mhz a 20 MHz. En los sistemas LTE se alcanza una velocidad máxima de bajada de unos 100 Mbps y de subida de unos 50 Mbps. Gracias a esta alta velocidad de datos, puede soportar aplicaciones que requieren mucho ancho de banda, como los juegos en línea, la transmisión en directo de vídeo de alta definición o la voz sobre IP (ARCOTEL, 2018, p.8).

La tecnología 4G está basada en una red IP, la cual posee latencias muy bajas, cuenta con un canal más amplio y una agregación de portadoras de hasta 100Mhz. Los dos modos habituales de LTE son LTE FDD y LTE TDD.

Bluetooth

Está basado en el estándar IEEE 802.15.1, definiéndola como una tecnología inalámbrica de comunicación de bajo costo y consumo energético ideal para la transmisión de datos entre dispositivos móviles en un rango corto de 8 a 10 metros, llamada Red de Área Personal (PAN) que opera en la banda de 2.4 GHz. Soporta dos tipos de topología: piconet y scatternet. Sus velocidades varían según las versiones que van 1 Mbps a 32 Mbps; Bluetooth Special Interest Group (SIG) supervisa las especificaciones de Bluetooth (IONOS, 2015).

La versión de ultra bajo consumo y bajo costo de este estándar se denomina Bluetooth Low Energy (*BLE*) que surgió en la especificación Bluetooth 4.0, que posee dos actualizaciones (4.1 y

4.2) y ha sido diseñada para dispositivos de baja potencia con recursos. Existe una acogida masiva de esta tecnología, debido a que la mayoría de los teléfonos inteligentes de gama alta son compatibles con BLE, facilitando una integración rápida y sin infraestructura de los dispositivos BLE con Internet; convirtiéndola en una tecnología ideal para IoT (Bluetoooh, 2020). En su última versión 5.0, el SIG de Bluetooth adapto más a la tecnología para un ambiente IoT, cuadruplicando su cobertura, duplicando la velocidad y aumentando la capacidad de transmisión de datos con el mismo bajo consumo de energía (Rodriguez, 2022).

Una vez detalladas las tecnologías a considerar para la capa física, la tabla 4-2 presenta los criterios cuantizados de la escala de Likert para evaluar el consumo de potencia de transmisión de cada tecnología; adicionalmente la tabla 5-2 presenta una comparativa de las diferentes tecnologías empleadas en la capa de percepción, con el fin de determinar la más adecuada para el sistema propuesto al inicio de la investigación.

Tabla4-2: Escala de Likert para evaluación del consumo de potencia de las tecnologías IoT

Nivel	Descripción
1	Muy bajo – hasta 10mW
2	Bajo – hasta 100 mW
3	Medio- hasta 1000 mW
4	Alto – mayor a 1000 mW

Realizado por: López, Diego; 2022

Tabla5-2: Comparativa de tecnologías disponibles en la capa de percepción

Características	Zigbee	6LoWPAN	Bluetooth	Wi-Fi	SigFox	Z-Wave	LoRa	Celular	NB-IoT
Estándar	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.11n	SigFox	Z-Wave	LoRaW AN	(2G) GSM/GPRS/ESDGE (3G) UMTS/HS PA (4G) LTE	NB IoT
Bandas de frecuencia	868 MHz (Europa) 915 MHz (USA) 2.4	868 MHz (Europa) 915 MHz (USA) 2.4 GHz (Global)	2.4 GHz	2.4 GHz 5 GHz	868 MHz (Europa) 915 MHz (USA)	868 MHz (Europa) 908 MHz (USA)	433/ 868 / 915 MHz	850-190 MHz (GSM)	B2 (1900 MHz), B3 (1800 MHz), B5 (850

	GHz (Global)								MHz) y B28 (700 MHz) en América Latina
Red	WPA N	WPAN	WPAN	WLAN	LPWA N	WPA N	WWAN	WWAN	LPWAN
Topología	árbol, malla, estrella	malla y estrella	estrella	estrella	estrella	malla	estrella	NA	estrella
Ancho de banda	2 MHz	5MHz - Banda 2,4 GHz 2 MHz- Banda de 915 MHz 600 KHz- Banda 868 MHz	1 MHz	2 canales de: 20 MHz y 40 MHz	100 Hz	400 KHz	125-500 KHz	0,2 MHz- GSM 25 MHz - 3G 100 MHz - 4G	180 KHz
Velocidad de datos	250 Kbps	250 Kbps	1 Mbps	MIMO 2x2 canal 20 MHz: 130-144 Mbps canal 40 MHz: 270-300 Mbps	100 bps (UL) 600 bps (DL)	9,6 a 100 Kbps	0,3 a 50 Kbps	150 Kbps (GSM) 384 Kbps (EDGE) 2 Mbps (3G) 42 Mbps (HSPA) 100Mbps (4G)	200 Kbps (UL) 20 kbps (DL)
Alcance	corto 10-100 m	corto 10 -100 m	corto 15 - 30m	2.4 GHz 45 m interiores 90 m exteriores 5 GHz 15 m interiores 30	largo 10 Km (urbanos) 50 Km (rural)	corto 30 m	urbanos 2 - 5 Km suburbanos hasta 15 Km rurales hasta 45 Km	5 a 100 Km	urbano 1 Km rural 10 Km

				m exterior es					
Consumo de potencia	2	2	1	3	3	3	3	4	3

Realizado por: López, Diego; 2022

2.1.2.2. Protocolos en la capa de red

Los protocolos en esta capa tienen el propósito de ayudar a transportar los datos proporcionados por el nivel de percepción hacia la capa de aplicación. En la cual solo interviene un único protocolo el cual es el Protocolo de Internet (IP), el cual se detalla a continuación.

Protocolo de Internet

El protocolo de Internet (IP), publicado en 1974 por la organización IEEE y especificado como estándar en el RFC791; representa al componente fundamental de la familia de estándares TCP/IP en los que se basa el funcionamiento de Internet, que junto al protocolo de control de transmisiones (TCP), resulta imprescindible en redes informáticas para el intercambio de mensajes. Este protocolo permite el direccionamiento y el desglosamiento de paquetes de información en las redes informáticas. Se define una estructura de paquetes que agrupa los datos a enviar por parte del protocolo IP, describiendo el origen y destino de la información y excluyéndolas de los datos útiles, a este formato de paquetes se los conoce como datagramas IP (IONOS, 2018).

Al no estar orientado a la conexión, el protocolo conlleva a que los paquetes viajen a través de la red por diversas rutas hasta llegar a su destino, pero no aseguran su recepción; implicando que, si no es recibido, el paquete permanecerá en la red por un tiempo determinado. La fragmentación de los paquetes solo ocurre en caso de ser necesario, estableciendo un tamaño máximo del datagrama de 65365 bytes de longitud (KeepCoding, 2022).

Dirección IP

Son el identificador único que permite enviar información entre dispositivos en una red: contienen información de ubicación y permiten a los dispositivos ser accesibles para la

comunicación. Internet necesita una forma de diferenciar entre diferentes computadoras, enrutadores y sitios web. Existen 2 versiones de las direcciones IP: IPv4 e IPv6 (JUNIPER, 2015).

IPv4

Es la primera versión de las direcciones IP y la que más se emplea en la actualidad, debido a la continua demanda de direcciones IP por parte de los usuarios la vuelven limitada a solo 4.3 millones de direcciones. Una dirección IPv4 es un número de 32 bits formado por cuatro octetos (números de 8 bits) en una notación decimal, separados cada uno por un punto, y teniendo 256 valores posibles en cada octeto que van del 0 a 255 (IBM, 2019).

Existen cinco clases de direcciones: A, B, C, D y E. Las clases A, B y C tienen una longitud de bits diferente para direccionar el host de la red. Las direcciones de clase D están reservadas para fines militares, mientras que las de clase E se reservan para un uso futuro (JUNIPER, 2015).

En una red IPv4, los hosts pueden de tres distintas formas:

- **Unicast:** se envía un paquete de un host a un host individual.
- **Broadcast:** se envía un paquete de un host a todos los hosts en la red.
- **Multicast:** se envía un paquete de un host a un grupo seleccionado de hosts, posiblemente en redes distintas

IPv6

Es la nueva versión de las direcciones IP, las cuales poseen una longitud de 128 bits, ofreciendo un máximo de 3.4×10^{38} direcciones. Están especificadas en el RFC 2462, su formato consta de 8 hexetos, de 2 bytes cada uno y cada hexeto es separado por dos puntos (IBM, 2019).

Existen tres tipos de direcciones IPv6: unicast, multicast y anycast. Los conceptos se mantienen para los dos primeros tipos, la dirección anycast es muy similar a la dirección multicast, pero los paquetes se entregarán solo a un host aleatorio, en lugar de a todo el grupo (IONOS, 2018).

A continuación, se muestran los formatos de cabeceras de IPv4 e IPv6.

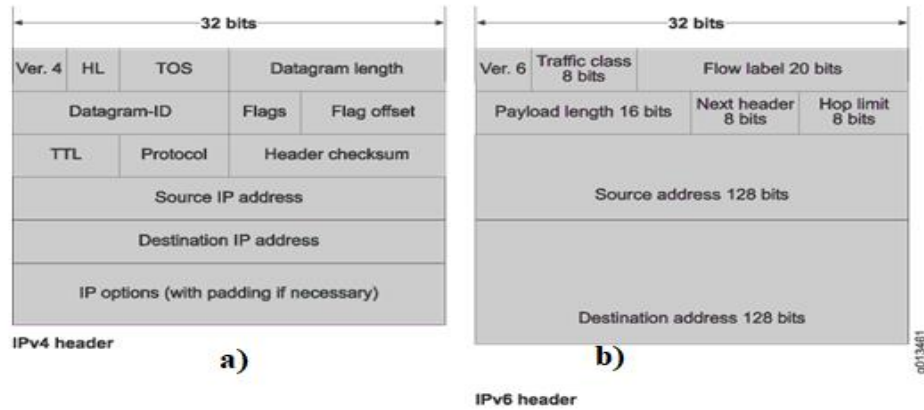


Ilustración 24-2: Cabeceras de las direcciones IP. a) Cabecera IPv4. b) Cabecera IPv6.

Fuente: JUNIPER, 2015

La tabla 6-2 presenta una comparativa en donde se detallan los campos en común y los campos nuevos de las diferentes versiones de las direcciones del protocolo.

Tabla6-2: Comparativa entre cabeceras IPv4 e IPv6

Campo IPv4	Campo IPv6	Función
Campos con la misma funcionalidad y el mismo nombre en IPv4 e IPv6		
Versión		Indica la versión del protocolo IP en uso. Valor de 4 IPv4 y 6 para IPv6
Dirección de origen		Identificador de capa de red del emisor del paquete. 32 bits en IPv4 y 128 bits en IPv6
Dirección de destino		Identificador de capa de red del receptor del paquete. 32 bits en IPv4 y 128 bits en IPv6.
Campos con la misma funcionalidad, pero diferente nombre		
Tipo de servicio	Clase de tráfico	Empleado en la clasificación y señalización del tráfico. Ambos protocolos usan la técnica de Servicios Diferenciados (DSCP) de 6 bits
Largo total	Longitud de la carga útil	Especifica longitud del paquete IP. En IPv4 la longitud incluye tanto el encabezado IP como los datos. En IPv6, la longitud incluye los datos más los encabezados de extensión, pero no incluye el encabezado IP principal.
Tiempo para vivir (TTL)	Límite de salto	Ambos campos aseguran que los

		paquetes no circulen por la red indefinidamente.
Protocolo	Siguiente encabezado	Define el protocolo que se transporta en la porción de carga útil. En IPv6, también podría indicar la existencia de un encabezado de extensión.
Campos existentes en IPv4 y eliminados en IPv6		
Longitud del encabezado de Internet		En IPv4, este campo se utiliza en los casos en que el encabezado tiene una longitud variable de 20 a 60 bytes, pero es 20 bytes por defecto. No es necesario en IPv6, porque el encabezado posee una longitud fija de 40 bytes.
Identificación, banderas, desplazamiento de fragmentos		En IPv4, estos campos se utilizan al realizar la fragmentación. En IPv6, solo la fuente del paquete realiza la fragmentación mediante el encabezado de extensión de fragmentación.
Suma de comprobación del encabezado		IPv6 clasifica a este campo como redundante y ya no es necesario porque hay campos de suma de comprobación en los protocolos de capa superior.
Opciones		IPv6 usa los encabezados de extensión en vez de el parámetro opciones, por lo que este campo no es necesario.
Relleno		No es necesario en IPv6 debido a que posee un tamaño fijo.
Campos nuevos en IPv6 que no existen en IPv4		
	Etiqueta de flujo	Identifica que un paquete es parte de una secuencia y debe manejarse de la misma manera que todo el flujo de tráfico.

Realizado por: López, Diego; 2022

Sin duda IPv6 se presenta como una opción más eficiente en cuanto a escalabilidad, y formato de cabecera. En Ecuador la versión de direcciones IP predominantes es IPv4 debido a que la nueva versión IPv6 se encuentra en inicios de adopción en los sistemas informáticos del país, por lo cual es la versión del protocolo a usar en el proyecto.

Una vez definido el método de direccionamiento, se procederá comprender los métodos de garantía de entrega de datos a través de la red, segmentación de paquetes de información y control del reensamblaje de estos, a través de los protocolos TCP y UDP.

TCP

El Protocolo de control de transmisión, define cómo establecer y mantener una comunicación de red a través de la cual los programas de aplicación pueden intercambiar datos, brindando soporte a varios protocolos de la capa de aplicación como: HTTPS, POP3, SMTP y FTP. Su cabecera consta de 20 bytes, en la cual se encuentran campos como: puerto de origen, puerto de destino, número de secuencia, número de ACK, y los diferentes FLAGS de TCP como SYN, ACK, RST, FIN, URG y otros (De Luz, 2021).

Es un protocolo orientado a conexión, estableciendo una conexión previa con dicho servidor para la comunicación cliente-servidor, llamada 3 way handshake.

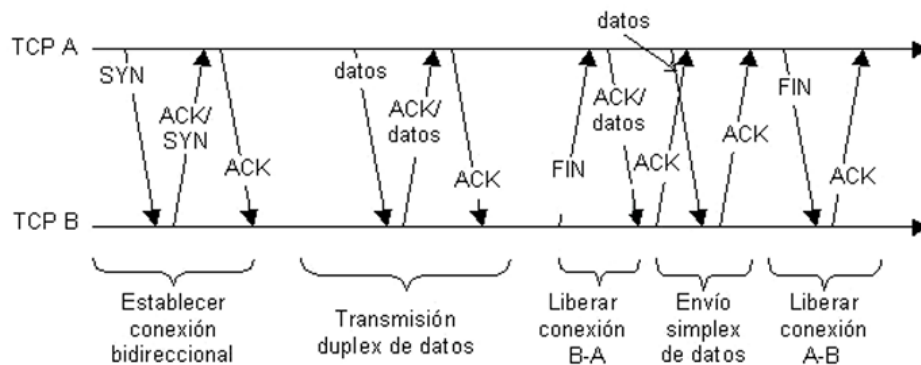


Ilustración 25-2: Proceso de intercambio de paquetes en una conexión TCP

Fuente: Baeza, 2009, p.4

La ilustración 25-2 detalla el proceso de intercambio de mensajes en una conexión TCP. El cliente A transmite el primer paquete con el bit SYN activado hacia un servidor B solicitando el inicio de una conexión, el servidor B acepta la conexión enviando un paquete con el bit ACK activado y el cliente responde al servidor enviando un paquete con el bit ACK activo; o también servidor puede rechazar la conexión enviando un paquete con el bit RST activado. Si el servidor acepta la solicitud de conexión (Vienažindyte, 2019).

Una vez establecida la conexión se pueden intercambiar datos con paquetes TCP, ya que se usa un sistema de envío continuo, un equipo puede activar el bit de ACK de un paquete TCP mediante el cual envía datos para confirmar al mismo tiempo los datos previamente recibidos. La desconexión puede ser iniciada por cualquiera de los equipos implicados, enviando un paquete TCP con el bit FIN activado y el otro equipo responde con un paquete TCP con ACK activado (Vienažindyte, 2019).

UDP

El Protocolo de Datagramas de Usuario representa a otro protocolo fundamental en Internet, permite el envío de datagramas sin necesidad de establecer previamente una sincronización entre el origen y el destino como en TCP, tan solo es necesario tener abierto un puerto en el destino para que acepte los datagramas del origen; brinda soporte a protocolos como DNS, DHCP, VoIP, entre otros (Diffen, 2015).

La cabecera de UDP consta de un tamaño de 8 bytes en cada datagrama, está conformada por: puerto de origen y puerto de destino de la conexión, longitud del datagrama y el checksum de dicho datagrama para comprobar que no tiene errores ni la cabecera ni los datos del datagrama (Baeza, 2009, p.6).

Una vez comprendidos los protocolos de transmisión de paquetes, se concluye que el protocolo más adecuado para el sistema propuesto es TCP, debido a que presenta compatibilidad con varios protocolos de la capa de aplicación que se mencionarán en la siguiente sección, además de la necesidad de registrar todos los datos posibles del consumo realizado en la vivienda, para poder estimar un valor a pagar con mayor exactitud en su costo y también consumo diario o mensual.

2.1.2.3. Protocolos en la capa de aplicación

Como se mencionó, en esta capa se define los servicios y funciones que proporciona la aplicación IoT implementada, tales como: hogares inteligentes, ciudades inteligentes, salud inteligente a los clientes. Un manejo competente de ancho de banda y consumo energético para la transmisión de información a través de protocolos livianos, seguros y eficientes, son requeridos por los dispositivos IoT. A continuación, se detallan los diferentes protocolos involucrados en esta capa.

MQTT

Es un protocolo *Push* basado en publicación/suscripción, creado por IBM en 1999 fue planeado para enviar datos con precisión a través de redes con baja velocidad de transmisión y ancho de banda. En MQTT Para la comunicación, se intercambia una serie de mensajes de control de manera específica. Hay catorce paquetes de control. Cada uno de ellos contiene tres partes, como se ilustra en la ilustración 26-2.

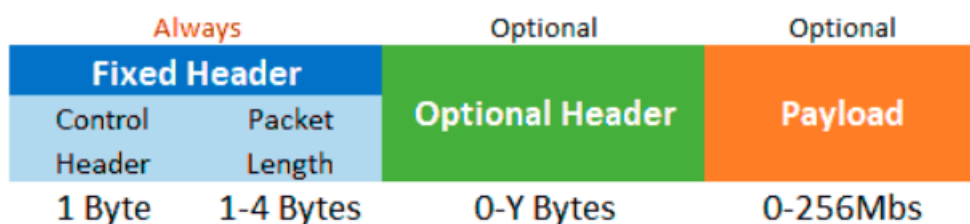


Ilustración 26-2: Estructura de un mensaje MQTT

Fuente: Llamas, 2019

- **Cabecera fija.** Posee una longitud de 2 a 5 bytes, el primer byte representa el tipo de mensaje y el segundo byte representa la longitud restante donde los 7 primeros bits informan el número de bytes y el último bit muestra la continuidad del paquete.
- **Cabecera variable.** Es un campo opcional que brinda información adicional necesaria para ciertos mensajes.
- **Carga útil.** Representa al contenido real del mensaje con un máximo de 256 Mb.

MQTT envía los diferentes tipos de mensaje y códigos de control, los cuales se muestran en la tabla 7-2.

Tabla 7-2: Tipos de mensaje en MQTT

Tipo de mensaje	Código
CONNECT	0X10
CONNACK	0X20
PUBLISH	0X30
PUBACK	0X40
PUBREC	0X50
PUBREL	0X60
PUBCOMP	0X70
SUSBSRBE	0X80
SUBACK	0X90
UNSUBSCRIBE	0XA0
UNSUBACK	0XB0
PINGREQ	0XC=
PINGRESP	0XD0
DISCONNECT	0XE0

Fuente: Oliveira, 2017

Funcionamiento

Los clientes se conectan a un servidor denominado broker, el cual acepta mensajes publicados por clientes y los difunde entre los clientes suscritos. Los mensajes se disponen en topics organizados jerárquicamente para filtrar los mensajes que son enviados a cada cliente. Un cliente puede publicar un mensaje en un determinado topic, así mismo otros clientes pueden suscribirse a este topic y el broker se encargará de entregar los mensajes suscritos (Makwana y Dipa, 2017, p.2).

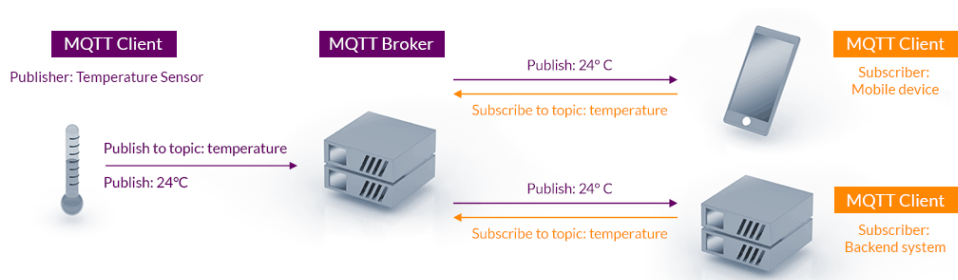


Ilustración 27-2: Funcionamiento del broker en MQTT

Fuente: Downey, 2020

La ilustración 28-2 muestra el proceso de funcionamiento de MQTT a través del flujo de mensajes, donde los clientes inician una conexión TCP/IP con el broker, el cual mantiene un registro de los clientes conectados. La conexión permanece abierta hasta ser finalizada por el cliente. Cuando funciona sobre TLS se emplea el puerto 1883 y el 8883 por defecto (Makwana y Dipa, 2017, p.3).

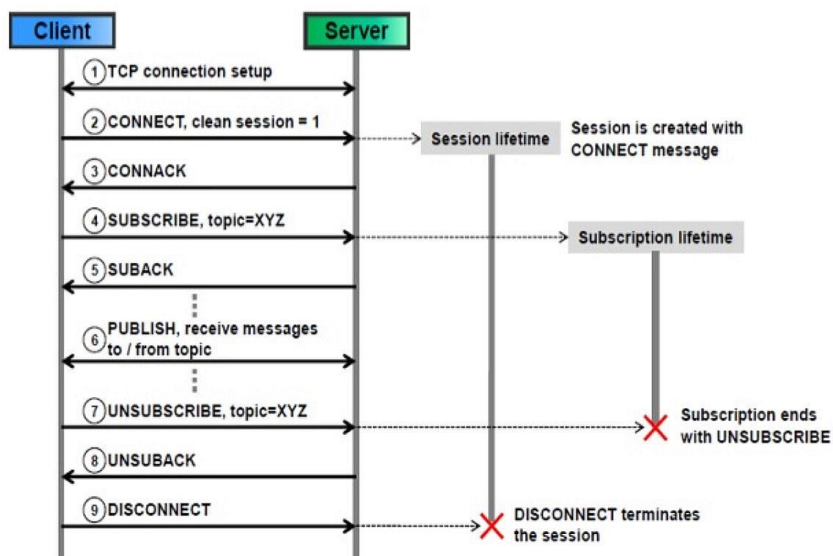


Ilustración 28-2: Flujo de mensajes en el funcionamiento de MQTT

Fuente: DEVELOPEDIA, 2021

Para el envío de mensajes por parte del cliente, se hace uso de los mensajes tipo publish, los cuales contienen el topic y el payload. Para la vinculación y desvinculación al topic se emplean mensajes tipo SUBSCRIBE y UNSUSCRIBE, que el servidor responde con SUBACK y UNSUBACK. Por otro lado, para asegurar que la conexión se mantenga activa los clientes envían frecuentemente mensajes tipo PINGREQ, el cual es respondido por el servidor a través de un mensaje PINGRESP y por último el cliente finaliza la sesión en el servidor enviando un mensaje tipo DISCONNECT (Makwana y Dipa, 2017, p.3).

Calidad de Servicio (QoS)

Este protocolo dispone de un mecanismo de calidad de servicio, el cual proporciona robustez ante fallos en el envío de mensajes al cliente (Llamas, 2019). Se establecen 3 niveles de calidad de servicio que son:

- **QoS 0 (Como máximo una vez):** El mensaje es enviado sol una vez y en caso de fallo no se garantiza su entrega.
- **QoS 1 (Al menos una vez):** El mensaje se envía hasta que se garantiza la entrega. En caso de fallo, el suscriptor puede recibir algún mensaje duplicado.
- **QoS 2 (Exactamente una vez):** Cada mensaje es enviado al suscriptor una única vez, garantizando su entrega.

HTTP

El Protocolo de Transferencia de Hipertexto tiene la finalidad de posibilitar la transmisión de información entre un cliente y un servidor web, empleando el esquema petición-respuesta para ayudar a que el proceso sea más eficiente. El cliente envía una petición con un determinado formato al servidor y este le devuelve una respuesta a través del navegador web; es un protocolo que carece de estado, es decir que carece de registro de visitas (IONOS, 2013).

Este protocolo está íntimamente relacionado con la dirección URL (Uniform Resource Locator) que localiza al recurso en Internet y HTML (Hypertext Markup Language) que define la estructura de las páginas web (Bernes y Fielding, 1996).

Funcionamiento

Al ser un protocolo basado en el modelo cliente-servidor, el cliente envía mensajes y peticiones a través de un agente de usuario, el cual puede ser navegador web o cualquier otro programa; un servidor recepta los mensajes para gestionarlos y enviar una respuesta. La arquitectura HTTP se compone de tres elementos:

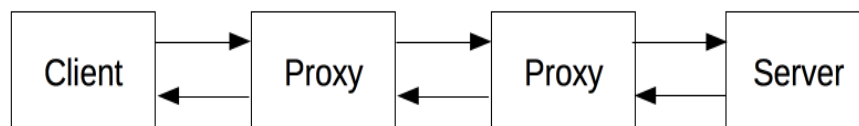


Ilustración 29-2: Arquitectura HTTP

Fuente: Developer, 2015

- **Agente del usuario:** Es la herramienta que representa al usuario, ya sea mediante un navegador web o un bot. El navegador siempre iniciará una comunicación a través de la petición y el servidor procesará la solicitud para emitir una respuesta con hojas de estilo, scripts, imágenes, videos, etc.

- **Proxy:** Actúan como intermediarios en la comunicación entre el agente del usuario y el servidor, realizando filtrado, autenticación, registro de eventos, balanceo de peticiones, etc.
- **Servidor:** Entrega los datos solicitados por el cliente, el cual está formado por varios elementos u otros programas que gestionan varios computadores.

El protocolo HTML en principio se creó para para solicitar documentos HTML a un servidor web, pero actualmente se explota este protocolo para funciones como:

- Solicitar cualquier tipo de archivo en la web como: texto, vídeo, código de programación, etc.
- Carga de archivos y actualizaciones de servidores lejanos.
- Controlar servicios web por parte de una API Rest.
- En comunicaciones M2M como protocolo de comunicación entre servicios web.
- Los reproductores multimedia.
- Las operaciones de acceso a bases de datos en la web.

Versiones de HTTP

- a) **HTTP1:** Descrita en el RFC1945, incluía una cabecera que especificaba la solicitud del cliente y la respuesta del servidor; adicionalmente se incorporó la cabecera Content-Type la cual permitía transferir archivos diferentes a HTML. Esta versión poseía tres características principales: conexión efímera, sin estado e independiente del tipo de archivo.
- b) **HTTP1.1:** Descrita en el RFC2068, considerada como el primer estándar oficial es la versión que se emplea en todos sistemas hoy en día, presenta cambios significativos con respecto a su predecesora.
 - Keepalive: Añadiendo a la cabecera el comando keepalive, el cliente puede optar a mantener la conexión más allá de la solicitud
 - HTTP pipelining: Permite enviar al cliente la siguiente solicitud sin necesidad de esperar a recibir la respuesta de la primera.
 - En chats, el navegador puede actualizar la ventana usando el tipo MIME multipart/replace.
 - También se pueden transferir datos del cliente al servidor.
 - Con el nuevo método TRACE, puede rastrearse la ruta entre el cliente y el servidor web.
 - Caché: existen nuevos mecanismos para guardar contenido temporalmente.

- **Host:** La especificación host en el header permite que la solicitud HTTP funcione también si hay varios dominios alojados en la misma dirección IP, lo cual es el caso actualmente en la mayoría de las páginas webs debido al shared hosting
- c) **HTTP2:** HTTP. HTTP/1.1 está pensado para procesar solicitudes una tras otra en una misma conexión, de manera que cuanto más compleja sea una página web, más tardará en cargarse y mostrarse. Por esta razón, Google desarrolló un nuevo y experimental protocolo, el SPDY o Speedy, que despertó un gran interés entre los desarrolladores y permitió que en 2015 se publicara la versión HTTP/2 del protocolo (Developer, 2015). Este estándar incluye, entre otras, las siguientes mejoras, que tienen como objetivo acelerar la carga de las páginas web:
- **Datos binarios.** El protocolo trabaja con datos binarios en lugar de archivos de texto.
 - **Multiplex.** El cliente y el servidor pueden enviar y procesar varias solicitudes HTTP simultáneamente.
 - **Compresión.** Los *headers* se comprimen, puesto que suelen ser idénticos en muchas solicitudes HTTP y, de este modo, se evitan las redundancias.
 - **Server Push.** Cuando el servidor prevé qué datos le pedirá el cliente, los envía directamente a la caché del cliente, sin esperar a recibir la solicitud HTTP correspondiente.

CoAP

El protocolo de aplicación restringida se basa en el modelo cliente-servidor, permitiendo a los usuarios realizar solicitudes de transferencias web según las necesidades del momento. Por otra parte, también permite a los servidores de apoyo responder a las solicitudes que llegan (Grafika, 2014, p.3). Los nodos de los dispositivos en el ecosistema IoT están habilitados para interactuar únicamente a través de CoAP.

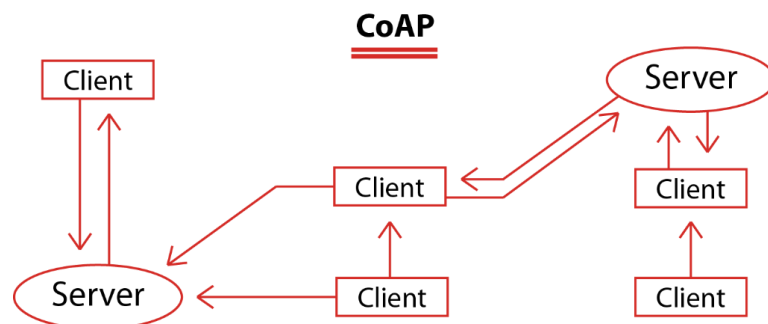


Ilustración 30-2: Esquema de funcionamiento de CoAP

Fuente: Al-Masri et al. ,2020 : p.3

CoAP sigue el mismo principio de funcionamiento que HTTP. Sin embargo, consigue su funcionalidad a través de transacciones asíncronas mediante el protocolo UDP. Al implementar REST posee acciones POST, GET, PUT y DELETE. Esta es la razón por la que la seguridad de la API es de mayor grado mientras CoAP está activo, ya que es un protocolo con certificación RPK y PSK (Al-Masri et al. ,2020 : p.3). CoAP es compatible con 4 tipos de mensajes:

Mensajes de acuse de recibo (ACK): Son los que confirman la conclusión o el fracaso de un evento.

- Mensajes confirmables (CON): Son los mensajes que se reenvían en el tiempo de espera hasta que no llega la confirmación de envío con éxito.
- Mensajes de reinicio (RST): Indica que se recibió un mensaje específico, pero falta información contexto para procesarlo correctamente.
- Mensajes non-confirmable (NON): No requieren una confirmación por parte del receptor, y la información solicitada será enviada en otro mensaje del tipo NON. Estos mensajes son poco fiables y se utilizan para enviar información no crítica.

Arquitectura

La WWW y el ecosistema de restricciones son los dos elementos fundamentales de la arquitectura del protocolo CoAP. Aquí, el servidor supervisa y ayuda en la comunicación que se produce utilizando CoAP y HTTP, mientras que los dispositivos proxy cubren el vacío existente para estos 2 ecosistemas, haciendo que la comunicación sea más fluida (Al-Masri et al. ,2020 : p.3).

CoAP permite a los clientes HTTP (también llamados clientes CoAP) hablar o intercambiar datos/información entre sí dentro de las limitaciones de recursos (Shelby et al., 2012). Su arquitectura se compone de los siguientes elementos:

- Puntos finales: Son los nodos de los que el anfitrión tiene conocimiento.
- Cliente: Envía solicitudes y responde a las solicitudes entrantes.
- Servidor: Recibe y reenvía las solicitudes. También recibe y reenvía los mensajes recibidos en respuesta a las solicitudes que había procesado.
- Remitente: Crea y envía el mensaje original.
- Destinatario: Recibe la información enviada por el cliente o reenviada por el servidor.

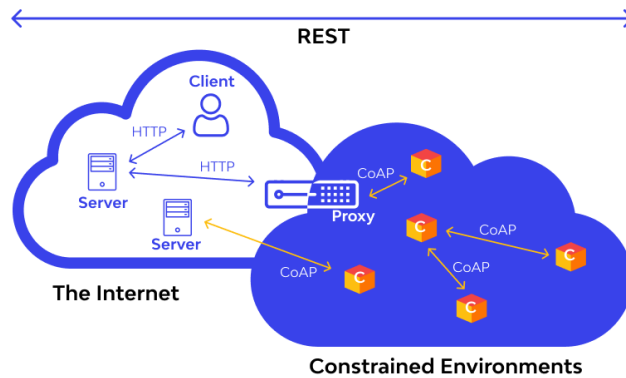


Ilustración 31-2: Arquitectura CoAP

Fuente: Bassi, 2021

Funcionalidad

La función principal de CoAP es actuar como HTTP siempre que los dispositivos restringidos formen parte de la comunicación. Al mismo tiempo que llena el vacío de HTTP, permite que dispositivos como actuadores y sensores interactúen a través de Internet (Bassi, 2021).

Los dispositivos, involucrados en el proceso, son administrados y controlados considerando los datos como un componente del sistema. El protocolo CoAP puede realizar sus funciones en un entorno con un ancho de banda reducido y una congestión extrema, ya que consume menos energía y ancho de banda de red (Rajeesh y Mohan, 2020, p.2).

Características

Las principales características del protocolo CoAP son:

- Protocolo web que satisface los requerimientos de las aplicaciones M2M. Conexión UDP con fiabilidad opcional que soporta transmisiones unicast y multicast.
- Transmisión asíncrona de mensajes.
- Bajo overhead y facilidad para mapear a HTTP.
- Identificación de recursos mediante URI (Uniform Resource Identifier) y Content-Type.

Capa de CoAP

El protocolo funciona a través de dos capas:

1. Modelo de mensajes CoAP

Hace posible las transacciones UDP en los puntos finales en formato confirmable (CON) o no confirmable (NON). Cada mensaje CoAP cuenta con un ID distinto para mantener a raya las posibilidades de duplicación de mensajes (Shelby et al., 2012). Las 3 partes clave que intervienen en la construcción de esta capa son la cabecera binaria, la opción informática y la carga útil.

Como se explicó previamente, los textos confirmables son mensajes fiables y fáciles de construir que son rápidos y se reenvían hasta la recepción de una confirmación de entrega (ACK) con el ID del mensaje.

2. Modelo de solicitud/respuesta de CoAp

Esta capa se encarga de las solicitudes de mensajes CON y NON. La aceptación de estas peticiones depende de la disponibilidad del servidor, los casos son:

- Si está inactivo, el servidor atenderá la petición de inmediato. Si se trata de una CON, el cliente recibirá un ACK por ella. Si el ACK se comparte como un Token y difiere del ID, es esencial mapearlo adecuadamente haciendo coincidir los pares petición-respuesta (PickData, 2019).
- Si hay un retraso o espera, el ACK se envía, pero como un texto vacío. Cuando llega su turno, la solicitud se procesa y el cliente recibe un nuevo CON (PickData, 2019).

A continuación, se mencionan los rasgos clave del modelo de solicitud/respuesta:

- Los códigos de petición o respuesta para CoAP son los mismos que para el HTTP, excepto por el hecho de que están en formato binario (Tokens de 0-8 bytes) en el caso de CoAP (Gimenez, 2013, p.34).
- Los métodos de solicitud para realizar llamadas (GET, PUT, POST y DELETE) se declaran en el proceso (Gimenez, 2013, p.34).
- Una respuesta CON puede ser almacenada en un mensaje ACK o reenviada como CON/NON (Gimenez, 2013, p.34).

XMPP

El Protocolo Extensible de Mensajería y Presencia (XMPP) es un protocolo estándar abierto que se utiliza para crear aplicaciones en tiempo real y emplea una amplia gama de tecnologías de comunicación de servicios, como la mensajería instantánea, chat-multiparty, las llamadas de voz y vídeo, la colaboración, el middleware ligero y el enrutamiento generalizado de datos XML (Saint-Andre, 2004).

Arquitectura

El protocolo XMPP establece un formato para el intercambio de datos entre dos o más dispositivos que se comunican, que en el caso del ecosistema IoT puede ser entre sensores/actuadores (D2D) o entre un sensor/actuador y un servidor/nube (D2S/C). (Moffit, 2010, p.22) sostiene que los sistemas que utilizan el protocolo XMPP en Internet son en su mayoría accesibles para todos y, por tanto, dichos sistemas forman una red federada de sistemas interconectados. El sistema XMPP consta de servidores, clientes, componentes y plugins de servidor; los componentes son descritos como

algo externo a los servidores con los que los clientes pueden comunicarse como un nuevo servicio. Un servicio de chat multiusuario es un ejemplo de tal componente. El propósito de los plugins de servidor es mayormente similar al de los componentes, pero los plugins pueden cambiar el comportamiento central del servidor con una sobrecarga reducida en comparación con los componentes, así como puede acceder a las estructuras de datos que se utilizan en el servidor interno (VEPROF, 2021).

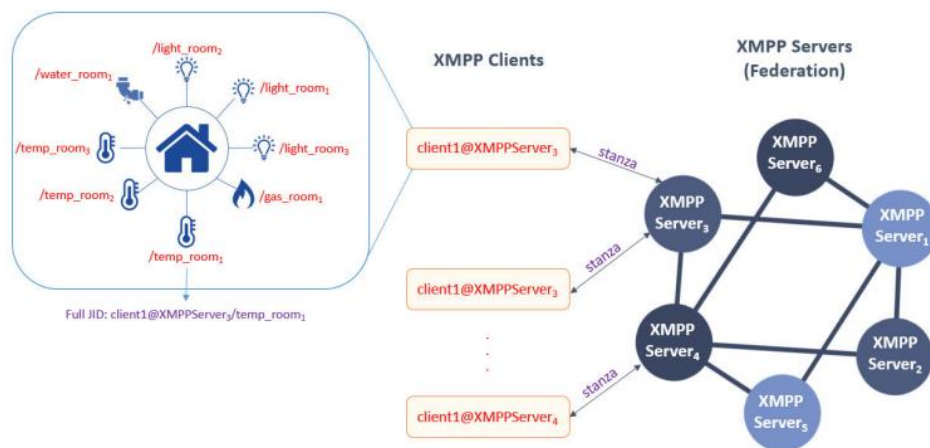


Ilustración 32-2: Arquitectura XMPP

Fuente: Malik et al., 2018, p.6

Jabber como originalmente se denomina a esta arquitectura, permite que las aplicaciones ofrezcan capacidades de mensajería instantánea, mediante el uso de XML como formato de intercambio de datos subyacente que se ejecuta sobre TCP/IP. Proporciona un conjunto de servicios esenciales denominados servicios centrales de XMPP y servicios extensibles denominados protocolos de extensión de XMPP (XEP), cuyo objetivo es ampliar los servicios centrales de XMPP (Malik et al., 2018, p.6).

Uno de los principales protocolos de extensión de XMPP es el PubSub XEP, que permite a XMPP soportar el modelo de publicación y suscripción, a través de PubSub XEP, es posible utilizar XMPP como protocolo de mensajería para sistemas IoT.

El PubSub es una extensión del protocolo del núcleo de XMPP que permite a las entidades XMPP crear temas (nodos) y publicar información en esos nodos. Las entidades suscritas recibirán entonces notificaciones con o sin carga útil. Las entidades XMPP se asocian con JIDs (Jabber IDs) cuando se ejecutan en una red. Un JID tiene la forma de una dirección de correo electrónico con un nombre de dominio completamente calificado y/o un recurso válido (por ejemplo, xmpp_user@xmpp_server/resource) donde xmpp_user es el nombre de usuario del cliente, xmpp_server es un nombre de dominio completamente calificado y resource es un identificador

utilizado para identificar el dispositivo del cliente en la red. Un JID desnudo es una dirección sin el recurso, mientras que un JID que incluye un identificador de recurso se denomina JID completo (RF-Wireless, 2015).

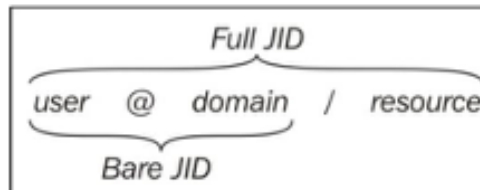


Ilustración 33-2. Estructura mensaje XML

Fuente: RF-Wireless, 2015

Se pueden asociar múltiples recursos (por ejemplo, JID completos) a un nombre de usuario que indique diferentes dispositivos utilizados o asociados a la misma "cuenta" o usuario.

XMPP proporciona encriptación punto a punto (TLS) que está incorporada dentro de la especificación principal (RF-Wireless, 2015).

A continuación, se presenta una comparativa de los diferentes protocolos de la capa de aplicación, mostrando sus características más relevantes en la tabla 8-2.

WebSocket

Este protocolo fue desarrollado para cumplir con el intercambio constante de datos entre el cliente y el servidor que no se soportaba en HTTP. El protocolo consiste en un canal de comunicación bidireccional completo que funciona a través de un único socket, además de tener una comunicación asíncrona a diferencia de HTTP, es decir, ambas partes pueden enviar datos en cualquier momento mientras exista conexión (Bayılmış et al., 2022).

Arquitectura

El funcionamiento del protocolo WebSocket consta básicamente de dos fases. Se trata de las fases de handshake y de transferencia de datos. Una conexión WebSocket se inicia mediante peticiones HTTP para el handshake. A continuación, los mensajes se transmiten a través de TCP. Este principio de funcionamiento se muestra en la Ilustración 34-2. Primero, el cliente transmite una solicitud de conexión al servidor a través de HTTP; si el servidor soporta WebSocket, se envía un mensaje afirmativo a través de HTTP y se lanza un handshake (Google, 2011).

Una vez establecida la conexión, el cliente y el servidor comienzan a utilizar el protocolo WebSocket para las comunicaciones. La conexión entre estos dos nodos continúa hasta que el apretón de manos finaliza, mientras la conexión esté activa, ambos nodos pueden enviarse

mensajes en cualquier momento, incluso al mismo tiempo. En esta fase, no es necesario enviar mensajes mediante paquetes de petición o respuesta; los mensajes pueden contener datos de texto o binarios (Google, 2011).

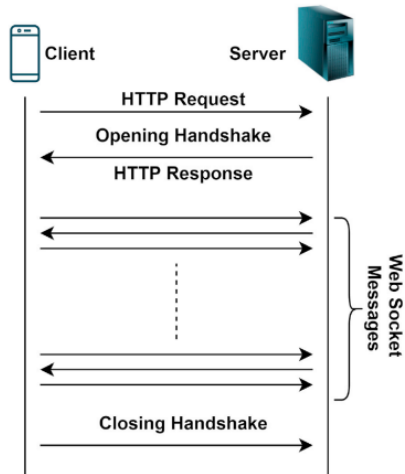


Ilustración 34-2: Procedimiento de handshaking en WebSocket

Fuente: Bayılmış et al., 2022

Las cabeceras de los mensajes se reducen al menor tamaño posible que minimice el consumo de ancho de banda. Es posible reenviar los mensajes de forma secuencial dividiéndolos en múltiples tramas, es decir, en streaming. Cada cabecera de trama puede tener un tamaño de 2 a 10 bytes si es enviada por el servidor y de 6 a 14 bytes si es enviada por el cliente (el cliente debe añadir una clave de enmascaramiento para evitar ataques de envenenamiento de caché) (Bayılmış et al., 2022). Los URIs de WebSocket son los mismos que los URIs de HTTP, con la diferencia del esquema, ya que WebSocket no utilizan un esquema HTTP/HTTPS; en su lugar utiliza ws/wss (web socket secure) para conectar con los hosts/endpoints (Vikram, 2021).

A continuación, la Ilustración 35-2 presenta el esquema de una URI de WebSocket y WebSocket Secured.

```
ws://domain-name[:port]/resource-path/resource-id
{secured} wss://domain-name[:port]/resource-path/resource-id
```

Ilustración 35-2: Esquemas de Websocket

Fuente: Vikram, 2021

Tabla 8-2: Comparativa de protocolos de la capa de aplicación

Características	HTTP	CoAP	MQTT	XMPP	WebSocket
Año de estandarización	1997	2013	1999	2002	2011
Patrón de mensajes	Petición/ Respuesta	Petición/ Respuesta	Publicación/ Suscripción	Publicación/ Suscripción; Petición/ Respuesta	Publicación/ Suscripción; Petición/ Respuesta
Arquitectura	Cliente/ Servidor	Árbol	Árbol	Cliente/ Servidor	Árbol
Transporte	TCP	UDP	TCP (MQTT-S: UDP)	TCP	TCP
Capa de Red	IPv4 o IPv6	IPv6	IPv4 o IPv6	IPv4 o IPv6	IPv4 o IPv6
Comunicaciones M2M	No	Si	Si	Si	Si
Mensajes asíncronos	No	Si	Si	Si	Si
Priorización de datos	No	No	No	Si	Si
Soporte de QoS	No	Si	Si	No	Si
Cache de mensajes	Si	Si	Si	Si	Si
RESTful	Si	Si	No	No	Si
Niveles de QoS	Basadas en TCP	2 niveles	3 niveles	No existe	No existe
Dominio de comunicación	Dispositivo- Nube; Nube-Nube	Dispositivo- Dispositivo	Dispositivo- Nube	Dispositivo- Nube; Nube-Nube	Dispositivo- Nube; Nube-Nube
Direccionamiento	URI	URI	Solo tópicos	Jabber ID	WebSocket URI
Formato del Payload	Indefinido	JSON, XML	Indefinido	XML	Indefinido
Tamaño máximo del mensaje	Indefinido	64KB (UDP)	256 MB	Indefinido; 64 KB (tamaño del Stanza)	Tramas múltiples de 125 Bytes
Lenguajes usados en: [Broker/Servidor]	C, C++, C#, PHP, Java, LUA, Javascript, Ruby, Python, Perl.	Java, C, C++, C#, Python, JavaScript, Ruby, LUA.	Java, C, C++, C#, Python, Ruby.	Java, C, C++, C#, Python, Go.	C, C++, C#, PHP, Java, LUA, Javascript, Ruby, Python, Perl.

Realizado por: López, Diego; 2022

Una vez expuestos los protocolos de la capa de aplicación se procede a la revisión del procesamiento y alojamiento de datos en esta capa.

2.1.3. *Procesamiento de datos*

La concepción de IoT dirige todo el procesamiento de datos hacia el Internet para alivianar la carga de procesamiento a los dispositivos finales. Existen diferentes métodos de procesamiento de datos en IoT, los cuales dependen de las necesidades de cada entorno al que se vaya a aplicar; los cuales se describen a continuación.

2.1.3.1. *Cloud Computing*

Es el acceso bajo demanda a recursos informáticos a través de Internet tales como: aplicaciones, servidores, alojamiento de datos, herramientas de desarrollo, funciones de red; que están ubicados en un centro de datos remoto orquestado por un proveedor de servicios en la nube (CSP) (IBM, 2022).

La computación en nube ofrece una gama de servidores competentes, tiempo compartido y virtualización de activos, con énfasis en el plan de acción. A través de esta visión del mundo, los usuarios y las empresas pueden acceder a los modelos de infraestructura como servicio, plataforma como servicio y software como servicio a petición de cualquier lugar del planeta utilizando esta visión del mundo (DeepShah, 2021, p.2).

Cuando un proveedor de la nube hace que sus administraciones sean accesibles a la población en general en una premisa de pagar más a medida que surgen los costos, esto se denomina Nube Pública, mientras que la Nube Privada se refiere a las etapas internas de la Nube solicitadas y utilizadas exclusivamente por organizaciones explícitas. Las nubes pueden aumentar o disminuir para abordar los problemas de los clientes, al tiempo que ofrecen condiciones de ejecución más adaptables y aisladas para los programas (IBM, 2022).

Arquitectura

La arquitectura se divide en dos partes: front-end y back-end. Para conectar el front-end y el back-end se emplea una red o Internet. La ilustración 36-2 muestra el diagrama de la arquitectura de la computación en nube.

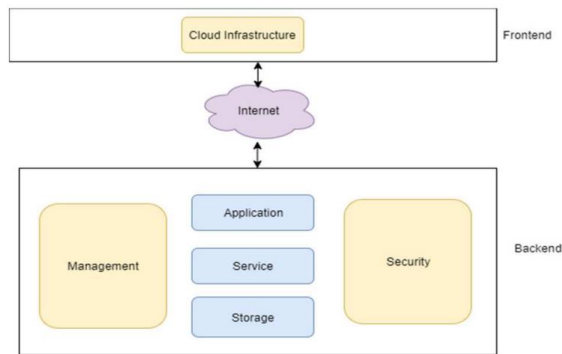


Ilustración 36-2: Arquitectura de Cloud Computing.

Fuente: DeepShah, 2021, p.2

El front end es el lugar donde los clientes se conectan a Internet para obtener datos. Esto incluye los ordenadores, las asociaciones de ordenadores, las aplicaciones y algunas técnicas diferentes por las que los clientes pueden conectarse con el sistema de computación en nube (DeepShah, 2021, p.3).

El back end reúne todo lo necesario para las organizaciones de computación en nube, este es el lado del proveedor de la condición de computación en nube. Está provisto de servidores, PCs, estructuras de almacenamiento de datos, aplicaciones y cualquier otro equipo que se espera que pase en varias organizaciones de computación en nube, ahí se alojan varias aplicaciones que están dotadas con su propio servidor (DeepShah, 2021, p.3).

- **Aplicación:** Una aplicación en el backend es una pieza de programación o una fase a la que se acerca un cliente; ofreciendo ayuda en el backend de acuerdo con las especificaciones del cliente.
- **Servicio:** La organización del backend implica las tres clases fundamentales de organizaciones basadas en la nube: SaaS, PaaS e IaaS. Del mismo modo, controla la clase de organización a la que se acerca el cliente.
- **Nube de tiempo de ejecución:** En el backend, la nube de tiempo de ejecución implica el plan de juego de una etapa/entorno de ejecución a la máquina virtual.
- **Almacenamiento:** El almacenamiento en el backend implica el plan de juego de un curso de acción de acumulación versátil y flexible de forma similar a la asociación de datos apartados.
- **Infraestructura:** La estructura del backend insinúa el engranaje de la nube y las partes de la programación como los servidores, la acumulación, los dispositivos de red, la programación de la virtualización, etc.
- **Gestión:** La gestión en el backend sugiere la asociación de partes del backend como aplicaciones, organizaciones, nubes en tiempo de ejecución, amasamiento,

establecimiento y otras estrategias de seguridad, entre otras.

- **Seguridad:** La seguridad del backend implica el plan de varios procedimientos de seguridad en el backend para salvaguardar los recursos de la nube, los sistemas, los datos y el establecimiento para los clientes finales.
- **Internet:** Una afiliación a la web hace de conductor o de aumento entre el frontend y el backend, estableciendo la participación y la correspondencia entre ambos.

2.1.3.2. Fog Computing

La computación en la niebla es una estructura de PC descentralizada situada entre la nube y los aparatos que producen información. Su propósito es reducir la carga de datos en el canal de comunicación de los dispositivos y la nube; por lo tanto, determina de forma rápida cuál será el destino de estos datos según sus características, de manera que el proceso sea más eficiente (Aicad, 2020).

Arquitectura

La arquitectura de la red implica el uso de servicios de dispositivos finales (conmutadores, enrutadores, multiplexores, etc.) por razones de computación, almacenamiento y procesamiento; su diseño implica piezas de organización física y lógica, software y equipos para montar una organización extensa de muchos aparatos interconectados. La dispersión de los centros de niebla (físicos y geológicos), así como la geografía y las convenciones utilizadas, son partes significativas de la construcción de un diseño de niebla. El transporte de las capacidades en las distintas capas, el tipo y el número de convenciones utilizadas y los límites forzados en las distintas capas son elementos que contribuyen a la ingeniería de la niebla. La ilustración 37-2 presenta un diagrama del diseño de la informática de niebla (DeepShah, 2021, p.4).

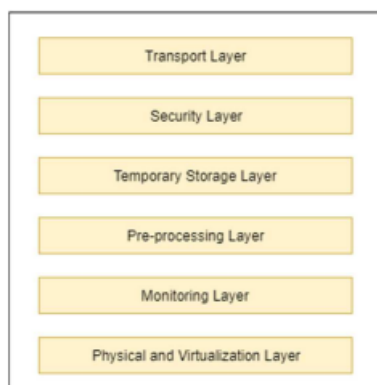


Ilustración 37-2: Arquitectura de Fog Computing

Fuente: DeepShah, 2021, p.4

Esta arquitectura está conformada por las siguientes capas:

- **Capa física y de virtualización:** Consta de concentradores (físicos y virtuales), los hubs que están dispuestos en diferentes áreas realizan la captación de información.
- **Capa de monitorización:** El nodo realiza diferentes observaciones identificadas con diferentes asignaciones como la medida del tiempo trabajado, la temperatura.
- **Capa de preprocesamiento:** Lleva a cabo una serie de ejercicios de información, la mayoría de los cuales están relacionados con la investigación. La información se depura y se analiza en busca de cualquier información no deseada; una vez filtrada la información sólo se emplea la información útil.
- **Capa de almacenamiento temporal:** Vinculada a la dispersión y replicación de la información que no es duradera. En este nivel se utiliza la virtualización de almacenamiento. En el momento en que la información se envía a la nube desde esta capa, se borra de la capa breve.
- **Capa de seguridad:** Esta capa está destinada a la integridad, protección, decodificación y el cifrado de la información.
- **Capa de transporte:** La función principal de este nivel es la transmisión de información segura preparada y de precisión a la capa de la nube para su almacenamiento a largo plazo. La parte de la información se acumula y se transfiere para su eficacia.

El Edge computing se ha excluido de esta lista de métodos de procesamientos IoT, ya que la concepción IoT implica todas las tareas del tratamiento de la información en el Internet, mas no en los dispositivos de borde (Gillis, 2017). A continuación, la tabla 9-2 presenta un comparativa entre los métodos de procesamiento anteriormente mencionados.

Tabla9-2: Comparativa de diferentes técnicas de análisis de datos

Características	Cloud Computing	Fog Computing
Arquitectura	Centralizada	Descentralizada
Latencia	Alta	Media
Locación del servicio	Dentro del data center en el cloud vía internet	Borde de la red cloud
Procesamiento de datos	Respuesta en largo plazo debido a la locación del servicio	Respuesta en corto plazo debido a la locación del servicio

Tiempo de respuesta	Algunos minutos	Milisegundos
Capacidad de cómputo	Alta, no proporciona ninguna reducción de datos mientras se envían o procesan.	Baja, reduce la cantidad de datos enviadas al cloud computing.
Numero de saltos Usuario-Servidor	Múltiples	Uno
Tiempo de almacenamiento de datos	Meses o años dependiendo el contrato del servicio	Temporal
Escalabilidad	Alta, en función de las capacidades del servidor en el cloud	Limitada, según las capacidades del servidor local

Realizado por: Lopez, Diego; 2022

2.2. Energía eléctrica

Es un tipo de energía que se basa en el movimiento de los electrones entre dos puntos al existir una diferencia de potencial entre ellos, dando lugar a la generación de corriente eléctrica. Tiene la capacidad de transformarse en otros tipos de energía, tales como energía lumínica, térmica o mecánica (Repsol, 2022).

2.2.1. Corriente eléctrica

Es el flujo de carga eléctrica a través de un material conductor, debido al desplazamiento continuo y ordenado de los electrones a lo largo del conductor. Existen dos tipos de corrientes: continua y alterna (Solar, 2019). El presente estudio está enfocado al control de variables energéticas únicamente de corriente alterna.

2.2.2. Corriente alterna (AC)

En este tipo de corriente electrones varían constantemente su sentido de movimiento por lo cual el valor de voltaje no permanece constante en el tiempo. La corriente alterna mas más usada en la senoidal y en las viviendas los valores característicos son 230V de tensión y 50 Hz de frecuencia (Solar, 2019).

2.2.3. Potencia Eléctrica

La potencia eléctrica se define como el trabajo eléctrico realizado por unidad de tiempo, la cual se representa en Watts. También conocida como Potencia Instantánea “ $p(t)$ ”, es el producto del voltaje instantáneo multiplicado por la corriente instantánea que fluye a través de él, esta potencia instantánea puede ser positiva o negativa. Un valor positivo significa que la potencia fluye hacia el dispositivo, por otro lado, un valor negativo muestra que la potencia sale del dispositivo (Navarro, 2007, p.160).

En circuitos de corriente alterna se emplean tres conceptos de potencia, los cuales son: potencia activa, reactiva y aparente.

2.2.3.1. Potencia activa (P)

Se denomina potencia activa a la potencia que empleada en un circuito o sistema eléctrico/electrónico para realizar una transformación de energía eléctrica a otro tipo de energía utilizable y se disipa en forma de calor. Se obtiene a partir del promedio de la potencia instantánea $p(t)$ en un período o ciclo de trabajo y se mide en Watts (Universidad Nacional de Rosario, 2022, p.3). La ecuación 1-3 representa el cálculo de la potencia activa a partir de la potencia instantánea.

$$p(t) = \int_0^T p(t) dt \quad (1-3)$$

2.2.3.2. Potencia Reactiva(Q)

Está presente en los circuitos capacitivos y/o inductivos, es primordial funcionamiento de los campos eléctricos y magnéticos, usuales en sistemas como transformadores o motores; pero no es una potencia útil ya que no genera trabajo alguno ni convierte la energía eléctrica en otro tipo de energía, se mide en voltamperio reactivos (VAR) (García, 2022).

2.2.3.3. Potencia Aparente(S)

Es la potencia total que se consume en un sistema de corriente alterna, se mide en voltamperio (VA) y se obtiene a partir de la suma vectorial de la potencia activa (P) y la potencia reactiva (Q); esta relación se puede obtener a través del triángulo de potencias en la ilustración 38-2.

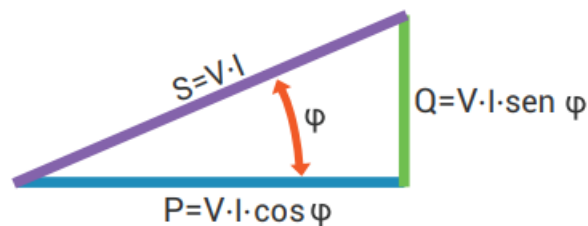


Ilustración 38-2: Triángulo de Potencias en corriente alterna

Fuente: Tecnológico de Monterrey, 2022, p.4

2.2.4. Energía consumida

Es la cantidad la energía eléctrica empleada para distintos propósitos como el funcionamiento de electrodomésticos o maquinas industriales en un determinado lapso, por lo general se toma a la hora como referencia y se mide en Kilowatts hora (KWh) (SmartGridsInfo, 2020).

2.2.4.1. Dispositivos de uso diario en el hogar

En base a los electrodomésticos de mayor uso en el hogar para el cumplimiento las necesidades básicas de los residentes, expuestos por (Pérez, 2021). La Tabla 10-2 presenta una clasificación de aquellos elementos según su respectiva línea.

Tabla10-2: Clasificación de electrodomésticos del hogar

Clasificación	Electrodoméstico
Línea Blanca	Cocina de inducción
	Microondas
	Lavaplatos
	Refrigerador
	Secadora
	Calefactor
	Aire acondicionado
	Plancha
	Secador de cabello
	Ducha eléctrica
	Aspiradora
	Ventilador
Línea Marrón	Televisor
	Equipo de sonido
	Teléfono móvil
	Teléfono Fijo
	Consola de videojuegos
	Computadora de escritorio
	Computadora personal
Impresora	

Realizado por: López, Diego; 2022

2.2.4.2. Medidas de ahorro energético en el hogar

La siguiente lista muestra las principales medidas para el ahorro energético.

- **Apagar las luces al salir de las habitaciones:** Es un problema muy frecuente en los hogares y a la larga se ve reflejado en el aumento del costo del servicio de electricidad (El Español, 2020).
- **Apagar los aparatos en stand-by:** El consumo de electricidad en stand-by, es el que

hacen los aparatos cuando no están en funcionamiento, pero permanecen conectados a la red (El Español, 2020).

- **Ahorrar energía con la lavadora:** La lavadora es de los elementos que más energía consume dentro del hogar, para mitigar este problema, se debe elegir los programas de lavado a menor temperatura o el lavado en frío y comprar una lavadora que consuma la menor cantidad de agua posible (El Español, 2020).
- **Desconectar los dispositivos cuando estén cargados por completo:** Mas allá de que la batería de este dispositivo se dañará, se producirá un consumo innecesario (El Español, 2020).
- **Establecer rutinas de encendido de las luminarias:** Hace referencia a aprovechar más la luz natural y solo usar la artificial cuando la ocasión lo amerite (El Español, 2020).

Hardware en capa de percepción

Sensores

Son el componente indispensable en los sistemas IoT, los cuales son piezas de hardware que son capaces de percibir el estado del entorno físico al que están conectados. Estos dispositivos pueden ser cámaras, sensores de temperatura, humedad, gas, energía (Castañeda, 2020).

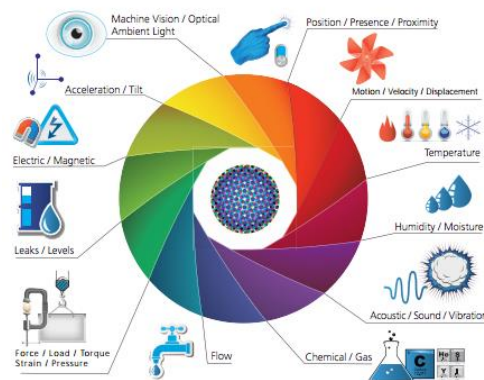


Ilustración 39-2: Parámetros captados por los sensores

Fuente: Aprendiendo Arduino, 2019

Para el presente trabajo se hará énfasis en los sensores del tipo energético, los cuales permitan la percepción de variables eléctricas como: voltaje, corriente, potencia, frecuencia; la tabla 11-2 presenta una comparativa de los diferentes sensores de variables energéticas de corriente alterna.

Tabla 11-2: Comparativa de sensores para la capa de percepción

Características	PZEM-016	PZEM-004T v3	HLW8032	ACS712	ZMPT101B
Variables de medición en corriente alterna	voltaje, corriente, potencia (activa), energía consumida, frecuencia, factor de potencia	voltaje, corriente, potencia (activa), energía consumida, frecuencia, factor de potencia	voltaje, corriente, potencia	corriente	voltaje
Soporte a tarjeta de desarrollo	Si	Si	Si	Si	Si
Voltaje de funcionamiento	5V	5V	5V	0-25V	3.3 -5V
Vac máximo de funcionamiento	260 Vac	260 Vac	380Vac	No mide	250 Vac
Iac máximo de funcionamiento	10 A sin módulo de adaptación de corriente 100 A con módulo de adaptación de corriente	10 A sin módulo de adaptación de corriente 100 A con módulo de adaptación de corriente	10 A	30 A	No mide
Precisión de los datos	± 0,5 %	± 1%	±6 %	±3%	± 0,2 %
Tipo de conexión	RS485	TTL 4 pines	TTL 4 pines	TTL 3 pines	TTL 4 pines
Precio	\$25	\$16	\$11	\$8	\$7

Realizado por: López, Diego; 2022

Tarjetas de desarrollo

Una tarjeta de desarrollo es una placa o circuito que contiene un microcontrolador principal que corre o ejecuta una serie de instrucciones de un programa suministrado o un microprocesador con un sistema operativo instalado para la ejecución de las instrucciones. Posee un diseño electrónico alrededor de esta unidad de procesamiento que permite la programación del componente suministra el voltaje adecuado para el correcto funcionamiento del controlador y proporciona acceso a las entradas y salidas del microcontrolador o microprocesador para la conexión de sensores y actuadores (Concepcion, 2019). La tabla 12-2 presenta una comparativa de las diferentes tarjetas de desarrollo para el sistema IoT propuesto.

Tabla 12-2: Comparativa de tarjetas de desarrollo

Características	Arduino Uno R3	Arduino Mega 2560 R3	Arduino Nano	BeagleBone Black Rev C	Raspberry Pi 4 Model B	NodeMCU ESP 8266-12F
Precio	\$25	\$35	\$19	\$90	\$120	\$9
Tamaño	6,8 x 5,4 cm	10,2 x 5,4cm	4,5 x 1,8 cm	8,6 x 5,5 cm	8,5 x 5,6 cm	4,8 x 2,6 cm
Velocidad del reloj	16 MHz	16 MHz	16 MHz	1 GHz	1 GHz	80 MHz
Voltaje de funcionamiento	5V	5V	5V	5 V	5,1 V	3,3 V
Canales de alimentación	2	2	1	2	2	2
Unidad de procesamiento	Micro-controlador	Micro-controlador	Micro-controlador	Micro-procesador	Micro-procesador	Micro-controlador
Sistema Operativo	No posee	No posee	No posee	Debian	Debian	XTOS
Potencia de consumo	0,5W	0.37W	0,14 W	6 W	6,4 W	0,24 W
Numero de puertos seriales (Tx, Rx)	1	4	1	6	6	1
Numero de GPIO	20	54	14	92	40	17
Numero de pines PWM	6	15	6	8	4	17 emulados por software
Adición de periféricos	Si	Si	Si	Si	Si	Si
Conectividad inalámbrica incorporada	No posee	No posee	No posee	Wi-Fi 2,4GHz / 5GHz IEEE 802.11.b/g/n Bluetooth 5.0, BLE	Wi-Fi 2,4GHz / 5GHz IEEE 802.11.b/g/n/ac Bluetooth 5.0, BLE	Wi-Fi 2,4GHz / 5GHz IEEE 802.11.b/g/n

Realizado por: Lopez, Diego; 2022

Actuadores

Es un dispositivo que transforma un cierto tipo de energía, en este caso eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un elemento externo. Este recibe la orden del controlador y en base a esa señal se genera la orden para activar un elemento final de control como, por ejemplo, un tomacorriente inteligente o un dimmer (Aprendiendo Arduino, 2019).

Hardware en capa de red

Router

Es el dispositivo encargado de guiar y dirigir los datos de red mediante paquetes que contienen varios tipos de datos, como archivos, comunicaciones y transmisiones simples como interacciones web; permite interconectar segmentos de red o redes enteras (Cisco, 2022).

Hardware en capa de aplicación

Servidor basado en hardware: Es una máquina física que conforma una red informática en la que, adicional al sistema operativo, funcionan uno o varios servidores basados en software. Se lo denomina también como “host”, por lo que todo ordenador puede usarse como host mediante el respectivo software para servidores (IONOS, 2020).

Tabla 13-2: Comparativa de opciones para servidor como hardware

Características	Raspberry Pi 4 Model B	Beagle Bone Black Rev C	HP Pavilion DM4
RAM	4GB DDR2	0,5 GB DDR3L	6GB DDR2
Procesador	ARM Cortex-172 4x 1,5GHz	ARM Cortex-A8 1 GHz	Intel Core i3 2x 2,1 GHz
Sistema Operativo	Debian	Debian	Linux
Puertos USB	2x USB 2.0, 2x USB3.0	1 x USB 2.0, 1 x mini USB2.0	3x USB 2.0
Pantalla	No posee	No posee	Pantalla LCD de 14''
Teclado	No posee	No posee	Posee teclado alfanumérico
Mouse	No posee	No posee	Mouse path integrado
Capacidad de Almacenamiento	Tarjeta SD, SDHC o MicroSD, no posee almacenamiento nativo	tarjeta microSD, MMC, no posee almacenamiento nativo	500 GB disco duro
Conectividad	Wifi, Bluetooth, Ethernet	Wifi, Bluetooth, Ethernet	Wifi, Bluetooth, Ethernet
Precio	\$90	\$120	\$70

Fuente: López, Diego; 2022

Software en capa de percepción

Entorno de Desarrollo Integrado

Es un sistema de software enfocado hacia el diseño de aplicaciones combinando herramientas comunes para programadores en una misma interfaz de usuario gráfica (GUI), también conocido como IDE, posee las siguientes características:

- **Editor de código fuente:** Ayuda a digitar el código de software con funciones que permiten desarrollarlo de una manera fluida y controlada (RedHat, 2019).
- **Automatización de compilaciones locales:** Automatizan tareas simples y repetitivas como parte de la creación de una compilación local del software para su uso por parte del desarrollador, como la compilación del código fuente del ordenador en un código binario, el empaquetado de ese código y la ejecución de pruebas automatizadas (RedHat, 2019).

- **Depurador:** Prueba otros programas y resalta la ubicación del error en el código original de forma gráfica (RedHat, 2019).

Tabla 14-2: Comparativa de entornos de desarrollo IoT

Características	Contíki	AWS IoT Core	Node Red
Descripción	Sistema operativo de código abierto	Servicio en la nube para la conexión de dispositivos IoT	Motor de flujos con enfoque IoT de código abierto
Lenguajes soportados	C	Depende del SDK (kits de desarrollo de software)	Javascript
Costo	Gratuito	Gratuito (hasta millón de mensajes) y Paga (más de millón de mensajes)	Gratuito
Aplicaciones	Redes de sensores, domótica, sistema de alarma, conexión de sistemas computacionales.	Gestión de almacenamiento de información en la nube, Red de sensores, automatización.	Proyectos IoT, servicios en línea, automatización, manejo de sistemas de almacenamiento, aplicaciones web, red de sensores.
Nivel de complejidad de uso	Alta	Media	Media
Compatibilidad con sistemas operativos	Sistema operativo propio	Windows, Linux, MAC, Android	Windows, Linux, MAC, Android
Soporte brindado por comunidad	Medio	Medio	Alto
Integración de bases de datos externas	Permite	No permite	Permite

Fuente: López, Diego; 2022

Software en capa de red

Analizador de paquetes de red

Es un software informático que se emplea para el rastreo, interceptación y registro del tráfico de red que fluye a través de esta. Analiza el tráfico de la red y crea un informe personalizado para la gestión adecuada de la red, esta herramienta es empleada por varias organizaciones para el manejo eficiente de las redes informáticas (LaRosa, 2019).

Tabla 15-2: Analizadores de paquetes de red

Herramienta	Costo	Espacio requerido	RAM requerida	Compatibilidad
Wireshark	Gratis	15 MB	500 MB	Windows, Linux, Solaris, Mac
TCPdump	Gratis	1 MB	1GB	Linux, Windows
Snoop	Gratis	Nativo del sistema operativo	1GB	Solaris
Windump	Gratis	1MB	512 MB	Mac, Linux, Solaris
Paessler PRTG	Paga	100 GB	4 GB	Windows, Mac, Android
SolarWinds	Paga	20 GB	4 GB	Windows, Solaris, Linux

Realizado por: López. D; 2022

Software en capa de aplicación

Servidor basado en software: Es un software que brinda un servicio particular que otros programas denominados “clientes” pueden usar a nivel local o a través de una red y el tipo de servicio depende del tipo de software del servidor. El modelo de comunicación está basado en el esquema cliente-servidor (IONOS, 2020).

Base de datos: Es un compendio estructurado de datos, que se alojan de forma lógica en un sistema informático. Una base de datos está gestionada por un sistema de gestión de bases de datos (DBMS); los datos y el DBMS junto con las aplicaciones asociadas a ellos se los denomina “base de datos” (Oracle, 2022).

Existen varios tipos de bases de datos, pero las dos principales son:

- **Bases de datos relacionales:** Sus elementos se organizan como un conjunto de tablas con columnas y filas, proporcionando una manera más eficiente y flexible de acceder a información estructurada (Oracle, 2022).
- **Bases de datos NoSQL:** NoSQL se refiere a una base de datos no relacional, estas bases de datos están orientadas a los documentos y le permiten almacenar y recuperar datos en formatos que no sean tablas (Oracle, 2022).

A continuación, la Tabla 14-2 presenta una comparativa entre las bases de datos relacionales y no relacionales.

Tabla 16-2: Comparativa de bases de datos relacionales y no relacionales

Características	Bases SQL	Bases No SQL
Datos	Estructurados almacenados en tablas	No estructurados almacenados en archivos JSON.
Esquema	Estático	Dinámico
Lenguaje	SQL	No SQL
Escalabilidad	Gran escalabilidad vertical	Gran escalabilidad horizontal
Modelo	Relacional	No Relacional
Búsqueda	Indexada	Semántica
Redundancia de datos	Reduce redundancia	Permite redundancia
Enfoque	Aplicaciones con datos consistentes	Aplicaciones con Big Data
Velocidad de respuesta	Depende de la carga de datos a procesar	Priorizan velocidad sobre integridad de datos

Realizado por: López, D; 2022

CAPITULO III

3. MARCO METODOLÓGICO

El presente capítulo aborda los procedimientos realizados para el diseño e implementación del sistema IoT de monitoreo y control del consumo energético en redes de baja tensión para uso residencial. Así como requerimientos del sistema IoT, selección de las tecnologías y protocolos de comunicación. Exponiendo cada una de las etapas y procesos, como diseño esquemático de los elementos de hardware y programación a nivel de software.

3.1. Metodología

3.1.1. *Metodología Inductiva*

Establece una manera de razonamiento, partiendo de una serie de observaciones particulares que conducen a la elaboración de leyes y conclusiones generales (Arrieta, 2022). Esta metodología se caracteriza por:

- Observación de fenómenos y hechos.
- Generalización a partir de la observación.
- Conclusiones probables.
- Su objetivo es la generación de nuevo conocimiento.

La Ilustración 1-3 muestra los procesos resultantes de aplicar el método inductivo en el desarrollo del proyecto.

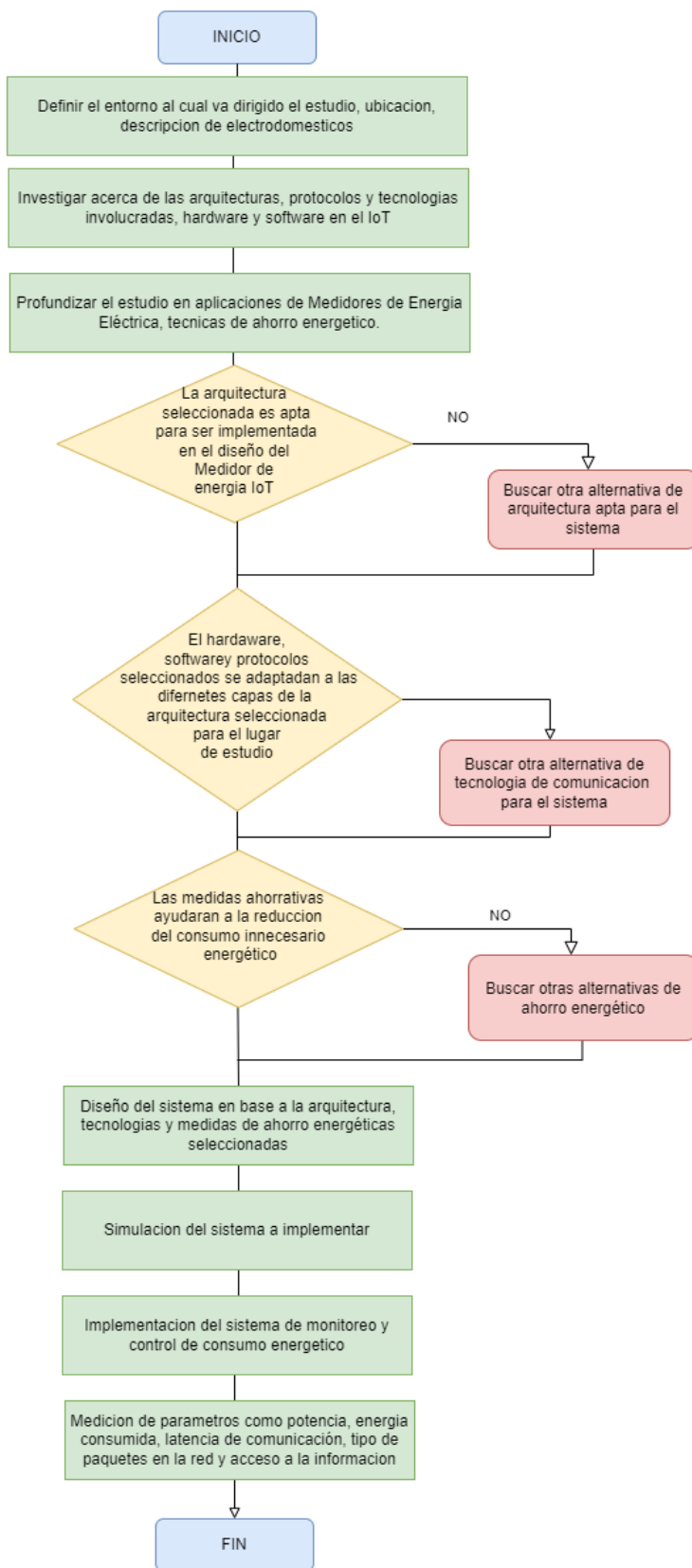


Ilustración 1-3: Diagrama de procesos del proyecto en base al método inductivo

Fuente: Lopez, D; 2022

3.1.2. Metodología Deductiva

Decreta una forma de razonamiento y explicación de la realidad a partir de leyes o teorías generales hacia casos particulares (Arrieta, 2022). Esta metodología se caracteriza por:

- A partir de generalizaciones se establecen conclusiones.
- La conclusión de un razonamiento está incluida en las premisas.
- Empleada cuando las causas de un fenómeno no se pueden observar.
- Conclusiones válidas y estrictas.
- Al verificar conocimiento previo no genera nuevo conocimiento.

La Ilustración 2-3 muestra los procesos resultantes de aplicar el método deductivo en el desarrollo del proyecto.

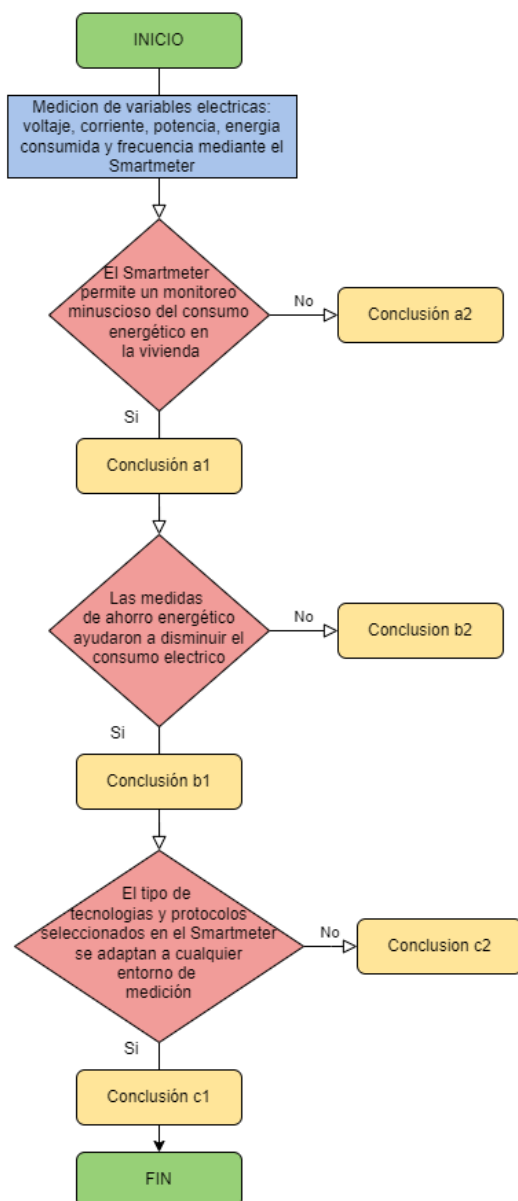


Ilustración 2-3: Diagrama de procesos del proyecto en base al método deductivo

Fuente: López, D; 2022

3.2. Estudio del escenario de implementación del sistema

El presente proyecto está destinado a desarrollarse en el tercer piso de una vivienda ubicada en la ciudad de Ambato- Ecuador, en la parroquia Huachi-Chico, barrio La Esperanza. El tercer piso está dividido en ocho secciones, con 14 lámparas de 9 Watts cada una, 17 tomacorrientes, 1 teléfono convencional, 1 lavadora, 1 refrigeradora, 1 router del proveedor de Internet, 1 televisión, 1 equipo de sonido y 3 computadoras portátiles en total. La tabla 1-3 muestra la distribución detallada de los diferentes elementos en el hogar.

Tabla 1-3: Descripción de las secciones del piso 3 de la vivienda

Sección del Piso 3	Luminarias	Tomacorrientes	Dispositivos
Corredor	3	0	Ninguno
Cuarto Máster	2	4	2 televisores, 1 portátil, 1 router para servicio de Internet, 1 teléfono convencional
Cuarto A	1	2	1 portátil
Cuarto B	1	2	1 portátil
Baño	1	1	Ninguno
Cocina	2	3	1 refrigeradora
Área de Lavado	1	2	1 lavadora
Sala	3	3	1 equipo de sonido

Realizado por: Lopez, Diego; 2022

3.3. Requerimientos del diseño del sistema

Se deben considerar las siguientes características para llevar a cabo el diseño del sistema IoT de monitoreo y control del consumo energético en redes de baja tensión.

- Arquitectura que brinde una comunicación fluida y eficiente en el sistema hacia su núcleo de procesamiento.
- Protocolo IoT que permita compatibilidad y comunicación con las capas superiores en el modelo de comunicación.
- Procesamiento y análisis de la información de manera remota.
- Envío de información al servidor remoto para el respectivo procesamiento y accionar de la información.
- Registro del consumo energético del tercer piso de la vivienda de forma horaria, diaria, mensual y anual en la base de datos.
- Uso de sensores que brinden la mejor relación entre costo y prestaciones para el censado de las variables energéticas de la vivienda.
- Control del consumo energético de las cargas en el piso de la vivienda a través de diferentes acciones para la reducción del costo de la planilla eléctrica.

- Aplicabilidad del sistema de monitoreo y control en cualquier domicilio que posea una red monofásica de 110v.

3.3.1. Análisis comparativo de Arquitecturas IoT

En base al grado de cumplimiento ante los parámetros expuestos en la Tabla 3-2, se ha seleccionado la arquitectura basada en APIS ya que brinda simplicidad y eficiencia al sistema mediante el uso de la metodología REST. Debido a la separación entre el cliente y el servidor brindan una alta escalabilidad del servicio brindado, también se adapta en todo momento al tipo de sintaxis o plataformas de trabajo y permite realizar una migración de un servidor a otro o practicar cambios en la base de datos en todo momento.

3.3.2. Análisis comparativo de tecnologías de comunicación IoT

Tomando en cuenta el análisis realizado a la Tabla 5-2, se concluye emplear la tecnología IEEE 802.11n (WiFi) debido a que presenta las mejores características, soporta velocidades de transmisión muy altas, opera en 2 bandas de frecuencia y principalmente es la tecnología más usada en domicilios debido a los equipos empleados por los proveedores de internet para la creación de redes LAN, consumo de potencia moderado y brinda cobertura tanto al interior como al exterior de la vivienda.

3.3.3. Análisis comparativo de protocolos IoT

Este análisis de protocolos se deriva en 3 secciones: red, transporte y aplicación.

3.3.3.1. Análisis de protocolos de capa de red

En base a la Tabla 6-2, sin duda IPv6 es la mejor opción, pero actualmente en Ecuador el despliegue de IPv6 es limitado por lo cual el protocolo más usado en esta capa es IPv4. Siendo este el protocolo de red a usar en el sistema IoT.

3.3.3.2. Análisis de protocolos de capa de transporte

Debido a la necesidad de mantener una integridad de las variables eléctricas censadas, así como una mayor compatibilidad con los diferentes protocolos en la capa subsecuente, la elección del protocolo TCP permite satisfacer todas estas necesidades para el sistema IoT de monitoreo y control del consumo energético.

3.3.3.3. Análisis de protocolos de capa de aplicación

En base a la Tabla 8-2, se escoge al protocolo HTTP debido a que permite implementar una metodología RESTful, brindando una amplia compatibilidad con varios lenguajes de

programación, sin embargo, el aspecto más importante es que el protocolo HTTP está enfocado a entornos domésticos a diferencia de MQTT, CoAP o XMPP que se enfocan más a entornos industriales. Adicionalmente, se selecciona WebSocket debido a la disponibilidad del canal para aplicaciones que requieran recibir actualizaciones en tiempo real como el control del brillo de las luminarias.

3.3.4. Análisis comparativo de sistemas de computing.

Considerando las características expuestas en la tabla 9-2, se opta por emplear un procesamiento en la niebla o Fog Computing, debido a que las tareas que se desempeñaran no requieren un procesamiento excesivo para mudarlo a la nube. Este procesamiento representa una menor latencia en el sistema, así como un alojamiento de datos en base a las características del servidor propio que se implemente.

3.3.5. Análisis comparativo de sensores para el Smartmeter

En base a la comparativa realizada en la tabla 11-2 se selecciona al módulo PZEM-004T v3. Este módulo ofrece censar parámetros como: voltaje (Volts), corriente (Amp), potencia activa (Watt), frecuencia (Hz), energía consumida (KWh) y factor de potencia; con una precisión del $\pm 1\%$ a un precio muy accesible.



Ilustración 3-3: Modulo PZEM 004Tv3

Fuente: Soffar, 2017.

3.3.6. Análisis comparativo de tarjetas de desarrollo

En base a la Tabla 12-2 se seleccionaron 2 placas de desarrollo, las cuales son Arduino Mega 2560 y Node MCU ESP8266 12F debido al bajo consumo energético y adición de periféricos que presentan. Arduino Mega 2560 R3 presenta una gran disponibilidad de pines GPIO y sobre todo un número considerable de puertos seriales, los cuales serán de gran utilidad para receptor la información de cada sensor en el sistema, los pines PWM serán de gran utilidad para la regulación de intensidad luminaria del sistema; sin embargo la desventaja de esta placa reside en su carencia

de un módulo incorporado de comunicación inalámbrica, por ello se optó por adicionar el módulo NODE MCU ESP8266, debido a su compatibilidad con el protocolo IEEE 802.11.n el cual se seleccionó previamente.



Ilustración 4-3: Arduino Mega 2560 R3

Fuente: Arduino, 2022

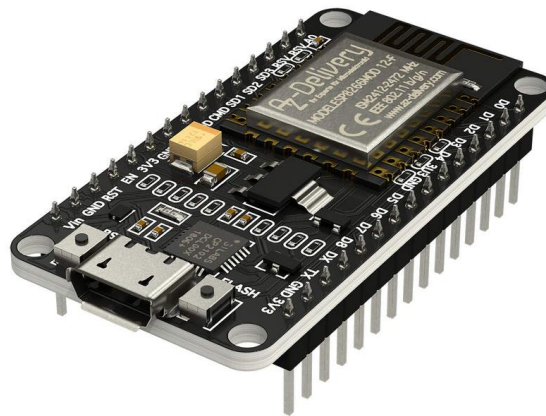


Ilustración 5-3: Node MCU ESP8266

Fuente: 10Gtek, 2022

3.3.7. Análisis Comparativo de hardware como servidor

Debido a la selección de Fog computing, se consideraron los siguientes dispositivos para desempeñar la función de servidor, en base a la comparativa realizada en la Tabla 13-2 se selecciona a la portátil HP Pavilion DM4, debido a su capacidad de procesamiento, memoria RAM, almacenamiento, disponibilidad de pantalla, teclado, mouse path, conectividad y precio la convierten en la mejor opción para implementar un servidor en ella.

3.3.8. Análisis Comparativo de entornos de desarrollo IoT

3.3.8.1. Capa de percepción

En base a la previa selección de tarjetas de desarrollo, se escoge al entorno de desarrollo integrado de Arduino para programar los respectivos procesos en la capa de percepción, debido a su compatibilidad con estas tarjetas, amplia gama de librerías para los sensores escogidos, gran soporte de comunidad en foros y su lenguaje de programación basado en C++.

3.3.8.2. Capa de Aplicación

Considerando las características expuestas en la tabla 14-2, se selecciona el entorno de desarrollo “NODE RED” en base a sus prestaciones como aplicaciones en el entorno IoT, conectividad de varios elementos hardware, programación en base a eventos, automatización de procesos y manejo de bases de datos externas; convirtiéndola en una herramienta ideal para el desarrollo del sistema IoT.

3.3.9. Análisis comparativo de componentes de actuadores para la integración de actuadores en el Smartmeter

En base a las recomendaciones mencionadas en el capítulo previo, se optó por seleccionar dispositivos que permitan el control del consumo energético en tomacorrientes y en las luminarias del hogar.

3.3.9.1. Enchufe Inteligente

Cumple la misma función de enchufe, salvo que puede conectarse a un smartphone a través de Wi-Fi, permitiéndolo ser controlado a distancia y gestionar de una mejor manera el consumo de energía de esa toma de corriente, a través de una aplicación para smartphones (Cielo, 2022). Es una de las maneras más eficaces de controlar el consumo innecesario de energía en aparatos que no desempeñen funciones útiles (Nor’ Adil, 2022).

La tabla 2-3 presenta una comparativa de los diferentes Smart Plugs existentes en el mercado para la incorporación al sistema propuesto.

Tabla 2-3: Comparativa de tomacorrientes inteligentes.

Características	TP-Link Kasa Smart Wi-Fi Mini Plug EP10	Wemo WiFi Smart Plug	Ring Outdoor Smart Plug	Wyze Plug
Voltaje de operación	100-120 Vac	100-120 Vac	100-120 Vac	100-120 Vac
Frecuencia de operación	60 Hz	60 Hz	60 Hz	60 Hz
Corriente máxima	15 A	15 A	15 A	15 A
Potencia máxima	1800 W	1800 W	1875 W	1800 W
Versión de wifi compatible	802.11b/g/n	802.11b/g/n	802.11b/g/n/ac	802.11b/g/n
Precio unitario	\$7	\$20	\$30	\$12

Fuente: López, Diego; 2022

Se selecciona al tomacorriente inteligente de la marca TP-Link debido a su compatibilidad con el estándar de WiFi seleccionado, voltaje, corriente, potencia y precio económico para la integración en el diseño del sistema IoT.



Ilustración 6-3: Tomacorriente Inteligente Kasa EP10

Fuente: Kasa Smart, 2022

3.3.9.2. Dimmer

Es un dispositivo que posibilita la regulación de la intensidad de las luminarias, a través de una perilla. Accionando la perilla, la intensidad aumenta o disminuye gradualmente hasta conseguir el grado de iluminación deseada por el usuario (Faro, 2022).

A continuación, la Tabla 3-3 presenta una lista de dimmers comerciales con sus precios.

Tabla 3-3: Precios y especificaciones de dimmers comerciales

Marca	Modelo	Wi-Fi	Amperaje	Precio unidad
TP-LINK	HS220	Si	6 A	\$40
Lutron	GE-18021	Si	12 A	\$36
SUNTHIN	Smart Dimmer	Si	20 A	\$29
TREATLIFE	Smart Dimmer	Si	30 A	\$63

Realizado por: López, D; 2022

Debido al alto coste de los dispositivos, se optó por elaborar un dimmer propio, mediante el uso de microcontroladores como el ESP8266 para su control mediante Wi-Fi. La ilustración 7-3 muestra el diagrama de conexión del dimmer.

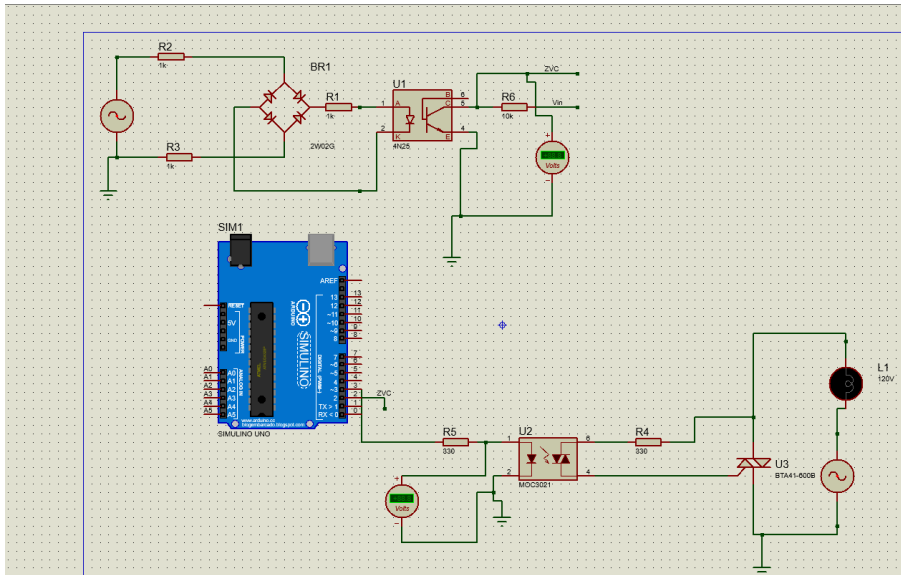


Ilustración 7-3: Esquema de conexión del dimmer

Fuente: López. D, 2022

La tabla 4-3 detalla los componentes empleados en el circuito de la ilustración 7-3.

Tabla 4-3: Lista de materiales del dimmer

Componente	Cantidad	Precio unitario	Precio total
ESP8266 Node MCU	1	\$9	\$9
Resistor 10K Ω	1	\$0,10	\$0,10
Resistor 4,7 K Ω	2	\$0,12	\$0,24
Resistor 330 Ω	2	\$0,10	\$0,20
MOC 3021	1	\$2	\$2
4N35	1	\$1,5	\$1,5
Diodo 1N4003	4	\$0,2	\$0,8
TRIAC BT 136	1	\$1,5	\$1,5
Total			\$15,14

Realizado por: López. D, 2022

3.4. Diseño del sistema de monitoreo del consumo energético

El esquema del sistema de monitoreo energético IoT es detallado a continuación en la ilustración 8-3 con sus respectivos procesos.

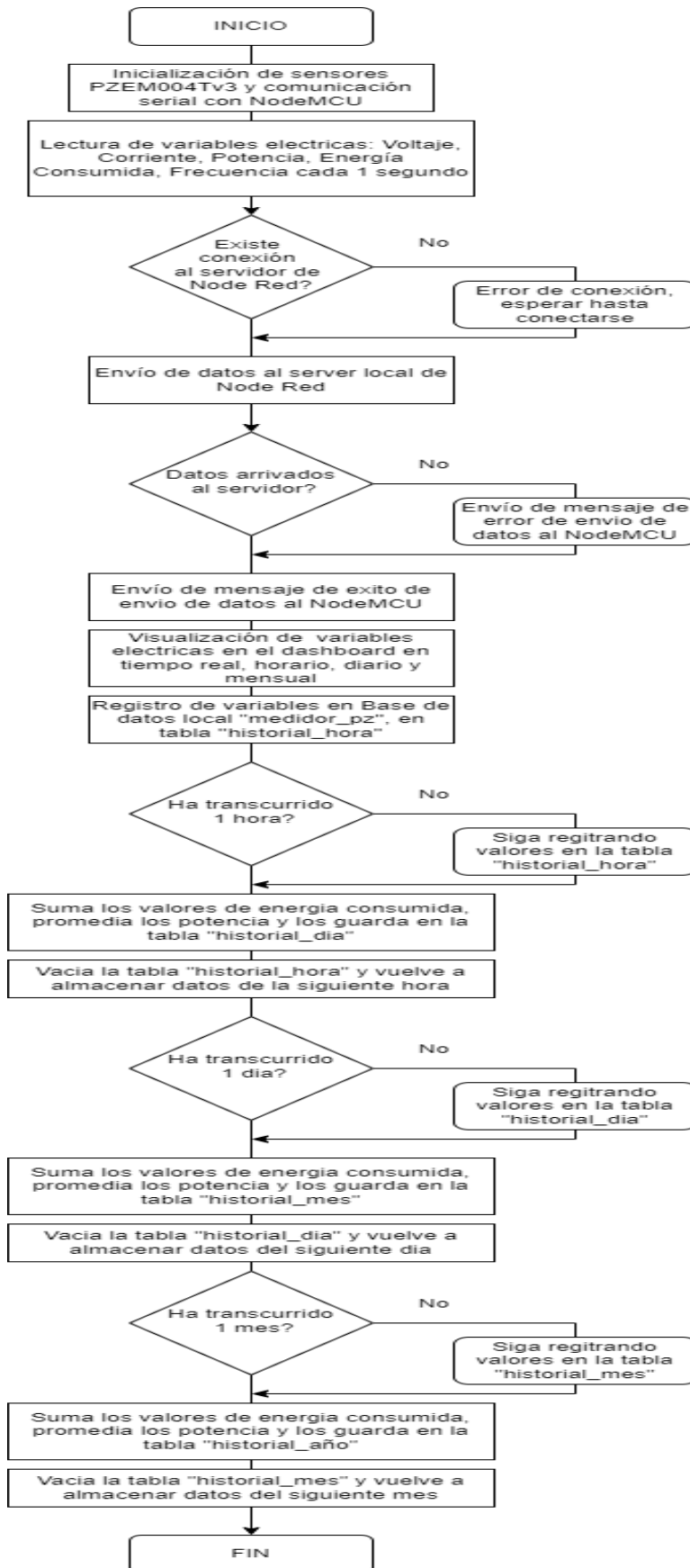


Ilustración 8-3: Diagrama de procesos del sistema de monitoreo del consumo energético IoT

Fuente: López, D; 2022

A continuación, se detallan los procesos implicados en el funcionamiento del sistema de monitoreo IoT en cada una de sus capas.

3.4.1. Capa de censado

En base al sensor PZEM 004T v3 seleccionado previamente, se optó por adicionar un sensor más del mismo tipo, con el propósito de monitorear el consumo del piso 3 de la vivienda ubicada en el barrio LA ESPERANZA del sector HUACHI CHICO en la ciudad de AMBATO, de una manera más detallada al dedicar el primer sensor denotado como “PZEM1” hacia el consumo de las luminarias y el sensor “PZEM 2” hacia el consumo de los diferentes dispositivos electrónicos, ambos sensores son conectados a través de comunicación serial hacia el Arduino MEGA 2560. La ilustración 9-3 muestra el esquema de conexión de los sensores hacia las líneas del breaker del piso 3 de la vivienda.

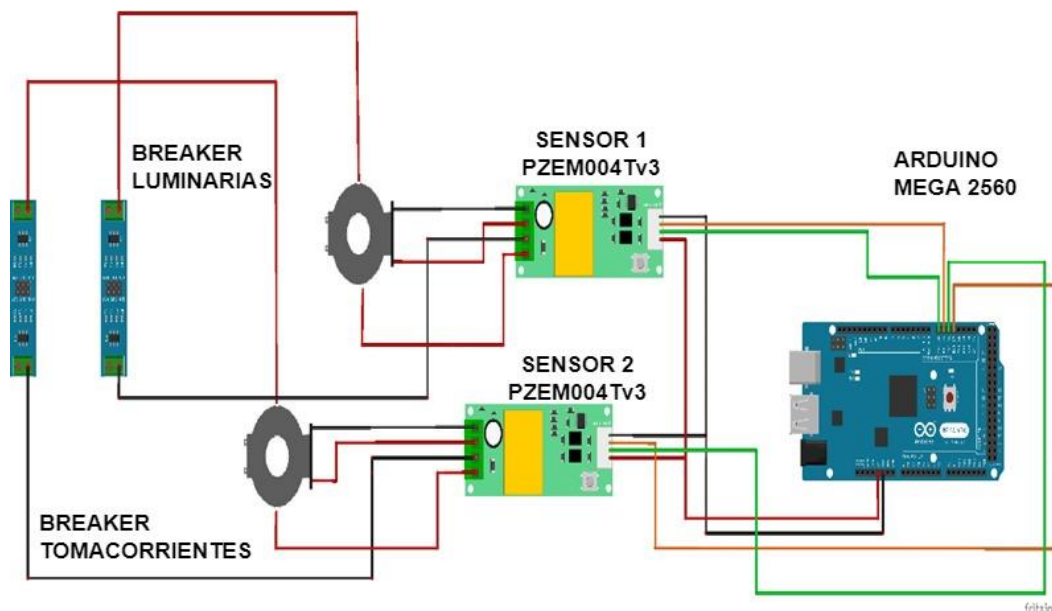


Ilustración 9-3: Esquema de conexión en nivel de censado

Fuente: Lopez. D, 2022

3.4.2. Capa de red

Una vez captada la información de los sensores PZEM004Tv3 por parte del Arduino MEGA2560, se procede a enviar los datos al ESP8266 por medio del protocolo HTTP 1.1 hacia el servidor instalado en la computadora portátil HP, la cual recibe y procesa los datos en NODE RED. La Ilustración 10-3 presenta el esquema de conexiones entre el Arduino Mega 2560 R3 con el NODE MCU ESP8266 12F a través de la comunicación serial por el puerto Serial 0 de ambos dispositivos.

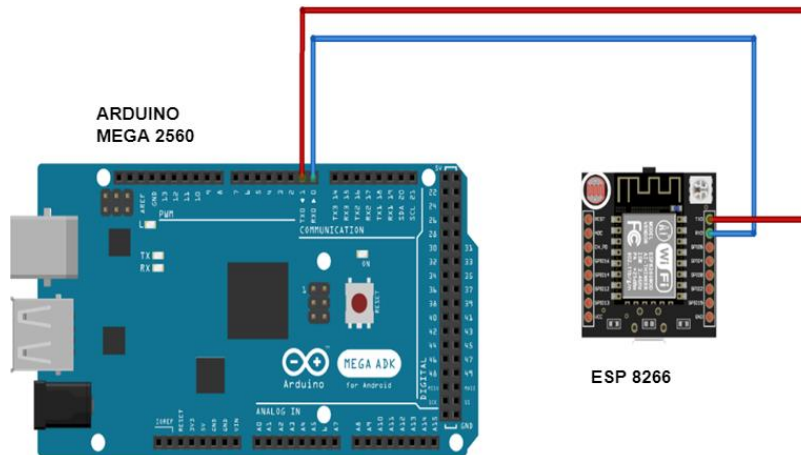


Ilustración 10-3: Esquema de comunicación serial Arduino-ESP8266.

Fuente: Lopez, D, 2022

3.4.3. Capa de aplicación

El servidor de Node-Red se encarga de la recepción y análisis de los datos enviados por el Node MCU los cuales son receptados en un formato String. Para manipular la información de mejor manera se realiza la conversión de formato de String a JavaScript Objet Notation (JSON), de tal forma que se puedan extraer las diferentes propiedades del objeto y asignarlas a las diferentes variables de tiempo real, horario, diario y mensual. Estas variables serán mostradas en el dashboard y a su vez serán almacenadas en el servidor de bases de datos local XAMPP a su respectiva tabla; y finalmente el usuario podrá acceder desde cualquier dispositivo final ya sea por un navegador o por Node-Red Remote la cual se conecta al servidor y muestra la información gracias al uso de APIs.

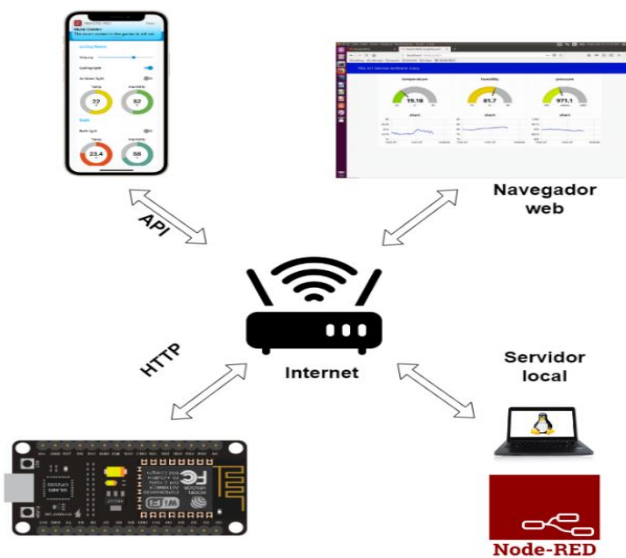


Ilustración 11-3: Esquema comunicacional del sistema de monitoreo IoT

Fuente: López, D; 2022

3.5. Simulación del prototipo del sistema de monitoreo energético IoT

Para la simulación del entorno se empleó el simulador de entornos IoT Packet Tracer, debido a las limitaciones del software solo se verifico pruebas de conectividad o ping. A continuación, la Ilustración 12-3 presenta el escenario del sistema de monitoreo IoT.

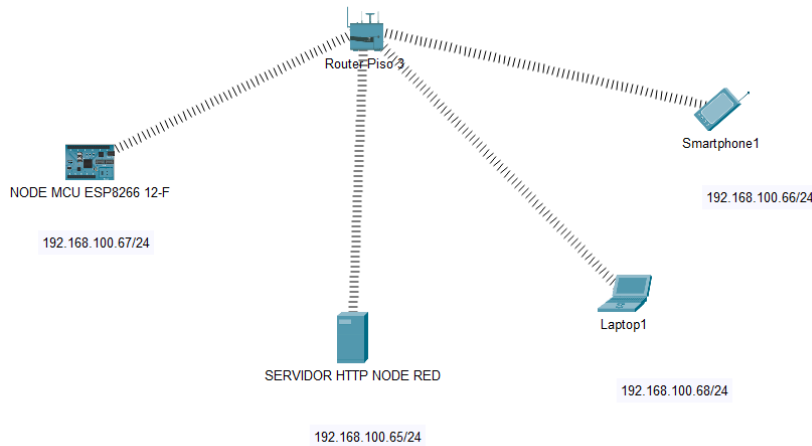


Ilustración 12-3: Escenario del sistema de monitoreo IoT en Packet Tracer

Fuente: López, D; 2022

Una vez realizado la conexión y configuración de direccionamiento en el simulador, se proceden a realizar pruebas de conectividad desde el Node MCU hacia el servidor de Node-Red y de uno de los dispositivos finales hacia el servidor de Node-Red. La figura 13-3 muestra el resultado de la prueba de conectividad del sistema de monitoreo IoT.

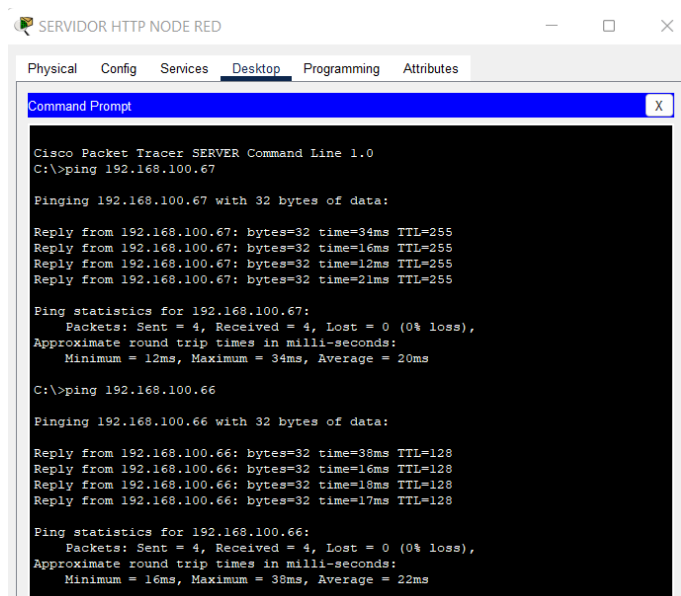


Ilustración 13-3: Resultado de prueba de ping del sistema de monitoreo IoT

Fuente: López, D; 2022

3.6. Implementación del prototipo IoT

En base a los resultados obtenidos en la simulación mostrada en la ilustración 13-3, se procede a la fase de implementación del sistema de monitoreo IoT.

3.6.1. Conexión física del hardware

Tomando como referencia la ilustración 9-3 se procede a conectar los sensores PZEM004Tv3.0 hacia los breakers del piso 3, seguidamente se conectan las tarjetas de desarrollo vía serial tal y como lo indica la ilustración 10-3.

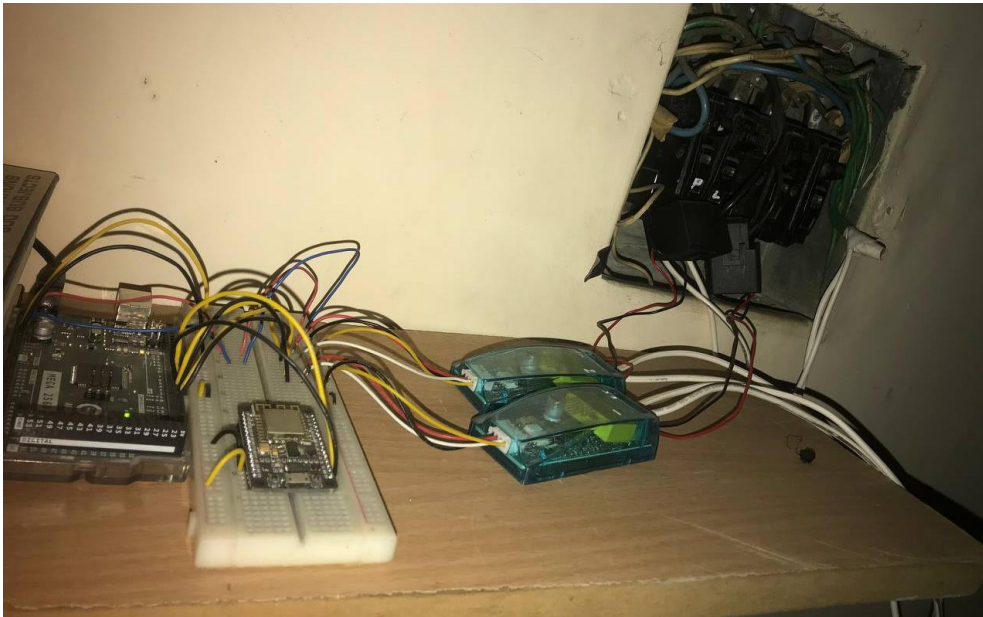


Ilustración 14-3: Implementación del prototipo del sistema de monitoreo de consumo energético.

Fuente: López, D; 2022

3.6.2. Programación de tarjetas de desarrollo

La ilustración 15-3 muestra el diagrama de procesos ejecutados en la tarjeta Arduino mega 2560.

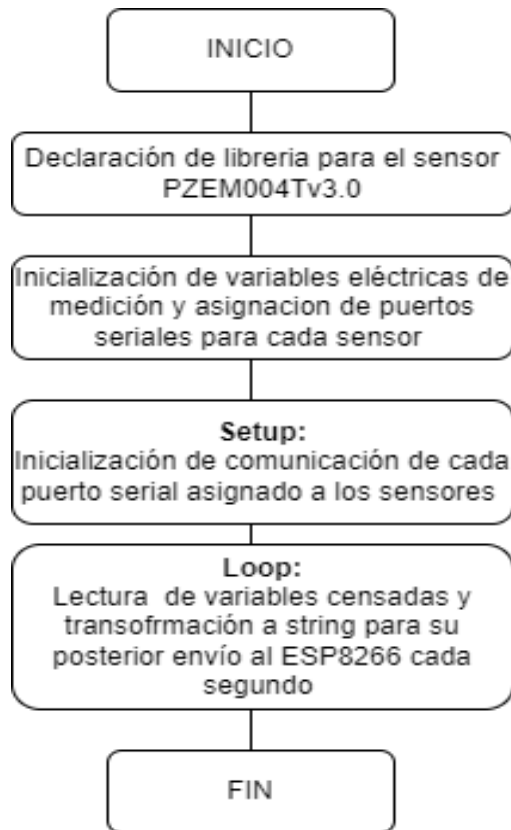


Ilustración 15-3: Diagrama de procesos en Arduino Mega 2560

Fuente: López, D; 2022

A continuación, se detallan las librerías y funciones en la tarjeta de desarrollo Arduino Mega 2560 de manera breve.

Librería empleada:

- **PZEM004Tv3.0.h:** Permite el monitoreo de las variables eléctricas como voltaje, corriente, potencia, energía consumida, factor de potencia y frecuencia mediante la comunicación serial con el sensor PZEM004Tv3.0

Variables y constantes globales:

- **add_pz1&2:** Constante e identificador único de cada sensor PZEM004Tv3.0, escrita en formato hexadecimal.
- **v, i, p, e, fr1&2:** Variables eléctricas tipo flotante de voltaje, corriente, energía consumida y frecuencias captadas por los sensores.
- **timmerDelay:** Constante en milisegundos, la cual define el tiempo de muestreo de las variables eléctricas.

Setup:

- **Serial.begin ():** Inicializa el puerto serial 0 y comunicación con monitor en computadora
- **Serial2.begin():** Inicializa el puerto serial 2
- **Serial3.begin():** Inicializa el puerto serial 3

Loop:

- **PZEM004Tv30. pzem#(Serial#):** Inicializa el sensor PZEM004Tv3.0 en el puerto serial asignado.
- **pzem#. setAddress():** Asigna el identificador previamente definido como dirección del sensor
- **pzem#. voltage():** Lectura del voltaje captado por el sensor PZEM004Tv3.0 en Volts.
- **pzem#. current():** Lectura de la corriente captada por el sensor PZEM004Tv3.0 en Amperes.
- **pzem#. power():** Lectura de la potencia captada por el sensor PZEM004Tv3.0 en Watt.
- **pzem#. energy():** Lectura de la energía captada por el sensor PZEM004Tv3.0 en KWh.
- **Serial.print("data"):** Envía los datos censados a través del Serial0 hacia el ESP8266 mediante una cadena de texto donde cada dato estará separado por un “!”.

La ilustración 16-3 muestra el diagrama de procesos ejecutados en la tarjeta Arduino mega 2560.

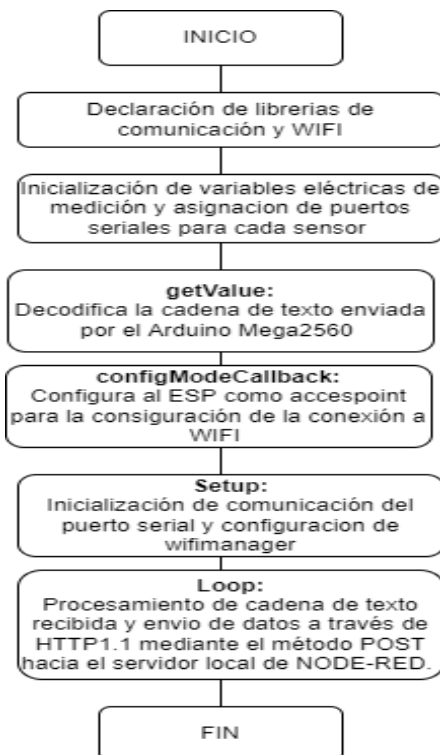


Ilustración 16-3: Diagrama de procesos involucrados en el ESP8266

Fuente: López, D;2022

Librería empleada:

- **ESP8266WiFi.h:** Permite el funcionamiento como del ESP como una estación, por lo que podemos conectarlo a la red WiFi. Y también puede funcionar como un punto de acceso wireless (SoftAP), para establecer su propia red WiFi.
- **DNSServer.h:** Redirige el tráfico a la página web o portal donde configurar el WiFi. El servidor web se encargará de proporcionar la página web donde introducirás el SSID y la contraseña de la red WiFi.
- **WebServer.h:** Simplificar la creación de un servidor web utilizando un módulo ESP8266.
- **String.h:** Permite el manejo de cadenas: copiar, cambiar caracteres, comparar cadenas, etc; a través de funciones.
- **ESP8266HTTPClient.h:** Permite la interacción con otros endpoints usando verbos del protocolo HTTP.
- **WiFiManager.h:** Permite la conexión a la red WiFi deseada sin tener que introducir manualmente el código mediante el IDE de Arduino.

Variables y constantes globales:

- **sen, vo, in, po, en, fre1&2:** Variables tipo string de valores decodificados al recibir la cadena de texto.
- **url:** Variable que detalla el localizador de recursos uniforme formado el protocolo, host y recurso.

getValue:

- Función que decodifica la cadena de texto ingresada al ESP8266 via comunicación serial, la cual identifica el separador “!”, extrayendo el valor de cada variable y asignándola a su respectiva variable tipo string previamente mencionada.

configModeCallback:

- Función que permite la configuración del modo access point del ESP para la selección de la red a conectarse.

Setup:

- **Serial.begin ():** Inicializa el puerto serial que va conectado al Arduino Mega 2560
- **wifiManager.setAPCallback(configModeCallback):** Configura las instancias de wifi manager.

Loop:

- **WiFiClient client:** Inicializa el cliente Wifi.
- **HTTPClient http:** Inicializa el cliente Http.
- **http.begin(client, url):** Especifica el destino de la petición POST
- **http.addHeader("Content-Type","application/x-www-form-urlencoded"):** Especifica que la petición que será enviada contendrá informacion de texto plano.
- **datos_a_enviar:** Variable tipo string que contiene a todas las variables eléctricas concatenadas a ser enviadas al servidor.
- **codigo_respuesta:** Variable tipo int que devuelve la respuesta del servidor de Node-Red, 200 para éxito y diferente a 200 como fracaso.
- **http.end():** Finaliza la comunicación via HTTP y libera recursos.

La ilustración 17-3 muestra el tráfico enviado por el ESP8266 hacia el servidor de NODE-RED vía HTTP POST.

The screenshot shows a Wireshark capture of network traffic. The main pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. Packet 169 is highlighted, showing a POST request to 'update-sensor' with a body of form-encoded data. The details pane for packet 169 shows the following structure:

- Frame 13: 169 bytes on wire (1352 bits), 169 bytes captured (1352 bits) on interface WavicleMPC_ [D5059510-81FF-4582-9483-ECF507308E65], id 0
- Ethernet II, Src: Express13:da:2b:98 (cd:ac:3f:da:2b), Dst: DESKTOP-GPKQTHJ.local (3c:1b:b5:4b:bc:c3)
- Internet Protocol Version 4, Src: ESP-3FDA2B (192.168.1.121), Dst: DESKTOP-GPKQTHJ.local (192.168.1.136)
- Transmission Control Protocol, Src Port: 50420 (50420), Dst Port: vsat-control (1880), Seq: 235, Ack: 1, Len: 115
- Hypertext Transfer Protocol
- HTML Form URL Encoded: application/x-www-form-urlencoded
 - Form item: "sen1" = "1"
 - Form item: "voi" = "112.23"
 - Form item: "in1" = "0.50"
 - Form item: "pot1" = "50.12"
 - Form item: "en1" = "50.12"
 - Form item: "frec1" = "60.00"
 - Form item: "sen2" = "2"
 - Form item: "vo2" = "121.37"
 - Form item: "in2" = "1.50"
 - Form item: "po2" = "22.50"
 - Form item: "en2" = "22.60"
 - Form item: "frec2" = "59.60"

Ilustración 17-3: Información enviada desde ESP8266-NODE-RED vía HTTP POST.

Fuente: López, D; 2022

3.6.3. Programación del servidor

La ilustración 18-3 muestra el diagrama de procesos a realizarse en la computadora HP Pavilion para su funcionamiento como servidor.



Ilustración 18-3: Diagrama de procesos a ejecutar en el servidor local

Fuente: López, D; 202

Instalación del Sistema Operativo:

Debido a las características de hardware de la computadora seleccionada previamente se optó por la instalación de Ubuntu 20.0.4 para mejorar el rendimiento de esta. La instalación del sistema operativo se llevó a cabo mediante el uso de una imagen ISO en una memoria flash.

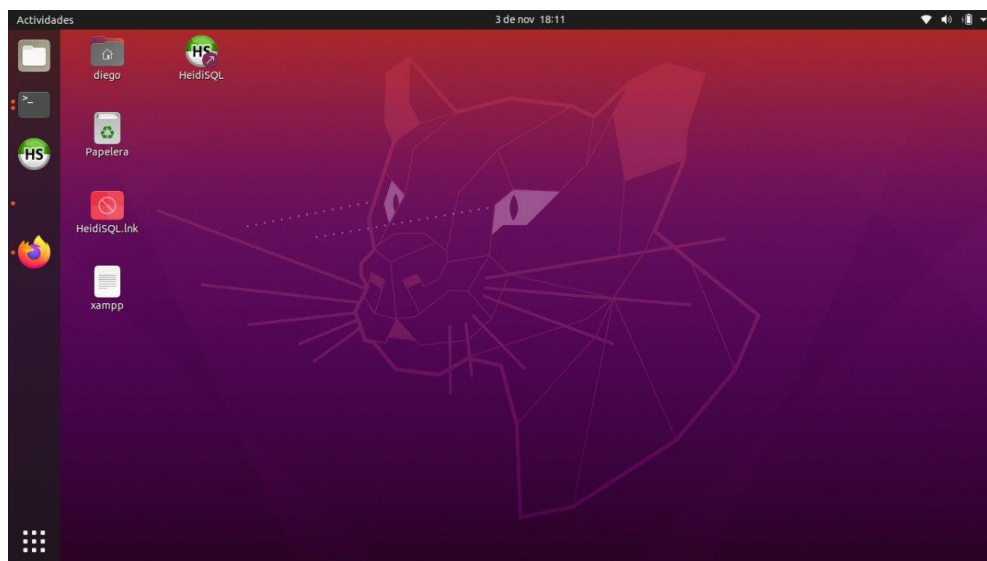


Ilustración 19-3: Sistema operativo Ubuntu 20.0.4 instalado en la computadora HP Pavilion

Fuente: López, D; 2022

Instalación del servidor basado en software:

Una vez instalado el sistema operativo se procede a la instalación de XAMPP, principalmente por el servicio de base de datos de Maria DB que ofrece el software. La instalación del software se llevó a cabo por las siguientes etapas.

- Descarga del software ejecutable de instalación a través del navegador web en base a la arquitectura del computador (64bits).
- Asignación de permisos de lectura, ejecución y escritura al archivo mediante la ventana de comandos (chmod 777).
- Asignación de dirección IP estática mediante DHCP en el router.
- Ejecución del archivo de instalación de XAMPP.

Se asigno la IP estática 192.168.100.65/24 en base a la dirección MAC del equipo HP Pavilion, para evitar conflictos en la ejecución del software XAMPP.

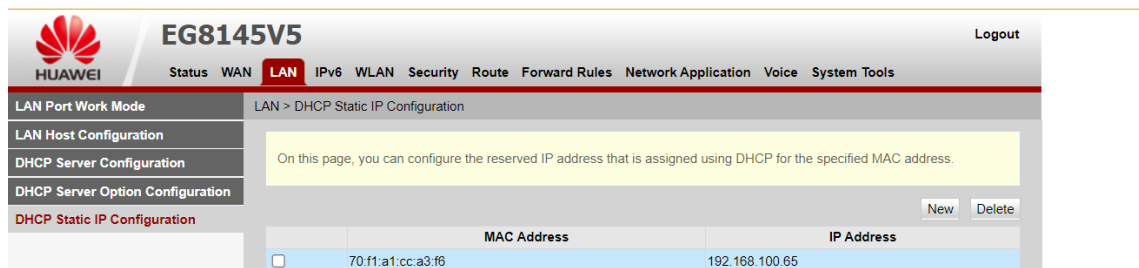


Ilustración 20-3: Asignación de dirección IP estática en el router de la vivienda del piso 3.

Fuente: López, D; 2022

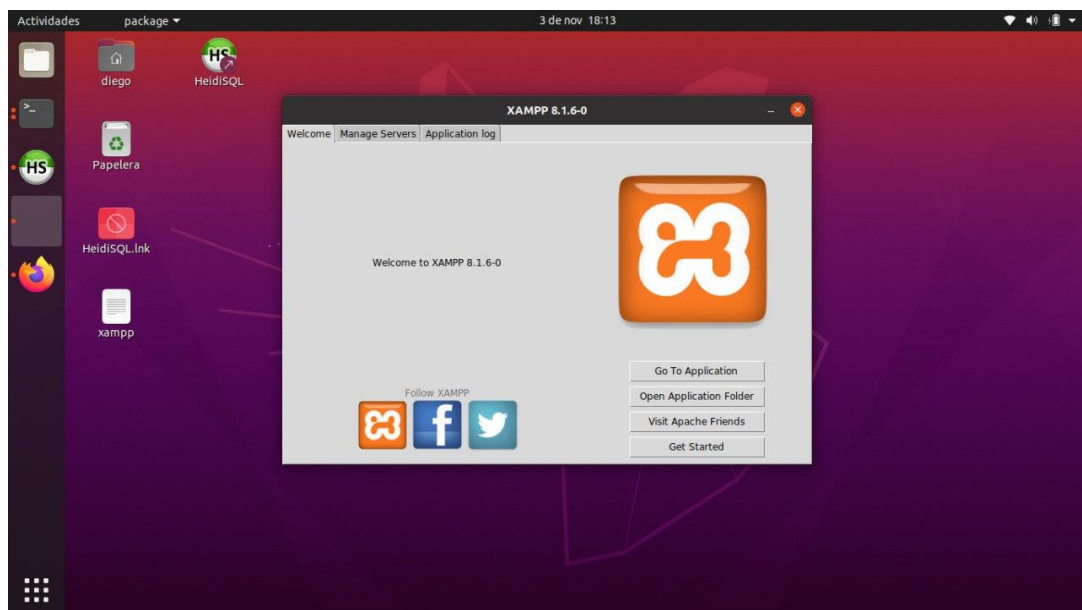


Ilustración 21-3: Servidor XAMPP instalado en la computadora HP Pavilion

Fuente: López, D; 2022

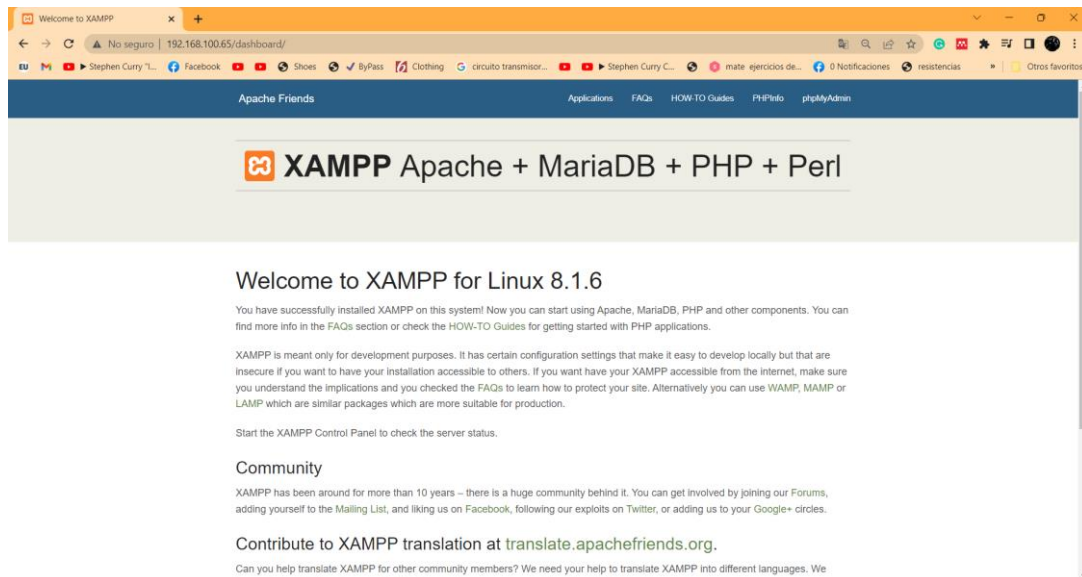


Ilustración 22-3: Interfaz del panel de servicios de XAMPP.

Fuente: López, D; 202

Configuración de la base de datos:

Una vez instalado el servidor XAMPP se procede a la creación de la base de datos, esta configuración se la realiza accediendo al panel de servicios, en la opción “phpMyAdmin”. Una vez en el servicio de base de datos, se crea la nueva base de datos de nombre “medidor_pz” la cual contendrá cinco tablas, que son: estado_actual, historial_hora, historial_dia, historial_mes, historial_año.

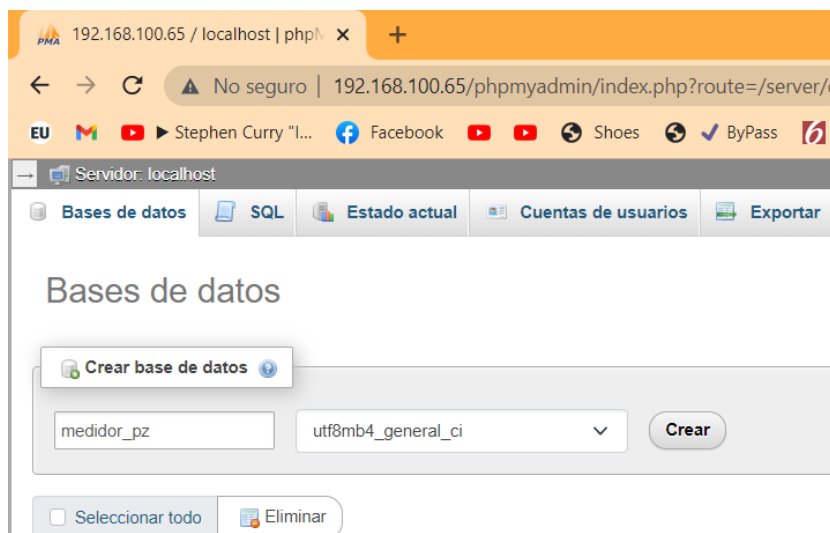


Ilustración 23-3: Creación de nueva base de datos en phpMyAdmin.

Fuente: López, D; 2022

Ya creada la base de datos se procede a crear cada una de sus tablas, especificando el número de columnas que serán los atributos de cada una.

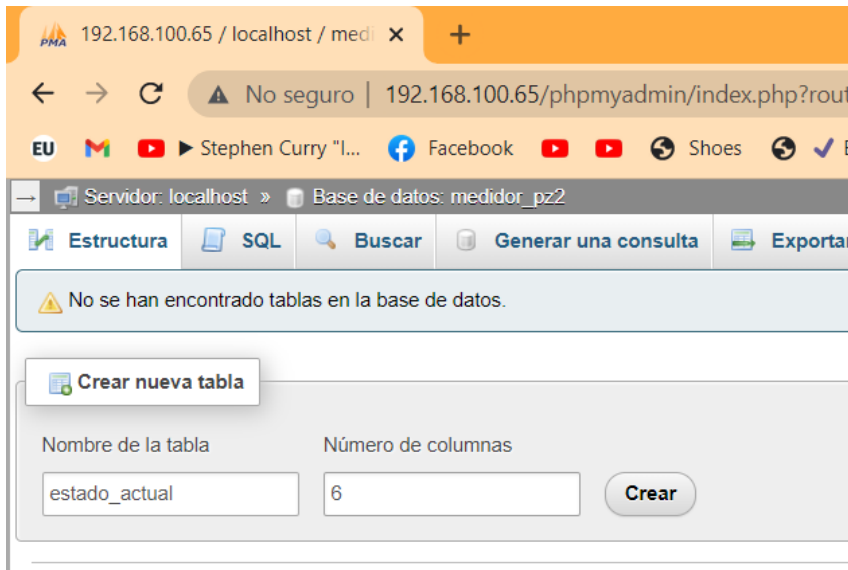


Ilustración 24-3: Creación las diferentes tablas de la base de datos en phpMyAdmin.

Fuente: López, D; 2022

Una vez creadas todas las tablas, se procede a relacionarlas mediante el establecimiento de llaves primarias y foráneas, la ilustración 25-3 presenta el diagrama entidad relación de la base de datos “medidor_pz”.

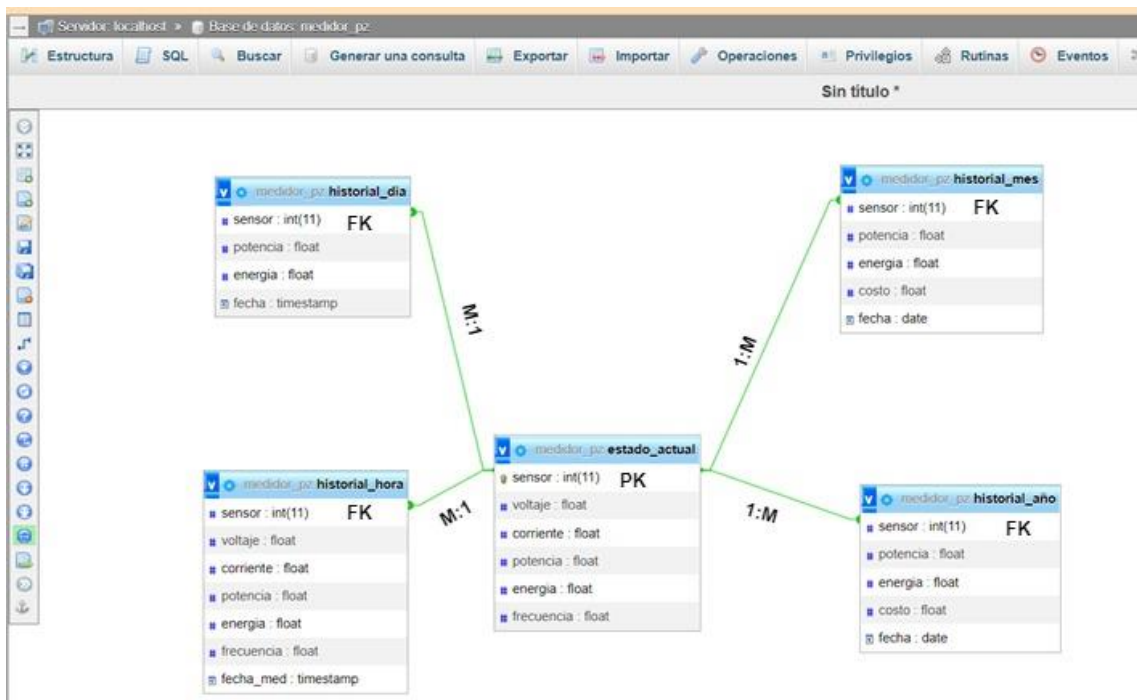


Ilustración 25-3: Diagrama entidad relación de la base de datos “medidor_pz”

Realizado por: Lopez, D; 2022

Seguidamente se procede a registrar los sensores involucrados en el sistema de monitoreo del consumo energético, donde “1” denota al sensor ubicado en el breaker de las luminarias y “2” al

sensor ubicado en el de los tomacorrientes, esta tabla ira actualizándose periódicamente cada 1 segundo.

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0.0006 segundos.)

SELECT * FROM `estado_actual`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la

Opciones extra

	sensor	voltaje	corriente	potencia	energia	frecuencia
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	126.3	0.02	0	11.84	59.9
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	126.3	0.17	10.6	115.23	59.9

Ilustración 26-3: Registro manual de sensores 1 y 2 en la tabla “estado_actual”

Realizado por: Lopez, D; 2022

Instalación del entorno de desarrollo IoT

Seguidamente se procede a instalar el servidor de Node-Red a través del uso de comandos en la terminal de Linux, se debe asegurar tener conexión a internet para ello. Adicionalmente se debe instalar Node.js como npm para el correcto funcionamiento del servidor.

Ya concluidas las acciones mencionadas se procede a iniciar el servicio con el comando “node-red-start”, en caso de detención “node-red-stop” y finalmente en caso de reinicio “node-red-restart”.

```

diego@diego-HP-G42-Notebook-PC:~$ node-red-start
Start Node-RED
Once Node-RED has started, point a browser at http://192.168.100.65:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop           to stop Node-RED
Use node-red-start         to start Node-RED again
Use node-red-log           to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org
[sudo] contraseña para diego:
Starting as a systemd service.
  
```

Ilustración 27-3: Inicialización del servidor Node-Red

Realizado por: Lopez, D; 2022

Configuración del sistema de monitoreo del consumo energético

Se procede a la programación de los diferentes nodos que conformaran el sistema, la ilustración 28-3 detalla el diagrama de procesos involucrados en esta fase.

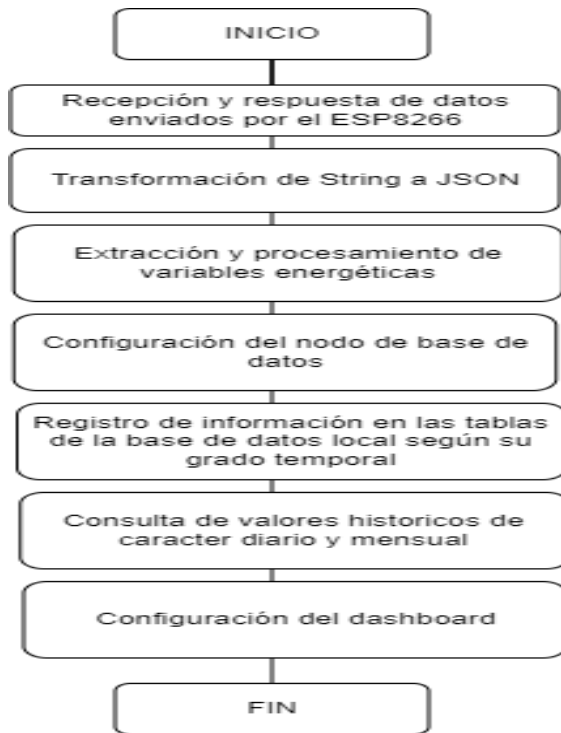


Ilustración 28-3: Inicialización del servidor Node-Red

Realizado por: Lopez, D; 2022

Configuración de la entrada de datos enviados por el ESP8266

A continuación, se detallan brevemente los nodos mostrados en la ilustración 29-3.

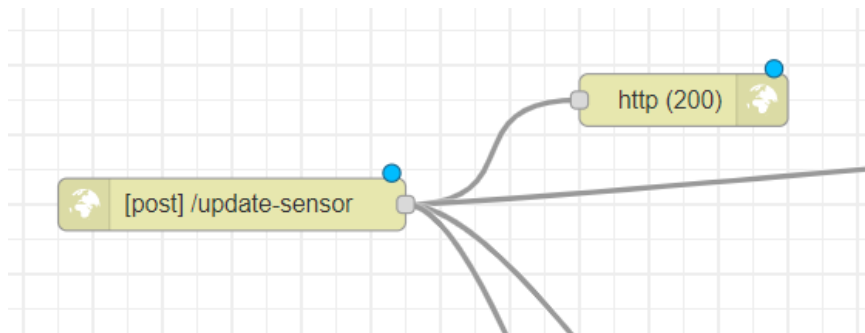


Ilustración 29-3. Nodos de recepción y confirmación de recibimiento de datos

Realizado por: Lopez, D; 2022

- **http in:** Crea una instancia en el servidor con el nombre "update-sensor", que recibirá toda la información enviada por el ESP8266 vía HTTP POST.
- **http response:** Devuelve una respuesta al ESP8266, indicando una recepción exitosa o recepción fallida.

Transformación de String a Json

Ya receptados los datos en un formato de cadena de texto o String, se procede a transformarlos a una notación JSON o JavaScript Object Notation, por motivos de facilidad en la manipulación de

datos. Este nodo se configuro con los parámetros mostrados en la Ilustración 30-3.

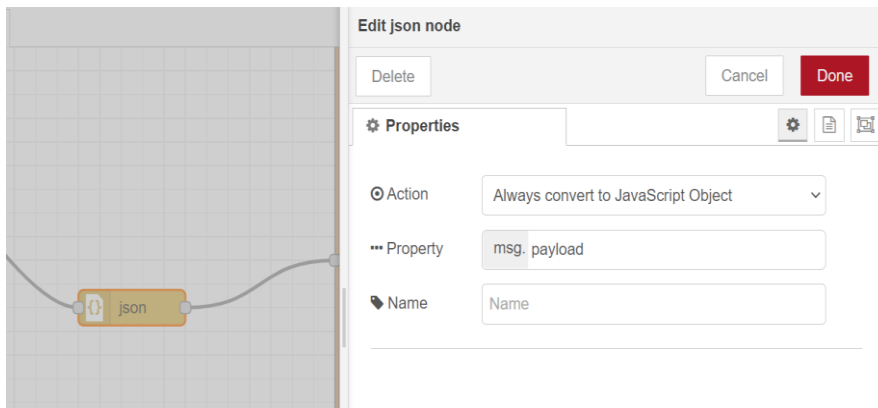


Ilustración 30-3: Nodos de recepción y confirmación de recibimiento de datos

Realizado por: Lopez, D; 2022

Extracción y procesamiento de variables energéticas

Para esta fase se empleó el nodo function en el cual se programa en lenguaje JavaScript, la extracción y asignación de los valores de las variables recibidas hacia variables de Flow. Estas variables serán accesibles en toda la pestaña o Flow y posteriormente son enviadas hacia las instancias del dashboard.

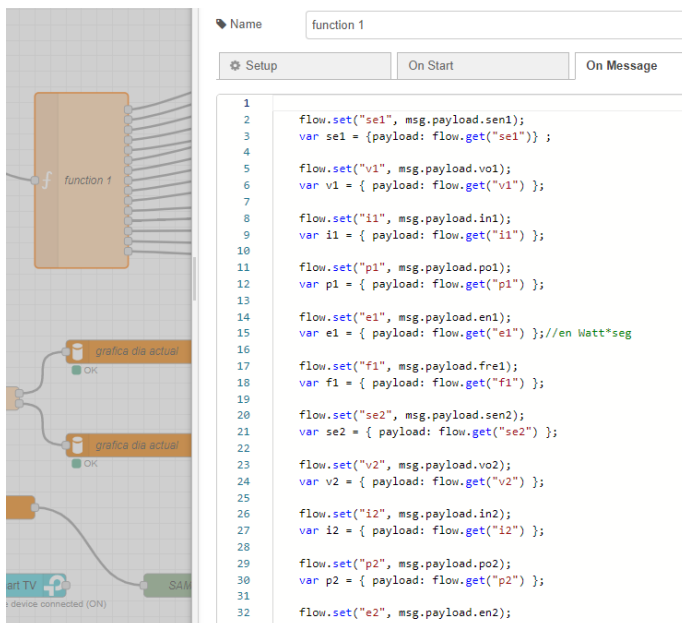


Ilustración 31-3: Nodo de extracción y procesamiento de variables energéticas

Realizado por: Lopez, D; 2022

Configuración del nodo base de datos

Este nodo permite la comunicación lógica entre el servidor Node-Red y la base de datos local instalada en el computador. De esta forma se escriben los datos recopilados en las diferentes tablas de la base de datos.

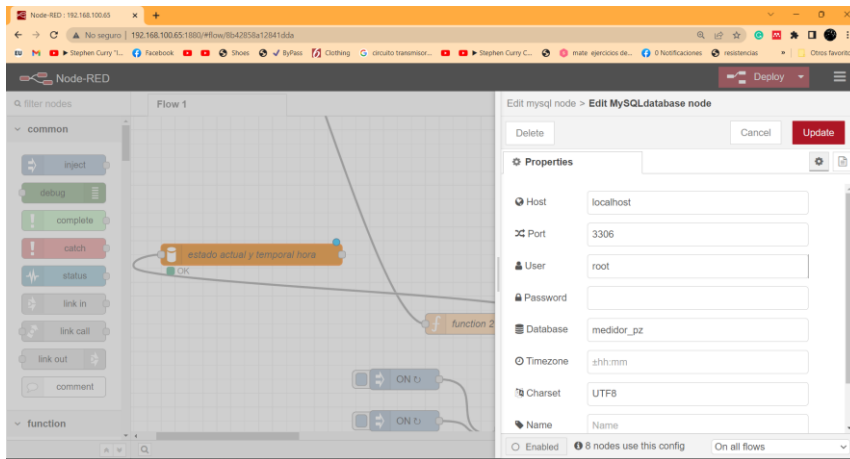


Ilustración 32-3: Nodo de base de datos SQL

Realizado por: Lopez, D; 2022

Registro de información en las tablas de la base de datos

Mediante el uso de disparadores temporales, se programa un evento cada hora, día y mes. Cada evento generara una consulta en la base de datos la cual responderá con la información solicitada y la almacenara en una variable tipo Flow para mayor facilidad en el procesamiento de los datos y visualización en el dashboard.

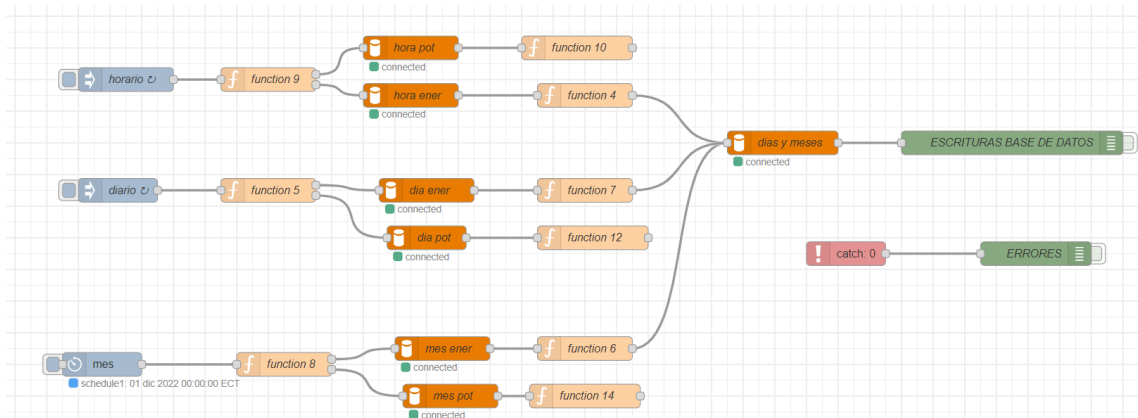


Ilustración 33-3: Nodos de consulta, escritura de promedios de potencia y energía consumida horaria, diaria y mensual

Realizado por: Lopez, D; 2022

Name: function 9

Setup On Start On Message On Stop

```

1 var sql1 = "";
2 var sql2 = "";
3 var output1 = [];
4 var output2 = [];
5 sql1 = "SELECT ROUND(AVG( potencia),2) AS 'pot_h' FROM historial_hora GROUP BY sensor";
6 output1.push({ topic: sql1 });
7
8 sql2 = "SELECT (SUM( energia) / (3600*1000)) AS 'ener_h' FROM historial_hora GROUP BY sensor";
9 output2.push({ topic: sql2 });
10
11
12 return [output1, output2];
13

```

Ilustración 34-3: Funciones de consulta horaria de parámetros eléctricos.

Realizado por: Lopez, D; 2022

Al accionarse el evento horario Node-Red envía dos consultas a la base de datos para la suma de todas las potencias y energía consumida captada en 1 hora. Cabe resaltar que la energía consumida captada por el sensor cada segundo está dada en Wattios por segundo, por ello se realiza la conversión a Wattios por hora, como se detalla en la Ilustración 34-3. Una vez obtenida la consulta, se procede a asignar los resultados a variables tipo Flow, Ilustración 35-3.

Name: function 10

Setup On Start On Message

```

1
2
3 flow.set("p1_h", msg.payload[0].pot_h);
4 var p1_h = {payload: flow.get("p1_h")};
5 flow.set("p2_h", msg.payload[1].pot_h);
6 var p2_h = {payload: flow.get("p2_h")};
7
8
9 return msg;

```

Ilustración 35-3: Asignación de parámetros consultados a variables tipo flow.

Realizado por: Lopez, D; 2022

Posteriormente se realiza la inserción de los datos almacenados en las variables tipo Flow hacia la tabla “historial_dia” de la base de datos, obteniendo así el consumo de una hora en el día.

```

Name: function 4
Setup On Start On Message On Stop
1 var sql = "";
2 var outputs = [];
3 var se1_h = flow.get("se1");
4 //var p1_d = flow.get("p1");
5 var se2_h = flow.get("se2");
6 //var p2_d = flow.get("p2");
7
8 flow.set("e1_h", msg.payload[0].ener_h);
9 var e1_h = flow.get("e1_h");
10 flow.set("e2_h", msg.payload[1].ener_h);
11 var e2_h = flow.get("e2_h");
12
13 var p1_h = flow.get("p1_h");
14
15 var p2_h = flow.get("p2_h");
16 //INSERT INTO historial_dia(sensor, potencia, energia, fecha) VALUES(1, 2, 3, DATE_ADD(CURRENT_TIMESTAMP(), INTERVAL - 1 HOUR));
17 sql = "INSERT INTO historial_dia(sensor, potencia, energia, fecha) VALUES (" + se1_h + "," + p1_h + "," + e1_h + ", DATE_ADD(CURRENT_TIMESTAMP(), INTERVAL - 1 HOUR) )";
18 outputs.push({ topic: sql });
19
20 sql = "INSERT INTO historial_dia(sensor, potencia, energia, fecha) VALUES (" + se2_h + "," + p2_h + "," + e2_h + ", DATE_ADD(CURRENT_TIMESTAMP(), INTERVAL - 1 HOUR) )";
21 outputs.push({ topic: sql });
22
23 sql = "TRUNCATE TABLE historial_hora";
24 outputs.push({ topic: sql });
25
26 return [outputs];

```

Ilustración 36-3: Inserción de los parámetros horarios hacia la tabla “historial_dia”.

Realizado por: Lopez, D; 2022

Una vez completadas las 24 inserciones en la tabla historial_dia, se procede a realizar la suma de la energía consumida y el promedio de las potencias captadas en el día a través de una consulta hacia la base de datos; las cuales se muestran en la Ilustración 37-3.

```

Name: function 5
Setup On Start On Message On Stop
1 var sql1 = "";
2 var output1 = [];
3 var sql2 = "";
4 var output2 = [];
5 sql1 = "SELECT ROUND( SUM( energia), 3 ) AS 'ener_d' FROM historial_dia WHERE DAY(fecha)=(DAY(CURDATE()-1) GROUP BY sensor";
6 output1.push({ topic: sql1 });
7 sql2 = "SELECT ROUND(AVG( potencia),2) AS 'pot_d' FROM historial_dia WHERE DAY(fecha)=(DAY(CURDATE()-1) GROUP BY sensor";
8 output2.push({ topic: sql2 });
9
10
11 return [output1,output2];
..

```

Ilustración 37-3: Funciones de consulta diaria de parámetros eléctricos.

Realizado por: Lopez, D; 2022

Consecuentemente se asignan los valores obtenidos en la consulta a variables tipo Flow, como se muestra en la Ilustración 38-3.

```

Name: function 12
Setup On Start On Stop
1
2
3 flow.set("p1_d", msg.payload[0].pot_d);
4 var p1_d = {payload: flow.get("p1_d")};
5 flow.set("p2_d", msg.payload[1].pot_d);
6 var p2_d = {payload: flow.get("p2_d")};
7
8
9 return msg;

```

Ilustración 38-3: Asignación de parámetros consultados a variables tipo flow.

Realizado por: Lopez, D; 2022

Se registran los datos almacenados en las variables tipo Flow, en la tabla “historial_mes” de la base de datos, obteniendo así el consumo de un día en el mes.

```

1 var sql = "";
2 var outputs = [];
3 var se1_d = flow.get("se1");
4 var p1_d = flow.get("p1_d");
5 var se2_d = flow.get("se2");
6 var p2_d = flow.get("p2_d");
7
8 flow.set("e1_d", msg.payload[0].ener_d);
9 var e1_d = flow.get("e1_d");
10 flow.set("e2_d", msg.payload[1].ener_d);
11 var e2_d = flow.get("e2_d");
12
13 sql = "INSERT INTO historial_mes (sensor, potencia, energia,costo, fecha) VALUES (" + se1_d + "," + p1_d + "," + e1_d + "," + "ROUND((" + e1_d + " 0.002 ) ,3) , DATE_ADD(CURDATE(), INTERVAL - 1 DAY)";
14 outputs.push({ topic: sql });
15
16 sql = "INSERT INTO historial_mes (sensor, potencia, energia,costo, fecha) VALUES (" + se2_d + "," + p2_d + "," + e2_d + "," + "ROUND((" + e2_d + " 0.002 ) ,3) , DATE_ADD(CURDATE(), INTERVAL -1 DAY) ";
17 outputs.push({ topic: sql });
18
19
20 return [outputs];

```

Ilustración 39-3: Inserción de los parámetros diarios hacia la tabla “historial_mes”.

Realizado por: Lopez, D; 2022

Ya completados todos los días del mes se generará un evento al inicio del nuevo mes, enviando una consulta a la base de datos; obteniendo de igual forma el promedio de potencias y el total de la energía consumida.

```

1 var sql1 = "";
2 var output1 = [];
3 var sql2 = "";
4 var output2 = [];
5
6 sql1 = "SELECT ROUND( SUM( energia), 3 ) AS 'ener_m' FROM historial_mes WHERE MONTH(fecha)=(MONTH(CURDATE()-1) GROUP BY sensor";
7 output1.push({ topic: sql1 });
8
9 sql2 = "SELECT ROUND(AVG( potencia),2) AS 'pot_m' FROM historial_mes WHERE MONTH(fecha)=(MONTH(CURDATE()-1) GROUP BY sensor";
10 output2.push({ topic: sql2 });
11
12 return [output1,output2];
13

```

Ilustración 40-3: Funciones de consulta mensual de parámetros eléctricos.

Realizado por: Lopez, D; 2022

Se extraen los valores consultados hacia la base de datos mediante el uso de variables tipo Flow como se muestra en la Ilustración 41-3.

```

1
2
3 flow.set("p1_m", msg.payload[0].pot_m);
4 var p1_m = {payload: flow.get("p1_m")};
5 flow.set("p2_m", msg.payload[1].pot_m);
6 var p2_m = {payload: flow.get("p2_m")};
7
8
9 return msg;

```

Ilustración 41-3: Asignación de parámetros consultados a variables tipo flow.

Realizado por: Lopez, D; 2022

Finalmente se insertan los valores almacenadores en la base de datos, obteniendo así el consumo realizado en un mes del año. Adicionalmente se realiza en cálculo del costo a pagar por el consumo mensual y lo almacena en la tabla “historial_año”.

```

1 var sql = "";
2 var outputs = [];
3 var se1_m = flow.get("se1");
4 var p1_m = flow.get("p1_m");
5 var se2_m = flow.get("se2");
6 var p2_m = flow.get("p2_m");
7
8 flow.set("e1_m", msg.payload[0].ener_m);
9 var e1_m = flow.get("e1_m");
10 flow.set("e2_m", msg.payload[1].ener_m);
11 var e2_m = flow.get("e2_m");
12
13 sql = "INSERT INTO historial_año (sensor, potencia, energia,costo, fecha) VALUES (" + se1_m + "," + p1_m + "," + e1_m + "," + "ROUND((" + e1_m * 0.0924 + "),2)" + ", DATE_ADD(CURDATE(), INTERVAL -1 MO
14 outputs.push({ topic: sql });
15
16 sql = "INSERT INTO historial_año (sensor, potencia, energia,costo, fecha) VALUES (" + se2_m + "," + p2_m + "," + e2_m + "," + "ROUND((" + e2_m * 0.0924 + "),2)" + ", DATE_ADD(CURDATE(), INTERVAL -1 ME
17 outputs.push({ topic: sql });
18
19
20 return [outputs];

```

Ilustración 42-3: Inserción de los parámetros mensuales hacia la tabla “historial_año”.

Realizado por: Lopez, D; 2022

A continuación, se muestran los resultados en las tablas “estado_actual”, “historial_hora”, “historial_dia”, “historial_mes”, “historial_año” obtenidos al poner en funcionamiento el sistema de monitoreo del consumo energético en el piso 3 de la vivienda previamente descrita.

SMARTMETER\medidor_pz\estado_actual\ - HeidiSQL 12.0.0.6468

Host: 192.168.100.65 Base de datos: medidor_pz Tabla: estado_actual

medidor_pz.estado_actual: 2 filas en total (aproximadamente)

sensor	voltaje	corriente	potencia	energia	frecuencia
1	125,7	0,02	0	14,92	59,9
2	125,8	2,03	237,9	144,34	60

Ilustración 43-3: Resultados de variables censadas en la tabla “estado_actual”.

Realizado por: Lopez, D; 2022

SMARTMETER\medidor_pz\historial_hora\ - HeidiSQL 12.0.0.6468

Host: 192.168.100.65 Base de datos: medidor_pz Tabla: historial_hora

medidor_pz.historial_hora: 4.088 filas en total (aproximadamente), limitado a 1.000

sensor	voltaje	corriente	potencia	energia	frecuencia	fecha_med
1	128,4	0,02	0	14,92	59,9	2022-11-17 00:40:36
2	128,5	0,33	21,3	144,1	59,9	2022-11-17 00:40:36
2	128,5	0,33	20,9	144,1	59,9	2022-11-17 00:40:34
1	128,4	0,02	0	14,92	59,9	2022-11-17 00:40:34
1	128,5	0,02	0	14,92	59,9	2022-11-17 00:40:33
2	128,5	0,33	20,9	144,1	59,9	2022-11-17 00:40:33
2	128,6	0,33	21,5	144,1	59,9	2022-11-17 00:40:32
1	128,5	0,02	0	14,92	59,9	2022-11-17 00:40:32
2	128,5	0,33	21	144,1	59,9	2022-11-17 00:40:31
1	128,4	0,02	0	14,92	59,9	2022-11-17 00:40:31
1	128,4	0,02	0	14,92	59,9	2022-11-17 00:40:30
2	128,5	0,35	22,9	144,1	59,9	2022-11-17 00:40:30

Ilustración 44-3: Resultados de variables censadas en la tabla “historial_hora”.

Realizado por: Lopez, D; 2022

sen...	potencia	energia	fecha
1	0	0,0127227	2022-11-16 02:00:00
2	23,27	0,122229	2022-11-16 02:00:00
1	3,16	0,011668	2022-11-16 01:00:00
2	40,64	0,112119	2022-11-16 01:00:00
1	6,74	0,00943108	2022-11-16 00:00:00
2	23,81	0,0913612	2022-11-16 00:00:00
1	1,42	0,0124083	2022-11-15 23:00:00
2	21,95	0,119205	2022-11-15 23:00:00
1	1,42	0,0122347	2022-11-15 22:00:00
2	21,95	0,117519	2022-11-15 22:00:00
1	0,34	0,0120445	2022-11-15 21:00:00
2	22,25	0,115674	2022-11-15 21:00:00
1	0,34	0,0124035	2022-11-15 20:00:00

Ilustración 45-3: Resultados de variables censadas en la tabla “historial_dia”.

Realizado por: Lopez, D; 2022

sensor	potencia	energia	costo	fecha
2	59,21	0,708	0,065	2022-08-19
1	4,23	0,076	0,007	2022-08-19
2	62,33	0,184	0,017	2022-08-18
1	0,75	0,02	0,002	2022-08-18
1	0,75	0,031	0,003	2022-08-14
2	62,33	0,271	0,025	2022-08-14
1	3,71	0,062	0,006	2022-08-13
2	47,46	0,531	0,049	2022-08-13
1	6,06	0,059	0,005	2022-08-12
2	76,96	0,507	0,047	2022-08-12
2	76,96	0,503	0,046	2022-08-11

Ilustración 46-3: Resultados de variables censadas en la tabla “historial_mes”.

Realizado por: Lopez, D; 2022

sensor	potencia	energia	costo	fecha
1	33,64	0,59	0,05	2022-07-01
2	231,33	2,27	0,21	2022-07-01
1	5,37	1,626	0,15	2022-08-01
2	63,56	12,194	1,12	2022-08-01
1	143.769.000	4,098	0,38	2022-09-01
2	259.395.000	37,481	3,45	2022-09-01

Ilustración 47-3: Resultados de variables censadas en la tabla “historial_año”.

Realizado por: Lopez, D; 2022

Consulta de valores históricos de carácter diario y mensual

La presente sección permite al usuario obtener reportes de la información almacenada en la base de datos de forma diaria y mensual. El reporte de carácter diario presenta el valor a pagar por el consumo realizado por las luminarias, tomacorrientes y total en el día seleccionado, adicionalmente una gráfica de barras que representan el consumo en KWh por cada hora del día.

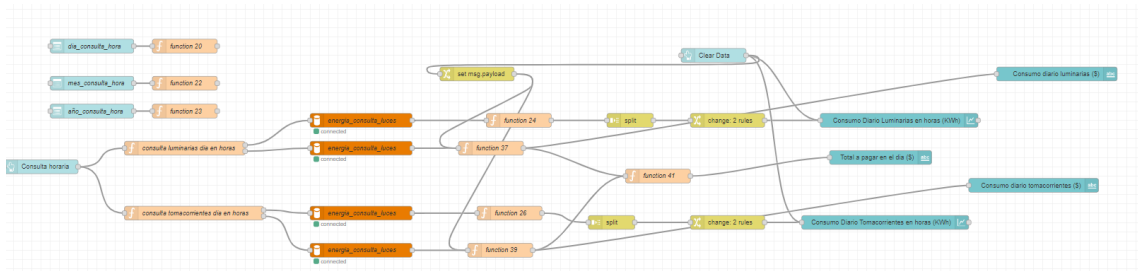


Ilustración 48-3: Esquema de consulta diaria del consumo energético a la base de datos "medidor_pz".

Realizado por: Lopez, D; 2022

Finalmente, el reporte mensual refleja de igual forma los valores a pagar por luminarias, tomacorrientes y total; así como sus respectivas graficas de barras en las cuales se representa el consumo respectivo de cada día del mes seleccionado.

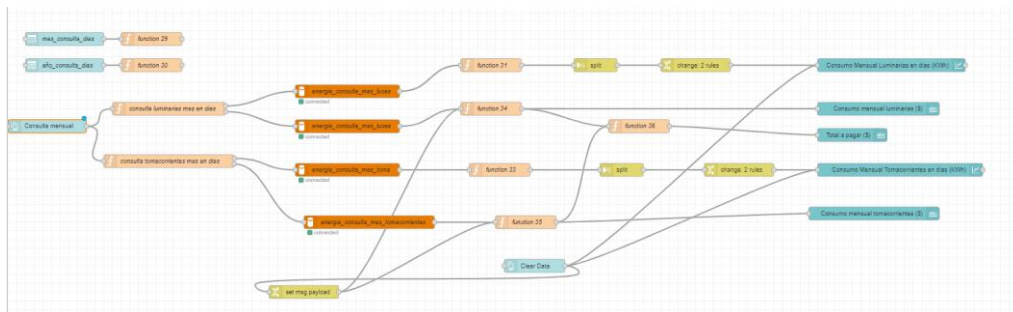


Ilustración 49-3: Esquema de consulta mensual del consumo energético a la base de datos "medidor_pz".

Realizado por: Lopez, D; 2022

Configuración del dashboard

En la opción dashboard se configuraron seis pestañas con sus respectivos grupos, los cuales representan los atributos a ser mostrados y controlados mediante el dashboard.

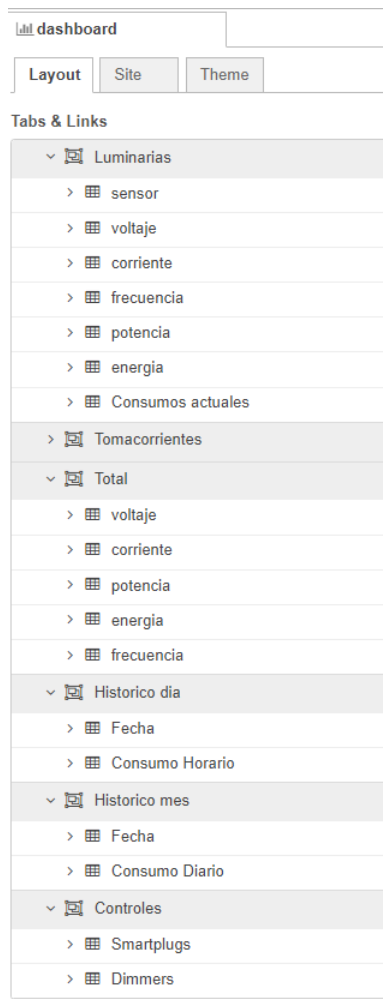


Ilustración 50-3: Configuración del dashboard del sistema de monitoreo del consumo energético

Realizado por: Lopez, D; 2022

3.7. Diseño del sistema de control del consumo energético

La ilustración 51-3 describe las etapas del sistema de control del consumo energético, donde cada etapa se detallará brevemente a continuación.

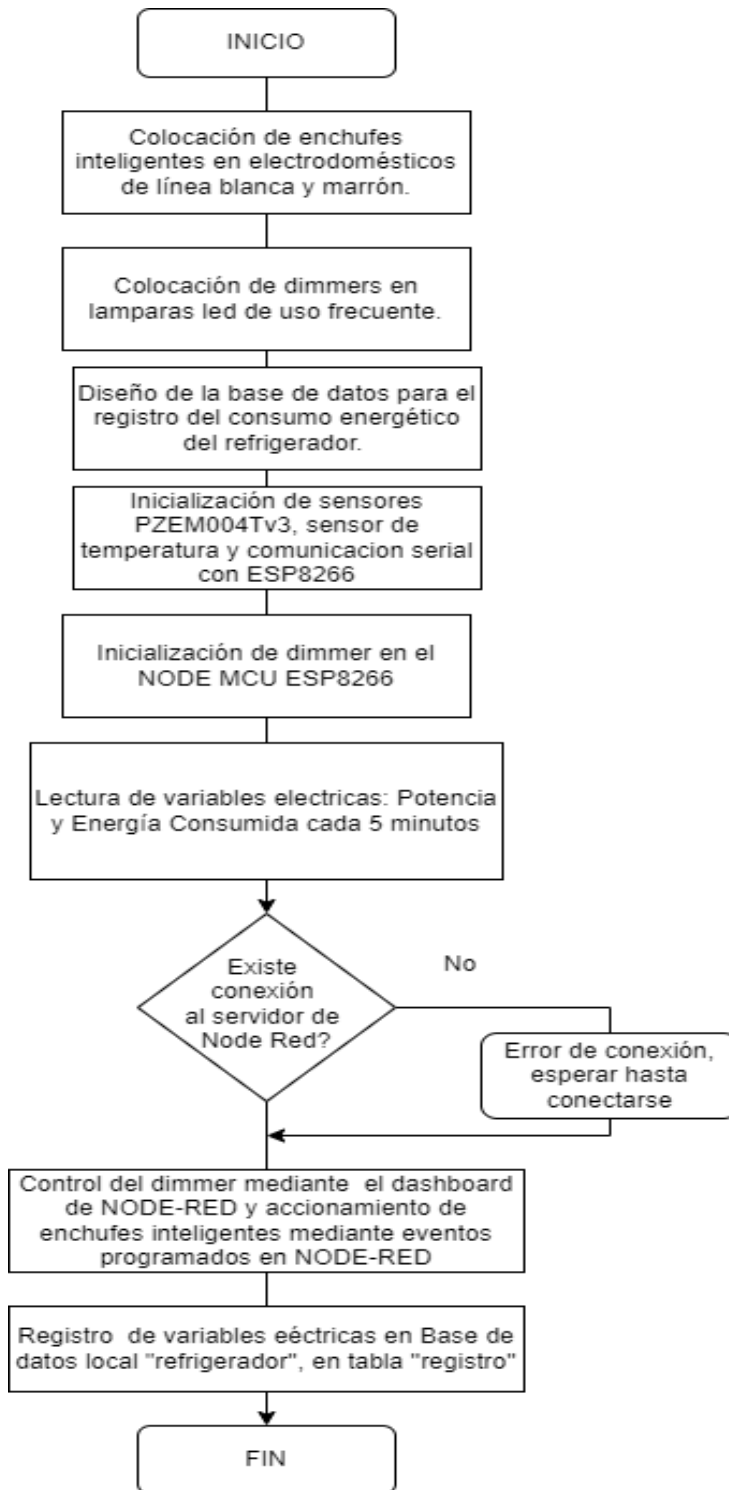


Ilustración 51-3: Diagrama del sistema de control del consumo energético.

Realizado por: Lopez, D; 2022

3.7.1. Colocación de enchufes inteligentes

En base a la tabla 10-2, se seleccionan los electrodomésticos existentes en el piso 3 de la vivienda descrita. Los cuales son refrigerador, stereo, televisión y lavadora, cada enchufe contara con una dirección IP estática.

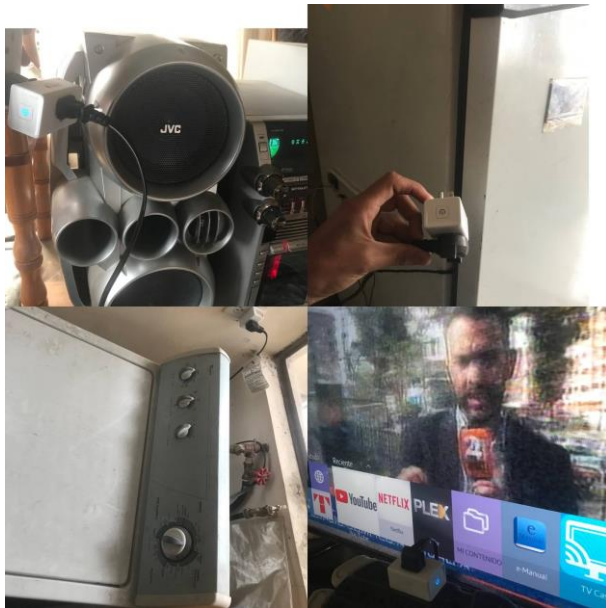


Ilustración 52-3: Colocación de enchufes inteligentes en electrodomésticos de línea blanca y marrón.

Realizado por: Lopez, D; 2022

Una vez colocados los dispositivos, se procede a configurarlos mediante la aplicación KASA de la compañía TP-LINK. Cada enchufe se conecta a internet vía WiFi y configurado con un nombre e icono para su distinción como se muestra en la ilustración 53-3.

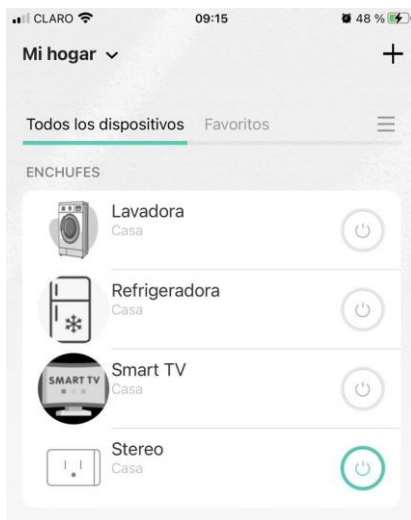


Ilustración 53-3: Configuración de los tomacorrientes inteligentes mediante aplicación Kasa.

Realizado por: Lopez, D; 2022

La tabla 5-3 describe la pertenencia de electrodoméstico y asignación de direcciones IP estéticas de los tomacorrientes inteligentes.

Tabla 5-3: Esquema de direccionamiento IP de los tomacorrientes.

Dirección MAC	Electrodoméstico	Dirección IP estática
6C:5A:B0:BC:D8:CE	Lavadora	192.168.100.71 /24
6C:5A:B0:BC:E4:53	Televisión	192.168.100.72 /24
6C:5A:B0:BC:EB:41	Stereo	192.168.100.73 /24
6C:5A:B0:BC:FB:B1	Refrigerador	192.168.100.74 /24

Realizado por: Lopez, D; 2022

Ya descrito el direccionamiento, se procede a configurar el pool de direcciones IP estáticas en el router del piso 3. La ilustración 54-3 muestra el proceso de configuración de las direcciones IP estáticas en base a la dirección MAC de cada enchufe inteligente.

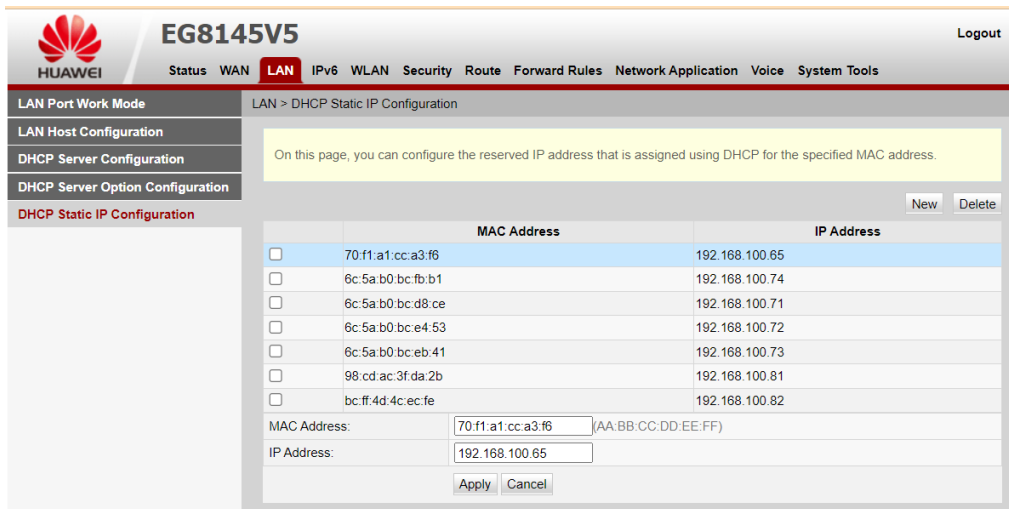


Ilustración 54-3: Configuración del pool de direcciones IP estáticas en el router del piso 3 de la vivienda.

Realizado por: Lopez, D; 2022

3.7.2. Colocación de dimmers en lámparas de uso frecuente

Se seleccionan únicamente las lámparas del cuarto máster en base a la frecuencia de uso por parte de los residentes del piso 3. La tabla 6-3 muestra el direccionamiento IP estático de los dimmers del cuarto máster de la vivienda del piso 3.

Tabla 6-3: Esquema de direccionamiento IP de los tomacorrientes.

Dirección MAC	Electrodoméstico	Dirección IP estática
98:CD:AC:3F:DA:2B	Dimmer1	192.168.100.81 /24
BC:FF:4D:4C:EC:FE	Dimmer2	192.168.100.82 /24

Realizado por: Lopez, D; 2022



Ilustración 55-3: Implementación de dimmers en luminarias del cuarto máster.

Realizado por: Lopez, D; 2022

3.7.3. Configuración del sistema de regulación de intensidad de luminarias

Para la regulación de la intensidad de las luminarias, se han configurado 2 canales de escucha a través del protocolo WebSocket en Node-Red como se muestra en la ilustración 56-3, la cual establece un canal de comunicación bidireccional en el cual recibirá los valores del slider que irán variando conforme a lo que usuario lo desee.

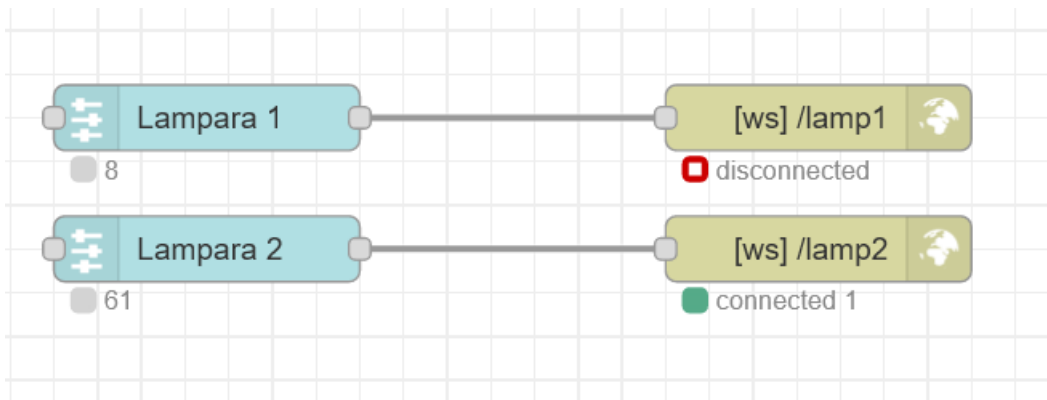


Ilustración 56-3: Configuración canales WebSocket para luminarias del cuarto máster.

Realizado por: Lopez, D; 2022

3.7.4. Diseño de la base de datos para el registro del consumo eléctrico de electrodomésticos

Para la verificación del uso eficiente de los tomacorrientes, se ha determinado únicamente el monitoreo del refrigerador a través de variables como la energía consumida y temperatura dentro del electrodoméstico, debido a la presencia de alimentos perecibles es importante mantener una temperatura óptima para su conservación. Diferente al accionamiento de los demás electrodomésticos en los cuales no se requiere la conservación de alimentos para el consumo humano.

Se procede a la creación de la base de datos “refrigerador” para almacenar las variables mencionadas en el servidor de XAMPP, opción phpMyAdmin.

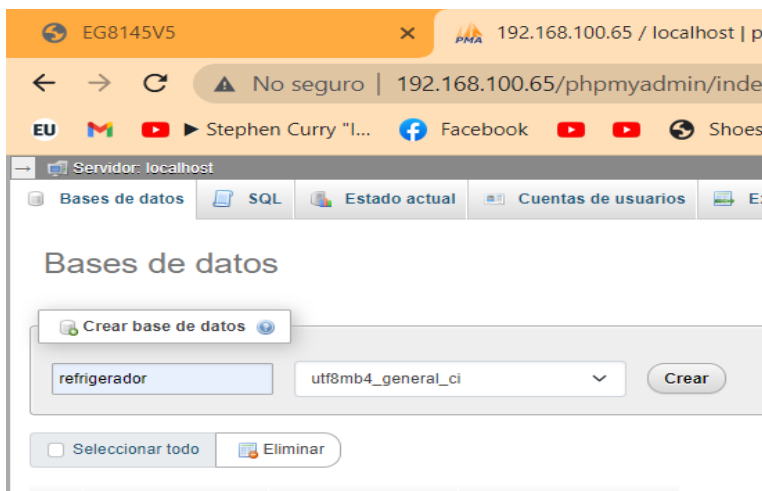


Ilustración 57-3: Creación de base de datos para el registro de variables de consumo y temperatura del refrigerador.

Realizado por: Lopez, D; 2022

Ya configurada la base de datos se procede a crear la tabla “estado_actual” la cual tendrá tres atributos: id, temperatura y consumo; en la que su llave primaria será “id” para ser asociada con sus respectivas tablas.

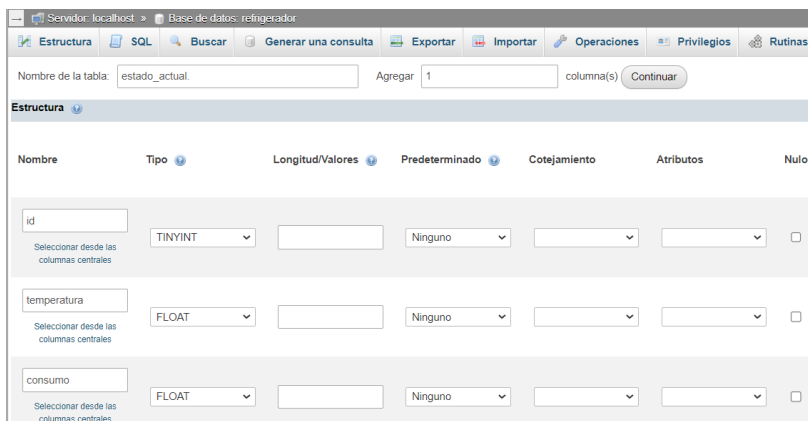


Ilustración 58-3: Creación de tabla “estado_actual” en la base de datos “refrigerador”.

Realizado por: Lopez, D; 2022

Seguidamente las Ilustraciones 59-3 hasta la 62-3 muestran la creación de las diferentes tablas para el almacenamiento según el grado temporal de los datos censados.

Nombre de la tabla: Agregar columna(s)

Estructura

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice
<input type="text" value="id"/> <small>Seleccionar desde las columnas centrales</small>	TINYINT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	...
temperatura <small>Seleccionar desde las columnas centrales</small>	FLOAT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	...
consumo <small>Seleccionar desde las columnas centrales</small>	FLOAT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	...

Ilustración 59-3: Creación de tabla “historial_hora” en la base de datos “refrigerador”.

Realizado por: Lopez, D; 2022

Nombre de la tabla: Agregar columna(s)

Estructura

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice	A,J
<input type="text" value="id"/> <small>Seleccionar desde las columnas centrales</small>	TINYINT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	...	<input type="checkbox"/>
temperatura <small>Seleccionar desde las columnas centrales</small>	FLOAT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	...	<input type="checkbox"/>
consumo <small>Seleccionar desde las columnas centrales</small>	FLOAT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	...	<input type="checkbox"/>

Ilustración 60-3: Creación de tabla “historial_dia” en la base de datos “refrigerador”.

Realizado por: Lopez, D; 2022

Nombre de la tabla: Agregar columna(s)

Estructura

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos
<input type="text" value="id"/> <small>Seleccionar desde las columnas centrales</small>	TINYINT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>
temperatura <small>Seleccionar desde las columnas centrales</small>	FLOAT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>
consumo <small>Seleccionar desde las columnas centrales</small>	FLOAT	<input type="text"/>	Ninguno	<input type="text"/>	<input type="text"/>

Ilustración 61-3: Creación de tabla “historial_mes” en la base de datos “refrigerador”.

Realizado por: Lopez, D; 2022

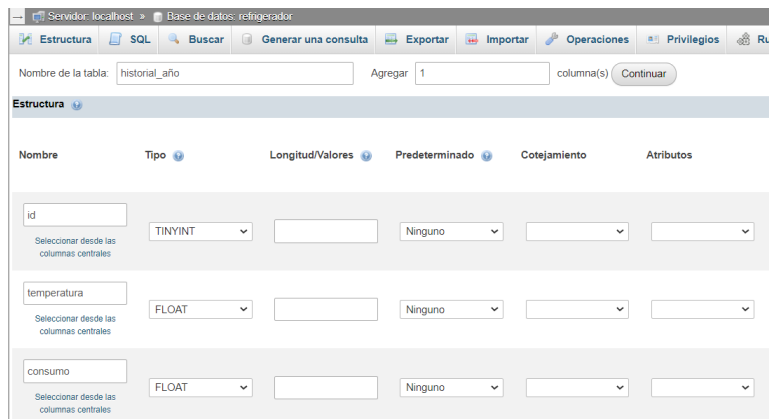


Ilustración 62-3: Creación de tabla “historial_año” en la base de datos “refrigerador”.

Realizado por: Lopez, D; 2022

Finalmente, la ilustración 63-3 muestra el diagrama entidad- relación de la base de datos creada asociando la tabla estado actual con las demás tablas de almacenamiento temporal, mediante el uso de una llave primaria y llaves foráneas.

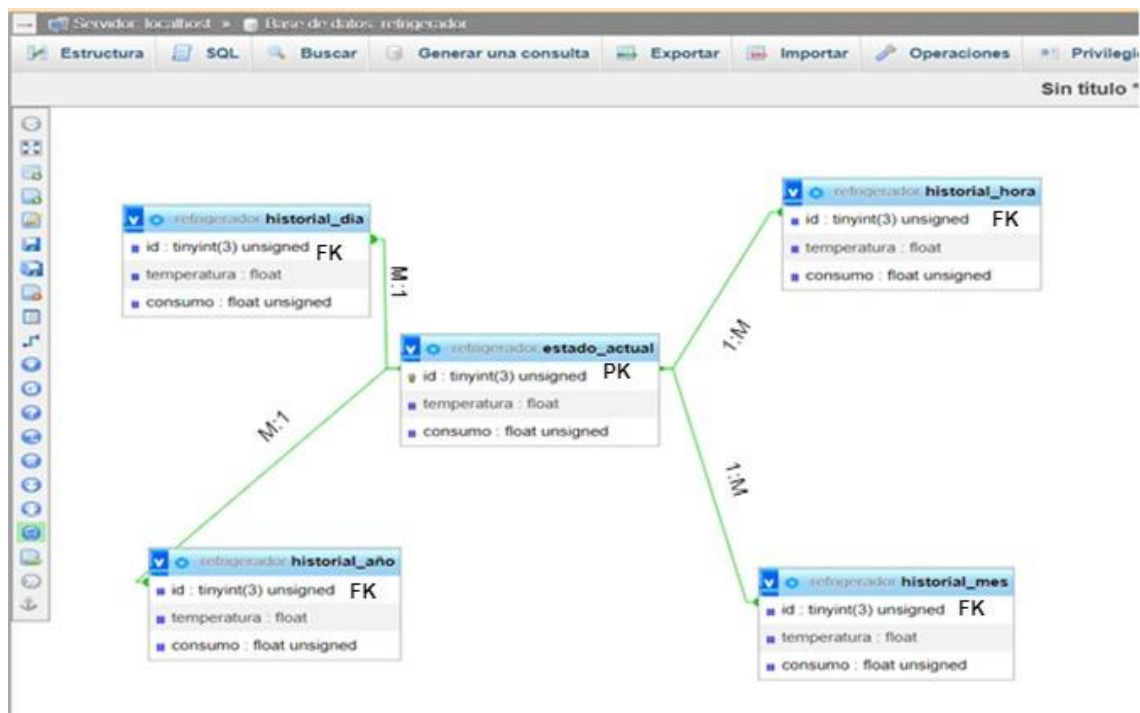


Ilustración 63-3: Diagrama entidad- relación de la base de datos “refrigerador”.

Realizado por: Lopez, D; 2022

3.7.5. Configuración del sistema de control de cargas a través de tomacorrientes inteligentes

Ya concluida la estructuración de la base de datos se procede a la configuración de los distintos procesos para cada actuador del piso 3 en Node-Red. La ilustración 64-3 muestra los procesos configurados en Node-Red.

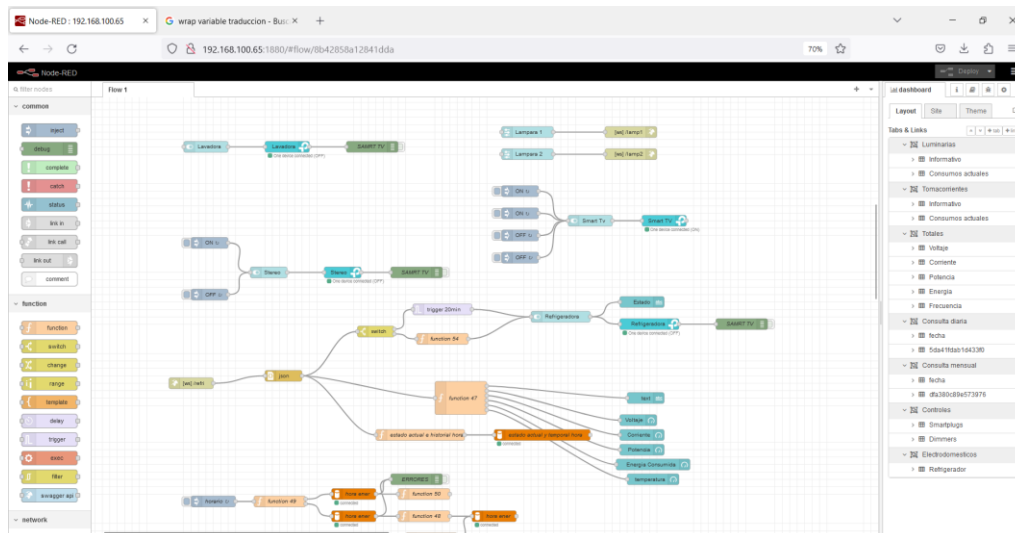


Ilustración 64-3: Configuración de procesos de los actuadores en Node-Red.

Realizado por: Lopez, D; 2022

Los tomacorrientes inteligentes fueron configurados según su frecuencia de uso e importancia de funcionamiento. El nodo de lavadora fue configurado sin horarios ya que las horas de lavado no son rutinarias, caso contrario que con los nodos de SmartTV y Stereo que fueron configurados en base al horario del noticiero nacional que los moradores de la vivienda sintonizan; finalmente el nodo de refrigerador es accionado en base la temperatura en su interior.

La tabla 6-4 muestra una comparativa de los sensores posibles de uso en el monitoreo de temperatura en el interior del refrigerador, exponiendo brevemente sus características.

Tabla 6-4: Comparativa de sensores de temperatura.

Característica	DHT11	DHT22
Rango de temperaturas (°C)	0-50	-40-125
Rango de humedad (%)	20-80	0-100
Voltaje de funcionamiento (V)	3-5	3-5
Precisión de datos	±2%	±0.5%
Precio (\$)	4.35	9.74
Dimensiones (mm)	15.4x12x5.5	15.1x25x7.7

Realizado por: Lopez, D; 2022

Una vez expuestas las diferencias entre los 2 módulos, se selecciona el sensor DHT22 en base al rango de temperatura de percepción, pero sobre todo a la precisión de los datos que brinda el módulo ya que este rango de precisión influye directamente en el accionar del tomacorriente

inteligente y por ende al consumo energético del refrigerador. Adicionalmente para el censado de voltaje, corriente, potencia y energía consumida se adicione otro sensor PZEM004Tv3 exclusivamente para el electrodoméstico.

Brown; et al. en 2014 menciona que la reducción de la temperatura a valores menores o iguales a 4°C ayuda a prolongar significativamente la vida de almacenamiento de los alimentos perecibles, dando más oportunidades de uso antes de la eliminación. Adicionalmente en 2022 CSIRO establece que la temperatura para la mantención correcta de alimentos perecibles en un plazo de tiempo que va desde los 2 hasta los 14 días según el alimento a temperaturas por debajo de 5 °C.

En base las primicias mencionadas se establece una orden de apagado del refrigerador cuando este alcance una temperatura de 3.8°C y se encienda para enfriar el ambiente cuando la temperatura supere los 5°C. Concluyendo las acciones del tomacorriente inteligente, se registran los consumos energéticos y las temperaturas que fueron censadas cada segundo, mostrando los valores en tiempo real del electrodoméstico en el dashboard. La ilustración 65-3 muestra la implementación del sistema de monitoreo del consumo energético y temperatura, así como también la colocación del tomacorriente inteligente para su accionamiento en base a los límites de temperatura establecidos.

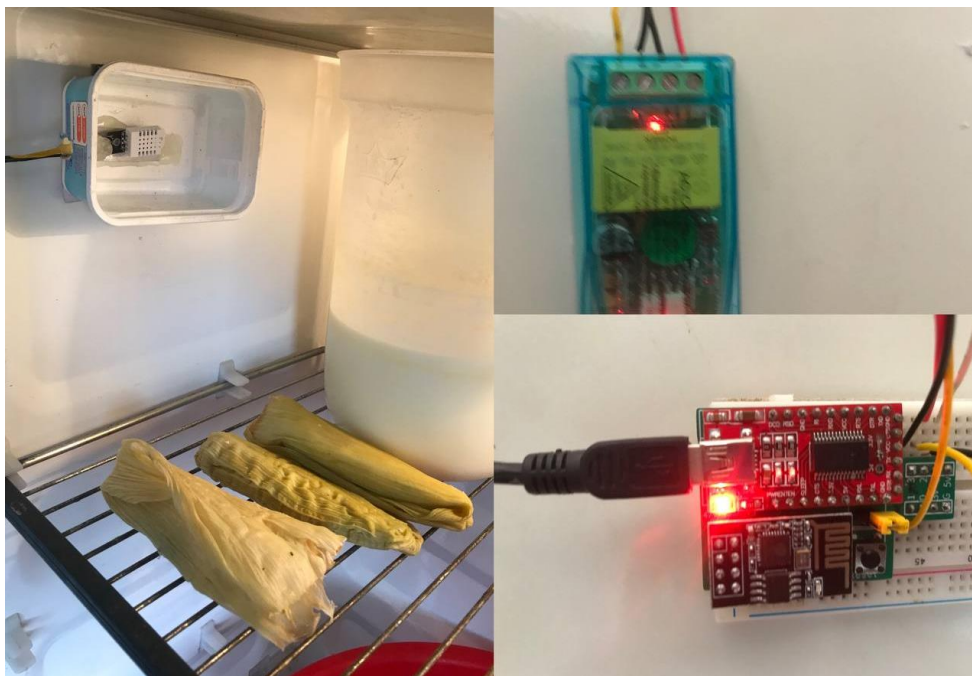


Ilustración 65-3: Implementación del sistema de control del consumo energético en refrigerador.

Realizado por: Lopez, D; 2022

CAPÍTULO IV

4. MARCO DE RESULTADOS Y DISCUSIÓN

En el presente capítulo se detallan los resultados obtenidos al realizar las pruebas de la primera y segunda fase del sistema IoT de monitoreo y control del consumo energético. Realizando una lista de chequeo, comparativa del consumo diario y mensual de ambas fases, así como el rendimiento del sistema final implementado mediante factores como latencia y control de los actuadores para la verificación de los objetivos propuestos.

Se ha denominado primera fase al sistema de monitoreo del consumo energético y como segunda y última fase al sistema de monitoreo y control del consumo energético. La muestra de datos será presentada de forma mensual y una muestra diaria que estará conformada por 24 submuestras que comprenden desde las 00h hasta las 23h00.

4.1. Lista de chequeo

Para la verificación del cumplimiento del segundo objetivo propuesto en el desarrollo del proyecto, la tabla 1-4 presenta una lista de características a ser enmarcadas para evaluar el sistema de monitoreo y control del consumo energético en redes de baja tensión.

Tabla 1-4: Checklist de aspectos cualitativos del sistema de monitoreo y control del consumo energético

Característica	Cumple	No Cumple
La información captada por el sensor 1 es arribada en el servidor de Node-Red	X	
La información captada por el sensor 2 es arribada en el servidor de Node-Red	X	
La información recolectada por los sensores es almacenada en la base de datos local.	X	
La información recolectada por los sensores es visualizada en tiempo real a través del Dashboard.	X	
El sistema permite consultas de datos históricos de carácter diario.	X	
El sistema permite consultas de datos históricos de carácter mensual.	X	
El sistema permite el control del accionamiento de tomacorrientes inteligentes de forma programada.	X	
El sistema permite el control del accionamiento de tomacorrientes inteligentes de forma remota a través del Dashboard.	X	
El sistema permite la regulación de los dimmer de forma remota a través del Dashboard.	X	
El sistema de monitoreo y control del consumo energético es escalable	X	
El sistema de monitoreo y control del consumo energético es accesible desde cualquier parte en el mundo.	X	

Realizado por: Lopez, D; 2022

4.2. Resultados de la primera fase

Una vez registrada toda la información de los dos sensores hacia la base de datos pertinente al mes, se procede a mostrar las distintas pestañas que conforman el dashboard.

La ilustración 1-4 muestra la información del sistema de monitoreo del consumo energético plasmada en la pestaña “luminarias”. Esta pestaña está compuesta por siete grupos, donde seis de los cuales muestran datos captados por el sensor en tiempo real, a excepción del grupo “consumo actual” el cual plasma el consumo del día actual en KWh en cada hora que conforma el día y siendo actualizado automáticamente de manera horaria.

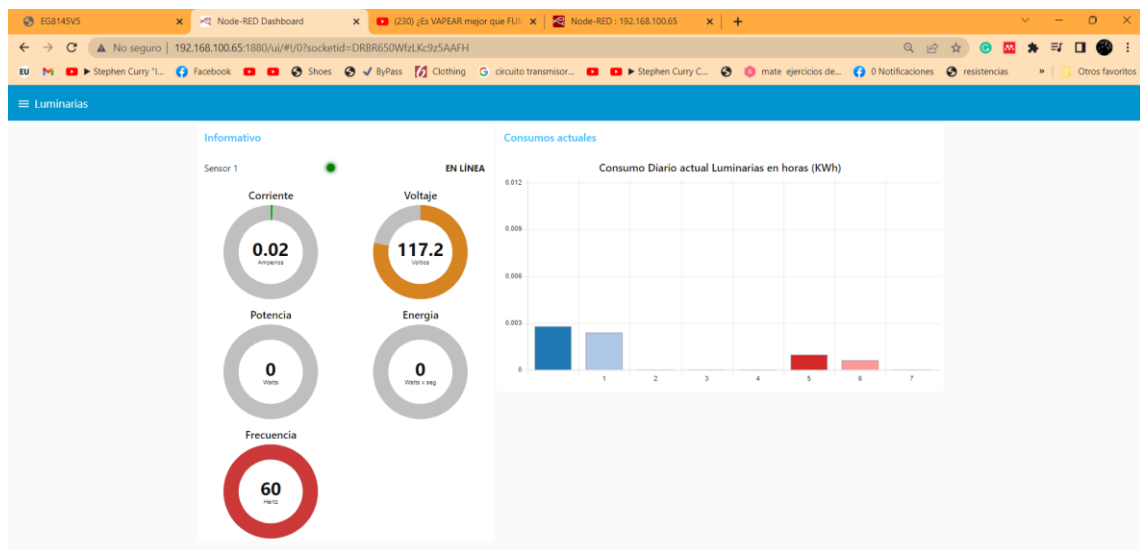


Ilustración 1-4: Pestaña de datos en tiempo real del consumo de “Luminarias”.

Realizado por: Lopez, D; 2022

De igual forma la Ilustración 2-4 presenta el consumo en tiempo real de las diferentes variables energéticas de los “Tomacorrientes”, así como los consumos realizados en el día de manera horaria que irán actualizándose cada hora.

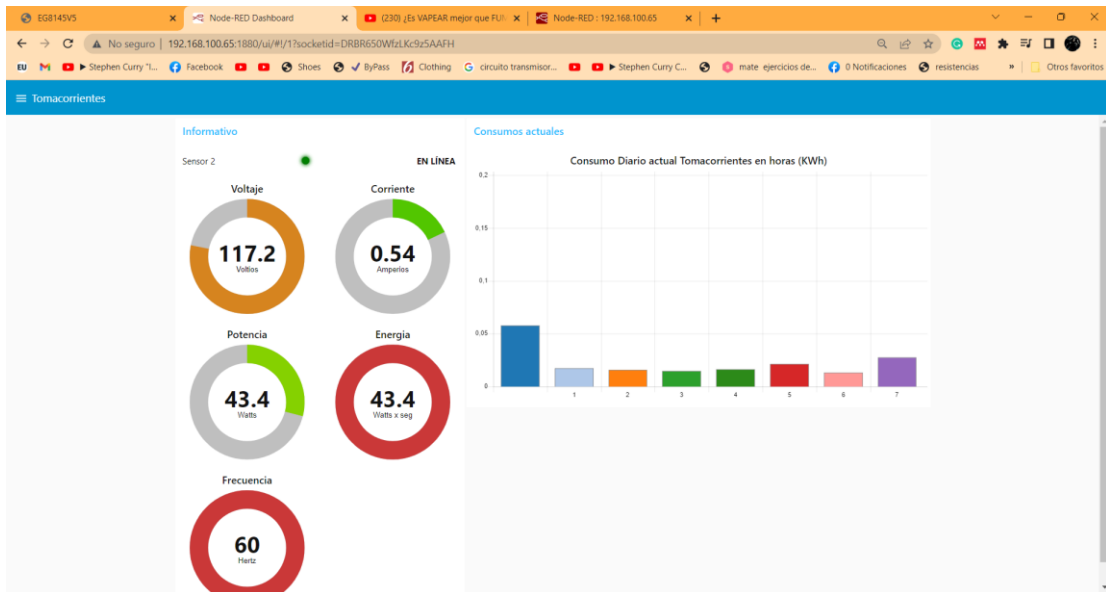


Ilustración 2-4: Pestaña de datos en tiempo real del consumo de “Tomacorrientes”.

Realizado por: Lopez, D; 2022

Finalmente se presenta la pestaña de “Totales” en la ilustración 3-4, la cual muestra el voltaje, corriente y frecuencia promedio, así como la potencia y energía total, datos en tiempo real que se consumen en el piso 3 de la vivienda descrita.

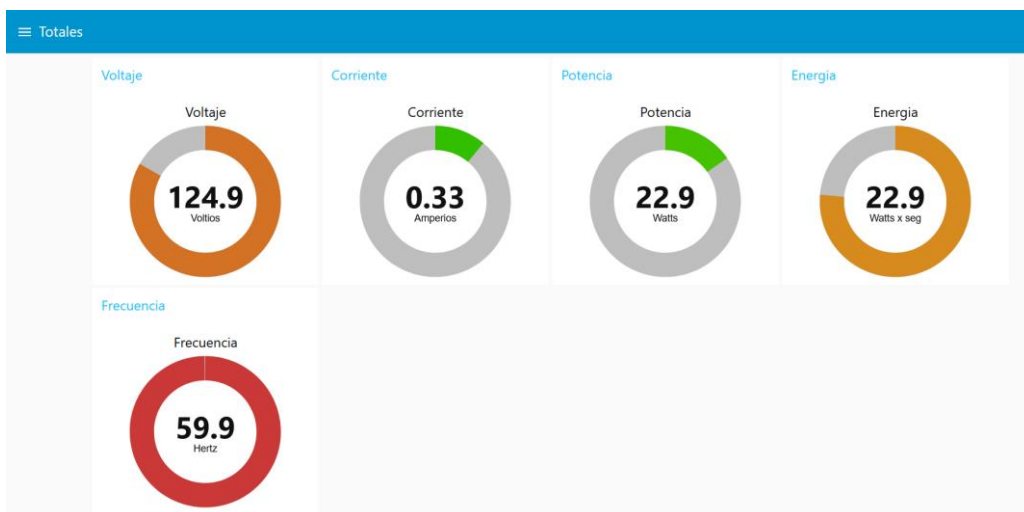


Ilustración 3-4: Pestaña de datos en tiempo real “Totales” del consumo en la vivienda.

Realizado por: Lopez, D; 2022

Procediendo a la parte de consultas históricas de los consumos realizados en el piso 3, la ilustración 4-4 representa la pestaña “Consulta diaria” la cual brinda un informativo del consumo diario durante las 24 horas del día, así como los valores a pagar.

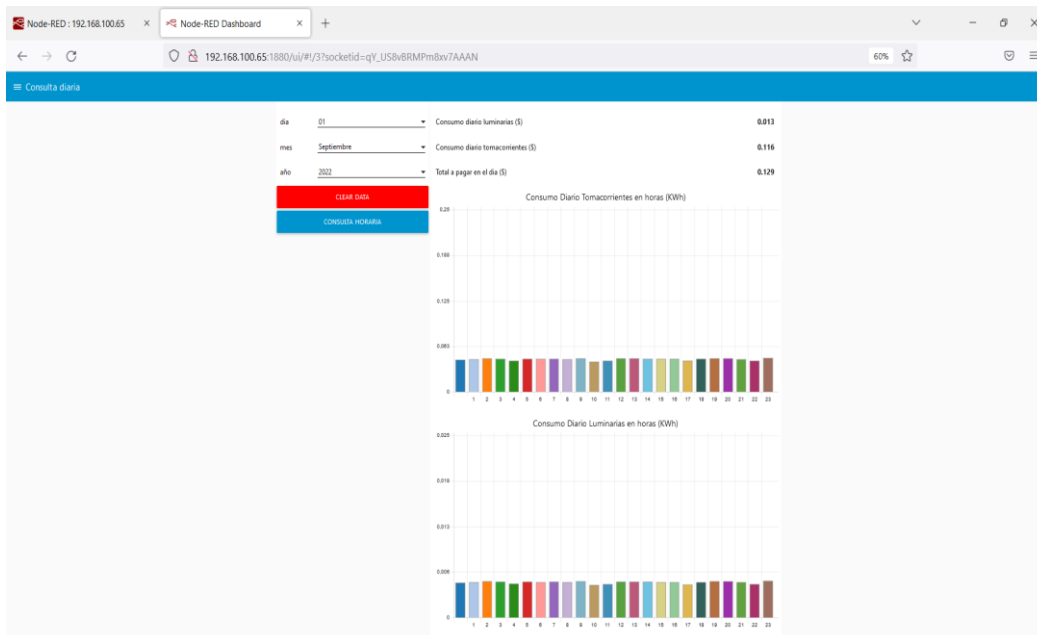


Ilustración 4-4: Pestaña de “Consulta diaria” de los consumos del piso 3 de la vivienda.

Realizado por: Lopez, D; 2022

De igual forma la ilustración 5-4 representa la pestaña “Consulta diaria” la cual brinda un informativo del consumo diario durante las 24 horas del día, así como los valores a pagar.

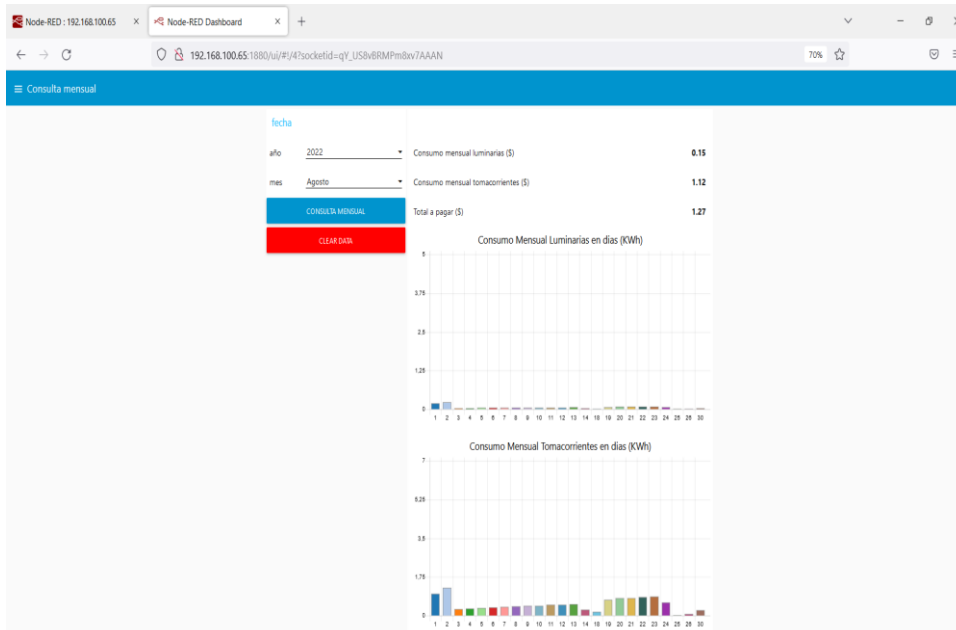


Ilustración 5-4: Pestaña de “Consulta mensual” de los consumos del piso 3 de la vivienda.

Realizado por: Lopez, D; 2022

Finalmente, la tabla 2-4 realiza una comparativa de los consumos eléctricos históricos obtenidos en el sistema IoT implementado con respecto a los registrados en la planilla de la empresa eléctrica.

Tabla 2-4: Checklist de aspectos cualitativos del sistema de monitoreo y control del consumo energético

Mes	Sistema IoT implementado		Empresa Eléctrica	
	Consumo (Kwh)	Valor a pagar (\$)	Consumo (Kwh)	Valor a pagar (\$)
Septiembre	41,58	3,83	43,28	3,98
Octubre	54,28	5,05	53,87	4,95

Fuente: Lopez, D; 2022

4.3. Resultados de la segunda fase

Tabla 3-4: Checklist de aspectos cualitativos del sistema de monitoreo y control del consumo energético

Mes	Sistema IoT implementado 2022-2023	
	Consumo (Kwh)	Valor a pagar (\$)
Noviembre	137,5	12,36
Diciembre	220,47	20,28
Enero	213,23	19,62

Fuente: Lopez, D; 2022

4.4. Resultados del rendimiento del sistema IoT

La evaluación del rendimiento del sistema final IoT se realizó en base a parámetros como latencia, monitoreo de variables eléctricas, consumo eléctrico, regulación de intensidad de luminarias y accionamiento de cargas mediante el uso de tomacorrientes inteligentes.

4.4.1. Latencia

En la evaluación del parámetro de latencia, se midió el tiempo de acceso hacia el dashboard ubicado en la capa de aplicación. Mediante el uso de la herramienta Google PageSpeed Insights, se obtuvieron los siguientes resultados mostrados en la tabla 4-4

Tabla 4-4: Resultados de pruebas de latencia hacia el dashboard mediante el uso de herramienta web.

Herramienta Web	Tiempo de carga completa del dashboard	
	Móvil	Ordenador
Google PageSpeed Insights	3500 ms	900 ms

Fuente: Lopez, D; 2022

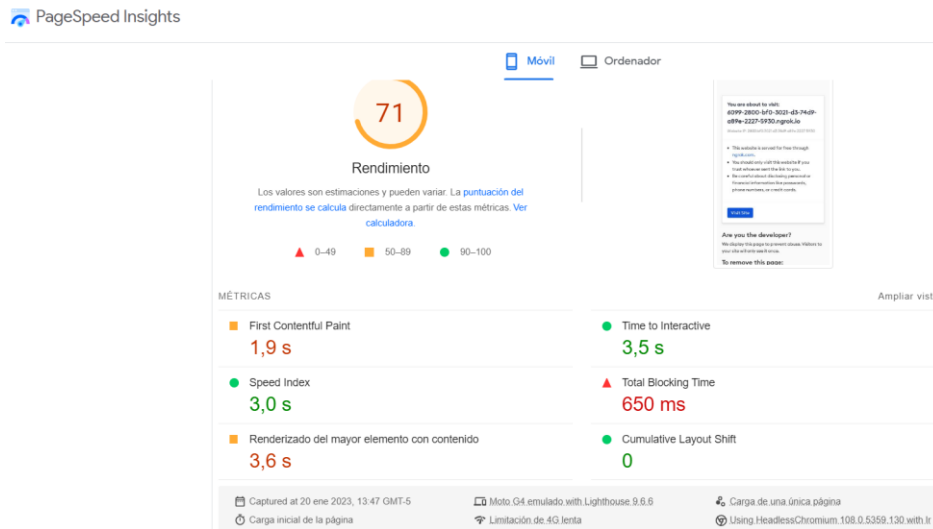


Ilustración 6-4: Reporte de latencia en Google PageSpeed Insights desde un móvil-dashboard.

Realizado por: Lopez, D; 2022

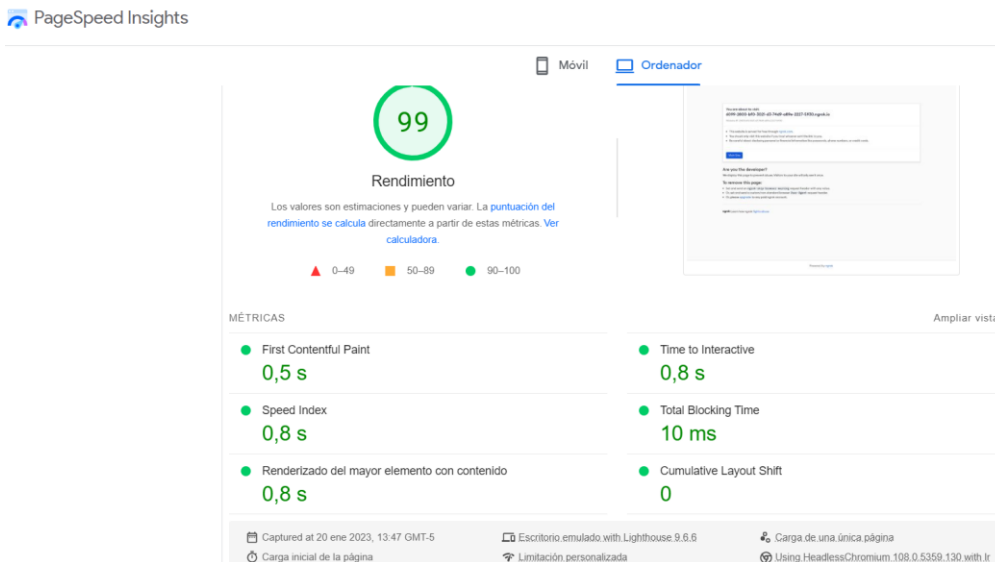


Ilustración 7-4: Reporte de latencia en Google PageSpeed Insights desde un ordenador-dashboard.

Realizado por: Lopez, D; 2022

Seguidamente al comunicarse Node-Red con los actuadores desde el dashboard se registró la siguiente medición presentada en la tabla 5-4.

Tabla 5-4: Resultados de pruebas de latencia dashboard-dimmer.

Herramienta	Latencia
Arduino Serial Monitor	32,25 ms

Fuente: Lopez, D; 2022

```

COM7
[WS-Client] client disconnected.
[WS] Disconnected!
[WS-Client] connect ws...
[WS-Client] connected to 192.168.100.65:1880.
[WS-Client][sendHeader] sending header...
[WS-Client][sendHeader] handshake GET /lamp1 HTTP/1.1
Host: 192.168.100.65:1880
Connection: Upgrade
Upgrade: websocket
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: CU8IiPO2+GrPOhpMhbgRqQ==
Sec-WebSocket-Protocol: arduino
Origin: file://
User-Agent: arduino-WebSocket-Client

[write] n: 251 t: 1416785
[WS-Client][sendHeader] sending header... Done (28894us).
[WS-Client][handleHeader] RX: HTTP/1.1 101 Switching Protocols
[WS-Client][handleHeader] RX: Upgrade: websocket
[WS-Client][handleHeader] RX: Connection: Upgrade
[WS-Client][handleHeader] RX: Sec-WebSocket-Accept: OZL/vLoFlzlABKazaXVlQxlyRmK=
[WS-Client][handleHeader] RX: Sec-WebSocket-Protocol: arduino
[WS-Client][handleHeader] Header read fin.
[WS-Client][handleHeader] Client settings:
[WS-Client][handleHeader] - cURL: /lamp1
[WS-Client][handleHeader] - cKey: CU8IiPO2+GrPOhpMhbgRqQ==
[WS-Client][handleHeader] Server header:
[WS-Client][handleHeader] - cCode: 101
[WS-Client][handleHeader] - cisUpgrade: 1
[WS-Client][handleHeader] - cisWebsocket: 1
[WS-Client][handleHeader] - cAccept: OZL/vLoFlzlABKazaXVlQxlyRmK=
[WS-Client][handleHeader] - cProtocol: arduino
[WS-Client][handleHeader] - cExtensions:
[WS-Client][handleHeader] - cVersion: 0
[WS-Client][handleHeader] - cSessionId:
[WS-Client][handleHeader] Websocket connection init done.
[WS][0][headerDone] Header Handling Done.
[WS] Connected to url: /lamp1
[WS][0][sendFrame] ----- send message frame -----
[WS][0][sendFrame] fin: 1 opCode: 1 mask: 1 length: 9 headerToPayload: 0
[WS][0][sendFrame] text: Connected
[WS][0][sendFrame] pack to one TCP package...
[write] n: 15 t: 1416908
[WS][0][sendFrame] sending Frame Done (3258us).

```

Ilustración 8-4: Reporte de latencia en Arduino Serial Monitor, NodeRed-Actuador.

Realizado por: Lopez, D; 2022

Una vez ya establecidos los informes de latencia se concluyen las latencias totales para dispositivos móviles y ordenadores; incluyendo el tiempo de interacción desde un móvil y computador en la tabla 6-4.

Tabla 6-4: Resultados de pruebas de latencia hacia el dashboard mediante el uso de herramienta web.

Latencia total del sistema de monitoreo y control del consumo energético		
Tiempo de conexión del actuador al sistema	Móvil	Ordenador
	3532 ms	932 ms
Tiempo de interacción del dashboard-dimmer	Móvil	Ordenador
	310 ms	230 ms
Tiempo de interacción del dashboard-tomacorriente inteligente	Móvil	Ordenador
	350 ms	220 ms
Latencia total dimmer	3852 ms	1162 ms
Latencia total tomacorriente inteligente	3842 ms	1152 ms

Fuente: Lopez, D; 202

En base a los resultados mostrados en la tabla 6-4 se obtiene una latencia total al acceder al servicio desde un dispositivo móvil de 3852 milisegundos para controlar los dimmers y 3842

milisegundos al controlar los tomacorrientes inteligentes. Adicionalmente si se controla a los actuadores desde un ordenador se registra una latencia total de 1162 milisegundos para los dimmers y 1152 milisegundos para los tomacorrientes inteligentes.

4.4.2. Monitoreo de variables eléctricas

En base a la comparación de variables eléctricas medidas con instrumentos comerciales y las realizadas por el sistema de monitoreo y control presentadas en la tabla 7-4 se determina la precisión del sistema implementado.

Tabla 7-4: Medidas realizadas por el sistema IoT implementado respecto a un equipo comercial.

Ubicación	Variables	Equipo comercial	Sistema IoT implementado	Error relativo (%)
Luminarias	Voltaje (V)	127.7	127.6	0.08
	Corriente (A)	0.38	0.37	2.63
	Potencia (W)	32.02	31.63	1.22
	Frecuencia (Hz)	60.02	60	0.03
Tomacorrientes	Voltaje (V)	127.7	129	1.02
	Corriente (A)	1.04	1.03	0.96
	Potencia (W)	122.18	122.2	0.02
	Frecuencia (Hz)	60.02	60	0.03

Fuente: Lopez, D; 2022

4.4.3. Consumo Eléctrico

Para la evaluación de este apartado se observó la notificación del estado del sistema en el dashboard y el registro de datos de consumo ante 3 cortes de energía controlados de 1 minuto de duración cada uno.



Ilustración 9-4: Reporte de estado del sistema IoT ante cortes eléctricos.

Realizado por: Lopez, D; 2022

La tabla 8-4 presenta la lista de cortes eléctricos realizados para determinar el tiempo de

reconexión del sistema IoT.

Tabla 8-4: Evaluación del sistema ante cortes de energía eléctrica.

Corte Eléctrico	Inicio de corte eléctrico	Fin de corte eléctrico	Tiempo de recuperación del sistema IoT (segundos)
1	2023-01-20 15:17:00	2023-01-20 15:18:00	17
2	2023-01-21 14:18:00	2023-01-21 14:19:00	16
3	2023-01-21 14:30:00	2023-01-21 14:31:00	17

Fuente: Lopez, D; 2022

En base a los resultados obtenidos de la exposición del sistema IoT a la suspensión de energía eléctrica, es capaz de reintegrarse al funcionamiento en un tiempo promedio de 16.66 segundos.

4.4.4. Control de intensidad luminaria

Para verificar la regulación de la intensidad en luminarias se regularon los valores de ciclo de trabajo en las sliders de las luminarias contemplando seis casos, la intensidad de las luminarias es apreciada en la gráfica en tiempo real “Corriente Luminarias”.

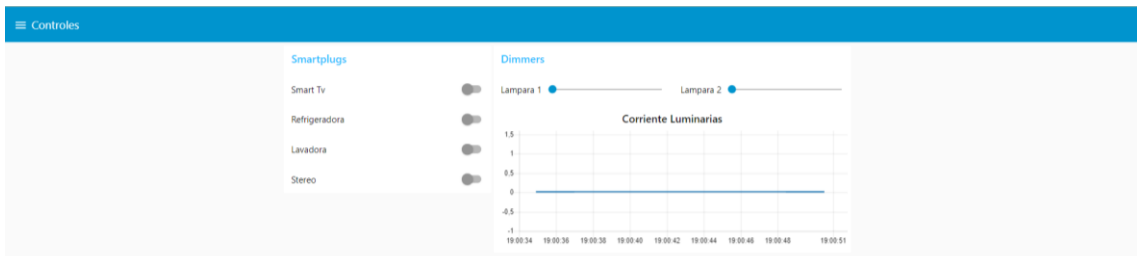


Ilustración 10-4: Regulación de intensidad luminaria de la lampara 1 al 100% de brillo.

Realizado por: Lopez, D; 2022

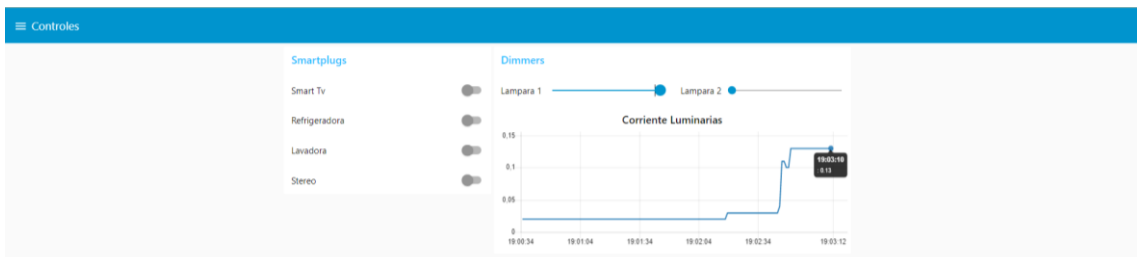


Ilustración 11-4: Regulación de intensidad luminaria de la lampara 2 al 100% de brillo.

Realizado por: Lopez, D; 2022

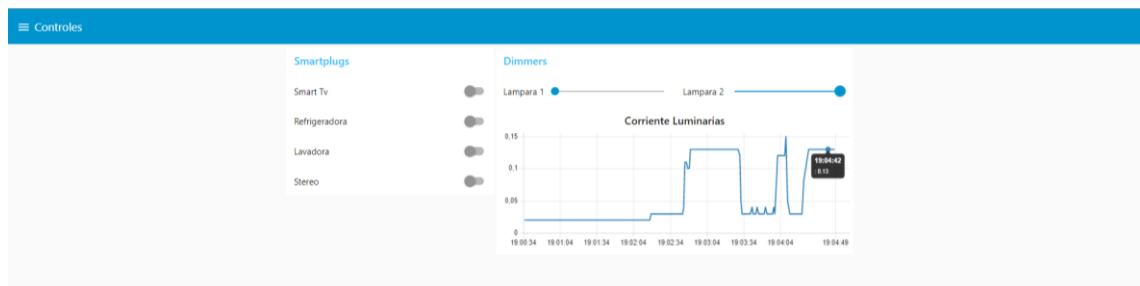


Ilustración 12-4: Regulación de intensidad luminaria de la lampara 2 al 100% de brillo.

Realizado por: Lopez, D; 2022

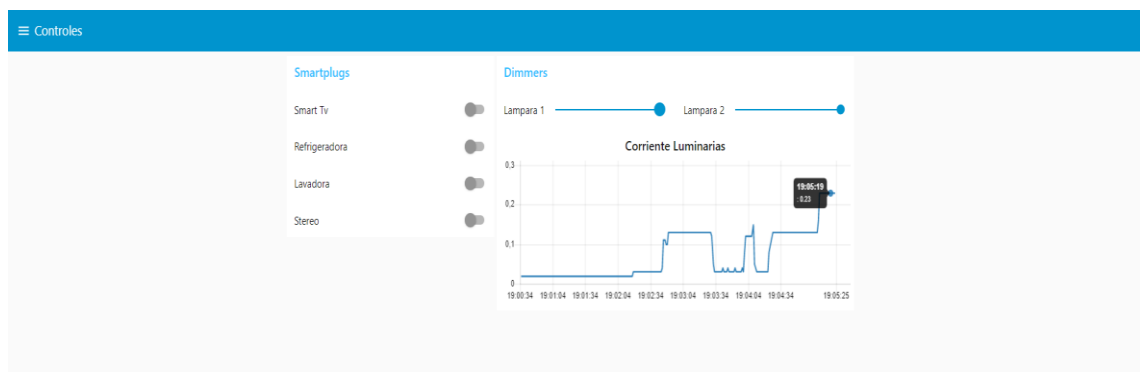


Ilustración 13-4: Regulación de intensidad luminaria de las lámparas 1 y 2 al 100% de brillo.

Realizado por: Lopez, D; 2022

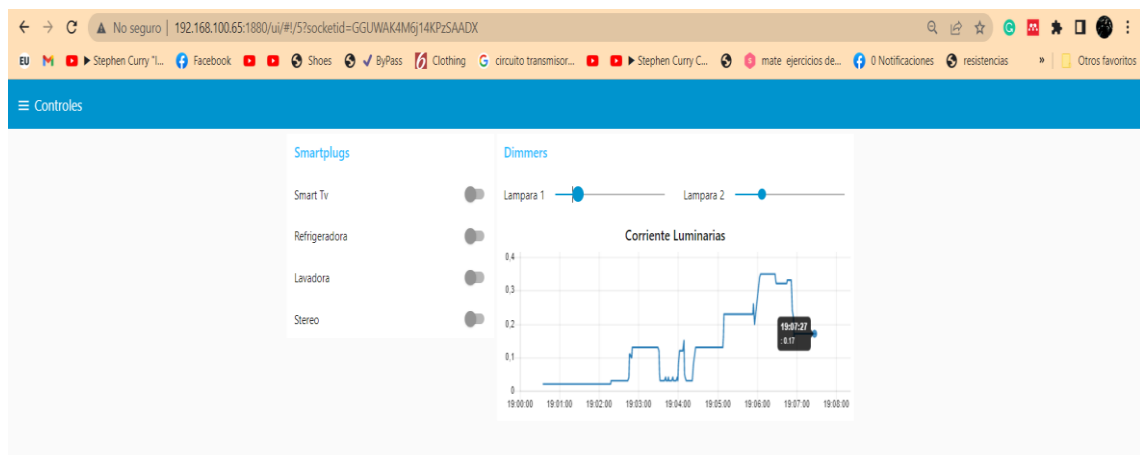


Ilustración 14-4: Regulación de intensidad luminaria de las lámparas 1 y 2 al 30% de brillo.

Realizado por: Lopez, D; 2022

Tiempo de reconexión al servidor

La tabla 9-4 presenta los tiempos de recuperación de conexión cuando se presenta un corte eléctrico en los dimmers.

Tabla 9-4: Evaluación de respuesta en cortes eléctricos para los dimmers inteligentes.

Corte eléctrico	Dimmer	Tiempo de reconexión al servidor (ms)
1	1	400
	2	400
2	1	360
	2	360
3	1	410
	2	410

Fuente: Lopez, D; 2022

En base a los tiempos obtenidos en la tabla 9-4 se establece un tiempo promedio de reconexión al servidor de 390 milisegundos ante un corte eléctrico para los dimmers.

La tabla 10-4 presenta los tiempos de recuperación de conexión cuando se presenta un corte eléctrico en los tomacorrientes inteligentes.

Tabla 10-4: Evaluación de respuesta en cortes eléctricos para el sistema de control de consumo energético en base a la temperatura.

Corte eléctrico	Tiempo de reconexión al servidor (ms)
1	620
	620
2	550
	550
3	590
	590

Fuente: Lopez, D; 2022

En base a los tiempos obtenidos en la tabla 10-4 se establece un tiempo promedio de reconexión al servidor de 586,6 milisegundos ante un corte eléctrico para los tomacorrientes inteligentes.

4.4.5. Accionamiento de los tomacorrientes inteligentes

En base a los eventos programados en la sección metodológica, la ilustración 15-4 muestra una comparativa de la obtención de resultados como el consumo diario (kWh), costo diario (\$) y la fecha; de los cuales el evento de control en base a la temperatura del refrigerador fue aplicado en los días 25-27 y los días del 28-30 fueron censados sin aplicar dicho evento.

historial_dia (6r × 3c)		
CONSUMO DIA	COSTO DIA	FECHA
0,564	0,05	2023-01-25 00:00:00
0,618	0,06	2023-01-26 00:00:00
0,505	0,05	2023-01-27 00:00:00
1,318	0,12	2023-01-28 00:00:00
0,831	0,08	2023-01-29 00:00:00
0,792	0,07	2023-01-30 00:00:00

Ilustración 15-4: Comparativa de resultados obtenidos al realizar la medición del consumo energético.

Realizado por: Lopez, D; 2022

La ilustración 15-4 denota un ahorro del consumo energético del 57,36 % cuando se aplica el evento ahorrativo de energía en el electrodoméstico con respecto a cuando no funcionan este sistema y solo está conectado a la toma eléctrica de la casa.

CONCLUSIONES

En base al estudio de las arquitecturas, protocolos y componentes que comprenden la tecnología IoT, se seleccionó a la arquitectura basada en APIs debido al uso de la metodología REST, la tecnología Wi-Fi en base a su frecuencia de uso en los hogares, los protocolos HTTP y WebSocket debido a su propósito de uso doméstico, tarjetas de desarrollo como Arduino 2560 y NodeMCU ESP8266 y sensor PZEM004Tv3 debido a sus características.

Se diseñó e implementó un prototipo para el monitoreo del consumo energético en redes de baja tensión para uso residencial, haciendo uso del protocolo HTTP para la visualización en tiempo real en el dashboard de las variables eléctricas captadas por el sensor PZEM004Tv3 como: voltaje, corriente, potencia, energía consumida y frecuencia; así como el registro de dicha información de manera exitosa en la base de datos relacional local.

Se controló el consumo energético mediante el uso del protocolo WebSocket en los actuadores como dimmers para la regulación de intensidad luminaria en las lámparas del cuarto máster del piso 3 y tomacorrientes inteligentes en los electrodomésticos como SmartTV, Stereo, Lavadora y principalmente refrigerador. Estableciendo un control del actuador en base a la temperatura del refrigerador.

El rendimiento del prototipo IoT del monitoreo y control del consumo energético en redes de baja tensión para uso residencial se registró una latencia total al acceder al servicio desde un dispositivo móvil de 3852 milisegundos para controlar los dimmers, 3842 milisegundos al controlar los tomacorrientes inteligentes; 1162 milisegundos para los dimmers y 1152 milisegundos para los tomacorrientes inteligentes accediendo desde un ordenador. Con un error porcentual menor al 2,63% con respecto a equipos comerciales de medición de variables eléctricas, capacidad reintegración al funcionamiento promedio de 16.66 segundos y un ahorro del 57,36 % cuando se acciona el tomacorriente inteligente en base a la temperatura al interior del refrigerador.

RECOMENDACIONES

Se recomienda mantener la arquitectura basada en APIs, ya que la metodología REST permite al usuario aprovechar al máximo los recursos del servidor independientemente de la capacidad de procesamiento del consumidor; así como portabilidad y escalabilidad del sistema, también se debe considerar adicionar una tecnología de comunicación de respaldo en caso de que el servicio Wi-Fi se suspenda por cortes eléctricos.

Se recomienda sustituir el protocolo HTTP por Websocket para todo el sistema, debido que sus prestaciones permiten una mayor velocidad en la comunicación lo que resulta en una menor latencia, conexión continua entre cliente y servidor, lo que hace factible el envío de información al cliente en cualquier momento dando lugar a una comunicación en tiempo real, incluso sin que el cliente la solicite, también permite una comunicación full dúplex lo que lo hace ideal para comunicaciones M2M y finalmente agregar TLS a la sesión Websocket para brindar una mayor seguridad a la información.

Se recomienda agregar sensores de movimiento en el pasillo para activar las luminarias en dicha sección de la vivienda, accionamiento de cargas como ordenadores en base a su nivel de batería, así como un sistema de control de consumo energético de electrodomésticos como calefactores o aire acondicionado en base a nivel de temperatura existente en el dormitorio. Adicionalmente agregar un sistema de respaldo de energía para el servidor local en caso de un corte eléctrico en la vivienda.

En caso de requerir mayor capacidad de procesamiento para el sistema, se recomienda mudar todo el entorno a la nube para evitar sobrecarga en el servidor local; adicionalmente contratar un servicio permanente de asignación de dominio público en NGROK para evitar errores de conexión debido a la asignación temporal de dominios públicos.

BIBLIOGRAFÍA

10GTEK. *ESP8266 ESP-12F NodeMcu Lua WiFi Módulo Conector de Desarrollo, para Arduino IDE Micropython Pack de 3 [blog]*. 2022. [Consulta: 4 agosto 2022]. Disponible en: https://www.amazon.com/-/es/ESP8266-ESP-12F-Conector-Desarrollo-Micropython/dp/B09MK7RPHT/ref=sr_1_7?_mk_es_US=ÁMÁŽŮÑ&crid=2Q7YETJTNDURM&keywords=esp8266+12f&qid=1659638838&s=electronics&prefix=esp8266+12f%2Celectronics%2C227&sr=1-7.

AICAD. *Fog Computing: ¿Qué es? funciones, arquitectura y beneficios* [blog]. 2020. [Consulta: 19 junio 2022]. Disponible en: <https://www.aicad.es/fog-computing/#Queesselfogcomputing>.

AL-MASRI, Eyhab; et al. "Investigating Messaging Protocols for the Internet of Things (IoT)". *Journal of Internet of Things* [en línea], 2020, (Estados Unidos) 8(1), p. 32. ISSN 21693536. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9090208>.

ALTEXSOFT. *REST API: Best Practices, Concepts, Structure, and Benefits* [blog]. 2021. [Consulta: 27 abril 2022]. Disponible en: <https://www.altexsoft.com/blog/rest-api-design/>.

AMAZON WEB SERVICES. Amazon. *¿Qué es IoT? - Guía del Internet de las cosas para principiantes* [blog]. 2022.[Consulta: 17 abril 2022]. Disponible en: <https://aws.amazon.com/es/what-is/iot/>.

APRENDIENDO ARDUINO. *Sensores y Actuadores* [blog]. 2019. [Consulta: 31 julio 2022]. Disponible en: <https://aprendiendoarduino.wordpress.com/2018/11/14/sensores-y-actuadores-2/>.

ARCOTEL. Boletín estadístico. *Infraestructura del Servicio Móvil Avanzado* [en línea], 2018, Ecuador 3(1), p.16. Disponible en: https://www.arcotel.gob.ec/wp-content/uploads/2015/01/BOLETIN-ESTADISTICO-Diciembre-2018-v4_4.pdf.

ARDUINO. *Arduino Mega 2560 REV3* [blog]. 2022. [Consulta: 4 agosto 2022]. Disponible en: https://www.amazon.com/-/es/Arduino-Mega-2560-REV3-A000067/dp/B0046AMGW0/ref=sr_1_3?keywords=arduino+mega+2560+rev3&qid=1659669123&prefix=arduino+mega+%2Caps%2C175&sr=8-3.

ARRIETA, EVER. *Diferencia entre método inductivo y deductivo* [blog]. 2022. [Consulta: 2 octubre 2022]. Disponible en: <https://www.diferenciador.com/diferencia-entre-metodo-inductivo-y-deductivo/>.

AVSYSTEM. *Cellular IoT: What is the difference between NB IoT and LTE-M?* [blog]. 2018. [Consulta: 29 mayo 2022]. Disponible en: <https://www.avsystem.com/blog/Cellular-IoT/>.

AYŞE, B. "Compilation of Data Link Protocols: Bluetooth Low Energy (BLE), ZigBee and Z-Wave ". *Journal of IoT protocols* [en línea], 2019, (Estados Unidos) 1(1), p.6. [Consulta: 29 abril 2022]. ISBN 78-1-7281-3964-7. Disponible en: <https://ieeexplore.ieee.org/document/8907068>.

BAEZA, JORGE. Protocolos de Transporte TCP y UDP. *Universidad de Alicante* [en línea], 2009, (España), p.19. [Consulta: 16 mayo 2022]. Disponible en: <https://rua.ua.es/dspace/bitstream/10045/11606/1/Pr3-2009-10.pdf>.

BASSI, AGUSTIN. *Introducción a AMQP* [blog]. 2021. [Consulta: 16 mayo 2022]. Disponible en: https://www.gotoiot.com/pages/articles/amqp_intro/index.html.

BAYILMIŞ, CÜNEYT; et al. A survey on communication protocols and performance evaluations for Internet of Things. *Digital Communications and Networks* [en línea], 2022, (Turquía) 3(1), p.11. ISSN 23528648. Disponible en: <https://reader.elsevier.com/reader/sd/pii/S2352864822000347?token=E8C2515786FB9C9B7A4A1EA057D832F0FD3328769972ED69474C7E705FEF3DA057E86CAC7A03AAA8007DEC1F6C55C0E2&originRegion=us-east-1&originCreation=20221010190026>.

BERNES, T.; & FIELDING, R. RFC 1945. Hypertext Transfer Protocol [blog]. 1996.[Consulta: 16 mayo 2022]. Disponible en: <https://www.tech-invite.com/y15/tinv-ietf-rfc-1945.html>.

BLUETOOTH. Specifications Bluetooth Technology Website. [blog]. 2020. [Consulta: 28 abril 2022]. Disponible en: <https://www.bluetooth.com/specifications/>.

BOLAÑOS, HECTOR.; & RIOS, LUIS. Evaluación Y Elección De La Tecnología De Comunicación Más Adecuada Para Una Red De Sensores En Un Prototipo De Sistema De Riego En Cultivos De Flores Tipo Invernadero De Asoflorica Municipio De Piendamó Cauca [en línea] (Trabajo de titulación). (Pregrado) Fundación Universitaria de Popayán, Colombia. 2019, p. 85. [Consulta: 28 abril 2022]. Disponible en: <http://unividafup.edu.co/repositorio/items/show/715>.

CASTAÑEDA, LEONARDO. Sensores IoT Vs endNodes IoT [blog]. 2020.[Consulta: 13 junio 2022]. Disponible en: <https://www.logicbus.com.mx/blog/sensores-industriales-iot/>.

CAZALLA, MIGUEL. Análisis de vulnerabilidades en sistemas IEEE 802.15.4/ZigBee [en línea] (Trabajo de titulación). (Maestría) Universidad Abierta de Cataluña, Cataluña, España. 2019. p. 86. [Consulta: 13 junio 2022]. Disponible en: <http%3A%2F%2Fopenaccess.uoc.edu%2Fwebapps%2Fo2%2Fbitstream%2F10609%2F95166%2F6%2FmcazallabTFM0619memoria.pdf>.

CHÁVEZ, ALEX., & HERRERA, ESTEBAN. Diseño e Implementación de una Smart Meter de Energía Eléctrica enlazado en una plataforma de visualización para Monitoreo y Control del

consumo energético domiciliario basado en IoT [en línea] (Trabajo de Titulación).(Pregrado) Escuela Superior Politécnica del Ejercito, Latacunga, Ecuador. 2020. [Consulta: 14 junio 2022]. Disponible en: <http://repositorio.espe.edu.ec/bitstream/21000/23119/1/T-ESPEL-EMI-0396.pdf>.

CIELO. *Why Smart Plug is a Bad Choice for your Air Conditioner or Heat Pump?* [blog]. 2022. [Consulta: 4 agosto 2022]. Disponible en: <https://www.cielowigle.com/blog/smart-plug-air-conditioner/>.

CISCO. *¿Qué es un router? - Definición y usos* [blog]. 2022. [Consulta: 4 agosto 2022]. Disponible en: https://www.cisco.com/c/es_mx/solutions/small-business/resource-center/networking/what-is-a-router.html#~how-to-choose-small-business-routers.

CONCEPCION, RICARDO. *Tarjetas de Desarrollo (Episodio #8)* . [blog]. [Consulta: 4 agosto 2022]. Disponible en: <https://www.rjconcepcion.com/podcast/tarjetas-de-desarrollo-episodio-8/>.

CS NOTES. *802.11 how it works* [blog]. 2014. [Consulta: 9 mayo 2022]. Disponible en: <https://notes.eddyerburgh.me/computer-networking/802-11>.

DE LUZ, SERGIO. *Protocolos TCP y UDP: características, uso y diferencias* [blog]. 2021. [Consulta: 19 abril 2022]. Disponible en: <https://www.redeszone.net/tutoriales/internet/tcp-udp-caracteristicas-uso-diferencias/>.

DEEPSHAH. “A Comparative Study on Cloud, Fog and Edge Computing”. *IEEE Xplore* [en línea], 2021, (India) 1(1), pp. 2-4. [Consulta: 16 agosto 2022]. Disponible en: <https://ieeexplore.ieee.org/document/9707931/authors#authors>.

DEVELOPEDIA. *MQTT* [blog]. 2021.[Consulta: 15 mayo 2022]. Disponible en: <https://devopedia.org/mqtt>.

DEVELOPER. *Generalidades del protocolo HTTP* [blog]. 2015. [Consulta: 16 mayo 2022]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>.

DIFFEN. *TCP vs UDP Comparación* [blog]. 2015. [Consulta: 6 mayo 2022]. Disponible en: <https://es.diffen.com/tecnologia/TCP-vs-UDP>.

DOWNEY, DANIEL. *LAB 3 : Iot Broker. Mission 1 : Mosquitto MQTT Broker* [blog]. 2020. [Consulta: 15 mayo 2022]. Disponible en: <https://medium.com/@patcharaphon110341/lab-3-iot-broker-76c547c50be6>.

EBRARY. *IoT Reference Architectures, Introduction. What is IoT?, IoT Architecture, Three Layer (Tier) IoT Architecture, Four-Layer Architecture, Five Layer Architecture - Internet of Things, Threats, Landscape, and Countermeasures* [blog]. 2020. [Consulta: 25 abril 2022]. Disponible en: <https://ebrary.net/194544/computer-science/reference-architectures#170589>.

EL ESPAÑOL. Las 8 medidas de ahorro energético en el hogar más eficaces [blog]. 2020. [Consulta: 4 agosto 2022]. Disponible en: https://www.lespanol.com/como/medidas-ahorro-energetico-hogar-eficaces/494700888_0.html.

ELPROCUS. *MIMO - Learn About Multiple Input and Multiple Output Technology.* [blog]. 2021. [Consulta: 5 mayo 2022]. Disponible en: <https://www.elprocus.com/mimo-multiple-input-multiple-output-technology/>.

FARO. *Qué es un dimmer y cómo funciona.* [en línea]. 2022. [Consulta: 4 agosto 2022]. Disponible en: <https://faro.es/es/blog/que-es-un-dimmer-y-como-funciona/>.

GARCÍA, CARLOS. ¿Qué es la Potencia Activa y Cómo se Calcula? [blog]. 2022. [Consulta: 31 julio 2022]. Disponible en: <https://potenciacer0.com/potencia-electrica/activa#Que-es-la-potencia-activa>.

GAST, MATTHEW. *802.11 Wireless Networks Definitive Guide* [en línea] Louisiana-Estados Unidos: O'Reilly, 2005. [Consulta: 30 julio 2022]. Disponible en: <https://p303.zlibcdn.com/dtoken/0053ce29981c325ac9e336015b46bb98>.

GEEKOFGEEKS. *Arquitectura IoT de 3 capas* [blog]. 2021. [Consulta: 24 abril 2022]. Disponible en: <https://www.geeksforgeeks.org/3-layer-iot-architecture/>.

GILLIS, ALEXANDER. *What is IoT (Internet of Things) and How Does it Work?* [blog]. 2017. [Consulta: 19 junio 2022]. Disponible en: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>.

GIMENEZ, XAVI. Desarrollo y Estudio del Protocolo Observe para CoAP [en línea] (Trabajo de titulación). (Pregrado) Universidad Politécnica de Cataluña, Cataluña, España. 2013. p. 67. [Consulta: 30 julio 2022]. Disponible en: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj9j-iy3Yb8AhUzTTABHdCFD0UQFnoECA0QAQ&url=https%3A%2F%2Fupcommons.upc.edu%2Fbitstream%2Fhandle%2F2099.1%2F18314%2FXavi_Gimeno_Gimenez.pdf%3Fsequence%3D1%26isAllowed%3Dy&usq=AOvVaw379wmuEuPdV7i1FQBTZEql.

GOOGLE. *RFC 6455 -The WebSocket Protocol* [blog]. 2011. [Consulta: 10 octubre 2022]. Disponible en: <https://datatracker.ietf.org/doc/rfc6455/>.

GRAFIKA, JALAN. Data Transmission in Machine-to-Machine Communication Protocols for Internet of Things Application: A Review [en línea]. *IEEExplore*, 2014, (Estados Unidos) 1(1), p. 6. [Consulta: 13 junio 2022]. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8938420>.

GU, GUIFEN.; &PENG, GUILI. The Survey of GSM Wireless Communication System [en línea]. *IEE Xplore*, 2010, (China) 1(1), p. 4. [Consulta: 3 mayo 2022]. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6141552>.

HOFFMAN, CHRIS. *What's the Difference Between Ad-Hoc and Infrastructure Mode Wi-Fi?* [blog]. 2016. [Consulta: 9 mayo 2022]. Disponible en: <https://www.howtogeek.com/180649/htg-explains-whats-the-difference-between-ad-hoc-and-infrastructure-mode/>.

HOLOGRAM. *IoT Architecture Explained* [bloga]. 2020. [Consulta: 21 abril 2022]. Disponible en: https://www.hologram.io/whitepapers/iot-architecture?utm_source=google&utm_medium=cpc&utm_campaign=iot-usa&gclid=Cj0KCQjw3v6SBhCsARIsACyrRAM6ofHzMEoMqktdXQ5CYAbEdA8l_7ZwOfXzRX-Jtbq-ExLngkmYX6QaArtBEALw_wcB.

IBM. *Comparación de IPv4 y IPv6 - Documentación de IBM* [blog]. 2019. [Consulta: 5 mayo 2022]. Disponible en: <https://www.ibm.com/docs/es/i/7.1?topic=6-comparison-ipv4-ipv6>.

IBM. *What is SOA (Service-Oriented Architecture)?* [blog]. 2020 [Consulta: 21 abril 2022]. Disponible en: <https://www.ibm.com/cloud/learn/soa>.

IBM. *¿Qué es una API REST?* [blog]. 2021. [Consulta: 28 abril 2022]. Disponible en: <https://www.ibm.com/es-es/cloud/learn/rest-apis>.

IBM. *¿Qué es cloud?* [blog]. 2022. [Consulta: 18 junio 2022]. Disponible en: <https://www.ibm.com/co-es/cloud/learn/cloud-computing>.

ILIEV, GEORGI.; et al. Narrowband Interference Suppression in Wireless OFDM Systems. *Research Gate* [en línea], 2009, (Bulgaria) 5(1), p. 4. [Consulta: 2 mayo 2022]. Disponible en: https://www.researchgate.net/publication/260034429_Narrowband_Interference_Suppression_in_Wireless_OFDM_Systems/figures?lo=1.

IONOS. *HTTP: cuál es la función del protocolo en Internet* [blog]. 2013. [Consulta: 16 mayo 2022]. Disponible en: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/protocolo-http/>.

IONOS. *¿Qué es Bluetooth? Todo lo que debes saber* [blog]. 2015. [Consulta: 28 abril 2022]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-bluetooth/>.

IONOS. *¿Qué es el protocolo de Internet? Definición de Internet Protocol (IP).* [en línea]. 2018. [Consulta: 5 mayo 2022]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/internet-protocol-definicion-y-fundamentos/>.

IONOS. *¿Qué es un servidor informático? / Definición y explicación* [blog]. 2020. [Consulta: 4

agosto 2022]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/>.

IOTPOINT. *Z-Wave Tutorial* [blog]. 2019. [Consulta: 30 abril 2022]. Disponible en: <https://iotpoint.wordpress.com/z-wave-tutorial/>.

ITU. *Descripción General de Internet de los Objetos Y.2060- Y.4000. Y.2060 Y.4000* [blog]. 2012. [Consulta: 30 abril 2022]. Disponible en: <https://www.itu.int/rec/T-REC-Y.2060-201206-1/es>.

JUNIPER. *Junos Software for E Series Broadband Services Routers IP, IPv6, and IGP Configuration Guide* [blog]. 2015. [Consulta: 30 abril 2022]. Disponible en: https://www.juniper.net/documentation/en_US/junose15.1/information-products/topic-collections/swconfig-ip-ipv6/index.html?topic-64529.html.

JUNIPER. *Understanding the IEEE 802.11 Standard for Wireless Networks* [en línea]. 2018. [Consulta: 20 abril 2022]. Disponible en: https://www.juniper.net/documentation/en_US/junos-space-apps/network-director4.0/topics/concept/wireless-80211.html.

KALATZIS, NIKOS.; et al. Semantic interoperability for IoT platforms in support of decision making: An experiment on early wildfire detection. *Research Gate* [en línea], 2019, (Switzerland) 1(1), p. 39. [Consulta: 13 junio 2022]. ISSN 14248220. Disponible en: https://www.researchgate.net/profile/Nikos-Kalatzis/publication/330702152_Semantic_Interoperability_for_IoT_Platforms_in_Support_of_Decision_Making_An_Experiment_on_Early_Wildfire_Detection/links/5c5005f7458515a4c747dd44/Semantic-Interoperability-for-IoT-

KASA SMART. *Amazon.com: Kasa Smart Plug Mini 15A, Smart Home Wi-Fi Outlet funciona con Alexa, Google Home e IFTTT, no requiere concentrador, certificado UL, 2.4G WiFi solamente, paquete de 4 (EP10P4), color blanco: Herramientas y Mejoras del Hogar* [blog]. 2022. [Consulta: 4 agosto 2022]. Disponible en: https://www.amazon.com/-/es/funciona-concentrador-certificado-solamente-EP10P4/dp/B091FXLMS8/ref=sr_1_5?keywords=kasa&qid=1659642223&s=hi&srefix=Kasa%2Ctools%2C195&sr=1-5.

KEEPCODING. *¿Qué es el protocolo IP?* [blog]. 2022. [Consulta: 5 mayo 2022]. Disponible en: https://keepcoding.io/blog/que-es-el-protocolo-ip/#Caracteristicas_del_protocolo_IP.

KHARE, VARSHA.; et al. Comparative Study of 1G, 2G, 3G and 4G. *Journal of Engineering Computers & Applied Sciences* [en línea], 2013, (India) 3(1), p. 4. [Consulta: 5 mayo 2022]. ISSN 2319-5606. Disponible en: <http://borjournals.com/a/index.php/jecas/article/view/57/170>.

KUSHALNAGAR, N.; et al. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. *Research Gate* [en línea], 2007, (Estados Unidos) 1(1), p. 12. [Consulta: 5 mayo 2022]. Disponible en: <http://digilib.unila.ac.id/4949/15/BAB II.pdf>.

KYUN, CHEE.; et al. A Review of 6LoWPAN Routing Protocols. *Research Protocol* [en línea], 2012, (Malasia) 1(1), p.12. [Consulta: 5 mayo 2022]. Disponible en: https://www.researchgate.net/profile/Borhanuddin-Mohd-Ali/publication/266064775_A_Review_of_6LoWPAN_Routing_Protocols/links/5498f1120cf2eeefc30f9eaf/A-Review-of-6LoWPAN-Routing-Protocols.pdf?_sg%5B0%5D=8fHiA4vdOCV7Jj_444-j8zXGKn0cfLhVnl9EiCAu24ZT69cloWX-C.

LALLE, YANDJA.; et al. A Comparative Study of LoRaWAN, SigFox, and NB-IoT for Smart Water Grid. *IEE Xplore* [en línea], 2019, (Túnez) 3(1), pp. 1-6. [Consulta: 8 mayo 2022]. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9044961>.

LAPO, FREDDY. Diseño de un sistema de monitoreo de parámetros de la energía eléctrica, mediante dispositivos inteligentes aplicado a las IoT en uso residencial (Trabajo de titulación). (Pregrado) Escuela Superior Politécnica De Chimborazo, Riobamba, Ecuador. 2020. p. 17.

LAROSA, ALEXANDER. *Analizadores de Tráfico: ¿Qué hacen y cómo se usan?* [blog]. 2020.[Consulta: 4 agosto 2022]. Disponible en: <https://pandorafms.com/blog/es/analizadores-de-traffic/>.

LLAMAS, LUIS. *¿Qué es MQTT? Su importancia como protocolo IoT* [blog]. 2019. [Consulta: 15 mayo 2022]. Disponible en: <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>.

LOMBARDI, MARCO.; et al. Internet of things: A general overview between architectures, protocols and applications. *Information* [en línea], 2021, (Italia) 1(1), pp. 3-4. [Consulta: 15 mayo 2022]. ISSN 20782489. Disponible en: https://mdpi-res.com/d_attachment/information/information-12-00087/article_deploy/information-12-00087-v2.pdf?version=1614255980.

LORA ALLIANCE. *Homepage - LoRa Alliance* [blog]. 2014. [Consulta: 29 abril 2022]. Disponible en: <https://lora-alliance.org/>.

MAKWANA, ASHWIN.; & DIPA, SONI. A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT). *Research Gate* [en línea], 2017, (India) 1(1), pp. 2-3. [Consulta: 16 mayo 2022]. Disponible en:

https://www.researchgate.net/publication/316018571_A_SURVEY_ON_MQTT_A_PROTOCOL_OF_INTERNET_OF_THINGS_IOT.

MALIK, M. I.; et al. XMPP architecture and security challenges in an IoT ecosystem. *Australian Information Security* [en línea], 2018, (Australia), pp. 6. [Consulta: 16 mayo 2022]. Disponible en: <https://ro.ecu.edu.au/ism>.

MENCHACA, M. 6LoWPAN EN UNA RED DE SENSORES INALÁMBRICA [en línea] (Trabajo de titulación) (Pregrado). Universidad de Cantabria, Cantabria, España. 2012. pp. 30-31. [Consulta: 20 mayo 2022]. Disponible en: <https://repositorio.unican.es/xmlui/bitstream/handle/10902/1464/349717.pdf?sequence=1>.

MOFFIT, JACK. *Professional XMPP Programming with JavaScript and jQuery* [en línea]. Chicago- Estados Unidos: Wiley Publishing Inc., 2010, p. 22. [Consulta: 6 junio 2022]. Disponible en: https://books.google.ca/books?id=rVV_pCb-zzC&printsec=copyright#v=onepage&q&f=false.

MONTENEGRO, G.; et al. “Transmission of IPv6 Packets over IEEE 802.15.4 Networks. 4944”. *RFC* [en línea], 2007, (Estados Unidos) 3(1), p. 30. [Consulta: 28 abril 2022]. Disponible en: <https://www.rfc-editor.org/rfc/pdf/rfc4944.txt.pdf>.

MPYTHON. *Internet of Things-IoT basics* [blog]. 2017. [Consulta: 25 abril 2022]. Disponible en: <https://mpython.readthedocs.io/en/master/tutorials/advance/iot/>.

MUHAMMAD, BILAL. “A Review of Internet of Things Architecture, Technologies and Analysis Smartphone-based Attacks Against 3D printers”. Department of Computer Science [en línea], 2017, (China) 2(1), p. 5. [Consulta: 25 mayo 2022]. ISSN 03729311. Disponible en: <https://arxiv.org/pdf/1708.04560>.

MURDAN, Prakash.; & KULLOOTEE, Krishna. “Wi-Fi and LoRaWAN based interactive control system for household energy management”. *IEEE International Conference on Power Systems Technology* [en línea], 2020, (Mauricio) 2(1), p. 1. [Consulta: 25 mayo 2022]. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9230579>.

NAVARRO, RODOLFO. *Máquinas eléctricas y sistemas de potencia* [en línea]. Monterrey-México: Pearson Education, 2007, p. 160. [Consulta: 26 mayo 2022]. ISBN 9702608147. Disponible en: <https://books.google.com.co/books?id=ehxKXip1j6EC&printsec=frontcover&dq=maquinas+electricas&hl=es-419&sa=X&ved=0ahUKewi26K6m5MnOAhVFGx4KHaduADkQ6AEIMjAC#v=onepage&q=maquinas electricas&f=false>.

NISHANT, GARG. A review on comparative study of Bluetooth and ZigBee. *International Education & Research Journal* [en línea], 2013, (India) 3(1), p. 2. [Consulta: 26 mayo 2022]. Disponible en: https://www.seekdl.org/conferences/file/paper/20130716_123138.pdf.

NOR'ADIL, HEPBURN. *Why Smart Plugs Are Some Of The Best Energy-Saving Tech For Your Home* [blog]. 2022. [Consulta: 4 agosto 2022]. Disponible en: <https://www.slashgear.com/907849/why-smart-plugs-are-some-of-the-best-energy-saving-tech-for-your-home/>.

OLIVEIRA, BRUNO. *Un breve vistazo al protocolo MQTT* [blog]. 2017. [Consulta: 15 mayo 2022]. Disponible en: <https://medium.com/internet-das-coisas/iot-05-dando-uma-breve-analise-no-protocolo-mqtt-e404e977fbb6>.

ORACLE. *Qué es una base de datos.* [en línea]. 2022. [Consulta: 4 agosto 2022]. Disponible en: <https://www.oracle.com/mx/database/what-is-database/>.

OROSCO, FREDDY.; et al. Diseño, Simulación e Implementación de códigos de canal en sistemas OFDM. *Research Gate* [en línea], 2012, (Alemania) 2(1), p. 4. [Consulta: 5 agosto 2022]. Disponible en: https://www.researchgate.net/publication/305115171_Diseño_Simulación_e_Implementación_de_códigos_de_canal_en_sistemas_OFDM/figures?lo=1&utm_source=google&utm_medium=organic.

PÉREZ, J. *Cuáles son los electrodomésticos más usados en el hogar* [blog]. 2021. [Consulta: 9 septiembre 2022]. Disponible en: <https://www.robotina.us/electrodomesticos/Cuales-son-los-electrodomesticos-mas-usados-en-el-hogar-especialmente-en-la-cocina-20210501-0007.html>.

PICKDATA. *MQTT vs CoAP, the battle to become the best IoT protocol* [blog]. 2019. [Consulta: 13 junio 2022]. Disponible en: <https://www.pickdata.net/es/noticias/mqtt-vs-coap-mejor-protocolo-iot>.

PIZARRO, JAVIER. *Los niveles de madurez REST, ejemplo con HATEOAS y documentación con Swagger de un servicio con Spring Boot* [blog]. 2021. [Consulta: 27 abril 2022]. Disponible en: <https://picodotdev.github.io/blog-bitix/2021/07/los-niveles-de-madurez-rest-ejemplo-con-hateoas-y-documentacion-con-swagger-de-un-servicio-con-spring-boot/#nivel-0-transporte-http>.

PRISANTAMA, ALFADONI.; et al. "Tunneling 6LoWPAN protocol stack in IPv6 network". *Research Gate* [en línea], 2016, (Indonesia) 1(1), p. 3. [Consulta: 27 abril 2022]. ISSN 15517616. DOI 10.1063/1.4958501. Disponible en: https://www.researchgate.net/publication/305493766_Tunneling_6LoWPAN_protocol_stack_in_IPv6_network.

QUIÑONEZ, OSWALDO. *Internet de las Cosas (IoT)* [en línea]. Monterrey- México: Ibukku LLC, 2019, p. 13. [Consulta: 16 abril 2022]. Disponible en: <https://books.google.es/books?hl=es&lr=&id=vnnEDwAAQBAJ&oi=fnd&pg=PT2&dq=IoT+definicion&ots=oRqvTXmqvm&sig=dARVrowUz150hOCODkDIejp6Oo#v=onepage&q=IoTdefinicion&f=false>.

RAJEESH, KUMAR.; & MOHAN, KUMAR. “Survey on State of Art IoT Protocols and Applications”. *IEEE Xplore* [en línea], 2020, (India) 2(1), p. 3. [Consulta: 13 junio 2022]. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9212227>.

RANTCELL. *Comparison of 2G 3G 4G 5G Networks* [blog]. 2022. [Consulta: 5 mayo 2022]. Disponible en: <https://rantcell.com/comparison-of-2g-3g-4g-5g.html>.

REDHAT. *El concepto de IDE.* [blog]. 2019. [Consulta: 3 agosto 2022]. Disponible en: <https://www.redhat.com/es/topics/middleware/what-is-ide>.

REPSOL. *¿Qué es la energía eléctrica? Tipos, ventajas y aplicaciones* [blog]. 2022. [Consulta: 31 julio 2022]. Disponible en: <https://www.repsol.com/es/energia-innovacion/energia-futuro/transicion-energetica/energia-electrica/index.cshtml>.

RF-WIRELESS. *What is XMPP Protocol in IoT | XMPP Server | XMPP Client* [blog]. 2015. [Consulta: 13 junio 2022]. Disponible en: <https://www.rfwireless-world.com/IoT/XMPP-protocol.html>.

RFWW. *z-wave protocol stack | z-wave protocol layer basics* [blog]. 2016. [Consulta: 30 abril 2022]. Disponible en: <https://www.rfwireless-world.com/Tutorials/z-wave-protocol-stack.html>.

RODRIGUEZ, A. *Bluetooth 5.0, novedades y diferencias con las anteriores versiones* [blog]. 2022. [Consulta: 28 abril 2022]. Disponible en: <https://hardzone.es/tutoriales/rendimiento/bluetooth-5-0-caracteristicas-novedades/>.

SAINT-ANDRE, P. *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC3921* [blog]. 2004. [Consulta: 13 junio 2022]. Disponible en: <https://xmpp.org/rfc/rfc3921.html>.

SALAZAR, JORDI. *Redes Inalámbricas* [en línea]. Technická -República Checa: České vysoké učení technické, 2012, pp. 19-25. [Consulta: 13 junio 2022]. ISBN 9781447152927. Disponible en: https://upcommons.upc.edu/bitstream/handle/2117/100918/LM01_R_ES.pdf.

SALEEM, U.; et al. 2021. Design, Implementation, and Deployment of an IoT Based Smart Energy Management System. *IEEE Access* [en línea], pp. 16. ISSN 21693536. DOI 10.1109/ACCESS.2021.3070960. Disponible en:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9395090>.

SEMTECH. *LoRa and LoRaWAN* [blog]. 2020. Disponible en: <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>.

SHEA, S. *What is Z-Wave?* [blog]. 2018. [Consulta: 20 abril 2022]. Disponible en: <https://www.techtarget.com/iotagenda/definition/Z-Wave>.

SHEEBA, R.; et al., 2021. Real-time Monitoring of Energy Meters Using Cloud Storage. *IEEE Xplore* [en línea], 2021, (India) 1(1), p. 1. [Consulta: 29 abril 2022]. Disponible en: <https://ieeexplore.ieee.org/document/9640636>.

SHELBY, Z.; et al. *Constrained Application Protocol (CoAP)* [blog]. 2012. [Consulta: 13 junio 2022]. Disponible en: <https://tools.ietf.org/id/draft-ietf-core-coap-09.html>.

SIGFOX. *Coverage | Sigfox* [blog]. 2019. [Consulta: 29 abril 2022]. Disponible en: <https://www.sigfox.com/en/coverage>.

SINGH, KARAMJEET. *ZigBee: A Review ZigBee*. *Research Gate* [en línea], 2015, (India) 1(1), p. 2. [Consulta: 14 junio 2022]. Disponible en: https://www.researchgate.net/publication/280947104_ZigBee_A_Review.

SMARTGRIDSINFO. *Consumo Energía Eléctrica* [blog]. 2020. [Consulta: 31 julio 2022]. Disponible en: <https://www.smartgridsinfo.es/consumo-energia-electrica>.

SOFFAR, Heba. *3G Technology (third-generation technology) features and applications* [blog]. 2017.[Consulta: 3 mayo 2022]. Disponible en: <https://www.online-sciences.com/technology/3g-technology-third-generation-technology-features-and-applications/>.

SOLAR. *Corriente alterna (CA) - Frecuencias, concepto y definición* [blog]. 2019. [Consulta: 31 julio 2022]. Disponible en: <https://solar-energia.net/electricidad/corriente-electrica/corriente-alterna>.

SOSA, CLAUDIA.; et al. Enfoque para generar aplicaciones orientadas a servicios para IoT mediante el desarrollo dirigido por modelos. *CEUR Workshop Proceedings* [en línea], 2016, (México) 1(1), pp. 3-4. ISSN 16130073. Disponible en: http://ceur-ws.org/Vol-1807/01_ISW-LOD2016_1_10.pdf.

T-SYSTEMS. *Estas son las capas del Internet de las Cosas* [blog]. 2018. [Consulta: 23 abril 2022]. Disponible en: <https://www.t-systemsblog.es/estas-son-las-capas-del-internet-de-las-cosas/>.

T.W.S. *Fundamentos de la API REST - Tree Web Solutions* [blog]. 2021. [Consulta: 27 abril

2022]. Disponible en: <https://treewebsolutions.com/articles/fundamentals-of-rest-api-68>.

TECNOLÓGICO DE MONTERREY. *Energía eléctrica: conceptos y principios básicos Potencia real, aparente y reactiva* [en línea]. Monterrey- México: Departamento de Ingeniería Eléctrica, 2022, p. 4. [Consulta: 28 abril 2022]. Disponible en: https://repositorio.tec.mx/bitstream/handle/11285/631150/5_t1s1_c5_pdf_1.pdf?sequence=1&isAllowed=y.

UNIVERSIDAD NACIONAL DE ROSARIO. *Potencia: Tipos de potencia* [en línea], Rosario-Argentina: Departamento de Ingeniería, p. 3. [Consulta: 28 abril 2022]. Disponible en: https://www.fceia.unr.edu.ar/tci/utiles/Apuntes/Cap_9_2013_POT.pdf.

VEPROF. *Xmpp Overview* [blog]. 2021. [Consulta: 13 junio 2022]. Disponible en: <https://www.veprof.com/blog/technology/xmpp-overview>.

VIENAŽINDYTĖ, ILMA. *TCP vs. UDP, comparamos los dos protocolos* [blog]. 2019. [Consulta: 6 mayo 2022]. Disponible en: <https://nordvpn.com/es/blog/protocolo-tcp-udp/>.

VIKRAM, SAHU. *WebSockets: A Conceptual Overview* [blog]. [Consulta: 10 octubre 2022]. Disponible en: <https://dev.to/sahuvikramp/websockets-a-conceptual-overview-41m>.

WALI-UR, RAHMAN.; et al. IoT smart meter for electricity consumers [en línea] (Trabajo de titulación). (Pregrado) Universidad de BRAC. Dhaka, Bangladés. 2019. p. 4. [Consulta: 27 abril 2022]. Disponible en: <http://dspace.bracu.ac.bd/xmlui/handle/10361/12331>.

WALKER, A. *What is an API? Full Form, Meaning, Definition, Types & Example* [blog]. 2022. [Consulta: 27 abril 2022]. Disponible en: <https://www.guru99.com/what-is-api.html>.

WIRELESSOLUTION. “Review of the 802.11 standards”. *WIRELESSDMX* [en línea], 2014, (México) 1(1), p. 2. [Consulta: 27 abril 2022]. Disponible en: <https://wirelessdmx.com/wp-content/uploads/2016/07/Review-of-the-802.11-standards-2-spalter.pdf>.

YACCHIREMA, DIANA. ARQUITECTURA DE INTEROPERABILIDAD DE DISPOSITIVOS FÍSICOS PARA EL INTERNET DE LAS COSAS (IOT) [en línea] (Trabajo de titulación). (Doctoral) Universidad Politécnica de Valencia, Valencia, España. 2019. pp. 85. [Consulta: 28 abril 2022]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/129858/Yacchirema%20-%20Arquitectura%20de%20Interoperabilidad%20de%20dispositivos%20f%C3%ADsicos%20para%20el%20Internet%20de%20las%20C....pdf?sequence=1>.

BROWN,T; et al. “Reducing domestic food waste by lowering home refrigerator temperatures” *ScienceDirect* [en línea], 2014, (Estados Unidos) 1(1). [Consulta: 22 abril 2022]. ISSN 0140-7007. Disponible en:

<https://www.sciencedirect.com/science/article/abs/pii/S0140700713003629>.

CSIRO. *Storing perishable food* [blog]. 2022. [Consulta: 22 diciembre 2022]. Disponible en:
<https://www.csiro.au/en/research/production/food/refrigerating-foods>.



ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 03/04/2023

INFORMACIÓN DEL AUTOR/A (S)	
Nombres – Apellidos: Diego Alejandro López Freire	
INFORMACIÓN INSTITUCIONAL	
Facultad: Informática y Electrónica	
Carrera: Telecomunicaciones	
Título a optar: Ingeniero en Electrónica Telecomunicaciones y Redes	
f. Analista de Biblioteca responsable:	
Ing. Fernanda Arévalo M.	

