



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO E IMPLEMENTACIÓN DE SISTEMAS IoT A UN ENTORNO
INDUSTRIAL UTILIZANDO UN ORDENADOR DE PLACA REDUCIDA**

Luis Emilio Ortíz Alvarez

Asesorado por el Ing. José Rodrigo de León Velásquez

Guatemala, junio de 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO E IMPLEMENTACIÓN DE SISTEMAS IoT A UN ENTORNO
INDUSTRIAL UTILIZANDO UN ORDENADOR DE PLACA REDUCIDA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA

FACULTAD DE INGENIERÍA

POR

LUIS EMILIO ORTÍZ ALVAREZ

ASESORADO POR EL ING. JOSÉ RODRIGO DE LEÓN VELÁSQUEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, JUNIO DE 2022

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Armando Alonso Rivera Carrillo
EXAMINADOR	Ing. Kenneth Issur Estrada Ruiz
EXAMINADOR	Ing. Natanael Jonathan Requena Gómez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO E IMPLEMENTACIÓN DE SISTEMAS IoT A UN ENTORNO INDUSTRIAL UTILIZANDO UN ORDENADOR DE PLACA REDUCIDA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 18 de noviembre de 2021.



Luis Emilio Ortiz Alvarez

Guatemala, 23 de febrero de 2022.

Ingeniero
Oscar Argueta Hernández
Director Unidad de EPS
Unidad del Ejercicio Profesional Supervisado
Facultad de Ingeniería, USAC

Señor Director:

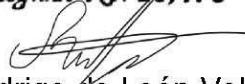
Por este medio tengo el gusto de informarle que he concluido con el asesoramiento y revisión del trabajo de graduación con título: **Diseño e implementación de sistemas IoT a un entorno industrial utilizando un ordenador de placa reducida**, desarrollado por el estudiante Luis Emilio Ortiz Alvarez con carnet 201701096, quien se encontraba incorporado en el programa del Ejercicio Profesional Supervisado en modalidad de 6 meses.

Después de revisar el contenido del informe final presentado doy mi entera aprobación del mismo.

Sin otro particular, le envío un cordial saludo.

Atentamente,

José Rodrigo de León Velásquez
Ingeniero Electrónico
Colegiado N°. 16,476


Ing. José Rodrigo de León Velásquez
Colegiado N°. 16,476
Asesor



Guatemala, 25 de Abril de 2022.
REF.EPS.DOC.195.05.2022.

Ing. Oscar Argueta Hernández
Director Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Argueta Hernández.

Por este medio atentamente le informo que como Supervisor de la Práctica del Ejercicio Profesional Supervisado (E.P.S.), del estudiante universitario **Luis Emilio Ortíz Alvarez** de la Carrera de Ingeniería Electrónica, Registro Académico No. **201701096** y **CUI 3101 71644 0614**, procedí a revisar el informe final, cuyo título es **“DISEÑO E IMPLEMENTACIÓN DE SISTEMAS IoT A UN ENTORNO INDUSTRIAL UTILIZANDO UN ORDENADOR DE PLACA REDUCIDA”**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todos”
Universidad de San Carlos de Guatemala
Ing. Kenneth Issur Estrada Ruiz
ASESOR - SUPERVISOR DE EPS
Unidad de Prácticas de Ingeniería y EPS
Ing. Kenneth Issur Estrada Ruiz
Supervisor de EPS
Área de Ingeniería Eléctrica
Facultad de Ingeniería

c.c. Archivo
KIER/ra



Guatemala 25 de abril de 2022.
REF.EPS.D.139.04.2022.

Ing. Armando Alonso Rivera Carrillo
Director Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Presente

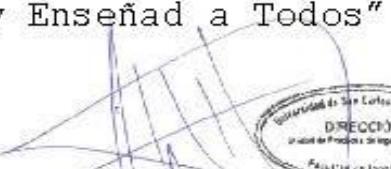
Estimado Ingeniero Rivera Carrillo.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado "**DISEÑO E IMPLEMENTACIÓN DE SISTEMAS IoT A UN ENTORNO INDUSTRIAL UTILIZANDO UN ORDENADOR DE PLACA REDUCIDA**" que fue desarrollado por el estudiante universitario, Luis Emilio Ortiz Alvarez, quien fue debidamente asesorado por el Ing. José Rodrigo de León Velásquez y supervisado por el Ing. Kenneth Issur Estrada Ruiz.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y Supervisor de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"


Ing. Oscar Argueta Hernández
Director Unidad de EPS

/ra



REF. EIME 38.2022.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Director de EPS, del Asesor, con el Visto Bueno del Coordinador de Área , al Informe final de EPS del estudiante Luis Emilio Ortíz Álvarez: **"DISEÑO E IMPLEMENTACIÓN DE SISTEMAS IoT A UN ENTORNO INDUSTRIAL UTILIZANDO UN ORDENADOR DE PLACA REDUCIDA"**, procede a la autorización del mismo.



Ing. Armando Alonso Rivera Carrillo

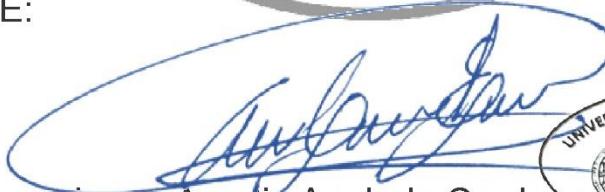
Guatemala, 23 de mayo de 2022.

Decanato
Facultad de Ingeniería
24189101- 24189102
secretariadecanato@ingenieria.usac.edu.gt

LNG.DECANATO.OI.484.2022

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado **DISEÑO E IMPLEMENTACIÓN DE SISTEMAS IoT A UN ENTORNO INDUSTRIAL UTILIZANDO UN ORDENADOR DE PLACA REDUCIDA**, presentado por **Luis Emilio Ortíz Alvarez**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



inga. Aurelia Anabela Cordova Estrada



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Decana

Guatemala, julio de 2022

AACE/gaoc

ACTO QUE DEDICO A:

Dios	Por concederme el don de la sabiduría y perseverancia para cumplir mis objetivos.
Mis padres	Irma Yolanda Alvarez Girón, por ser quien me cuidará, educará y animará desde pequeño a seguir adelante, por sus valores inculcados que me hacen ser la persona de hoy en día. Mainor Ortiz Garzo, por enseñarme el valor del trabajo y el esfuerzo.
Mis hermanos	Lcdo. Mynor Alfredo Ortiz Alvarez, por aconsejarme y apoyarme en el transcurso de mi vida personal y académica. Cristian Ortiz y Rony Ortiz, por su apoyo brindado desde niños hasta cumplir esta meta.
Mi tía	Hercilia Ortiz Garzo, por ser mi segunda madre y una luz en mi camino, quien me ha apoyado incondicionalmente y ha creído en mí. Por permitirme llegar hasta este momento, este logro también es suyo.
Mis tíos	Arcadia Alvarez, Alejandro Parada, por apoyar siempre a mi familia. Aracely Ortiz, por su consejo oportuno en la etapa final de mi carrera.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser la casa de estudio que permite cumplir los sueños de muchas personas.
Facultad de Ingeniería	Por ser mi centro de formación profesional, por todos los conocimientos impartidos en clases.
Mis amigos	Jesse Contreras y Edgar García, por compartir conmigo momentos divertidos y agotadores en este camino y en la realización de proyectos.
Inga. María Fernanda Barrera Ortíz	Por creer en mí y haberme dado la oportunidad de mostrar mis capacidades como profesional.
IELOU ENERGY	Por darme la oportunidad de realizar mis prácticas y haberme incluido en este proyecto. Especialmente al Ing. José Alvarez, por confiar en mis habilidades, por su invaluable apoyo y al Ing. Hamilton Ixcot, por haber compartido su conocimiento y experiencias en este trabajo.
Ing. José de León	Por su valiosa asesoría y por su tiempo dedicado en este proyecto.
Walter Rodríguez	Por su innegable apoyo y por sus conocimientos impartidos al iniciar con mi carrera.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN	XIX
OBJETIVOS	XXI
INTRODUCCIÓN	XXIII
1. GENERALES DE LA INSTITUCIÓN	1
1.1. Descripción de la institución	1
1.2. Reseña histórica	1
1.3. Visión	1
1.4. Misión	2
1.5. Ubicación	2
1.6. Organigrama	2
2. MARCO TEÓRICO	3
2.1. Internet de las Cosas	3
2.1.1. Internet Industrial de las Cosas	4
2.2. Revolución Industrial 4.0	4
2.2.1. Pilares de la Industria 4.0	4
2.2.1.1. Análisis de datos y <i>big data</i>	5
2.2.1.2. Robótica colaborativa	5
2.2.1.3. Simulación	5
2.2.1.4. Sistemas de integración vertical y horizontal	6

2.2.1.5.	Comunicación máquina a máquina	6
2.2.1.6.	Ciberseguridad.....	6
2.2.1.7.	Computación en la nube	7
2.2.1.8.	Manufactura aditiva.....	7
2.2.1.9.	Realidad aumentada	7
2.3.	Automatización	8
2.4.	Telemetría	9
2.5.	Eficiencia energética.....	9
2.6.	Auditorías energéticas	9
2.7.	Marco conceptual	10
2.7.1.	Ordenador de placa reducida.....	11
2.7.1.1.	Raspberry Pi	12
2.7.2.	Microcontroladores	16
2.7.2.1.	Tarjetas de desarrollo.....	17
2.7.2.2.	ESP32.....	18
2.7.3.	Protocolos de comunicación	22
2.7.3.1.	Protocolos industriales de comunicación	23
2.7.3.2.	Protocolos de red	26
2.7.3.3.	Protocolos del Internet de las Cosas	27
2.7.4.	Módulos de comunicación.....	29
2.7.4.1.	Módulo MAX485.....	30
2.7.4.2.	Conversor USB a RS485	31
2.7.5.	Sensores.....	33
2.7.5.1.	Sensor de DHT22	33
2.7.5.2.	Sensor magnético	34
2.7.5.3.	Sensores de temperatura	35
2.7.5.4.	Sensor de corriente	38
2.7.5.5.	Analizadores de energía	41

2.7.5.6.	Circuitos complementarios	42
2.7.6.	Software utilizado.....	45
2.7.6.1.	Sistema operativo Raspbian.....	45
2.7.6.2.	Node-RED.....	46
2.7.6.3.	Grafana	46
2.7.6.4.	Bases de datos	47
2.7.7.	Microservicios	47
2.7.8.	Servicios en la <i>web</i>	48
2.7.9.	Variables de medición	48
2.7.10.	Tecnologías complementarias	49
2.8.	Marco legal	51
2.8.1.	Normativa ISO/IEC 30141	51
2.8.2.	Normativa ISO/IEC 27033.....	51
2.8.3.	Normativa ISO 50001:2011	52
2.8.4.	Normativa UNE-EN 16247	52
3.	MARCO METODOLÓGICO	55
3.1.	Descripción del proyecto.....	55
3.2.	Arquitectura de aplicaciones IoT	56
3.2.1.	Requerimientos para la implementación	56
3.2.2.	Modelo de capas IoT.....	57
3.2.3.	Fases de integración del IoT en la industria	60
3.3.	Arquitectura del protocolo MQTT	61
4.	DESARROLLO DEL PROYECTO	63
4.1.	Equipos y especificaciones técnicas	63
4.2.	Medición y adquisición de datos	69
4.3.	Diagramas de conexiones	70
4.3.1.	Sistema de monitoreo de energía eléctrica	71

4.3.2.	Sistema de monitoreo de cámaras a bajas temperaturas.....	72
4.3.3.	Sistema de monitoreo de alta temperatura.....	73
4.4.	Conexiones de servicios	74
4.4.1.	Conexiones MQTT	74
4.4.2.	Diagrama de aplicaciones en plataforma de monitoreo	75
4.4.3.	Diagrama de aplicaciones para conexión remota....	76
4.5.	Diseño y fabricación de PCB expandida	77
4.6.	Diseño e impresión 3D	77
5.	FASE DE SERVICIO TÉCNICO PROFESIONAL, MANUAL DE CONFIGURACIONES BÁSICAS PARA ESTABLECER COMUNICACIÓN ENTRE LOS DISPOSITIVOS.....	79
5.1.	Análisis de la situación actual	79
5.2.	Manual de configuraciones	80
5.2.1.	Objetivos del manual.....	80
5.2.2.	Alcances del manual	80
5.2.3.	Configuraciones en Raspberry Pi.....	81
5.2.3.1.	Configuraciones generales en el sistema operativo Raspbian	81
5.2.3.2.	Configuraciones en conexión remota ...	83
5.2.3.3.	Programación Python – MODBUS	84
5.2.3.4.	Programación Node-RED – MODBUS	85
5.2.3.5.	Programación Node-RED – MQTT.....	88
5.2.3.6.	Configuración de nodos dashboard.....	89
5.2.3.7.	Programación Node-RED – base de datos	90

5.2.3.8.	Configuración de base de datos en MariaDB	94
5.2.3.9.	Configuraciones en Grafana.....	95
5.2.3.10.	Implementación de servidor ngrok.....	97
5.2.4.	Configuraciones en tarjeta de desarrollo ESP32	98
5.2.4.1.	Programación de sensores.....	99
5.2.4.2.	Programación Arduino – MODBUS	101
5.2.4.3.	Programación Arduino – MQTT.....	103
5.2.4.4.	Programación Arduino – OTA	106
6.	FASE DE INVESTIGACIÓN, PLAN DE CONTIGENCIA CONTRA POSIBLES FALLAS O BAJAS DEL SISTEMA	111
6.1.	Pérdidas de energía eléctrica en los dispositivos.....	111
6.2.	Desconexión entre la red de internet y los dispositivos	112
6.3.	Sistema de ciberseguridad	113
6.4.	Mantenimiento preventivo a los dispositivos	113
6.5.	Condiciones ideales para la instalación de los dispositivos....	114
7.	FASE DE ENSEÑANZA APRENDIZAJE	117
7.1.	Demostración del funcionamiento del sistema al departamento de la empresa	117
7.2.	Propuestas para optimizar el consumo energético.....	118
7.2.1.	Cálculos para banco de capacitores	118
7.2.2.	Dispositivos controladores de potencia	120
7.3.	Conferencias en los cursos de Electrónica 5 y 6, en la Facultad de Ingeniería USAC	122
8.	ANÁLISIS DE RESULTADOS	125
8.1.	Análisis financiero.....	125

8.1.1. Comparación con principales alternativas	128
8.2. La Industria 4.0 y sus nuevas posibilidades	130
8.3. Aplicación IoT para la supervisión energética en ambientes industriales	132
CONCLUSIONES	135
RECOMENDACIONES	137
BIBLIOGRAFÍA	139
APÉNDICES	149
ANEXOS	157

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Organigrama IELOU ENERGY	2
2. Estructura de un sistema automatizado	8
3. Diagrama de bloques general SBC	12
4. Esquemático de especificaciones mecánicas.....	13
5. Raspberry Pi 4 modelo B+	14
6. Pines GPIO de Raspberry Pi.....	15
7. Arquitectura básica de un microcontrolador	17
8. Módulo SoC ESP32	18
9. Tarjeta de desarrollo ESP32	19
10. Diagrama de bloques de funciones ESP32	20
11. Pines GPIO de ESP32	20
12. Lenguaje de programación y entorno de desarrollo integrado	22
13. Esquemático del módulo MAX485	30
14. Pinout módulo MAX485.....	31
15. Diagrama conversor USB a RS485	32
16. Conversor USB a RS485	32
17. Sensor DHT22	33
18. Sensor magnético de apertura	34
19. Sensor GY-906	35
20. Distintas presentaciones de los termopares	37
21. RTD utilizadas en la industria.....	37
22. Principio de la detección de corriente en conductores.....	39
23. Transformador de corriente toroidal	39

24.	Sensor transformador de corriente SCT013	40
25.	Sensor de corriente ACS712.....	41
26.	Analizador de redes CIRCUTOR CVM-C10	42
27.	Módulo MAX6675 para termocupla	43
28.	Módulo ADC ADS1115.....	43
29.	Terminales de un Puente de Wheatstone	44
30.	Transmisor de 4 – 20mA.....	45
31.	Las 7 capas de la arquitectura IoT	58
32.	Arquitectura IoT de 4 capas	58
33.	Arquitectura IoT de 3 capas	59
34.	Arquitectura de componentes IoT	60
35.	Topología entre clientes MQTT	62
36.	Jerarquía de clientes MQTT	62
37.	Localizaciones físicas en la Raspberry Pi 4.....	65
38.	Arquitectura IoT implementada	70
39.	Diagrama CT, analizador y Raspberry Pi	71
40.	Diagrama sensor magnético, DHT22 y ESP32.....	72
41.	Diagrama termocuplas, MAX6675 y ESP32	73
42.	Esquema MQTT con broker Mosquitto	74
43.	Esquema de aplicaciones	76
44.	Conexión remota desde ordenador hacia Raspberry Pi	77
45.	PCB expandida y case impreso en 3D	78
46.	Configuración de interfaces en Raspberry Pi	82
47.	Parámetros a ingresar en VNC	82
48.	Conexión generada desde remote.it.....	83
49.	Código en Python para comunicación MODBUS (parte 1)	84
50.	Código en Python para comunicación MODBUS (parte 2)	85
51.	Gestor de búsqueda de módulos en Node-RED	86
52.	Configuraciones en nodo Modbus-Read	86

53.	Configuraciones MODBUS en Node-RED	87
54.	Configuraciones del nodo mqtt in	88
55.	Configuraciones para un servidor MQTT en Node-RED	89
56.	Configuraciones del nodo mqtt out	89
57.	Configuraciones de distintos nodos <i>dashboard</i>	90
58.	Configuración nodo <i>inject</i>	91
59.	Programación para insertar datos en MariaDB	92
60.	Configuración del nodo MySQL	92
61.	Programación para generar diccionario http	93
62.	Configuración en nodo http request	93
63.	Configuraciones MySQL en Grafana	95
64.	Selección de datos en panel Grafana	96
65.	Creación de nuevos usuarios en Grafana	97
66.	Terminal de servicio ngrok	98
67.	Programación de sensor DHT22	99
68.	Programación de sensor magnético y conteo de aperturas	100
69.	Programación de sensor MAX6675	101
70.	Arduino sketch simple master MODBUS	102
71.	Arduino sketch simple slave MODBUS	103
72.	Programación inicial para ESP32 - MQTT	104
73.	Programación para conectar ESP32 a red WiFi	104
74.	Programación de cliente receptor en la ESP32	105
75.	Programación para suscribir la ESP32 al <i>broker</i>	105
76.	Programación para enviar datos desde la ESP32	106
77.	Pestaña principal en programación OTA	107
78.	Código para conexión WiFi en OTA	108
79.	Código de funciones OTA	109
80.	Sistema disipador de calor en Raspberry Pi	115
81.	Presentación del sistema de monitoreo a la empresa	117

82.	Fórmula de banco de capacitores	119
83.	Ejemplo de cálculo para banco de capacitores	119
84.	Capacitor industrial	120
85.	Arrancador suave marca SIEMENS	121
86.	Variador de frecuencia	122
87.	Presentación del sistema de monitoreo a la comunidad estudiantil....	123
88.	Revoluciones industriales.....	131

TABLAS

I.	Características técnicas de los protocolos	25
II.	Tipos de termopares y sus aplicaciones.....	36
III.	Temperatura y resistencia en una PT100.....	38
IV.	Diferencias entre RS232 y RS485.....	49
V.	Especificaciones técnicas TC4 marca CIRCUTOR	63
VI.	Especificaciones técnicas CVM-C5	64
VII.	Especificaciones técnicas Conversor USB a RS485	65
VIII.	Especificaciones enrutador Huawei B311-221	66
IX.	Especificaciones técnicas sensor DHT22.....	66
X.	Especificaciones técnicas del sensor magnético de apertura.....	67
XI.	Especificaciones técnicas termocupla	67
XII.	Especificaciones técnicas MAX6675	68
XIII.	Especificaciones técnicas ESP32	68
XIV.	Presupuesto de <i>hardware</i> para medición en un punto	125
XV.	Presupuesto de servicios y <i>software</i> de monitoreo.....	127
XVI.	Presupuesto de instalación en un punto.....	128
XVII.	Comparación de alternativas para sistemas de monitoreo	129

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperio
bit/s	Bit por segundo
C	Capacitancia
'0'	Cero lógico
ca	Corriente alterna
cc	Corriente continua
I_n	Corriente nominal
CO₂	Dióxido de carbono
fps	Fotogramas por segundo
GB	Gigabytes
GHz	Gigahertz
°C	Grados Celsius
Hz	Hertz
RH	Humedad relativa
KA	Kiloamperio
Kbit/s	Kilobits por segundo
KB	Kilobytes
kV	Kilovoltio
±	Más menos
Mbit/s	Megabits por segundo
MHz	Megahertz
<	Menor que
m	Metros

uA	Microamperios
uF	Microfaradios
mA	Miliamperios
mm²	Milímetros cuadrados
mV	Milivoltios
0x48	Número hexadecimal
Ω	Ohmios
%	Por ciento
“	Pulgadas
4K	Resolución de pantalla
s	Segundos
‘1’	Uno lógico
VAR	Voltamperio reactivo
V	Voltios
W	Watts

GLOSARIO

Actuador	Elemento electromecánico que activa funciones.
API	<i>Application Programming Interface.</i> Interfaz de programación de aplicaciones.
Backend	Procesos no visibles por el usuario encargados del funcionamiento de una página <i>web</i> .
BLE	<i>Bluetooth Low Energy.</i> Bluetooth de baja energía.
Bus de datos	Conexiones físicas para la transmisión de datos.
Carga eléctrica	Dispositivos, equipos o sistemas que consumen energía eléctrica en una instalación.
Case	Caja que alberga tarjetas electrónicas en su interior.
Ciber físico	Sistema controlado o monitorizado por algoritmos de computación relacionados con el internet.
CPU	<i>Central Processing Unit.</i> Unidad Central de Procesamiento, encargada de procesar datos.
CT	<i>Current Transformer.</i> Transformador de corriente, sensible a la inducción electromagnética.

Dashboard	Tablero de mandos.
DMA	<i>Direct Memory Access</i> , por sus siglas en inglés. Acceso directo a memoria.
Edge computing	Paradigma de la computación destinado a la mejora de datos para optimizar el tiempo de respuesta.
Embebido	Sistema diseñado para realizar funciones únicas.
Frontend	Aplicaciones <i>web</i> que interactúan con los usuarios.
FSK	<i>Frequency Shift Keying</i> . Modulación por desplazamiento de frecuencia, utilizada para la transmisión digital.
Gateway	Puerta de enlace o pasarela, interactúa con los dispositivos de una red para comunicarlos.
GNU/Linux	Serie de sistemas operativos basados en Unix.
GPU	<i>Graphics Processing Unit</i> , por sus siglas en inglés. Unidad de procesamiento gráfico.
Half duplex	Sistema de comunicación bidireccional capaz de recibir y enviar información de forma no simultánea.
Hardware	Partes tangibles de equipos eléctricos o electrónicos.

Host	Dispositivo conectado a una red.
IDLE	<i>Integrated Development Environment.</i> Entorno de desarrollo y aprendizaje integrado.
IEC	<i>International Electrotechnical Commission.</i> Comisión Electrotécnica Internacional.
IoT	<i>Internet of Things.</i> Internet de las cosas, hace posible la interconexión de dispositivos simples a la red.
IP	<i>Internet Protocol.</i> Protocolo de Internet, identifica a un dispositivo conectado a una red local o externa.
ISO	<i>International Organization for Standardization.</i> Organización Internacional de Normalización.
Kernel	Núcleo del sistema operativo.
LAN	<i>Local Area Network.</i> Red de área local, destinada para hogares o edificios.
Lenguaje máquina	Códigos entendibles por el microprocesador.
M2M	<i>Machine to Machine</i> , se refiere a la comunicación e intercambio de información entre máquinas.
Machine learning	Aprendizaje de las máquinas por medio de sistemas de inteligencia artificial.

Medidores	Instrumentos o dispositivos utilizados para medir una variable.
Middleware	<i>Software</i> intermedio, brinda servicios y funciones adicionales a las del sistema operativo.
Modelo OSI	<i>Open Systems Interconnection.</i> Modelo de interconexión de sistemas abiertos.
Nodo	Punto de intersección o unión de distintos bloques.
Nube	Paradigma de los servicios remotos provistos desde la red por medio de una conexión a internet.
Open software	Programas de código abierto utilizables por el público.
Ordenador	Computadora que ejecuta comandos, procesa datos y realiza operaciones lógicas programables.
OS	<i>Operating System</i> , por sus siglas en inglés. Sistema operativo.
Pinout	Asignación de pines, indica la disposición y función de los pines de un circuito integrado.
Plugins	Complementos, agregan una función nueva.
PoE	<i>Power over Ethernet.</i> Alimentación a través de Ethernet.

<i>Proxy</i>	Equipo intermediario entre las conexiones de un cliente y un servidor de destino.
<i>Pull</i>	Técnica para obtener datos cuando es requerido.
<i>QoS</i>	<i>Quality of Service</i> , por sus siglas en inglés. Calidad de servicio.
<i>RAM</i>	<i>Random Access Memory</i> , por sus siglas en inglés. Memoria de acceso aleatorio.
<i>Remoto</i>	Hace referencia a sistemas que pueden ser accedidos y que se encuentran a una distancia lejana.
<i>ROM</i>	<i>Read Only Memory</i> . Memoria de solo lectura, almacena información, datos e instrucciones.
<i>Semiduplex</i>	Tipo de comunicación bidireccional no simultánea.
<i>Sketch</i>	Bloques de código programados en Arduino IDE.
<i>Socket</i>	Interfaz para la entrada y salida de datos que permite la comunicación entre procesos, asociando la dirección <i>IP</i> y un número de puerto.
<i>Software</i>	Programas y rutinas en una computadora.
<i>Tiempo real</i>	Hace referencia a la actualización de información de forma inmediata.

Transductor	Dispositivo que convierte una magnitud de entrada en otra diferente en la salida.
Transiente	Fenómeno eléctrico que afecta a las señales de voltaje y corriente produciendo picos en rango de KA.
Túnel de red	Técnica utilizada para encapsular un protocolo de red sobre otro para establecer comunicación.
UART	<i>Universal Asynchronous Receiver-Transmitter</i> , por sus siglas en inglés. Transmisor-Receptor asíncrono universal.
UPS	<i>Uninterruptible Power Supply</i> . Sistema de alimentación ininterrumpida, baterías que proveen energía eléctrica por un periodo de tiempo.
URL	<i>Uniform Resource Locator</i> . Localizador de recursos uniforme, identifica las direcciones asignadas.
WAN	<i>Wide Area Network</i> . Red de área extensa.
WEP	<i>Wired Equivalent Privacy</i> . Privacidad equivalente a cableado, cifra la transmisión en redes inalámbricas.
WiFi	<i>Wireless Fidelity</i> . Fidelidad inalámbrica, conecta dispositivos a internet por medio de una red sin cables.
WPA	<i>WiFi Protected Access</i> , por sus siglas en inglés.

RESUMEN

La implementación de un sistema de monitoreo energético ambiental dentro del sector industrial consta de una serie de controles y visualizaciones que en la actualidad son requeridos por las empresas. El sistema propuesto está orientado en utilizar dispositivos electrónicos de bajo coste que provean conectividad a los sensores y equipos que no disponen de una, con el fin de almacenar y analizar los datos censados generando información de valor.

El primer capítulo muestra los aspectos generales de la empresa para la cual se desarrolló el proyecto. En el segundo capítulo se presentan las normativas en las cuales se fundamenta el desarrollo del proyecto, describiendo teóricamente cada concepto tecnológico, así como los detalles de los dispositivos, lenguajes de programación, protocolos y *software* utilizados; y en complemento el tercer capítulo define la metodología para implementar una arquitectura IoT segmentada en distintos niveles de aplicación.

Por consiguiente, el cuarto y quinto capítulo describen las características técnicas de cada dispositivo, la topología conformada en cada sistema y las configuraciones requeridas para establecer comunicación entre los dispositivos y la red de internet. Mientras tanto, en el sexto capítulo se consideran las posibles fallas del sistema y se plantea una solución para mitigar los efectos causados.

El séptimo y octavo capítulo convergen en la explicación y análisis del sistema de monitoreo implementado, finalizando con la comprensión de los beneficios que el IoT y la industria 4.0 brindan como valor agregado a las empresas que integran estas soluciones a sus modelos de negocio.

OBJETIVOS

General

Desarrollar diferentes sistemas basados en el concepto del Internet de las Cosas a través de la implementación de un ordenador de placa reducida en conjunto con tecnologías actuales, logrando obtener un sistema competente que opere de forma eficaz en ambientes industriales.

Específicos

1. Analizar los sistemas y dispositivos que se usan actualmente en la empresa y compararlos contra sistemas embebidos que pueden integrarse a un ordenador de placa reducida.
2. Detallar las diferentes aplicaciones IoT que pueden integrarse en un ordenador de placa reducida, considerando ventajas y desventajas que estas implican al ser llevadas a la práctica en el marco de la empresa.
3. Plantear soluciones para mejorar los procesos actuales conservando la calidad en los servicios finales, optimizando los factores energéticos y de producción.

INTRODUCCIÓN

La aplicación del Internet de las Cosas en los últimos años en Guatemala ha tenido un crecimiento considerable, llegando a implementarse en hogares y recientemente en la industria. Debido a las mejoras que esto conlleva a muchos procesos y servicios a nivel industrial se les incorporan controladores y monitoreos del tipo IoT.

La base del proyecto se fundamenta en especificar los distintos sistemas que se pueden integrar y presentar casos en los cuales se haya aplicado este tipo de tecnologías. Actualmente un factor importante para poder mejorar los procesos son los datos que se obtienen a través de sensores y que son almacenados para un análisis en tiempo real o posterior, con lo cual se pueden tomar decisiones más acertadas respecto a las variables que se requieran controlar. Sincrónicamente obteniendo un control digital centralizado contribuye a optimizaciones relacionadas a la siguiente revolución industrial 4.0, siendo los dispositivos inteligentes conectados la clave para alcanzar estos objetivos.

Esto implica una mejora consiguiendo reutilizar los dispositivos con los que la empresa disponga y por medio de sistemas en los que el mundo físico y virtual se encuentran para conectarse a una red general haciendo uso de protocolos de comunicación, beneficiando en que los medidores, actuadores y datos ya no estarán solamente de forma local en el lugar determinado en donde fueron instalados, sino que, se podrá acceder a estas mediciones y controles desde cualquier dispositivo que posea una conexión a internet.

1. GENERALES DE LA INSTITUCIÓN

1.1. Descripción de la institución

Destacamos como actividad principal la gestión y nuevos usos de energía, agregando a nuestros valores:

- Innovación, siempre creando y cambiando.
- Optimización, aprovechamos y utilizamos de forma eficiente los recursos disponibles.
- Sostenibilidad, creación de valor compartido para asegurar negocios a largo plazo.
- Agilidad, nos adaptamos rápido para capitalizar oportunidades.¹

1.2. Reseña histórica

En el 2015 nace IELOU ENERGY en Guatemala y en el 2019 inicia actividades en El Salvador para dar servicios de generación distribuida, ese mismo año hemos iniciado con el mismo servicio en Pradera Zacapa. Desde entonces estamos creciendo con servicios de generación distribuida y gestión energética en Guatemala, El Salvador, Honduras y Costa Rica.²

1.3. Visión

“Aprovechamos nuestra energía limpia y capacidades técnicas para minimizar la huella y maximizar el valor económico y reputacional de CMI”.³

¹ IELOU ENERGY. *Servicios energéticos distribuidos*. Consultado el 19 de diciembre de 2021.

² Ibid.

³ Ibid.

1.4. Misión

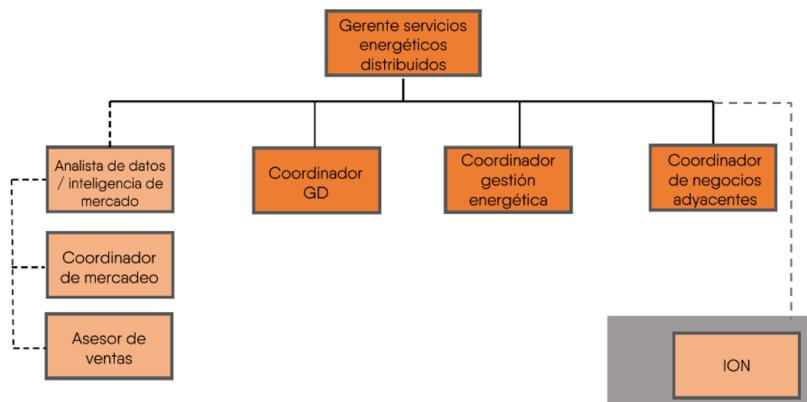
“Integraremos productos y servicios de energía inteligente a medida de tu negocio. Proporcionamos soluciones que te ayudan a ahorrar dinero y recursos naturales a través de autogeneración renovable y servicios de gestión energética”.⁴

1.5. Ubicación

“Contamos con oficinas ubicadas en Boulevard Los Próceres, Zona 10, Edificio Empresarial. Ciudad de Guatemala”.⁵

1.6. Organigrama

Figura 1. Organigrama IELOU ENERGY



Fuente: IELOU ENERGY, S.A.

⁴ IELOU ENERGY. *Servicios energéticos distribuidos*. Consultado el 19 de diciembre de 2021.

⁵ Ibid.

2. MARCO TEÓRICO

Para introducir el concepto del Internet de las Cosas y el impacto que ha tenido en el sector industrial, se hace necesario considerar algunos de los términos que han posicionado a las aplicaciones del tipo IoT como potenciales soluciones en las empresas. Con el fin de tener un panorama completo de los sistemas que el IoT puede interconectar, se iniciará definiendo algunos criterios de valor en el campo de la industria.

2.1. Internet de las Cosas

Es la columna vertebral sobre la cual se desarrollan otros conceptos que parten de las mismas bases, siendo el Internet de las Cosas o *Internet of Things* (IoT) una referencia a grandes cantidades de objetos conectados digitalmente. En términos teóricos, su objetivo es lograr que la mayoría de objetos estén permanentemente conectados al internet, mejorando su funcionalidad con otras posibles utilidades.

Si bien en un inicio estaba limitado a ordenadores, teléfonos inteligentes y *tablets*, en los últimos años ha aumentado exponencialmente el número y variedad de dispositivos conectados con finalidades aún más diversas. Entre los campos que han aplicado el IoT están:

- IIoT: Internet Industrial de las Cosas
- IoHT: Internet de las Cosas del hogar
- IoMT: Internet de las Cosas en la Medicina
- Agricultura 4.0 o agricultura de precisión

- Granjas inteligentes
- Ciudades inteligentes

2.1.1. Internet Industrial de las Cosas

Comprende el conjunto de sensores, actuadores, instrumentos, dispositivos autónomos y equipos conectados a través de internet a aplicaciones industriales. Esta red permite recopilar datos, realizar análisis y optimizar la producción aumentando la eficiencia y reduciendo los costes del proceso de fabricación y prestación de servicios.⁶

Las aplicaciones industriales son ecosistemas tecnológicos completos que conectan dispositivos y a estos con las personas que gestionan los procesos en líneas de montaje, logística o distribución a gran escala modificando su forma de actuar y relacionarse en los procesos sociales, industriales y de negocio.⁷

2.2. Revolución Industrial 4.0

En los últimos años ha surgido una cuarta revolución industrial conocida como Industria 4.0, haciendo énfasis en el uso de las tecnologías recientes, escalando a un nivel totalmente diferente con la aplicación de la interconectividad a través del Internet de las Cosas, el acceso en tiempo real y la introducción de los sistemas ciber físicos.

2.2.1. Pilares de la Industria 4.0

La Industria 4.0 integra conceptos tecnológicos esenciales que impulsan a su evolución. Los ejes en los cuales está articulada son cubiertos por nueve puntos clave:

⁶ IBERDOLA. *Internet Industrial de las Cosas (IIOT)*. Consultado el 19 de diciembre de 2021. Recuperado de <https://cutt.ly/5JJfD0n>.

⁷ Ibid.

2.2.1.1. Análisis de datos y *big data*

Analizar y administrar grandes cantidades de datos está permitiendo un mayor rendimiento en la optimización de procesos industriales, mejorando el consumo de energía y la calidad de la producción en empresas que saben cómo utilizar este paradigma tecnológico.⁸

2.2.1.2. Robótica colaborativa

Los robots colaborativos inteligentes son capaces de procesar información del ambiente y tomar decisiones en base a ella, tratando de implementar una forma de trabajo en conjunto con otras máquinas y con personal humano de forma segura, fiable y completamente autónoma.⁹

2.2.1.3. Simulación

Permite a los operadores probar y optimizar la configuración de las máquinas para el siguiente producto en la línea de producción virtual antes de un cambio real, a la vez el desarrollo y perfeccionamiento de prototipos se facilita haciendo uso de sistemas en computadoras que realizan análisis y proporcionan beneficios en la detección de fallas e identificación de soluciones.¹⁰

⁸ DEL VAL, José. *Industria 4.0: la transformación digital de la industria*. Consultado el 20 de diciembre de 2021.

⁹ Ibid.

¹⁰ Ibid.

2.2.1.4. Sistemas de integración vertical y horizontal

Se enfoca en los procedimientos utilizados para enlazar los distintos subsistemas o elementos en un único sistema que puede actuar como uno solo. El sistema horizontal se enfoca en la digitalización de toda la cadena de valor y suministro, es decir desde proveedores hasta clientes; mientras que el sistema vertical integra la gestión de la tecnología de la información en distintos niveles.¹¹

2.2.1.5. Comunicación máquina a máquina

Referido al uso de tecnologías que permiten la transferencia de información entre los productos y sistemas inteligentes, siendo la comunicación M2M la tecnología básica del Internet de las Cosas interconectando componentes que estaban desconectados y ahora se encuentran comunicados con la red.¹²

2.2.1.6. Ciberseguridad

Debido al aumento de la conectividad y el uso de protocolos se vuelve obligatorio proteger los sistemas industriales y líneas de fabricación de los ataques a la información, con lo cual se deben implementar comunicaciones seguras, fiables y sofisticadas para la gestión de identidades y accesos de equipos y usuarios.¹³

¹¹ DEL VAL, José. *Industria 4.0: la transformación digital de la industria*. Consultado el 20 de diciembre de 2021.

¹² Ibid.

¹³ Ibid.

2.2.1.7. Computación en la nube

Integra aplicaciones e infraestructuras utilizadas como servicios a través de la red. Los sistemas inteligentes generan una enorme cantidad de datos que son almacenados y procesados con la capacidad de ser accesibles *on-line* desde cualquier lugar, permitiendo el flujo de datos sin limitaciones y sustituyendo la inversión en unidades de almacenamiento físicas.¹⁴

2.2.1.8. Manufactura aditiva

Implica la producción rápida de prototipos en forma de capas de material superpuestas normalmente en forma de polvo, con la finalidad de obtener un modelo tridimensional que permite la creación de productos a medida que incluyen ventajas en la fabricación aditiva de formas complejas.¹⁵

2.2.1.9. Realidad aumentada

Siendo uno de los menos desarrollados en la industria 4.0, se prevé compatibilidad con la mayoría de las aplicaciones y servicios en diferentes campos, proporcionando a los operadores una visualización previa de cómo realizar determinados trabajos, con lo cual se incrementará la eficiencia y seguridad.¹⁶

¹⁴ DEL VAL, José. *Industria 4.0: la transformación digital de la industria*. Consultado el 20 de diciembre de 2021.

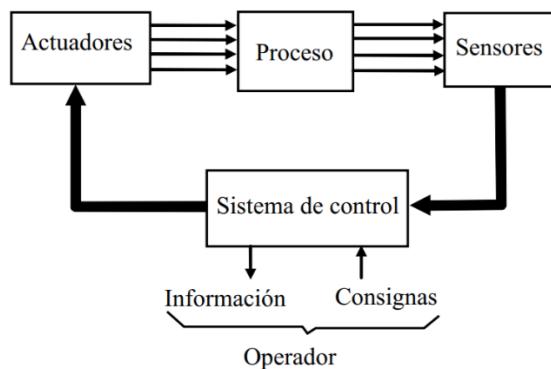
¹⁵ Ibid.

¹⁶ Ibid.

2.3. Automatización

Una máquina, proceso o sistema automatizado es aquel que tiene la capacidad de tomar decisiones de forma autónoma (sin intervención del operador) en respuesta a cambios que se han producido, realizando acciones adecuadas y cumpliendo la función que se le ha programado o diseñado.

Figura 2. **Estructura de un sistema automatizado**



Fuente: SANCHIS, Roberto; ROMERO, Julio; ARIÑO, Carlos. Capítulo 1: Introducción a los automatismos. Consultado el 21 de diciembre de 2021. Recuperado de <http://repositori.uji.es/xmlui/bitstream/handle/10234/24182/s31.pdf?sequence=6&isAllowed=y>.

Actualmente se ha seguido la tendencia de automatizar progresivamente los sistemas productivos de cualquier tipo, esto debido a que el desarrollo y abaratamiento de tecnologías utilizadas en estos sistemas ha hecho posible su implementación con el fin de obtener los siguientes objetivos:

- Mejorar la calidad y mantener un nivel uniforme.
- Producir las cantidades necesarias en el momento preciso.
- Mejorar la productividad y reducir costes.
- Flexibilidad en el sistema productivo (facilitar cambios en la producción).

2.4. Telemetría

Consiste en un sistema de comunicación automatizado con la función de recopilar datos de lugares remotos, disponiendo de características para recoger la información, procesarla y posteriormente transmitirla hacia el lugar en donde se encuentra el sistema de monitoreo con capacidad de ser operado vía remota facilitando el control de equipos, corregir errores o realizar modificaciones. Estos sistemas resultan útiles al evitar tener que desplazarse al sitio para obtener información, además de que ofrecen datos constantes en tiempo real.

2.5. Eficiencia energética

Es el conjunto de acciones que permiten hacer un uso óptimo de la cantidad de energía consumida en relación con los productos o servicios finales obtenidos, esto conlleva al hecho de minimizar la cantidad de energía necesaria para satisfacer la demanda sin afectar la calidad. Esto se puede llevar a cabo por medio de la implementación de distintas medidas de gestión, hábitos culturales y la sustitución de equipos por otros, que, con las mismas prestaciones tenga un menor consumo de electricidad.

2.6. Auditorías energéticas

Aplicada como un procedimiento por medio del cual se obtiene información objetiva y de valor acerca del consumo de energía de una instalación, con el fin de poder detectar cuales son los factores cruciales que afectan dicho consumo. Teniendo esta información se puede entender de qué forma se está utilizando la energía, identificando dónde, cuándo y cómo se está aprovechando o despilfarrando para poder establecer y gestionar las posibles estrategias y políticas de ahorro energético, desde el punto de vista técnico y económico.

Uno de los objetivos de una auditoría energética es proporcionar un informe técnico en el cual se detallen medidas adecuadas y enfocadas en:

- Identificar la situación actual, así como la operatividad y eficiencia de los equipos e instalaciones.
- Documentar los equipos principales e instalaciones existentes.
- Realizar mediciones y registro de los principales parámetros eléctricos, térmicos y de confort.
- Plantear recomendaciones a implementar, realizando el análisis financiero y técnico.
- Cuantificar los ahorros del factor energético y económico, obteniendo los períodos de retorno de inversiones.

Asimismo, un valor agregado de este proyecto es proveer de una plataforma *web* al auditor energético que le permita tener accesibilidad a toda la información almacenada de los lugares que requieran ser analizados, facilitando el acceso a los datos que pueden ser obtenidos remotamente y de esta forma evitar incurrir en gastos de logística al movilizarse hacia las instalaciones de interés.

2.7. Marco conceptual

Las definiciones de conceptos que complementan el estudio de nuevas tecnologías son importantes de considerar, debido a que existen términos básicos en el campo del Internet de las Cosas que tienen que ser conocidos previamente con el propósito de asegurar un entendimiento claro de las arquitecturas más complejas en las cuales interactúan los elementos esenciales en la implementación de este tipo de proyectos en la industria.

2.7.1. Ordenador de placa reducida

Los ordenadores de placa reducida o *Small Board Computer* (SBC) son diseñados para conectarse a un soporte o placa base que generalmente están conformados por un pequeño módulo procesador con una CPU y capacidad de puertos de entrada y salida estándar.¹⁷

Estas placas integran todos o la mayor parte de los componentes de una computadora normal, siendo la principal característica sus reducidas dimensiones, además de que, por lo general, el precio es considerablemente bajo. La potencia se ve reducida en comparación con placas convencionales, sin embargo, esto no implica una deficiencia en el uso cotidiano ni en el desarrollo de aplicaciones que requieran de mayores recursos para su funcionamiento.¹⁸

Las principales alternativas en ordenadores de placa reducida más comunes que se pueden encontrar en el mercado son:

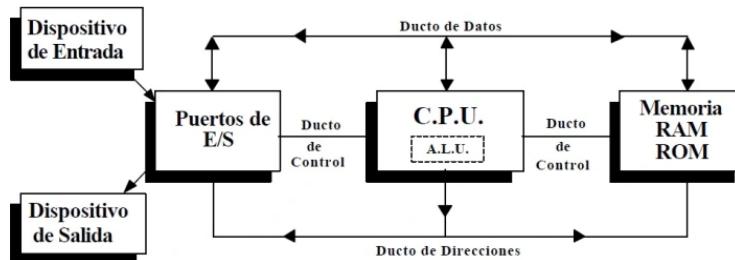
- Raspberry Pi
- BeagleBone Black
- PcDuino
- Pandaboard
- Jetson Nano
- Tinker Board S
- Orange Pi¹⁹

¹⁷ GARCÍA, Joaquin. *Introducción a las SBC*. Consultado el 23 de diciembre de 2021.

¹⁸ Ibid.

¹⁹ Ibid.

Figura 3. **Diagrama de bloques general SBC**



Fuente: Arquitectura PC. Arquitectura de Computadoras. Consultado el 23 de diciembre de 2021. Recuperado de <https://arquitecturadecomputadores252049.wordpress.com/2011/03/08/arquitectura-de-computadores/>.

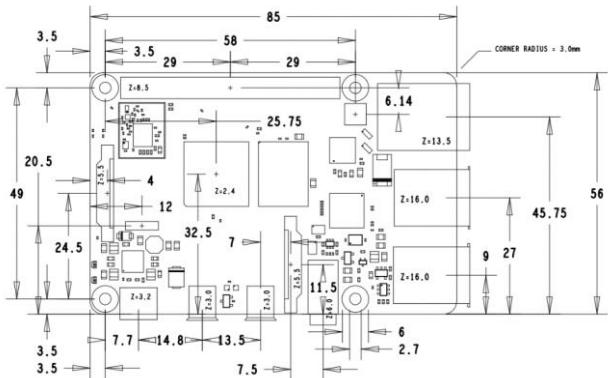
2.7.1.1. **Raspberry Pi**

Raspberry Pi es comúnmente llamado minicomputadora u ordenador de placa reducida, con relación a la descripción anterior. Entre sus características principales se encuentran que las placas son de bajo costo, bajo consumo y un tamaño reducido para adaptarse a cualquier tipo de proyecto electrónico. Generalmente se utiliza sobre el *hardware* un sistema operativo basado en el *kernel* de Linux, soportando muchas de las versiones *open software*.²⁰

Otras de las propiedades con las que disponen las placas Raspberry Pi es que integran periféricos que les permiten establecer comunicaciones con otros dispositivos por medio de protocolos como: GPIO, UART, SPI e I2C; además a través de estos se interconectan módulos convertidores que permiten la comunicación con dispositivos que utilizan distintos protocolos.

²⁰ Raspberry Pi. *Raspberry Pi Documentation*. Consultado el 23 de diciembre de 2021.

Figura 4. Esquemático de especificaciones mecánicas



Fuente: Raspberry Pi. Mechanical Drawings. Consultado el 23 de diciembre de 2021.
Recuperado de <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-mechanical-drawing.pdf>.

- Raspberry Pi 4 modelo B+

Es la versión más reciente de la *Raspberry Pi Foundation*, la cual puede encontrarse con distintas capacidades de memoria RAM 2GB, 4GB y 8GB, siendo directamente proporcional un mejor rendimiento con el costo de cada versión. El chip BCM 2711 de la empresa Broadcom fue diseñado para su uso en el miniordenador Raspberry Pi modelos 4B+ y 4A+, integrando un procesador de cuatro núcleos a 1,5 GHz con ARM Cortex-A72; esto permite que su rendimiento y capacidad de procesamiento en paralelo sea mucho mejor.

Incorpora conectores RJ45 con tecnología PoE (Power over Ethernet) y para la conectividad inalámbrica módulos WiFi 802.11 y Bluetooth 5.0, además de puertos USB 3.0 y 2.0, lector de tarjeta SD para almacenamiento no volátil, un puerto para cámara, salidas de audio estéreo, una GPU VideoCore VI de 32 bits y a diferencia de versiones anteriores, el modelo 4B+ soporta video en 2 puertos micro-HDMI que transmite en pantallas de 4K a 60Hz; el procesador tiene la

capacidad de ejecutar funciones complejas. Optando por diferentes configuraciones, los usuarios suelen utilizarla como servidor para control de bases de datos, asignación de direcciones lógicas para otros dispositivos, control de sitios *web*, entre otros.²¹

En las aplicaciones del Internet de las Cosas se utiliza como el puente entre equipos convencionales en la empresa y la conectividad con el internet a través del *Gateway*, a la vez puede funcionar como *Broker* de otras arquitecturas interconectadas con dispositivos que tengan la función de enviar datos o ser controlados por la misma Raspberry Pi.

Figura 5. **Raspberry Pi 4 modelo B+**



Fuente: Tecnobits. Qué hace cada pin GPIO de Raspberry Pi 4. Consultado el 23 de diciembre de 2021. Recuperado de <https://tecnobits.net/que-hace-cada-pin-gpio-de-raspberry-pi-4/>.

- **GPIO's en Raspberry Pi**

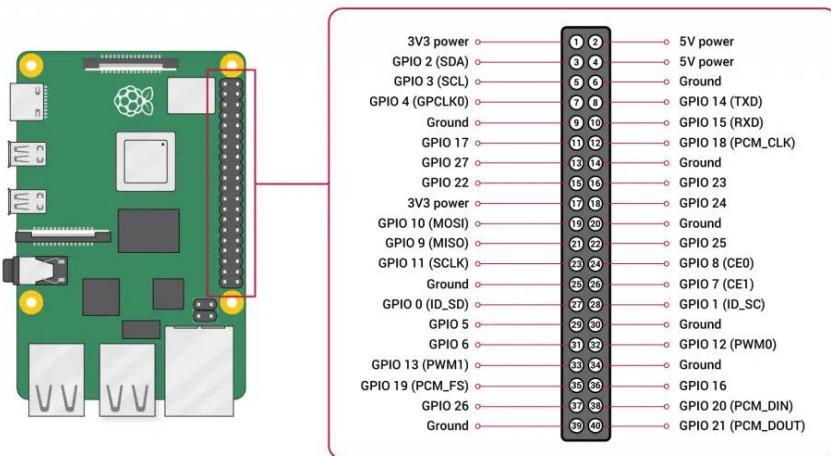
Las Entradas/Salidas de propósito general o *General Purpose Input/Output*, son un conjunto de pines físicos que pueden ser configurados para realizar

²¹ PC components. Características de la nueva Raspberry Pi 4 Mode B+. Consultado el 23 de diciembre de 2021.

distintas funciones. La Raspberry Pi integra 40 pines GPIO con posibilidad a ser conectados a sensores, luces, controladores y a una amplia gama de módulos electrónicos, de esta forma la Raspberry Pi no está limitada a una serie de puertos e interfaces para conectar dispositivos estándar.²²

Cada pin ofrece una conexión directa al *System on Chip* (SoC) en el centro de la Raspberry Pi, lo que le permite comunicarse con componentes externos. Los pines GPIO son digitales, lo que significa que pueden tener estados lógicos de 1 y 0, encendido o apagado respectivamente; pudiendo ser controlados con lenguajes de programación.²³

Figura 6. Pines GPIO de Raspberry Pi



Fuente: Hardware Libre. *GPIO: todo sobre las conexiones de la Raspberry Pi 4 y 3*. Consultado el 26 de diciembre de 2021. Recuperado de <https://www.hwlibre.com/gpio-raspberry-pi/>.

²² Hardwarelibre. *GPIO: todo sobre las conexiones en la Raspberry Pi 4 y 3*. Consultado el 23 de diciembre de 2021.

²³ Ibid.

La mayoría de los pines GPIO tienen asignada más de una función que pueden cumplir configurándose previamente en la programación. Los tipos de pines con los cuales podemos interactuar son:

- Alimentación: proveen voltajes de 5V y 3,3V con una corriente máxima de 50mA. Además, pines de tierra o *Ground*.
- Pines configurables: son pines de entrada o salida normales.
- Pines especiales: pueden ser GPIO y estar destinados para conexiones a interfaces y protocolos específicos. Entre ellos:
 - PWM
 - SPI
 - Serial (TX y RX)²⁴

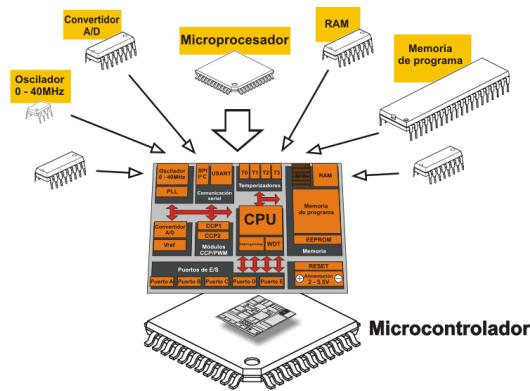
2.7.2. Microcontroladores

Un microcontrolador es un sistema integrado en un solo chip en el cual la memoria que ejecuta los programas no está en otro chip adicional, sino que, esta incorporada en el mismo, por lo cual un microcontrolador no requiere pines complementarios de direccionamiento ni datos que se comuniquen con una memoria RAM externa. Estos poseen una memoria *flash* del tipo reprogramable que puede ser accedida por medio de los pines de entrada y salida que están agregados alrededor del microcontrolador.

Algunas de las características esenciales son que la memoria utilizada es programable y no volátil, logrando que el tiempo de acceso a datos e instrucciones sea mucho más alto y por lo tanto la velocidad del reloj es baja, beneficiándose en un menor consumo de energía.

²⁴ Hardwarelibre. GPIO: todo sobre las conexiones en la Raspberry Pi 4 y 3. Consultado el 23 de diciembre de 2021.

Figura 7. Arquitectura básica de un microcontrolador



Fuente: Computo Integrado. Recursos de Microcontroladores y Familia de los PIC. Consultado el 27 de diciembre de 2021. Recuperado de <http://computointegradoits.blogspot.com/2012/03/recursos-de-microcontroladores-y.html>.

2.7.2.1. Tarjetas de desarrollo

Las diferentes versiones de microcontroladores ocupan distintas arquitecturas, algunas con más memoria y otras añaden nuevos recursos; cada una es propia de cada fabricante quienes han incorporado las funcionalidades de un microcontrolador en una tarjeta de desarrollo que presenta un dispositivo más práctico y robusto con mejoras en accesibilidad.

Una tarjeta de desarrollo es una placa o circuito que tiene integrado un microcontrolador principal que coordina la ejecución de un conjunto de instrucciones de un código programado. En complemento al microprocesador la tarjeta de desarrollo dispone de un diseño electrónico que permite: la programación del dispositivo, suministrar voltaje de funcionamiento y proporciona el acceso a los GPIO para la conexión con otros dispositivos.

Actualmente, en el mercado existe una gran variedad de tarjetas de desarrollo que nos permite optar por un modelo que se ajuste a nuestros requerimientos. Las más populares y utilizadas son:

- Arduino
- NodeMCU ESP8266 Y ESP32
- Bluepill STM32
- FPGA's

2.7.2.2. ESP32

Es considerada una de las innovaciones más importante que está orientada al desarrollo del IoT y como sucesora de la ESP8266, la ESP32 de la marca ESPRESSIF System es una de las primeras en permitir la conexión a una red WiFi y Bluetooth de modo dual, lo que impulsa la integración al mundo del IoT, además de su bajo costo, bajo consumo de energía y tamaño reducido.

Figura 8. **Módulo SoC ESP32**



Fuente: MarkerShop. ESP32-S by AiThinker Original. Consultado el 27 de diciembre de 2021.

Recuperado de <https://makershop.ie/ai-thinker-esp32-s>.

Una de las cualidades es que este SoC puede soldarse directamente en circuitos embebidos propios, conservando las mismas características y con funciones específicas que sean requeridas para la aplicación.

Internamente contiene al microprocesador Tensilica Xtensa LX6 de doble núcleo o de uno solo, con frecuencias de reloj hasta de 240MHz, la capacidad de la memoria ROM para funciones básicas y de arranque es de 448KB, una SRAM para datos e instrucciones de 520KB y hasta 16MB de memoria *flash*.

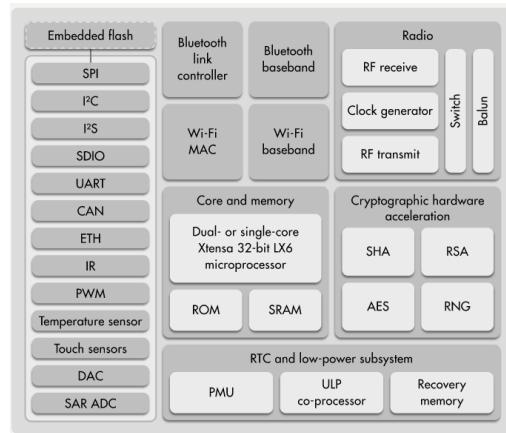
Figura 9. **Tarjeta de desarrollo ESP32**



Fuente: Talos Electronics. Programar ESP32 con IDE Arduino. Consultado el 27 de diciembre de 2021. Recuperado de <https://www.taloselectronics.com/blogs/tutoriales/programar-esp32-con-ide-arduino>.

La aplicación de la ESP32 en el Internet de las Cosas es muy diversa, comúnmente se utiliza para conectar sensores o actuadores que pueden enviar información o ser controlados a través del internet. Sin embargo, no está limitada a simples aplicaciones, debido a la amplia gama de funciones que esta posee, es posible gestionar arquitecturas aún más complejas, tales como operar en modo servidor local o conectarse directamente a servicios en la nube, así como el direccionamiento *IP* de cámaras.

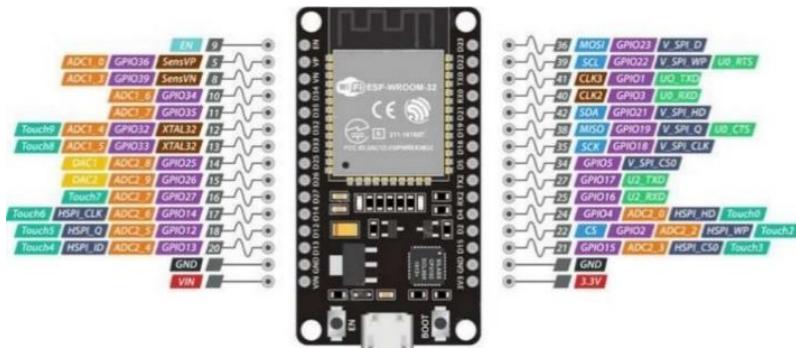
Figura 10. Diagrama de bloques de funciones ESP32



Fuente: The Internet of Things with ESP32. Features and Specifications Consultado el 27 de diciembre de 2021. Recuperado de <http://esp32.net/>.

La ESP32 dispone físicamente de 30 GPIO con conectividad a interfaces como: DMA, ADC, DAC, I²C, UART, CAN 2.0, SPI, I²S, RMII, PWM, entre otros.²⁵

Figura 11. Pines GPIO de ESP32



Fuente: ESP32, módulo ESP32-WROOM GPIO. DevKit Pinout. Consultado el 27 de diciembre de 2021. Recuperado de <https://descubreaduino.com/esp32-modulo-esp32-wroom-gpio-pinout/>.

²⁵ Esp32.net. *The Internet of Things with ESP32*. Consultado el 27 de diciembre de 2021.

Para la programación de códigos que pueden ser cargados y ejecutados en una tarjeta de desarrollo es necesario utilizar *software* proveído por cada fabricante o en algunos casos es posible lograr compatibilidad empleando un conjunto de librerías. Generalmente se utiliza una de las siguientes alternativas.

- Lenguajes de programación

Un lenguaje de programación se caracteriza por ser de uso general, de sintaxis compacta y notable; convirtiéndose en una herramienta informática con la cual se pueden desarrollar programas orientados a dispositivos electrónicos. Esto es posible por medio de compiladores e intérpretes que traducen o leen las líneas de código escritas obteniendo un equivalente en lenguaje máquina, con el fin de obtener un programa ejecutable. Entre ellos:

- Python o MicroPython
- Lenguaje C y C++
- Ensamblador²⁶

- Entornos de desarrollo integrado

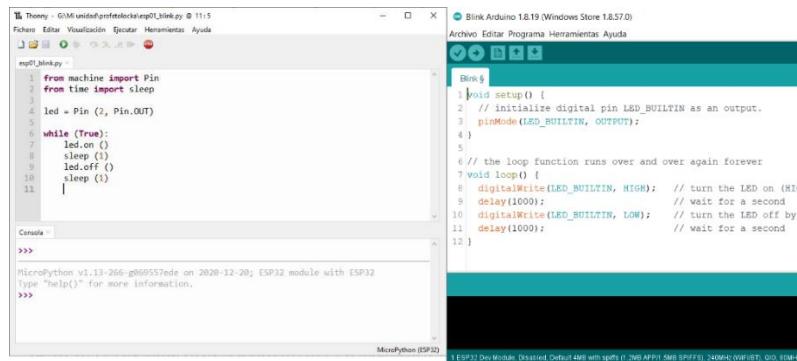
Es un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los fabricantes suelen complementar a las tarjetas de desarrollo con IDE y sintaxis propias basadas en algún lenguaje de programación, a pesar de ello algunos programadores prefieren utilizar otros lenguajes por simplicidad, costumbre o porque estos permiten utilizar algunas funciones más internas de la tarjeta que otros programas no contienen. Regularmente, cualquier IDE con las librerías correctas es compatible con distintas tarjetas de desarrollo. Entre ellos:

²⁶ CUAED. *Lenguajes de Programación*. Consultado el 27 de diciembre de 2021.

- Arduino IDE
- Energia
- PlatformIO
- Nanoframework

Tanto los lenguajes de programación como los IDE son opciones recomendables para iniciar la programación de una tarjeta destinada a un proyecto. La elección final deberá tomarla el programador basándose en su experiencia, el tiempo con el que disponga, la tarjeta a utilizar, entre otros criterios que deberá considerar para obtener los resultados propuestos.

Figura 12. Lenguaje de programación y entorno de desarrollo integrado



Fuente: elaboración propia, utilizando MicroPython a la izquierda y Arduino a la derecha.

2.7.3. Protocolos de comunicación

Para establecer la comunicación entre dos o más dispositivos que conforman una red se recurre a utilizar sistemas de reglas que permiten la transmisión de información entre ellos. Siendo los protocolos de comunicación el idioma que hablan los dispositivos entre sí, por consiguiente, las reglas o el

estándar que define la sintaxis, semántica y sincronización deben de ser las mismas para la correcta transferencia de información. Los protocolos de comunicación se han extendido a todo tipo de sistemas adaptando los estándares requeridos para cada aplicación, por este motivo la utilización de diferentes protocolos es necesaria para comunicar dispositivos en una arquitectura.

2.7.3.1. Protocolos industriales de comunicación

La constante evolución de la tecnología y su inmersión en el campo industrial contribuye en la optimización de los sistemas y procesos, logrando establecer una comunicación con las máquinas obteniendo datos y el control por medio de interfaces. En los últimos años, la automatización de procesos se ha incrementado, permitiendo la interconexión de sensores, actuadores y equipos de control; es aquí donde se hace obligatorio utilizar un protocolo para la comunicación de estos dispositivos.

Debido a la hostilidad en el ámbito industrial, los factores como la interferencia por ruido, electromagnetismo, entre otras causas que dificultan la transmisión correcta de información propician a la pérdida o mala interpretación de datos; los protocolos a utilizar deben ser robustos para soportar cualquier tipo de alteración en las señales y poder diferenciar cuando se generan errores, de esta forma algunos son modificados para ajustarlos a nuevos requerimientos que pueden ser creados a partir de propósitos específicos, o bien, desarrollados por los fabricantes para usos exclusivos en sus dispositivos. Entre ellos están:

- MODBUS: simplifica la transmisión de información entre distintos equipos electrónicos conectados a un mismo bus, en el cual existe un dispositivo maestro que realiza peticiones hacia varios dispositivos esclavos identificados numéricamente que responden con los datos solicitados.

- HART: su funcionamiento se basa en la corriente convencional 4-20mA y en la generación de una onda senoidal modulada en FSK. En otra configuración toda la información es transmitida digitalmente.
- DeviceNet: compuesto por un bus lineal, transporta en el mismo cable de red la alimentación de los dispositivos y los datos con una trama máxima de 8 octetos y con un límite de 64 nodos permitidos.
- AS-Interface: desarrollado por la firma Siemens, establece la comunicación entre un dispositivo de control con los sensores y actuadores que se encuentran en el nivel más bajo de automatización.
- PROFINET: permite la interconexión de dispositivos desde el nivel de campo hasta el nivel de gestión con sistemas informáticos e internet, logrando una cobertura en toda la planta industrial a través de enlaces.
- PROFIBUS: es un protocolo de campo abierto con capacidad de implementarlo en distintas áreas como en la fabricación, proceso y automatización de edificios, por lo que suele ser muy confiable. En la actualidad se ha dividido en tres versiones:
 - Profibus-PA (*Process Automation*)
 - Profibus-FMS (*Field Message Specification*)
 - Profibus-DP (*Periferia Descentralizada*)
- CAN (Controller Area Network): soporta fácilmente un control distribuido en tiempo real con un alto nivel de seguridad, con una capacidad de transmitir información hasta de 1 Mbps, por ese motivo su principal aplicación está en el sector automotriz.

- **ETHERNET INDUSTRIAL:** cumple con los estándares y potencial que ofrece Ethernet, pero incorpora medidas de seguridad con un alto nivel de confiabilidad en el funcionamiento de condiciones ambientales difíciles, interrupciones de la red y fallas de equipos.²⁷

Tabla I. Características técnicas de los protocolos

BUS DE CAMPO	TOPOLOGÍA	MEDIO FÍSICO	VELOCIDAD	DISTANCIA SEGMENTO	NODOS POR SEGMENTO	ACCESO AL MEDIO
P-NET	Anillo	Par trenzado apantallado	76'8 Kbps	1.200 m	125	Paso de testigo Maestro/esclavo
PROFIBUS	Bus lineal Anillo Estrella Árbol	Par trenzado apantallado Fibra óptica	Hasta 12Mbps	Hasta 9'6 Km y 90 Km	125	Paso de testigo Maestro/esclavo
WORLDFIP	Bus lineal	Par trenzado apantallado Fibra óptica	Hasta 1 Mbps y 5Mbps	Hasta 5 Km y 20 Km	64	Arbitro de bus
HART	Bus lineal	Cable 2 hilos	1'2Kbps	3.000 m	30	Maestro/esclavo
MODBUS	Bus lineal	Par trenzado	Hasta 19'2Kbps	1 Km	248	Maestro/esclavo
INTERBUS-S	Anillo	Par trenzado	500 Kbps	400 m	256	Paso de testigo
BITBUS	Bus lineal	Par trenzado Fibra óptica	Hasta 1'5Mbps	Hasta 1.200m	29	Maestro/esclavo
CAN	Bus lineal	Par trenzado	Hasta 1 Mbps	Hasta 1.000m	127-64	CSMA/CD con arbitraje de bit
SDS	Bus lineal	Cable de 4 hilos	Hasta 1 Mbps	500 m	64	CSMA
DEVICENET	Bus lineal	Par trenzado	Hasta 500 Kbps	Hasta 500 m	64	CSMA/CDBA
CONTROLNET	Bus lineal Árbol Estrella	Coaxial Fibra óptica	5 Mbps	Hasta 3.000m	48	CTDMA
SERIPLEX	Bus lineal	Cable 4 hilos apantallado	98 Kbps	1.500m	300	Maestro/esclavo
AS-i	Bus lineal Árbol - Estrella	Cable 2 hilos	167 Kbps	Hasta 200 m	32-62	Maestro/esclavo
LON WORKS	Bus Anillo Libre	Par trenzado Fibra óptica Red eléctrica Coaxial Radio Infrarrojos	Hasta 1'25 Mbps	Hasta 2.700 m	64	CSMA/CA
ARCNET	Bus Estrella	Par trenzado Fibra óptica Coaxial	2'5 Mbps	122 m	255	Paso de testigo
M-BUS	Bus lineal	Cable 2 hilos	Hasta 9'6 Kbps	1.000 m	250	Arbitro de bus

Fuente: HURTADO, José. Introducción a las Redes de Comunicación Industrial. Consultado el 28 de diciembre de 2021. Recuperado de

http://www.infopl.net/files/documentacion/comunicaciones/infoPLC_net_introduccic3b3n-a-las-redes-de-comunicac3b3n-industrial.pdf.

²⁷ Logicbus. *Protocolos de comunicación industriales*. Consultado el 28 de diciembre de 2021.

La mayoría de estos protocolos incorporan funcionalidades de protocolos de red con el fin de optimizar los procesos y la conectividad, tal es el caso de TCP/IP y algunas interfaces como RS232 y RS485, con lo cual aprovechan las ventajas operativas y de seguridad, llegando a ofrecer mejoras en la transferencia de información de los propios sistemas de control que pueden ser implementados en arquitecturas que utilizan determinados protocolos.

2.7.3.2. Protocolos de red

La interconexión de sistemas o redes de dispositivos con el internet utiliza múltiples protocolos para poder navegar en sitios, enviar correos electrónicos, consultar servidores, establecer conexiones remotas, entre otras aplicaciones.

Existen una gran cantidad de protocolos que interactúan en la red, no obstante, es oportuno conocer aquellos que resultan de utilidad en el desarrollo de proyectos relacionados al IoT.

- Ethernet: es un estándar orientado a la transmisión de datos para redes de área local, se encuentra en la capa 1 y 2 del modelo OSI convirtiéndolo en la base de la comunicación entre ordenadores.
- Internet Protocol (IP): enfocado al funcionamiento de los dispositivos conectados a una red. Gestiona el direccionamiento y enrutamiento, asigna direcciones IP a cada dispositivo y determina las vías para transferir los paquetes de información.
- Transmission Control Protocol (TCP): verifica que la transmisión de información sea correcta asegurándose de que los paquetes enviados lleguen a su destino, sin embargo, la velocidad de transferencia es baja.

- User Datagram Protocol (UDP): no es tan confiable como TCP debido a que no realiza revisiones para verificar si la transmisión de datos fue correcta, no obstante, es mucho más rápido y útil para aplicaciones que requieran altas velocidades de transferencia de información.
- Hypertext Transfer Protocol (HTTP): permite la comunicación entre navegadores y servidores *web*. Es utilizado en las peticiones que realizan los usuarios desde una página *web* hacia servicios remotos, por lo cual establece un modelo de comunicación del tipo cliente-servidor.
- File Transfer Protocol (FTP): empleado para compartir archivos entre ordenadores. Implementa el modelo cliente-servidor siendo necesario que tanto el cliente como el servidor compartan el mismo protocolo FTP.
- Simple Mail Transfer Protocol (SMTP): considerado como uno de los protocolos más importantes en internet. Utilizado para enviar y recibir correos electrónicos empleando una solicitud TCP con el servidor SMTP que inicia una conexión con el puerto asignado, verifica que la conexión TCP establecida sea correcta, entonces procede a enviar el correo.²⁸

2.7.3.3. Protocolos del Internet de las Cosas

La implementación del IoT continúa expandiéndose a diferentes campos, así como el aumento de proveedores de servicios, múltiples plataformas y millones de dispositivos nuevos que surgen año con año, lo que implica a los desarrolladores considerar distintos criterios antes de tomar una decisión previa respecto a qué protocolos y dispositivos utilizar.

²⁸ Redes Zone. *Protocolos básicos en redes*. Consultado el 28 de diciembre de 2021.

En este caso se mostrarán los protocolos IoT más comunes, con el objetivo de comprender las distintas alternativas que existen:

- Advanced Message Queuing Protocol (AMQP): es capaz de gestionar interoperabilidad en el *middleware* de mensajería y asiste a las aplicaciones y sistemas para que operen en conjunto, normalizando una mensajería a escala industrial.
- Constrained Application Protocol (CoAP): recomendado para redes con anchos de banda limitados. Está diseñado para dispositivos que poseen conectividad restringida, permitiéndoles establecer comunicación entre máquinas y a la vez formar parte de la red.
- Data Distribution Service (DDS): basado en la comunicación punto a punto con diversas aplicaciones, puede ser ejecutado desde pequeños dispositivos llegando a conectar redes de alto rendimiento.
- Message Queue Telemetry Transport (MQTT): enfocado en la comunicación ligera entre dispositivos, utilizado generalmente para conexiones de poco ancho de banda. Emplea un patrón de publicador-suscriptor en dispositivos pequeños logrando optimizar el ancho de banda y consumo de energía en la batería.²⁹

En complemento a estos protocolos se utilizan tecnologías adicionales que optimicen la funcionalidad entre los equipos de una topología.

²⁹ Azure. *Protocolos y tecnologías de IoT*. Consultado el 28 de diciembre de 2021.

- Redes de corto alcance y bajo consumo:
 - Bluetooth
 - NFC
 - RFID
 - Wi-Fi/802.11
 - Zigbee
- Redes de área extensa de bajo consumo:
 - 4G LTE para IoT
 - 5G para IoT
 - LoRaWAN
 - Sigfox³⁰

Debido a la gran diversidad de dispositivos IoT disponibles en el mercado es importante utilizar el protocolo adecuado en el contexto solicitado, esto dependerá en función de innumerables factores, tales como: el uso, distancia de conexión, duración de baterías, frecuencia de envío de datos, el ambiente y del nivel de arquitectura del sistema; por lo cual más de un protocolo puede estar presente en los dispositivos, puertas de enlace y centros de almacenamiento de datos en la nube.

2.7.4. Módulos de comunicación

Engloba una serie de dispositivos que tienen la capacidad de comunicar dos equipos, aunque el protocolo sea distinto es el propio módulo quien realiza la conversión de información decodificando en la entrada los datos provenientes de un dispositivo y en su salida entrega la misma información, pero en un formato diferente que sea entendible por el dispositivo receptor.

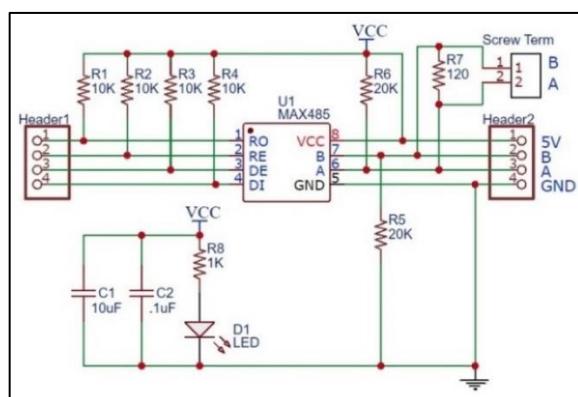
³⁰ Azure. *Protocolos y tecnologías de IoT*. Consultado el 28 de diciembre de 2021.

Por lo general, las empresas ya disponen de equipos que fueron instalados anteriormente, y para extraer la información generada se recomienda utilizar un módulo de comunicación que cumpla con las características de ambos equipos, tanto del emisor como del receptor. Siendo el protocolo MODBUS con la interfaz RS485 de estándar abierto e implementado por la mayoría de los fabricantes en sus productos, a continuación, se describen los módulos utilizados.

2.7.4.1. Módulo MAX485

Es un transceptor de baja potencia que puede comunicar interfaces RS485 y RS422 con tecnologías TTL, estableciendo una comunicación bidireccional con capacidad de escribir o recibir datos desde el dispositivo maestro para controlar a los dispositivos esclavos. Los datos inician en una serie de nivel TTL por parte del microcontrolador, mientras que el módulo RS485 se encarga de convertir las señales eléctricas entre TTL y la señalización diferencial.³¹

Figura 13. Esquemático del módulo MAX485



Fuente: MAX485 TTL to RS-485 Interface Module. Consultado el 29 de diciembre de 2021. Recuperado de <https://protosupplies.com/product/max485-ttl-to-rs-485-interface-module/>.

³¹ Thido. Módulo MAX485 Convertidor TTL a RS485. Consultado el 29 de diciembre de 2021.

Una de las características más importantes del módulo es que puede admitir hasta 32 dispositivos conectados en el mismo cable, utilizando un bus que pasa a través de cada dispositivo captando las señales generadas por cada uno. El voltaje de operación del módulo es de 5V, corriente de 500uA y con una velocidad máxima de 2,5Mbps.

Figura 14. Pinout módulo MAX485



Fuente: Módulo MAX485 Convertidor TTL a RS485. Consultado el 29 de diciembre de 2021. Recuperado de <http://www.electronicathido.com/detallesProducto.php?id=U2c0Rjl3Nk93clBWWjhQSURoSkJqZz09>.

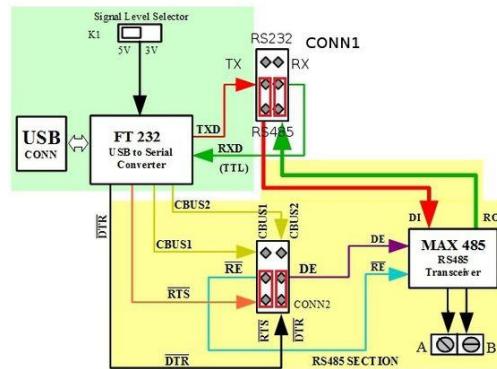
2.7.4.2. Conversor USB a RS485

Este módulo incorpora un conversor USB a serial TTL (CH340) y un chip serial a RS485, permite establecer la comunicación entre una computadora a través del puerto USB con cualquier dispositivo que integre la interfaz RS485. Además, cuenta con todas las características del módulo MAX485 y facilita el control que puede ser realizado desde un software en computadora sin requerir ninguna programación extra del usuario, solamente configurar los parámetros.

Para comunicarse entre dispositivos emplea un par de cables trenzados que admiten hasta 32 conexiones en un solo bus, operando con velocidades en el rango de 300 y 19200bit/s en tipos de comunicación *half-duplex* o *semiduplex*.³²

³² Naylamp Mechatronics. CONVERSOR USB A RS485. Consultado el 29 de diciembre de 2021.

Figura 15. **Diagrama conversor USB a RS485**



Fuente: Pinterest. Cross Platform RS485 Communication using USB2SERIAL (USB to RS485 Converter) in C. Consultado el 29 de diciembre de 2021. Recuperado de <https://www.pinterest.com/pin/389913280218791034/>.

No es necesario utilizar una fuente externa de alimentación, ya que toma el voltaje directamente del puerto USB 2.0 que es compatible con una gran variedad de sistemas operativos Windows y distribuciones Linux.

Figura 16. **Conversor USB a RS485**



Fuente: Altronics. Adaptador Conversor USB a RS-485. Consultado el 29 de diciembre de 2021. Recuperado de <https://altronics.cl/usb-to-rs485>.

2.7.5. Sensores

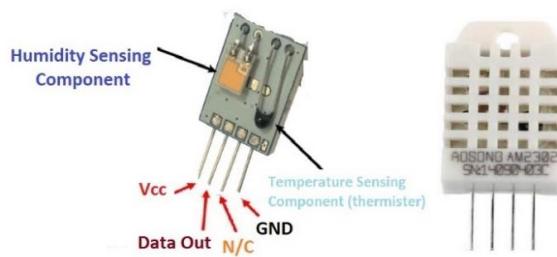
Un sensor es un dispositivo que detecta y permite medir magnitudes físicas externas, respondiendo a cada una de las variaciones y transformándolas en señales eléctricas, ópticas, digitales o de algún otro tipo. También son conocidos como transductores que pueden obtener en su salida magnitudes cuantificables en valores numéricos, los cuales posteriormente son analizados para tomar decisiones respecto al proceso censando.

En la actualidad la mayoría de los sensores están integrados en circuitos electrónicos que incorporan algún protocolo que facilita la conectividad para obtener los datos desde un microcontrolador, convirtiéndose en módulos de bajo coste y fácil implementación.

2.7.5.1. Sensor de DHT22

Es un sensor digital con capacidad de medir humedad relativa y temperatura al mismo tiempo, incorpora un sensor capacitivo de humedad y un termistor para medir el aire en el ambiente.

Figura 17. Sensor DHT22



Fuente: Introduction to DHT22. PINOUT & Description Consultado el 29 de diciembre de 2021.

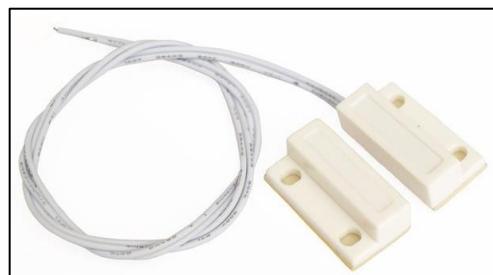
Recuperado de <https://www.theengineeringprojects.com/2019/02/introduction-to-dht22.html>.

Tiene como salida un pin digital para enviar datos, dispone de dos pines más uno de alimentación en el rango de 3V a 5V y otro para conexión a tierra. La única limitante es la frecuencia de censado, la cual se realiza cada dos segundos, por eso se recomienda utilizarlo en el monitoreo de ambientes controlados que no requieran acciones inmediatas.

2.7.5.2. Sensor magnético

Este sensor basa su funcionamiento en la presencia de un campo magnético que regularmente es generado por un imán con un rango de alcance de varios milímetros en función de la intensidad del campo. Entrega a la salida una señal digital que cambia a '1' ante la presencia de un campo y '0' en ausencia de dicho campo. En sus aplicaciones convencionales se utiliza para medir velocidades o detector de apertura de puertas y ventanas.

Figura 18. **Sensor magnético de apertura**



Fuente: Electrónica DIY. Sensor magnético de apertura para puerta o ventana. Consultado el 29 de diciembre de 2021. Recuperado de https://www.electronicadiy.com/products/sensor-magnetico-de-apertura-para-puerta-o-ventana?_pos=1&_sid=a075eeda3&_ss=r.

2.7.5.3. Sensores de temperatura

Existe una gran variedad de sensores de temperatura que pueden seleccionarse dependiendo del proceso que se requiera medir. Algunos de los utilizados en la industria integrables en sistemas IoT son:

- Pirómetros

Son instrumentos que miden temperaturas de cuerpos a distancia por medio de efectos ópticos y de radiación que operan en rangos de valores infrarrojos.

Figura 19. **Sensor GY-906**



Fuente: Electrónica DIY. Sensor de Temperatura Infrarrojo GY-906. Consultado el 29 de diciembre de 2021. Recuperado de https://www.electronicadiy.com/products/sensor-de-temperatura-infrarrojo-gy-906?_pos=6&_sid=caefebcf3&_ss=r.

- Termopares

También conocido como termocupla, basando su funcionamiento en el efecto termoeléctrico o efecto Seebeck. Consiste en una sonda formada por la unión de dos metales de diferente composición, en la cual los extremos presentan una junta caliente y una fría generando un voltaje del orden de los milivoltios que varía en función de la temperatura.

Para entender mejor estas variaciones de voltaje en función de la temperatura existen tablas para consultar los parámetros identificados por cada fabricante. A la vez, existen diferentes tipos de termopares que pueden funcionar a diferentes rangos de temperatura.

Tabla II. Tipos de termopares y sus aplicaciones

Termopar	Rango de temperatura [°C]	Metales	Aplicación
B	0 – 1700	+ Platino – 30% Rodio - Platino – 6% Rodio	Medición en altas temperaturas.
E	-200 – 900	+ Níquel – Cromo - Cromo – Níquel	Ideales para ambientes criogénicos.
J	0 – 750	+ Hierro - Cobre – Níquel	Recomendada para temperaturas promedio.
K	-200 – 1250	+ Níquel – Cromo - Níquel – Aluminio	Puede ser aplicada en diferentes procesos.
N	-200 – 1250	+ Níquel–Cromo–Silicio - Níquel – Silicio	Estabilidad y resistencia a la oxidación.
R	0 – 1450	+ Platino – Rodio - Platino	Presenta baja estabilidad y precios elevados.
S	0 – 1450	+ Platino – Rodio - Platino	Robustos y de buena exactitud.
T	-200 – 350	+ Cobre - Cobre – Níquel	Estable en temperaturas realmente bajas.

Fuente: elaboración propia.

Los termopares tipo J y K son los sensores más utilizados a nivel industrial debido a que son muy resistentes y con una buena precisión en múltiples intervalos de temperatura. Para la conexión están identificados dos cables con diferentes colores que pueden cambiar según el país o fabricante, estos son cables de extensión de grado *premium* que reducen hasta un 50 % el error de un cable normal producido por la resistencia de este.

Figura 20. **Distintas presentaciones de los termopares**



Fuente: Termocuplas. Consultado el 29 de diciembre de 2021. Recuperado de <http://loinsaperu.com/termocuplas-termometros-control-temperatura-sensores/>.

- **Sondas RTD**

El detector resistivo de temperatura o RTD es una resistencia eléctrica de un conductor que cambia de valor en función a la temperatura externa, también es conocida como PT100 debido a que su construcción es en platino 100. La conectividad puede ser a dos, tres o cuatro hilos; siendo la más utilizada a tres hilos para eliminar el error de la resistencia generada en el cable conductor.

Figura 21. **RTD utilizadas en la industria**



Fuente: Electrotec. Resistance Temperature Detectors. Consultado el 29 de diciembre de 2021. Recuperado de <https://electrotec.pe/blog/RTDvsTERMOCUPLA>.

Existen tablas que relacionan la salida de resistencia de la RTD con la temperatura. Normalmente están en el rango de 0 °C a 400 °C, correspondiendo para un valor de resistencia de 100 Ohmios una temperatura de 0 °C; tomando estos valores como punto de referencia cuando se requiere calibrar la RTD.

Tabla III. Temperatura y resistencia en una PT100

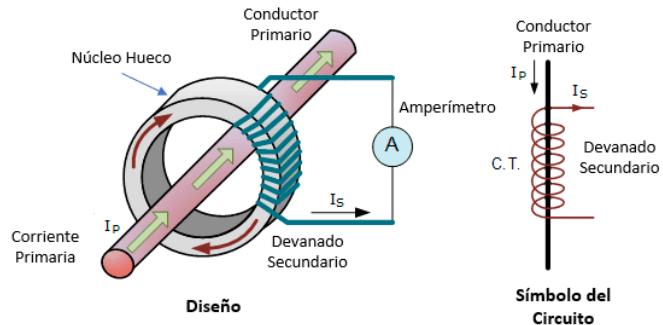
Temperature [Celsius]	PT100 resistance [Ohms]
0	100
10	103.9
20	107.79
30	111.67
40	115.54
50	119.4
60	123.24
70	127.08
80	130.9
90	134.71
100	138.51
110	142.29
120	146.07
130	149.83
140	153.58
150	157.33
160	161.05
170	164.77
180	168.48
190	172.17
200	175.86

Fuente: Dispositivos Termo Resistivos. Rtd pt100. Consultado el 29 de diciembre de 2021.
Recuperado de <https://es.slideshare.net/reymartinez1/rtd-pt100-14884304>.

2.7.5.4. Sensor de corriente

Son dispositivos empleados para medir el flujo de electrones en un conductor eléctrico que forma parte de un circuito. En el sector industrial se denominan transductores de corriente o transformadores de corriente (TC), debido a que su funcionamiento está basado en la inducción electromagnética y otros en el principio del efecto Hall, el cual consiste en detectar el flujo magnético causado por la corriente que circula en un conductor y este a la vez induce un flujo de corriente en la bobina de detección.

Figura 22. Principio de la detección de corriente en conductores



Fuente: Transformador de Corriente. Consultado el 30 de diciembre de 2021. Recuperado de <https://ingtelecto.com/transformador-de-corriente/>.

El CT toroidal suele ser el más utilizado en la industria para medir la alta corriente alterna que consumen las cargas eléctricas, puesto que es no invasivo y consta de una bobina de núcleo abierto por el cual se introduce la línea de corriente que se requiere medir. Así mismo existen CT de núcleo cerrado y de núcleo dividido, estos últimos permiten abrirlos, instalarlos y cerrarlos; evitando el corte de suministro eléctrico al circuito.

Figura 23. Transformador de corriente toroidal



Fuente: Donner. Transformador de corriente. Consultado el 30 de diciembre de 2021. Recuperado de <https://www.donner.mx/transformadores-corriente-nucleo-abierto/>.

Los CT tienen una clasificación según sus características para medir una corriente máxima, identificando en el devanado secundario un estándar de 5A y se expresa una relación de 100/5. Esto significa que la corriente primaria es 20 veces mayor que la corriente secundaria, por lo tanto, cuando circulen 100A en el conductor, se inducirá 5A en la bobina del sensor.

Para aplicaciones que requieran conectividad con microcontroladores existen alternativas de sensores y módulos de bajo coste que pueden ser de utilidad para medir la corriente eléctrica.

- SCT-013: es capaz de medir corriente alterna en un valor máximo de hasta 100A, es de núcleo dividido por lo cual no requiere esfuerzo de instalación. La salida entrega un rango de 0 a 50mA, esto hace necesario tener que utilizar el módulo convertidor de análogo a digital ADS1115 que integra un amplificador diferencial adecuado para señales bajas.

Figura 24. **Sensor transformador de corriente SCT013**



Fuente: Electrónica DIY. Sensor de corriente no invasivo SCT-013-000. Consultado el 30 de diciembre de 2021. Recuperado de https://www.electronicadiy.com/products/sensor-de-corriente-no-invasivo-sct-013-000?_pos=2&_sid=c7edc56f3&_ss=r.

- Módulo ACS712: utiliza el principio del efecto Hall para medir corriente alterna en rangos que van desde 5A, 20A hasta 30A. Sin embargo, para realizar las conexiones es necesario abrir el circuito de la carga y conectar en serie a una línea de la red las borneras del sensor.

Figura 25. **Sensor de corriente ACS712**



Fuente: Hardware Libre. ACS712: el módulo sensor de corriente. Consultado el 30 de diciembre de 2021. Recuperado de <https://www.hwlible.com/acs712/>.

2.7.5.5. **Analizadores de energía**

El analizador de energía eléctrica no es un sensor como tal, la diferencia principal radica en que un analizador es un dispositivo multifuncional que dispone de sensores internos para realizar mediciones y mostrar datos en pantalla de forma numérica. Adicionalmente cuenta con entradas aisladas de conectividad para otros sensores y salidas para protocolos de comunicación.

Las variables eléctricas fundamentales por medir son la tensión y la corriente, ya que a partir de estos datos es posible realizar internamente cálculos de otros parámetros, tales como: la potencia activa, aparente y reactiva, además del factor de potencia, frecuencia, energía, emisiones de CO₂, entre otros indicadores de importancia en el análisis de las redes eléctricas.

Es así como los analizadores son dispositivos esenciales para identificar posibles fallas en la red y por su versatilidad de instalación consigue medir el consumo de energía por línea en tableros principales o cargas específicas.

Figura 26. **Analizador de redes CIRCUTOR CVM-C10**



Fuente: Circutor. Programación del analizador de redes eléctricas CVM-C10. Consultado el 30 de diciembre de 2021. Recuperado de <http://circutor.es/es/formacion/formacion-practica/2538-programacion-del-analizador-de-redes-electricas-cvm-c10>.

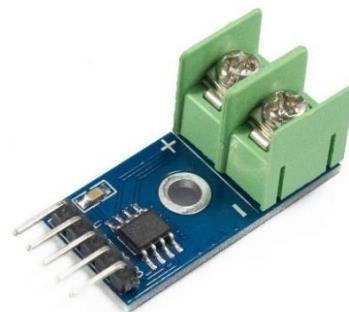
2.7.5.6. Circuitos complementarios

En algunos casos es necesario incorporar circuitos que mejoren el funcionamiento o conectividad de los sensores, con el objetivo de conseguir mejor precisión y exactitud en los datos de los procesos monitoreados. Considerando que algunos sensores no pueden ser conectados directamente a las tarjetas de desarrollo, es aconsejable utilizar módulos que permitan interconectar al sensor con la tarjeta.

- **Conversor de analógico a digital (ADC):** es un circuito que realiza la operación para expresar una señal física variante en un valor numérico amplificado que es entendible por los microcontroladores.

- MAX6675: utilizado para amplificar las señales de termocuplas y de RTD, facilita la conexión de la sonda con cualquier microcontrolador utilizando la interfaz SPI unidireccional.

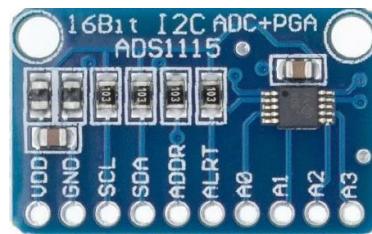
Figura 27. **Módulo MAX6675 para termocupla**



Fuente: TRANSMISOR PARA TERMOCUPLA K MAX6675. Consultado el 30 de diciembre de 2021. Recuperado de <https://naylampmechatronics.com/sensores-temperatura-y-humedad/331-transmisor-para-termocupla-k-max6675.html>.

- ADS1115: convertidor de señales analógicas a digitales, cuenta con una resolución de 16 bits con capacidad de leer cuatro señales individuales. En la salida dispone un bus I2C con dirección 0x48.

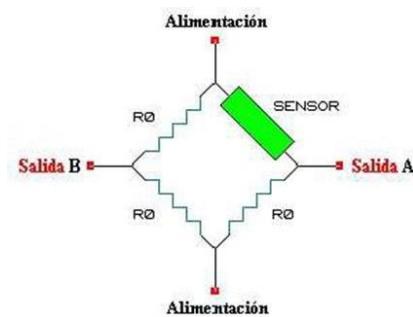
Figura 28. **Módulo ADC ADS1115**



Fuente: Módulo ADC ADS1115. Consultado el 30 de diciembre de 2021. Recuperado de https://www.electronicadiy.com/products/modulo-adc-ads1115?_pos=1&_sid=150f31b9f&_ss=r.

- Puente de Wheatstone: utilizado para encontrar valores de resistencias que normalmente suelen provenir de la variación de un sensor. Implementando este circuito se obtienen mejores resultados en la medición de señales analógicas, por lo general es complementado con un circuito de amplificador operacional para incrementar la señal de salida.

Figura 29. **Terminales de un Puente de Wheatstone**



Fuente: Circuitos Operacionales 19 – Puente de Wheatstone. Consultado el 30 de diciembre de 2021. Recuperado de <https://www.e-auto.com.mx/enew/index.php/85-boletines-tecnicos/6619-circuitos-operacionales-19-puente-de-wheatstone>.

- Transmisor de corriente estándar

Utilizado en ambientes industriales, el transmisor de corriente estándar es la mejor opción para interconectar sensores que se encuentran a grandes distancias del sistema principal. Integra electrónica de precisión para la lectura de sensores, comúnmente RTD, realizando un ajuste automático en la resistencia del cable del sensor conectado al tercer terminal.

Proporciona a la salida una corriente en el rango de 4 a 20 mA en función a la señal convertida proveniente del sensor. Siendo completamente una señal de corriente puede ser transmitida por cables a largas distancias.

Figura 30. **Transmisor de 4 – 20mA**



Fuente: Elkon. Transmisor de Temperatura RTD PT100 4-20mA 2 hilos 0 a 200°C 24V.

Consultado el 30 de diciembre de 2021. Recuperado de <https://www.elkon-bolivia.com/?product=transmisor-de-temperatura-rtd-pt100-4-20ma-2-hilos-0-a-200-c-24v>.

2.7.6. *Software utilizado*

En el desarrollo de proyectos IoT emplear distintas tecnologías permiten obtener plataformas para la visualización de datos, gráficos y paneles, además de controles administrativos, gestionar usuarios y almacenar la información recolectada; todos estos conceptos son integrados en una aplicación que anexa distintos sistemas y centraliza las funcionalidades en un solo sitio.

2.7.6.1. *Sistema operativo Raspbian*

Es una distribución libre de sistema operativo basada en Debian que esta optimizada para ser usada como el software base en la Raspberry Pi, integra muchas funcionalidades de GNU/Linux además de una interfaz gráfica avanzada que es amigable con el usuario permitiéndole modificar el sistema operativo sin necesidad ingresar directamente a los archivos de configuración desde el menú *raspi-config*. Incorpora gran cantidad de herramientas como la clásica terminal para ejecutar comandos Linux e incluye recursos de desarrollo IDLE para lenguajes de programación como Python o Scratch.

Así mismo ofrece compatibilidad con una variedad de dispositivos externos que se deseen conectar en los periféricos de la Raspberry Pi, al igual con los programas que se requieran descargar e instalar desde repositorios.³³

2.7.6.2. Node-RED

Es un *software* de programación orientado a la conexión de bloques configurables, con el objetivo de interconectar dispositivos de *hardware*, API y servicios en la *web* de formas muy simplificadas. El editor se ejecuta desde el navegador para facilitar la interacción y creación de flujos compuestos por la amplia gama de nodos seleccionables en la paleta de opciones.

Adicionalmente al espacio para crear los flujos de nodos, Node-RED incluye un tablero para visualizar de forma gráfica los valores de los datos que se están recibiendo en tiempo real, en el mismo tablero permite agregar botones para el control manual de algunos procesos. Por estas razones Node-RED es considerado como una herramienta para el desarrollo de plataformas IoT que integra programación, controles y visualizaciones en una sola aplicación.

2.7.6.3. Grafana

Es una aplicación multiplataforma de código abierto montada en la *web* utilizada para la visualización interactiva y analítica de datos que provee múltiples funcionalidades para la creación de tableros con cuadros, gráficos y alertas relacionadas a información de datos métricos originados por el monitoreo de diversos procesos. Incluye el control de paneles administrativos para distintos tipos de usuarios, permitiendo configurar los permisos y credenciales de cada uno agrupándolos en organizaciones o grupos de trabajo en paneles específicos.

³³ EcuRed. *Raspbian*. Consultado el 30 de diciembre de 2021.

La amplia conectividad con innumerable cantidad de *plugins* y distintas bases de datos relacionales y no relacionales como Prometheus, InfluxDB, MySQL, PostgreSQL, MongoDB, entre otras; así como la posibilidad de enlazarse con diversos servicios en la nube convierten a Grafana en una solución orientada al monitoreo de sistemas IoT, con la posibilidad de escalar y contratar el servicio de Grafana Enterprise que es una versión de paga con licencia, la cual ofrece capacidades adicionales.

2.7.6.4. Bases de datos

Encargadas de controlar el almacenamiento de forma ordenada de enormes cantidades de información o datos estructurados que pertenecen a un mismo formato, asignando cada una de las entradas en filas y columnas que conforman una serie de tablas. Siguiendo esta estructura se obtiene acceso a la información organizada con posibilidades de gestionar, modificar, actualizar, controlar e ingresar fácilmente nuevos datos. Alternativamente a utilizar medios físicos para almacenar la información, los servicios en la nube son opciones ampliamente utilizadas por su flexibilidad de incorporarse a cualquier sector.

2.7.7. Microservicios

En la actualidad, los servicios que concentraban distintas funcionalidades en uno solo han sido descentralizados, optando por conformar arquitecturas con pequeños servicios independientes enfocados en el desempeño de cada aplicación creada con componentes autónomos que ejecutan cada proceso del sistema como un servicio único, comunicándose entre ellos por medio de una interfaz que hace uso de API's. Puesto que, los microservicios se ejecutan de forma independiente cada uno es capaz de actualizar, implementar y escalar fácilmente en el caso de requerir mejoras para satisfacer la demanda.

2.7.8. Servicios en la web

Son servicios que están disponibles en la red por medio de conexiones a internet, en donde las aplicaciones están soportadas por un servidor que permite un enlace de consulta desde el cliente hasta el servicio requerido. Las funcionalidades de cada servicio *web* van desde el almacenamiento de información como bases de datos, análisis o visualizaciones de aplicaciones que pueden ser integradas a una arquitectura previamente desarrollada.

2.7.9. Variables de medición

En el desarrollo de proyectos IoT es importante definir cuáles serán las variables de interés para monitorear, que, a la vez, están sujetas a los procesos de las cuales provienen. Algunas de las más importantes para considerar son:

- Variables eléctricas:
 - Tensión
 - Corriente
 - Potencia activa
 - Potencia reactiva:
 - Potencia reactiva inductiva
 - Potencia reactiva capacitiva
 - Potencia aparente
 - Factor de potencia
 - Energía activa
- Variables ambientales:
 - Temperatura
 - Humedad relativa

Cada variable se relaciona estrechamente con el proceso en el cual está presente, mostrando para los equipos y dispositivos eléctricos las variables que interfieren directamente con el consumo de la energía eléctrica y para las variables ambientales de importancia en el monitoreo de frigoríficos, cámaras congeladas y procesos a altas temperaturas.

2.7.10. **Tecnologías complementarias**

- Interfaces RS485 y RS232: son estándares utilizados en la industria para efectuar comunicación serial entre ordenadores y dispositivos periféricos que transmiten datos. La interfaz RS485 presenta ventajas al utilizar la topología bus que le permite interconectar múltiples receptores y transmisores a la vez. Ambas interfaces presentan características que son útiles según sean los requerimientos de la aplicación.³⁴

Tabla IV. **Diferencias entre RS232 y RS485**

Parámetro	RS232	RS485
Configuración de línea	<i>Single-ended</i>	Diferencial
Modo de operación	<i>Simplex o full duplex</i>	Simplex o half duplex
Máxima longitud de cableado	12 m	1 200 m
Máxima velocidad de datos	20 kbits/s	10 Mbits/s
Niveles lógicos	± 5 a ± 15 V	± 1.5 a ± 6 V
Impedancia mínima	3 a 7 k Ω	12 k Ω
Sensibilidad de recepción	± 3 V	± 200 mV

Fuente: elaboración propia.

- VNC: es una aplicación que permite el acceso y control remoto de otro ordenador, de esta forma todas las funcionalidades están disponibles

³⁴ WEIS, Olga. *Virtual Serial Port*. Consultado el 2 de enero de 2022.

desde el control de periféricos como el teclado y puntero hasta la visualización de la pantalla y ejecución casi en tiempo real de los programas o comandos ingresados. Todo esto desde otra computadora, tableta o teléfono móvil donde el usuario conectado obtiene el control total del equipo remoto.

- Remote.it: hace posible las conexiones entre equipos conectados a distintas redes, accediendo de forma segura a los dispositivos como si estos estuvieran conectados a la misma red local. La conexión es establecida por medio de un *proxy* y un *socket* proporcionados al equipo local, consiguiendo que todo el tráfico generado sea enrutado al dispositivo y puerto de destino. Es importante mencionar que la conexión a internet en ambos equipos es obligatoria y que la sesión iniciada está limitada.
- Eclipse Mosquitto: es uno de los *brokers* más conocidos y utilizados como servidor de mensajería en el protocolo MQTT, este se encarga de captar los mensajes publicados por los dispositivos clientes y los difunde entre los dispositivos clientes suscritos a ese *topic* en específico.
- Ngrok: es un servicio multiplataforma que permite exponer una aplicación *web* alojada en un ordenador local a internet, siendo alcanzada por cualquier usuario que disponga de la dirección del túnel que haya generado ngrok. Es tan simple como especificar en la línea de comandos el puerto de escucha de la aplicación para que esta sea pública, por ese motivo es utilizado normalmente para realizar pruebas de desempeño de algún servicio que se encuentre en fase de desarrollo.
- Programación OTA: la programación *Over-the-Air* (OTA) es empleada en el mundo de los microcontroladores para reprogramar los bloques de

código en la tarjeta de desarrollo, pero sin utilizar la clásica comunicación serial con el cable USB conectado a la computadora, en su lugar aprovecha la conectividad WiFi para cargar un nuevo programa enviado desde el IDE. Este proceso facilita la actualización de código, debido a que en ocasiones la accesibilidad física a los dispositivos es complicada.

2.8. Marco legal

En esta sección serán definidas las normativas y regulaciones internacionales que han surgido directamente para el avance tecnológico y la transformación digital, así como otros factores de interés en la industria. Aunque algunas de estas normativas no son obligatorias de cumplir en el país, son consideradas debido a que estas involucran optimizaciones en los factores de producción, ambiental y económico que suelen ser de interés para las empresas.

2.8.1. Normativa ISO/IEC 30141

ISO / IEC 30141: 2018 Este documento proporciona una arquitectura de referencia de IoT estandarizada que utiliza un vocabulario común, diseños reutilizables y las mejores prácticas de la industria. Utiliza un enfoque de arriba hacia abajo, comenzando con la recopilación de las características más importantes de IoT, abstrayéndolas en un modelo conceptual genérico IoT, derivando una referencia basada en un sistema de alto nivel con la posterior disección de ese modelo en cinco vistas de arquitectura desde diferentes perspectivas.³⁵

2.8.2. Normativa ISO/IEC 27033

Proporciona una explicación de los parámetros de seguridad que deben cumplirse al hacer uso de la red o al conectar dispositivos a internet en la misma red o desde una red externa. Desarrolla los conceptos de seguridad desde las

³⁵ ISO. *Internet of Things (IoT)*. Consultado el 3 de enero de 2022. Recuperado de <https://www.iso.org/standard/65695.html>.

actividades de gestiones en los dispositivos, usuarios finales, aplicaciones y servicios en la *web*; hasta las formas en las que se transfiere información a través de protocolos o enlaces de comunicación, así como la identificación y análisis de riesgos en la red y además propone recomendaciones para implementar arquitecturas seguras.³⁶

2.8.3. Normativa ISO 50001:2011

Establece las bases para administrar Sistemas de Gestión de la Energía (SGEn), enfocadas en mejorar el desempeño del consumo de energía, aumentar la eficiencia energética y reducir el impacto ambiental; aplicando estas recomendaciones se obtiene un incremento de ventajas competitivas y disminución de costos operativos en los distintos sectores industriales que implementan buenas prácticas, sin verse obligados a alterar su productividad.³⁷

La estructura de la norma está centrada en cuatro hitos importantes: planificar, hacer, verificar y actuar. Entendiendo el comportamiento energético se busca controlar y monitorear procesos en base a objetivos por cumplir con la finalidad de tomar acciones orientadas a la mejora continua.

2.8.4. Normativa UNE-EN 16247

Correspondiente a las auditorías energéticas, establece que estas deben consistir en una inspección y análisis sistemático del uso y consumo de energía de los diversos procesos que realiza una empresa, con el objetivo de identificar el uso de energías y derivar propuestas para la mejora de la eficiencia energética.

³⁶ ISO. *Information technology*. Consultado el 3 de enero de 2022.

³⁷ ISO Tools. *La Norma ISO 50001 y la Gestión de la Energía*. Consultado el 3 de enero de 2022.

Las fases para realizar una auditoría energética están enfocadas en: preparación, inspección de las instalaciones, toma de datos, contabilidad energética, propuestas de mejora, informe final y resultados.³⁸

- Uso racional y eficiente de la energía en Guatemala

“Como órgano regulador del subsector eléctrico, la Comisión Nacional de Energía Eléctrica (CNEE), ha realizado múltiples esfuerzos para promover la eficiencia en la demanda de energía eléctrica sirva para atender el crecimiento de una demanda neta sin desperdicios”.³⁹

El común denominador del análisis exhaustivo y completo de experiencias exitosas en países que han implementado Planes de Eficiencia Energética es la vigencia de leyes de eficiencia energética y la creación de instituciones responsables de su cumplimiento y administración. El Proyecto de Ley de Eficiencia Energética comprende la creación de:

- El Consejo Nacional de Eficiencia Energética.
- El Plan Integral de Eficiencia Energética.
- Financiamiento para el Uso Eficiente de la Energía.
- Regulaciones y Otros Mecanismos para la Promoción del Uso Eficiente de la Energía.⁴⁰

La Eficiencia Energética se justifica ya que es la solución de más bajo costo y de menor plazo de implementación para reducir la tasa de demanda de energía respecto a la tasa de crecimiento de la economía. Representa ahorros de mayor nivel y menor costo, optimiza el consumo energético y es una medida de mitigación para reducir a corto plazo los impactos ambientales.⁴¹

³⁸ Ingertec. *UNE-EN 16247 Auditorias Energéticas*. Consultado el 3 de enero de 2022.

³⁹ CNEE. *Eficiencia Energética*. Consultado el 21 de diciembre de 2021. Recuperado de <https://www.cnee.gob.gt/wp/eficiencia-energetica/>.

⁴⁰ Ibid.

⁴¹ Ibid.

3. MARCO METODOLÓGICO

Continuando con el diseño y explicación de cómo el IoT ofrece soluciones escalables para las empresas, en esta sección serán definidos los procedimientos y técnicas que son aplicadas durante la integración de los distintos sistemas y la interacción entre ellos, tomando como referencia los requerimientos y objetivos por cumplir en el proyecto.

3.1. Descripción del proyecto

El proyecto propuesto está enfocado en cumplir con las características esenciales de un sistema de monitoreo en tiempo real del consumo de energía eléctrica en tableros principales, cargas específicas y variables ambientales, con la finalidad de que la empresa disponga de las bases necesarias para implementar este tipo de soluciones a mayor escala.

Las mediciones registradas de los procesos monitoreados deben ser almacenadas en una base de datos y a partir de esta información generar paneles de visualización en una plataforma *web* multifuncional que sea accesible por distintos usuarios con la facultad para realizar consultas y descargas de información, adicionando capacidades para enviar alertas cuando algún valor esté fuera del rango permitido.

Asimismo, demostrar que se puede llegar a obtener un sistema completamente funcional y eficaz al implementar *hardware* de bajo coste que, en conjunto con la utilización del *software* adecuado generan una solución final que puede ser escalable dentro de un entorno industrial.

3.2. Arquitectura de aplicaciones IoT

Cuando se requiere poner en práctica una arquitectura IoT está debe de cumplir con determinadas características para que la implementación sea factible, permitiendo que la tecnología sea descentralizada en la cual cada servicio es flexible, escalable, eficiente, seguro e independiente con capacidad de comunicarse entre ellos mismos y con los objetos que forman parte de la topología, facilitando el acercamiento entre procesos y personas.

3.2.1. Requerimientos para la implementación

Estos hitos deben considerarse previo a realizar la implementación del sistema. Proyectando un análisis del lugar, el ambiente y los dispositivos a utilizar para que puedan cumplir con lo siguiente:

- Conectividad y comunicación
- Gestión y control remoto de dispositivos
 - Desconectar o reconnectar dispositivos
 - Actualizar *software*
 - Cambiar credenciales de seguridad
 - Autorizar o denegar capacidades del *hardware*
 - Configurar parámetros de redes
- Monitoreo, almacenamiento y análisis de los datos
- Escalabilidad
- Alta disponibilidad
- Integración de sistemas
- Seguridad
 - Ataques cibernéticos
 - Alertas de fallas

3.2.2. Modelo de capas IoT

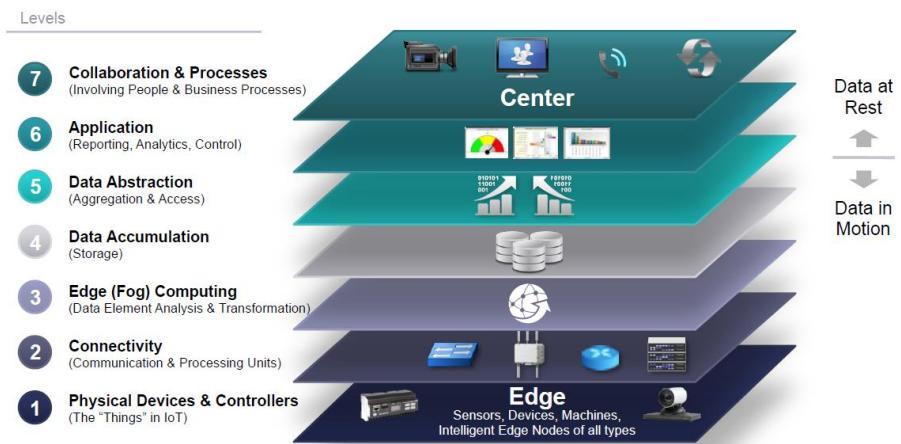
Una arquitectura IoT completa está formada por un modelo de 7 capas, las cuales integran distintos procesos y aplicaciones que van desde los objetos conectados hasta la colaboración de las personas en la toma de decisiones.

- Capa 1, objetos: sensores, controladores, actuadores, entre otros.
- Capa 2, conectividad: dispositivos para comunicación, protocolos y recolectores de información como; tarjetas de desarrollo y *gateways*.
- Capa 3, *edge computing*: análisis y transformación de los datos generados por los objetos de la capa 1.
- Capa 4, registro de los datos: bases de datos dedicadas al almacenamiento de la información.
- Capa 5, abstracción de los datos: manipulación y acceso a los datos generados por las capas inferiores.
- Capa 6, aplicaciones: plataformas para gestionar visualizaciones, reportes, análisis y control.
- Capa 7, colaboración y procesos: implica directamente la participación de las personas en la gestión de negocios basados en la información de valor obtenida hasta este punto.

La unión de las 7 capas constituye la estructura de un modelo para la arquitectura de un sistema IoT, englobando los aspectos físicos y virtuales en un único servicio que asegura el funcionamiento de todos los subsistemas que lo conforman.

Con la definición de este enfoque se pretende explicar la metodología para desarrollar un proyecto complejo, dividiéndolo en múltiples niveles que facilitan la integración del sistema en forma secuencial y ordenada.

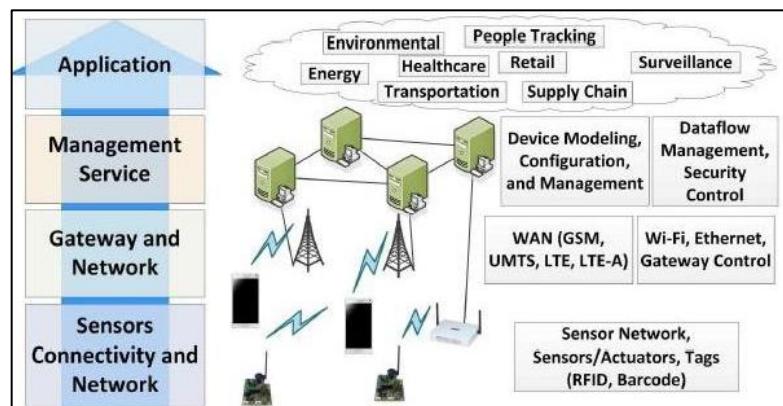
Figura 31. Las 7 capas de la arquitectura IoT



Fuente: Tech Data. Internet de las Cosas. Consultado el 5 de enero de 2022. Recuperado de <https://blog.techdata.com/ts/latam/internet-de-las-cosas-desde-abajo-dispositivos-y-controladores-f%C3%ADosicos>.

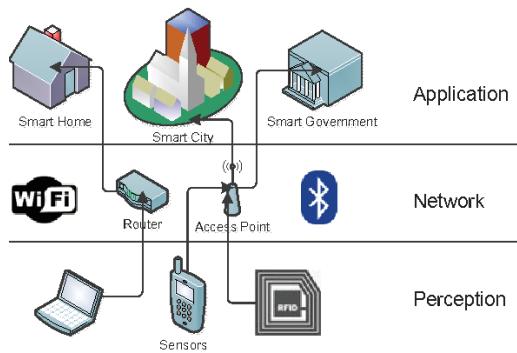
Otros tipos de arquitecturas:

Figura 32. Arquitectura IoT de 4 capas



Fuente: Arquitecturas IoT. Consultado el 5 de enero de 2022. Recuperado de <https://aprendiendoarduino.wordpress.com/tag/modelos-de-capas-iot/>.

Figura 33. Arquitectura IoT de 3 capas

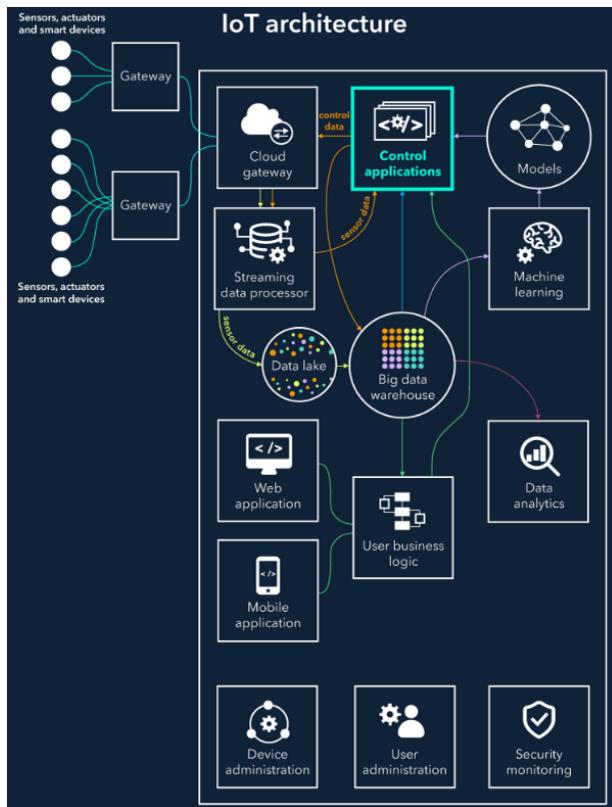


Fuente: Arquitecturas IoT. Consultado el 5 de enero de 2022. Recuperado de <https://aprendiendoarduino.wordpress.com/tag/modelos-de-capas-iot/>.

Los componentes que con frecuencia son utilizados en las topologías de red para arquitecturas IoT son:

- Objetos: equipos, máquinas, sensores para monitorear y actuadores para el control de procesos.
- Gateways/dispositivos inteligentes: interconectan a los objetos con la red, envían y reciben datos o comandos desde la nube.
- Pasarelas en nube: garantizan la transmisión de datos entre los gateways físicos y los servidores.
- Procesadores de datos: distribuyen en tiempo real los datos enviados por los sensores hacia los servicios establecidos.
- Bases de datos: almacenan toda la información monitoreada.
- Aplicaciones de usuario: para el control de actuadores y visualización del estado de los procesos censados.
- Análisis de datos: procesamiento manual o automático de la información.
- *Machine learning*: genera modelos para el pronóstico de líneas base a partir de los datos registrados.

Figura 34. **Arquitectura de componentes IoT**



Fuente: Arquitecturas IoT. Consultado el 5 de enero de 2022. Recuperado de <https://aprendiendoarduino.wordpress.com/tag/modelos-de-capas-iot/>.

3.2.3. **Fases de integración del IoT en la industria**

Previamente a la conceptualización de la arquitectura y selección de dispositivos y componentes a utilizar, se recomienda proceder con la etapa que consolida los pasos para la implementación.

- Fase 0: inicial. El primer paso es identificar aquellos procesos que requieren ser monitoreados, los equipos sin acceso a la red y los dispositivos con los que dispone la empresa.

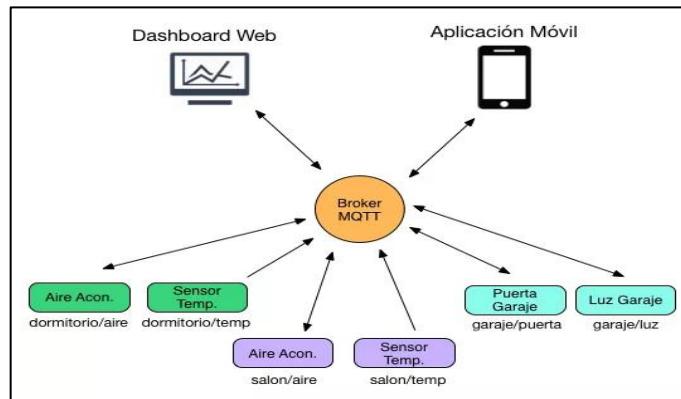
- Fase 1: conexión. Los procesos y equipos encontrados en la fase inicial son complementados con sensores y dispositivos inteligentes, que en conjunto envían la información censada del proceso y estado en tiempo real hacia las bases de datos. Es así como, conectar cosas se refiere a integrar nuevas funcionalidades a objetos desvinculados y proveer una forma para conectarse al internet permitiendo un acceso remoto.
- Fase 2: análisis y visualización. Para darle un sentido más completo al monitoreo, no basta simplemente con almacenar una gran cantidad de datos, es necesario analizarlos para notificar fallas o identificar posibles optimizaciones. A la vez generando información de valor que pueda ser mostrada a los usuarios en plataformas *web*.
- Fase 3: integración como una solución corporativa. Realizando la implementación de un plan piloto y teniendo en cuenta la escalabilidad que presentan este tipo de soluciones, es recomendable evaluar la posibilidad de gestionar un sistema propio a mayor escala que pueda ser replicado a cualquier otro tipo de proceso industrial.

3.3. Arquitectura del protocolo MQTT

Es el protocolo ideal utilizado para que las máquinas hablen entre ellas (M2M), permitiendo que los subsistemas se comuniquen con el servidor central. En la topología se definen a los equipos publicadores y suscriptores, que son llamados dispositivos clientes y el servidor es denominado como *broker* por el cual se transmiten y direccionan todos los mensajes. El protocolo se basa en la publicación/suscripción de dispositivos que generan una serie de sucesos al enviar y recibir mensajes que el *broker* se encarga de distribuir en base al *topic*

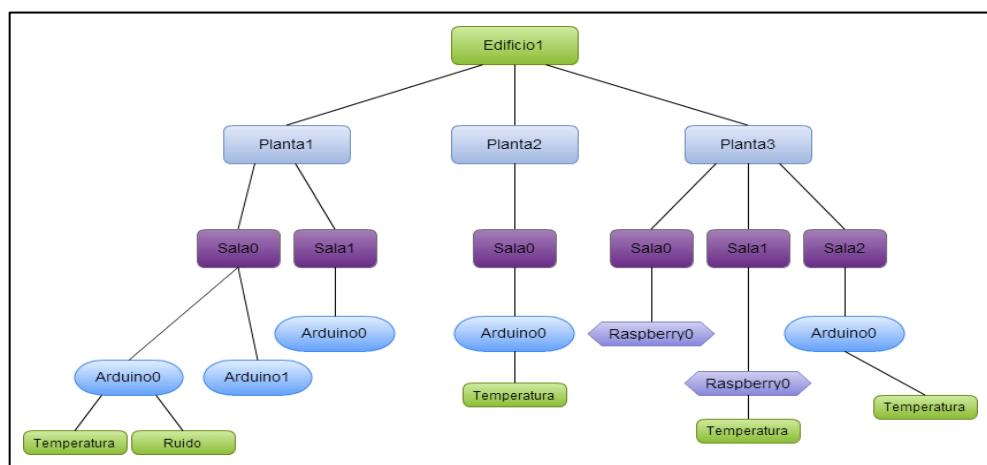
seleccionado. El *topic* es el elemento diferenciador que proporciona una dirección específica para que los publicadores sepan hacia donde enviar los datos.

Figura 35. **Topología entre clientes MQTT**



Fuente: Aprendiendo Arduino. MQTT. Consultado el 5 de enero de 2022. Recuperado de <https://aprendiendoarduino.wordpress.com/2018/11/19/mqtt/>.

Figura 36. **Jerarquía de clientes MQTT**



Fuente: Aprendiendo Arduino. MQTT. Consultado el 5 de enero de 2022. Recuperado de <https://aprendiendoarduino.wordpress.com/2018/11/19/mqtt/>.

4. DESARROLLO DEL PROYECTO

4.1. Equipos y especificaciones técnicas

El sistema de monitoreo implementado está constituido por dispositivos que ya se encontraban instalados en la empresa, los cuales miden el consumo eléctrico y fueron interconectados a través de protocolos de comunicación con dispositivos inteligentes. Asimismo, fue incorporado un sistema para monitorear el estado de cámaras congeladas, además se muestra el planteamiento para medir procesos a altas temperaturas. Los dispositivos utilizados son:

- Transformador de corriente 200/5 A: la instalación es realizada ocupando un TC para cada línea del circuito eléctrico a medir y a su salida las terminales de la bobina son conectadas al analizador de energía.

Tabla V. **Especificaciones técnicas TC4 marca CIRCUTOR**

Características eléctricas	Frecuencia	50 / 60 Hz
	Tensión de aislamiento	3 kVc.a.
	Corriente térmica de cortocircuito, I_{th}	60 I_n
	Corriente dinámica, I_{dyn}	2,5 I_{th}
	Tensión más elevada para el material	0,72 kVc.a.
	Clase térmica	B (130 °C)
	Tipo de encapsulado	Plástico V0 autoextinguible
	Factor de seguridad	FS 5
	Bornes secundarios precintables	Sí
	Terminales secundarios	IP 20
	Fijación en carril DIN	TC5 y TC6
Normas	IEC 61869-1, IEC 61869-2, BS2627	

Fuente: Circutor. Transformadores de corriente de perfil estrecho para medida. Consultado el 6 de enero de 2022. Recuperado de http://circutor.es/docs/FT_TC_SP.pdf.

- Analizadores de energía CVM-C5: utilizado para realizar las mediciones de tensión y corriente, conectando a las entradas aisladas las líneas y salidas de cada CT. Proporciona una interfaz RS485 para la comunicación bajo el protocolo MODBUS/RTU.

Tabla VI. **Especificaciones técnicas CVM-C5**

Círculo de alimentación	Tensión alimentación	85-265 Vc.a. / 50...60 Hz 95-300 Vc.c.
Círculo de medida	Tensión	300 Vc.a. f-n / 520 Vc.a. f-f
	Frecuencia	45 ... 65 Hz
	Corriente	.../5 A ó .../1 A MC: .../250 mA .../333 mV
	Muestreo	32 muestras/ciclo
Clase precisión	V, A	0,5%
	Potencia, Energía	1%
Salida	1 salida digital	Interfaz S0 Configurable hasta 1000 impulsos por kWh, kvarh, etc. (24 Vc.c. max, 50 mA, 5 imp/s, Max Ton/Toff configurable)
Entrada	1 entrada digital	Selección de tarifa, NPN, optoacoplada
Comunicaciones	Protocolo	RS-485 Modbus/RTU (según modelo)
	Velocidad	9600, 19200, 38400
	Bit, paridad, stop	8, n, 1
Características constructivas	Envolvente	Plástico VO autoextinguible
	Grado protección IP	Frontal: IP 51 Trasera: IP 21
	Dimensiones	96,7 x 96,7 x 62,60 mm
Condiciones ambientales	Temperatura de trabajo	-5...+45 °C
	Temperatura almacenamiento	-10...+50 °C
	Humedad relativa	5 ... 95%
	Altitud máxima	2000 m
Seguridad	Clase III según EN 61010 Protección frente a choque eléctrico por doble aislamiento Clase II	
Normas	IEC 664, VDE 0110, UL 94, IEC 801, IEC 348, IEC 571-1, EN 61000-6-3, EN 61000-6-1, EN 61010-1, CE, certificado UL	

Fuente: Circutor. Analizador de redes, Multímetro multifunción para panel. Consultado el 6 de enero de 2022. Recuperado de http://circutor.es/docs/FT_CVM-C5_ES.pdf.

- Conversor USB a RS485: empleado para comunicar la información generada por el analizador CVM-C5 (conectado a la entrada de la interfaz RS485) desde el puerto USB de la Raspberry Pi.

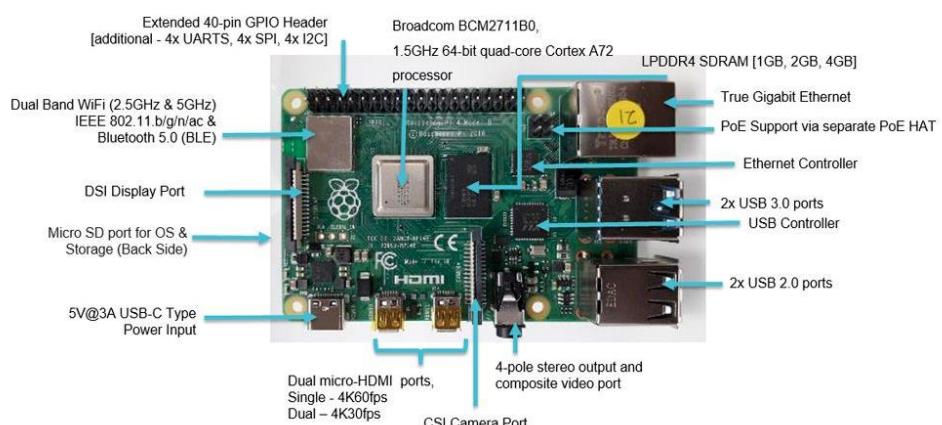
Tabla VII. **Especificaciones técnicas Conversor USB a RS485**

Chip USB a Serial	CH340G
Chip serial a RS485	MAX485
Interfaz	USB 2.0 tipo A macho
Velocidad	12 Mbps
Distancia máxima	1200 m
Dimensiones	54 mm x 16 mm

Fuente: elaboración propia, empleando Excel.

- Raspberry Pi 4 modelo B 4GB de RAM: es el dispositivo principal de la arquitectura IoT. Sobre ella se ejecutan la mayoría de los programas dedicados a la transferencia de datos, tableros de visualizaciones, acceso remoto y además contiene el *broker mosquitto* para la conexión de publicadores y subscriptores del protocolo MQTT de los subsistemas de monitoreo de cámaras congeladas y procesos a altas temperaturas.

Figura 37. **Localizaciones físicas en la Raspberry Pi 4**



Fuente: Omnidisco. Especificaciones y características de la nueva Raspberry Pi 4. Consultado el 6 de enero de 2022. Recuperado de <https://omnidisco.com/tecnologia/raspberry-pi-4-especificaciones-caracteristicas/>.

- Router Huawei: es el encargado de proveer la conectividad a la red, permitiendo a los dispositivos conectarse a internet. Por su facilidad de uso simplemente requiere de una tarjeta SIM con plan de datos o estar conectado a otra red WAN para suministrar una red LAN propia.

Tabla VIII. Especificaciones enrutador Huawei B311-221

<i>WiFi</i>	802.11 b/g/n
Conectividad SIM	4G LTE
Velocidad de descarga	150 Mbps
Dispositivos conectados	32
Conexión antena	Sí
Puerto WAN y Ethernet	1 conector para cada red

Fuente: elaboración propia, empleando Excel.

- Sensor DHT22: posicionado en el interior de las cámaras congeladas para monitorear los aumentos y descensos de temperatura, al mismo tiempo que censa la humedad relativa en ese ambiente. Por ser un sensor de bajo consumo de alimentación puede estar conectado directamente a la energía suministrada por la ESP32 y el pin de datos del sensor debe conectarse a un pin digital de lectura.

Tabla IX. Especificaciones técnicas sensor DHT22

Rango de medición de temperatura	- 40 °C a 80 °C
Precisión de medición de temperatura	< ± 0.5 °C
Resolución de temperatura	0.1 °C
Rango de medición de humedad	0 a 100 % RH
Precisión de medición de humedad	0.1 % RH
Resolución de humedad	0.1 % RH
Tiempo de censado	2s

Fuente: elaboración propia, empleando Excel.

- Sensor magnético: utilizado para el monitoreo de puertas. Consta de un imán y un sensor tipo switch que cambia de estado cuando se acerca o aleja un campo magnético, a través de esta detección se puede calcular desde la ESP32 las veces que la puerta ha sido abierta, el estado en el que se encuentra y el tiempo total que ha permanecido abierta.

Tabla X. **Especificaciones técnicas del sensor magnético de apertura**

Distancia de detección	15-25 mm
Potencia máxima	10 W
Tensión máxima	100 V
Corriente máxima	0.5 A

Fuente: elaboración propia, empleando Excel.

- Termocupla tipo J: ideal para monitorear procesos a altas temperaturas en el rango de 0 °C a 750 °C. Dispone de un vástago con una longitud considerable para introducir el sensor en distintas soluciones o ambientes.

Tabla XI. **Especificaciones técnicas termocupla**

Tipo	J
Diámetro de bulbo	3/16" (5 mm)
Longitud útil	4" (100 mm)
Material	Acero inoxidable
Conexión	5/16"
Alambre de extensión	2 m (fibra de vidrio enmallado)

Fuente: elaboración propia, empleando Excel.

- Módulo MAX6675: las terminales de entrada están diseñadas para la conexión de la señal proveniente de la termocupla, la cual es amplificada por el módulo y posteriormente leída por la ESP32.

Tabla XII. **Especificaciones técnicas MAX6675**

Voltaje de operación	5 V
Corriente de trabajo	50 mA
Resolución	12 bits (0 – 1023 °C)
Tipo de salida	SPI unidireccional

Fuente: elaboración propia, empleando Excel.

- ESP32: es el medio por el cual se puede obtener la información entregada a la salida de los sensores. Utilizando la conectividad WiFi se facilita la conectividad con otros sistemas dentro de la arquitectura, por ejemplo: siendo la ESP32 uno de los clientes del protocolo MQTT, se encuentra enviando datos frecuentemente hacia la Raspberry Pi. Es así como esta tarjeta de desarrollo conforma pequeños subsistemas de monitoreo al transmitir información de forma inalámbrica hacia un servidor central.

Tabla XIII. **Especificaciones técnicas ESP32**

Procesador	Tensilica Xtensa LX6 32 bit Dual-core a 160 MHz
Memoria RAM	520 kB
Memoria Flash	Hasta 16 MB
Memoria ROM	448 kB
Alimentación	2.2 a 3.6 V
Consumo de corriente	80 mA (promedio). 225 mA máximo
Consumo en modo sueño profundo	2.5 uA (10 uA RTC + memoria RTC)
WiFi	802.11 b/g/n WEP, WPA
Bluetooth	v4.2 BR/EDR Y BLE
Soft-AP	Sí
Encriptación por <i>hardware</i>	AES, SHA, RSA, ECC
Protocolos	CAN bus, Ethernet

Fuente: elaboración propia, empleando Excel.

4.2. Medición y adquisición de datos

Habiendo seleccionado y descrito los equipos a utilizar, es necesario conformar la topología entre dispositivos inteligentes, módulos y sensores para cada subsistema a implementar. Comprendiendo las funcionalidades de los dispositivos estos serán destinados para utilizarlos de forma óptima.

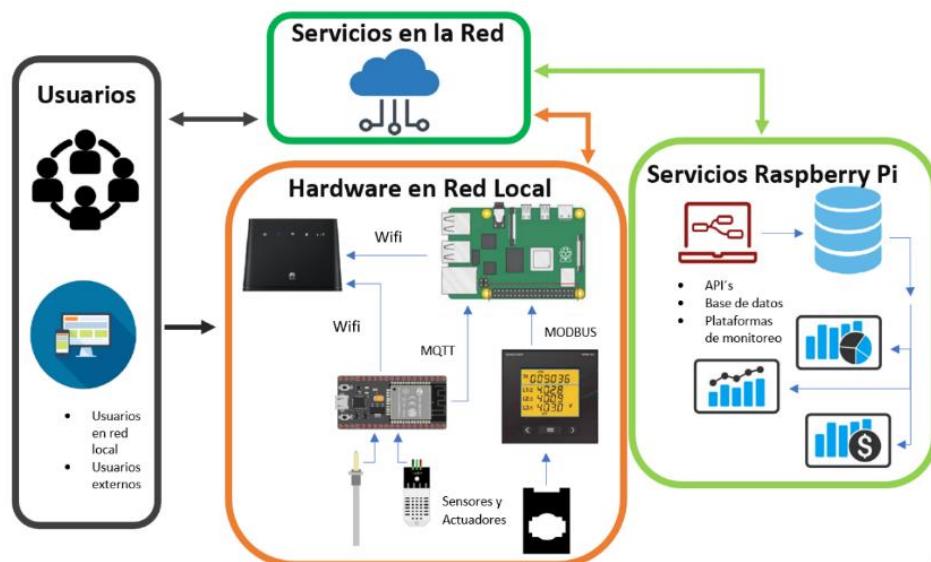
- Monitoreo del consumo de energía eléctrica (protocolo MODBUS/RTU)
 - Transformador de corriente
 - Analizadores de red eléctrica (esclavos)
 - Conversor de USB a RS485
 - Raspberry Pi 4 (maestro)
- Monitoreo de cámara congelada (protocolo MQTT)
 - Sensor DHT22
 - Sensor magnético
 - ESP32 (publicador 1)
 - Raspberry Pi 4 (*broker, suscriptor*)
- Monitoreo de proceso a alta temperatura (protocolo MQTT)
 - Termocupla tipo J
 - Módulo MAX6675
 - ESP32 (publicador 2)
 - Raspberry Pi 4 (*broker, suscriptor*)
- Conectividad remota
 - Router Huawei
 - Raspberry Pi 4 (servidor remote.it)

Las mediciones principales se realizan por los sensores que estarán monitoreando constantemente el cambio de las variables en los procesos, a la vez, los dispositivos inteligentes serán los encargados de registrar estos datos y compartirlos con los distintos servicios agregados en el sistema. Logrando constituir pequeños servicios que son agregados a un servicio único que integra todas las funcionalidades de una estación de monitoreo remota.

4.3. Diagramas de conexiones

Con el fin de ejemplificar en una forma gráfica la instalación de los sensores sobre los procesos a monitorear, las conexiones entre los dispositivos y protocolos, así como, los distintos servicios implementados para la interacción con los usuarios; se presentan en esta sección los diagramas utilizados en la arquitectura general y los subsistemas que la conforman.

Figura 38. Arquitectura IoT implementada



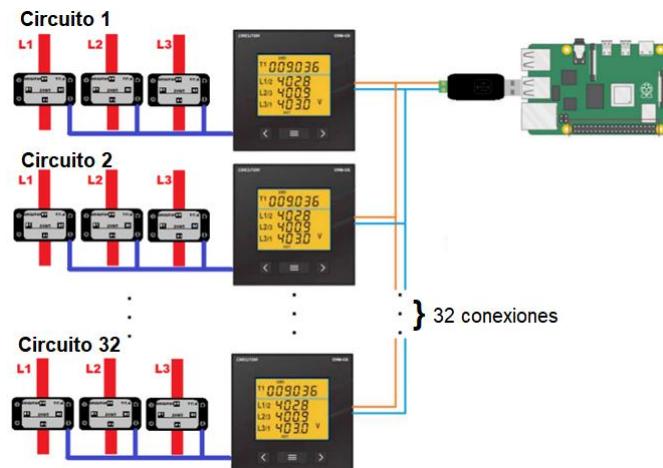
Fuente: elaboración propia, empleando Paint.

4.3.1. Sistema de monitoreo de energía eléctrica

Este sistema está basado en medir la corriente de cada fase en circuitos trifásicos, bifásicos o monofásicos; haciendo pasar cada línea de corriente alterna en un CT, estos envían las señales de corriente inducidas hacia el analizador de redes, el cual, en base a la diferencia de potencial y corriente censada realiza cálculos internos para obtener información de la red eléctrica.

Sin embargo, el sistema se encuentra en un estado transitorio y los valores cambian instantáneamente en función al consumo de energía. Debido a que el analizador no cuenta con una memoria para almacenar los datos, se requiere de un dispositivo adicional (en este caso la Raspberry Pi) que establezca una comunicación tipo *pull* y realice peticiones de los datos a través de la interfaz RS485 en conjunto con el protocolo MODBUS/RTU que especifica las direcciones de los parámetros, de esta forma se obtienen los datos y almacenan localmente o en la nube.

Figura 39. Diagrama CT, analizador y Raspberry Pi



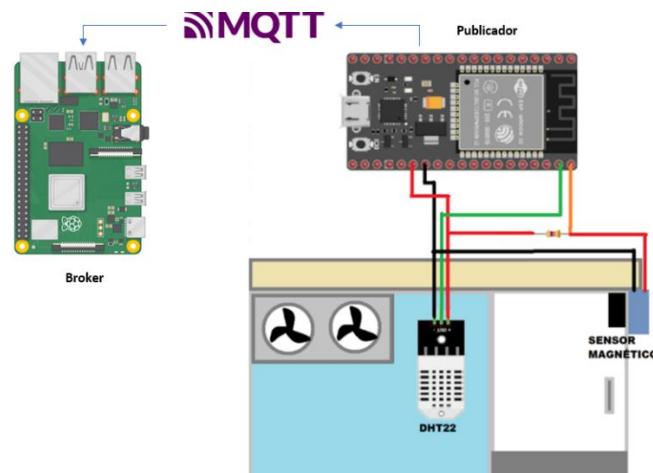
Fuente: elaboración propia, empleando Paint.

4.3.2. Sistema de monitoreo de cámaras a bajas temperaturas

Con el objetivo de realizar una comparativa entre el factor energético y las variables de control del proceso seleccionado, se propone el sistema de monitoreo de temperatura para correlacionar este valor con el obtenido de la energía eléctrica consumida por el sistema de enfriamiento en la cámara.

Teniendo en cuenta el factor de apertura y cierre de la puerta al ingresar a la cámara y que esto causa en la temperatura un aumento debido a que la cámara no está completamente cerrada, se da a conocer una relación en la cantidad de veces que la puerta es abierta y del tiempo en que permanece en ese estado. Siendo necesario tener un registro de esta actividad, la ESP32 obtiene los datos monitoreados por ambos sensores y los envía hacia la Raspberry Pi a través del protocolo MQTT. Logrando obtener indicadores del cambio de la temperatura y las causas que provocan estas alteraciones en el sistema que trata de regular la variable de control al nivel indicado.

Figura 40. Diagrama sensor magnético, DHT22 y ESP32



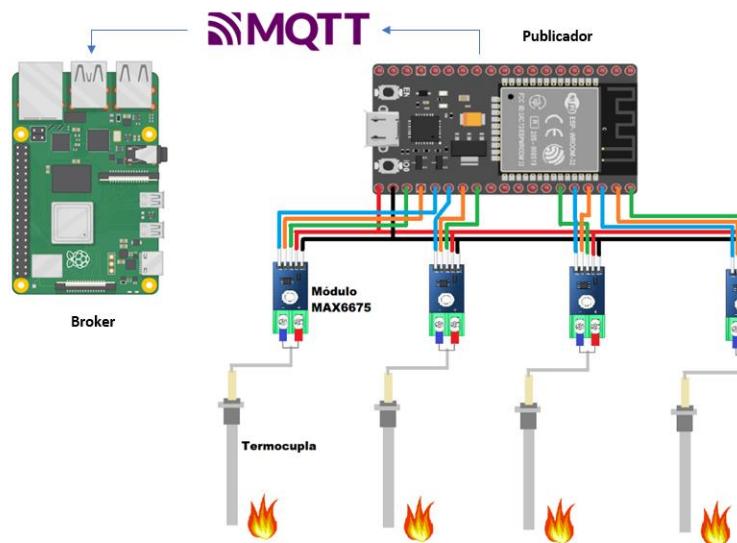
Fuente: elaboración propia, empleando Paint.

4.3.3. Sistema de monitoreo de alta temperatura

Continuando con la comparativa del consumo de energía eléctrica y el comportamiento de las variables de control en los procesos, este sistema se enfoca en monitorear altas temperaturas utilizando una termocupla de grado industrial para evitar un rápido deterioro y obtener mejores resultados en la precisión y exactitud de las mediciones.

El sistema propuesto consta de una termocupla que está conectada a un módulo MAX6675 el cual, se comunica con la ESP32 que dispone de conexiones para un total de 3 o 4 módulos de forma simultánea, dotando al sistema con un monitoreo de distintos procesos al mismo tiempo. La tarjeta de desarrollo obtiene las señales amplificadas que son convertidas en datos y enviados hacia el *broker* con un *topic* específico para diferenciar cada sensor y proceso censado.

Figura 41. Diagrama termocuplas, MAX6675 y ESP32



Fuente: elaboración propia, empleando Paint.

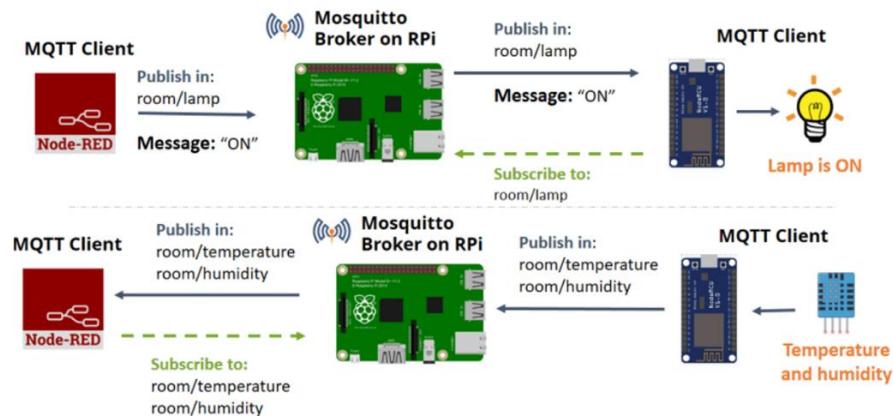
4.4. Conexiones de servicios

La arquitectura de *software* formada por el conjunto de aplicaciones ejecutadas como servicios en la Raspberry Pi permiten realizar múltiples tareas en tiempo real y conectarse a otros servicios de la misma red o en la nube, gestionando para el usuario una plataforma que opera en el *frontend* y que está constituida por diferentes aplicaciones funcionales en el *backend*.

4.4.1. Conexiones MQTT

Implementando este protocolo en la arquitectura para la comunicación entre los sistemas de monitoreo de bajas y altas temperaturas con el dispositivo central, en este caso la Raspberry Pi, la cual está configurada en modalidad de *broker* y a la vez incluye en Node-RED bloques cliente suscriptor y publicador MQTT. De esta forma obtiene, visualiza y controla los datos que son enviados a cada *topic* por los dispositivos publicadores, es decir; por la tarjeta de desarrollo ESP32.

Figura 42. Esquema MQTT con broker Mosquitto



Fuente: ESP8266 and Node-RED with MQTT. Consultado el 8 de enero de 2022. Recuperado de <https://randomnerdtutorials.com/esp8266-and-node-red-with-mqtt/>.

4.4.2. Diagrama de aplicaciones en plataforma de monitoreo

Muchas de las aplicaciones se encuentran alojadas en la Raspberry Pi, por lo cual la comunicación entre aplicaciones es más rápida y segura. Partiendo desde la adquisición de datos (MODBUS/RTU y los recibidos por MQTT), en Node-RED se configuran distintos bloques para formar nodos con opciones de redireccionar información a su propio tablero de monitoreo o enviarla a bases de datos locales o almacenamientos externos en la nube.

Con la información disponible en bases de datos, es posible establecer una conexión entre distintos servicios a través de API's para obtener los datos que sirven de entradas para generar cuadros, gráficos y realizar análisis en plataformas de monitoreo. Debido a que la aplicación Grafana se ejecuta de forma local en la Raspberry Pi, se hace necesario utilizar el servidor ngrok para que la plataforma sea alcanzable por redes externas. Ngrok genera un túnel desde sus propios servidores, permitiéndole a los usuarios conectarse con la dirección suministrada, siendo de utilidad en pruebas de funcionamiento.

Figura 43. Esquema de aplicaciones



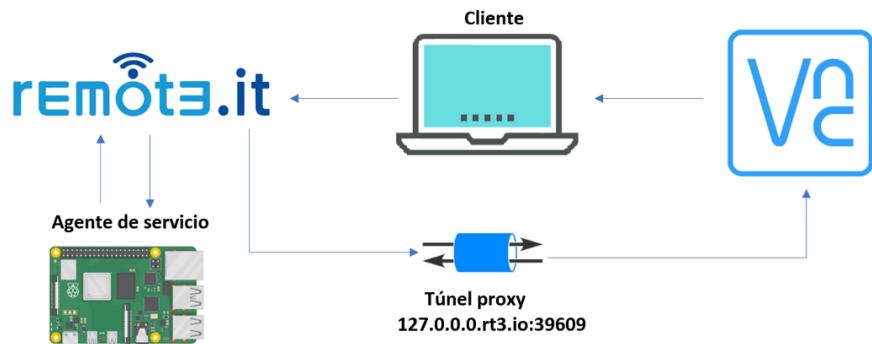
Fuente: elaboración propia, empleando Paint.

4.4.3. Diagrama de aplicaciones para conexión remota

El control remoto de dispositivos es un factor por implementar para tener la capacidad de realizar cambios, configuraciones, instalaciones o correcciones en los sistemas; sin tener que estar de forma presencial en el lugar de instalación.

Utilizando remote.it en conjunto con VNC es posible tener acceso remoto en la Raspberry Pi, esto funciona instalándole un agente de servicio de remote.it que mantiene una conexión para tráfico UDP en los puertos especificados; de esta forma, cuando un cliente solicite conectarse, la aplicación responderá con una dirección de un socket para ingresarla en VNC. De modo que, el cliente podrá visualizar y controlar a la Raspberry Pi desde la interfaz gráfica.

Figura 44. Conexión remota desde ordenador hacia Raspberry Pi



Fuente: elaboración propia, empleando Paint.

4.5. Diseño y fabricación de PCB expandida

La fabricación de una placa de circuito impreso es requerida para asegurar las conexiones entre la tarjeta de desarrollo y los sensores o controladores que pueden incorporarse en el momento o a largo plazo. Teniendo en cuenta la exigencia que presenta la instalación de dispositivos en sectores industriales, las desconexiones del cableado suelen ser susceptibles si las terminales no están aseguradas con los conectores adecuados, por estas razones se diseña una placa que permita ensamblar la ESP32 de forma fácil y segura. Asimismo, se consideran las borneras para la fijación de cables y un *dip-switch* para seleccionar distintos modos de configuraciones en resistencias y alimentación de sensores.

4.6. Diseño e impresión 3D

La impresión 3D es altamente recomendada para obtener modelos físicos de diseños en un corto tiempo y a un bajo coste, con el fin de proporcionar un case que se adecue a las dimensiones de la PCB expandida, se realiza un diseño en el *software* Inventor de Autodesk. Este case facilita la instalación del

dispositivo en distintas áreas de la empresa, a la vez, debe proteger al circuito y componentes de ciertos factores ambientales, brindando un grado mayor de seguridad al sistema.

Figura 45. **PCB expandida y case impreso en 3D**



Fuente: elaboración propia, empleando Autodesk.

5. FASE DE SERVICIO TÉCNICO PROFESIONAL, MANUAL DE CONFIGURACIONES BÁSICAS PARA ESTABLECER COMUNICACIÓN ENTRE LOS DISPOSITIVOS

5.1. Análisis de la situación actual

En las distintas visitas de campo y técnicas realizadas al lugar que fue destinado para implementar el sistema de monitoreo, se hizo necesario iniciar con una primera fase de identificación de equipos y conocimiento de los procesos que se llevan a cabo, así como, su control de operaciones; esto con el objetivo de seleccionar cuáles son los procesos y equipos que presentan mayor consumo de energía eléctrica y aplicar las mediciones a estos circuitos.

Una de las cualidades del sistema desarrollado es que logra incorporar antiguos equipos que no disponen de conectividad con la red, en este caso la empresa ya poseía analizadores de energía y transformadores de corriente instalados que solamente mostraban datos en pantalla de forma local. Por este motivo, el enfoque primordial fue proveer un canal de comunicación para estos dispositivos, aplicando el Internet de las Cosas, evitando generar costos adicionales por adquirir e instalar nuevos equipos.

Actualmente, los dispositivos instalados conforman un sistema de monitoreo de múltiples variables, tanto variables de energía eléctrica como ambientales, con posibilidad de censar cargas específicas o tableros completos. La escalabilidad que presenta el sistema le permite expandirse a otras mediciones que pueden ser integradas fácilmente al servidor principal, sin olvidar el acople con servicios en la nube para gestionar proyectos en distintos puntos.

5.2. Manual de configuraciones

Para completar la descripción del funcionamiento de un sistema capaz de operar en conjunto con otros servicios, se considera importante aplicar las configuraciones adecuadas en los dispositivos y programas utilizados, esto depende de protocolos implementados y de las funcionalidades de cada equipo.

5.2.1. Objetivos del manual

- Proveer un manual técnico para que la empresa disponga de las bases para desarrollar sistemas similares o alternativos.
- Describir las configuraciones básicas para establecer comunicación entre dispositivos, aplicaciones y servicios en la nube.
- Contribuir a la comunidad estudiantil con información acerca de nuevas tecnologías y el cómo poder implementarlas.

5.2.2. Alcances del manual

Es preciso mencionar que las soluciones presentadas son algunas de las alternativas que existen en la actualidad y que fueron seleccionadas en el desarrollo de este proyecto. Pudiendo optar por utilizar otro tipo de servicios que se consideren pertinentes, así como otros dispositivos de diferentes marcas.

Solamente se describe cómo realizar determinadas configuraciones respecto al *software* utilizado, queda a cargo del desarrollador del proyecto gestionar la instalación de los programas. Los enlaces para acceder a la documentación en sitios oficiales pueden encontrarse en la sección de Anexos.

En algunos casos la programación suele ser parecida o los resultados que se quieren obtener convergen en algún punto con lo descrito en este informe, por lo tanto, puede ser utilizado como una guía o bases para el desarrollo de otras aplicaciones relacionadas con el Internet de las Cosas.

5.2.3. Configuraciones en Raspberry Pi

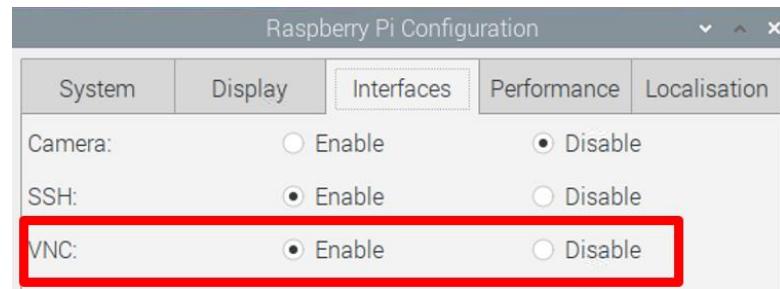
Siendo la Raspberry Pi un ordenador de placa reducida y el dispositivo principal sobre el cual se basa el desarrollo y ejecución de aplicaciones, además de que permite implementar otras configuraciones de protocolos y comunicación con módulos, se considera como la pieza fundamental para implementar servicios y conectividad entre sistemas.

5.2.3.1. Configuraciones generales en el sistema operativo Raspbian

El primer paso por considerar es proporcionar el sistema operativo a la Raspberry Pi, en este caso se utilizó la versión Debian llamada *buster*, sin embargo, existen más opciones a escoger en la página oficial de Raspberry Pi Foundation (ver la sección del Anexo 8).

Para realizar las configuraciones iniciales se requiere de un monitor, teclado y *mouse* conectados a la Raspberry Pi, de esta forma se ingresa desde el escritorio para conectarse a una red Ethernet o WiFi y posteriormente habilitar la opción VNC que permite la conexión desde otro dispositivo que esté en la red local. Cuando se habilita esta opción debemos establecer un nombre de usuario y una contraseña, ya que por seguridad será solicitada por el programa VNC previo a conceder el acceso a un usuario.

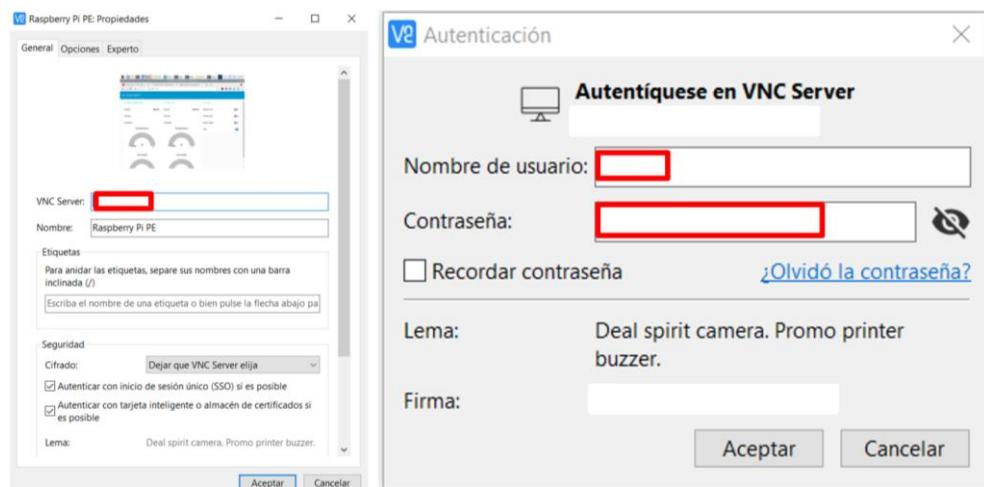
Figura 46. Configuración de interfaces en Raspberry Pi



Fuente: elaboración propia, empleando Debian Buster.

Con estas configuraciones realizadas ya no es necesario utilizar el monitor, teclado y *mouse* extra para controlar a la Raspberry Pi, simplemente debemos obtener la *IP* asignada a la Raspberry Pi e ingresarla en el cuadro de VNC Server, además del nombre de usuario y contraseña previamente establecidos. Para obtener la *IP* en una terminal Windows puede utilizar: *ping raspberry.local -4*.

Figura 47. Parámetros para ingresar en VNC



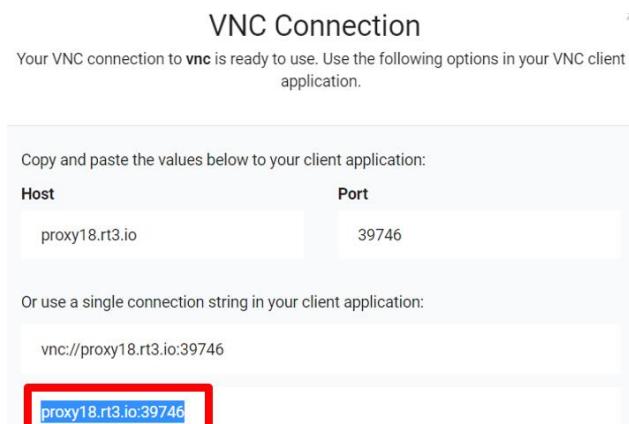
Fuente: elaboración propia, empleando VNC.

Habiendo establecido el control desde otro ordenador, procedemos a instalar y gestionar los programas complementarios en la Raspberry Pi, estos son: el agente de servicio `remote.it`, Node-RED, Grafana, la base de datos MariaDB, el *broker mosquitto MQTT* y el servidor `ngrok` (consultar Anexo 8).

5.2.3.2. Configuraciones en conexión remota

El servicio de `remote.it` requiere de una dirección de correo electrónico para identificar al usuario y el dispositivo a conectar, posteriormente de haber instalado el agente de servicio en la Raspberry Pi ingresamos a la aplicación *web* de `remote.it` en la cual aparecerá el dispositivo que hemos vinculado anteriormente. Pulsamos el botón `CONNECT` y seleccionamos la opción `VNC` que devolverá un mensaje con el *socket* generado para la conexión remota.

Figura 48. Conexión generada desde `remote.it`



Fuente: elaboración propia, empleando `remote.it`.

En las propiedades de la conexión VNC debemos reemplazar la *IP* por la dirección de *socket* obtenida, de esta forma podremos conectarnos desde cualquier red externa para controlar remotamente a la Raspberry Pi.

5.2.3.3. Programación Python – MODBUS

El lenguaje de programación Python incluye conectividad con múltiples paquetes que pueden ser instalados para incluir fácilmente instrucciones complejas en el código. En este caso la librería *minimalmodbus* proporciona un conjunto de funciones para establecer la comunicación entre la Raspberry Pi y el conversor USB a RS485, y a través de este realizar peticiones al analizador.

Inicialmente se tiene que definir un objeto con la función indicada en el código: añadiendo la dirección del puerto donde se haya conectado el conversor, posteriormente se configuran todos los parámetros de comunicación, teniendo en cuenta que deben estar sincronizados tanto en el analizador como en el programa; y para obtener los datos se utiliza la función de lectura que incluye la librería, en la cual se especifican las direcciones de los parámetros que representan a la variable medida. Esta tabla es propia del analizador CIRCUTOR CVM-C5 (ver anexo 5), los mapas de memoria MODBUS pueden cambiar en función del modelo o marca del dispositivo.

Figura 49. Código en Python para comunicación MODBUS (parte 1)

```
import minimalmodbus
import time
import serial

instrument = minimalmodbus.Instrument('/dev/ttyUSB0',4)

instrument.serial.baudrate = 9600
instrument.serial.bytesize = 8
instrument.serial.parity   = serial.PARITY_NONE
instrument.serial.stopbits = 1
instrument.serial.timeout  = 1
instrument.mode            = minimalmodbus.MODE_RTU
instrument.debug           = False
instrument.serial.xonxoff  = True
instrument.serial.rtscts   = False
instrument.serial.dsrdtr   = False
minimalmodbus.CLOSE_PORT_AFTER_EACH_CALL = True

usb1_on = True
```

Fuente: elaboración propia, empleando Python.

Figura 50. Código en Python para comunicación MODBUS (parte 2)

```
if instrument.debug == True:
    print(instrument)

test_reg = instrument.read_registers(0,2,3)
test_reg = list(map(lambda x: x / 10, test_reg))
print(test_reg)

print(" ")
print("Frecuencia: ",test_reg[1], "Hz")
print("Tensión F-N L1: ",test_reg[7], "V")
print("Tensión L1-L2: ",test_reg[3], "V")
print("Tensión L2-L3: ",test_reg[5], "V")
print("THD: ",test_reg[9], "%")

while 1:
    if usbl_on == True:
        try:
            test_reg = instrument.read_registers(1,0)
            print(test_reg)
            print("Registros de 1")
            test_reg = instrument.read_registers(40,40,3)

            print("Frecuencia: ",test_reg[1], "Hz")
            print("Tensión F-N L1: ",test_reg[7], "V")
            print("Tensión L1-L2: ",test_reg[3], "V")
            print("Tensión L2-L3: ",test_reg[5], "V")
            print("THD: ",test_reg[9], "%")

        while 1:
            if usbl_on == True:
                try:
                    test_reg = instrument.read_registers(1,0)
                    print(test_reg)
                    print("Registros de 1")
                    test_reg = instrument.read_registers(40,40,3)
                    resp = test_reg/10
                    print(test_reg)
                    test_reg = instrument.read_registers(0,10,4)
                    print(test_reg)
                    time.sleep(0.5)
                except:
                    print("Error en la lectura de registros")
            else:
                print("USB desconectado")
        except:
            print("Error en la lectura de registros")
    else:
        print("USB desconectado")

```

Fuente: elaboración propia, empleando Python.

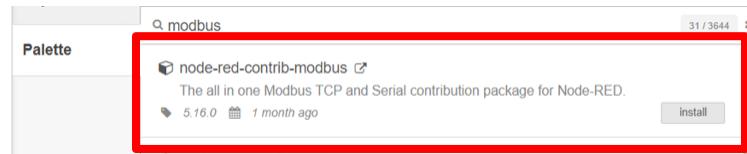
5.2.3.4. Programación Node-RED – MODBUS

La configuración de bloques en Node-RED facilita la creación de nodos que cumplen con diferentes funciones y, además, la ejecución de flujos realizada en paralelo permite desplegar múltiples procesos al mismo tiempo. Para ingresar a la interfaz de Node-RED debemos escribir en el navegador la dirección *IP* del ordenador y el puerto de la aplicación, que, por lo general, es 1880, conformando un *socket* de la siguiente forma: 127.0.0.0:1880 o localhost:1880.

Previo a iniciar con la configuración de bloques es necesario descargar el paquete de nodos MODBUS, ya que estos no se encuentran instalados por

defecto. La búsqueda de nuevos nodos se realiza en la administración de la paleta y en la sección de instalaciones se realiza la consulta de los módulos disponibles, encontrando una breve descripción y la versión del paquete.

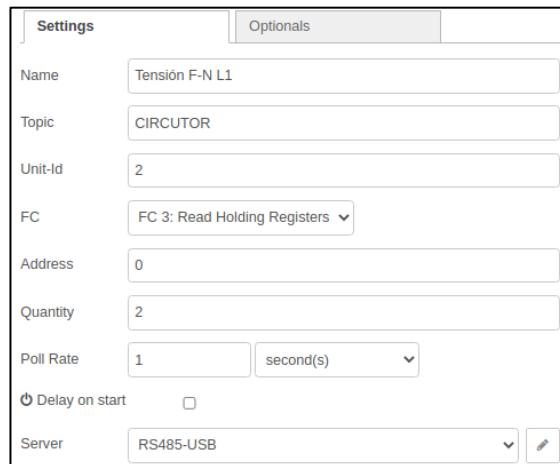
Figura 51. **Gestor de búsqueda de módulos en Node-RED**



Fuente: elaboración propia, empleando Node-RED.

Los bloques instalados se encuentran en la paleta ubicada en el lado izquierdo, disponiendo de una amplia gama de nodos con distintas funcionalidades. Acá seleccionamos el nodo *modbus-read* y lo movemos hacia el tablero de flujo, abrimos la configuración haciendo doble *clic* sobre el nodo.

Figura 52. **Configuraciones en nodo Modbus-Read**

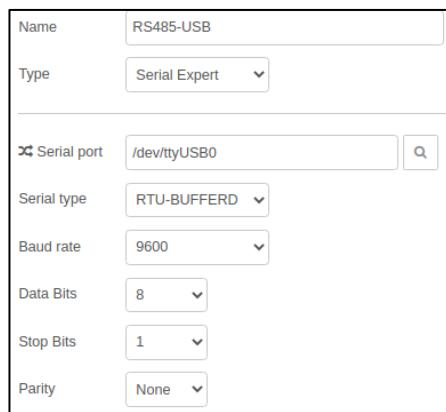


Fuente: elaboración propia, empleando Node-RED.

En esta sección se debe establecer un nombre para el nodo y respecto a la opción *Unit-Id* corresponde al número del dispositivo esclavo presente en la arquitectura, *Address* es el parámetro del mapa MODBUS de la variable a obtener, a la vez permite especificar un *Pull Rate* para solicitarle los datos al analizador de energía.

En la opción del servidor se configuran los parámetros para establecer la comunicación con el analizador, estos valores deben concordar en igualdad tanto en el dispositivo maestro como en el esclavo, de lo contrario no es posible sincronizar la transmisión de información.

Figura 53. **Configuraciones MODBUS en Node-RED**



Fuente: elaboración propia, empleando Node-RED.

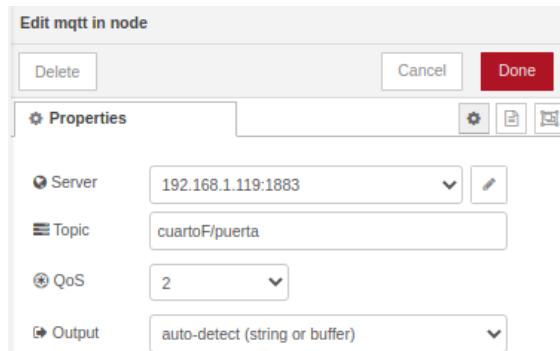
Al finalizar de establecer los parámetros en el nodo, presionamos los botones *Update* y *Done* secuencialmente, con lo cual estamos confirmando y guardando las configuraciones realizadas. Se recomienda agregar un nodo *debug* para visualizar la información obtenida, por último, presionamos el botón *Deploy* ubicado en la parte superior derecha y al realizar esto, todos los flujos configurados previamente iniciarán a ejecutarse con los cambios indicados.

5.2.3.5. Programación Node-RED – MQTT

Existen dos tipos de nodos para MQTT en Node-RED, estos son: *mqtt in* y *mqtt out*, los cuales corresponden a clientes suscriptores y publicadores, respectivamente. En esta sección se explicará la configuración para cada uno.

En el nodo *mqtt in* debemos especificar cuál será el nombre del *topic* al cual vamos a suscribirnos, así mismo seleccionar el nivel de servicio *QoS*; para garantizar la entrega de mensajes es recomendable optar por el nivel 2. En esta parte también podemos escoger, modificar o editar un servidor ya definido.

Figura 54. **Configuraciones del nodo mqtt in**

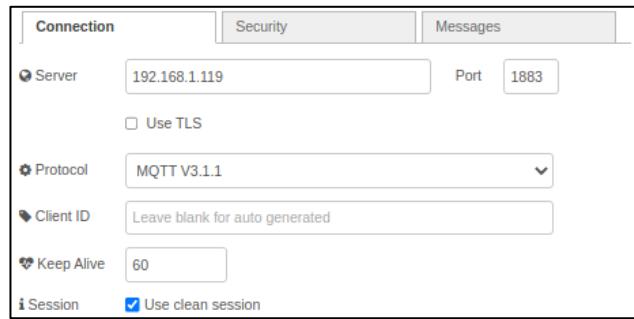


Fuente: elaboración propia, empleando Node-RED.

Si requiere establecer un nuevo servidor MQTT ingresamos a las configuraciones y editamos la dirección del *broker* que puede ser la *IP* en la cual instalo *mosquitto server* u otra dirección de servicio en la nube. El puerto por defecto de MQTT es el 1883, este puede ser cambiado por el desarrollador.

Al finalizar todas las configuraciones no olvide presionar los botones *Update* y *Done* para que todas las modificaciones realizadas queden guardadas.

Figura 55. **Configuraciones para un servidor MQTT en Node-RED**



Fuente: elaboración propia, empleando Node-RED.

Para configurar un nodo *mqqt out* como cliente publicador, simplemente escogemos el servidor y el *topic* al que estaremos enviando los mensajes.

Figura 56. **Configuraciones del nodo mqtt out**



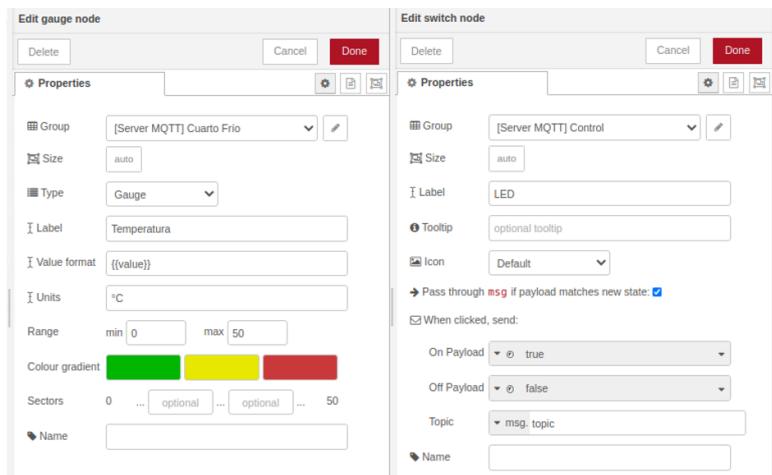
Fuente: elaboración propia, empleando Node-RED.

5.2.3.6. **Configuración de nodos dashboard**

Node-RED incluye una amplia variedad de nodos para generar gráficos, botones y otras visualizaciones en el *dashboard*; estos se adecuan a los datos que estemos midiendo y también permiten generar paneles de control con espacios personalizados que indican funciones externas o en el mismo tablero. En la configuración del nodo es posible seleccionar el grupo al cual pertenece, el

tipo de indicador, establecer un nombre a la variable, así como las unidades y un rango mostrado con colores para los valores registrados.

Figura 57. **Configuraciones de distintos nodos *dashboard***



Fuente: elaboración propia, empleando Node-RED.

5.2.3.7. Programación Node-RED – base de datos

Los bloques de función admiten la implementación de código basado en Node.js lo que facilita importar y exportar valores desde otros nodos, de esta forma se recolecta información de distintas fuentes, agruparla y enviarla a otra aplicación desde los nodos que conectan con estos servicios.

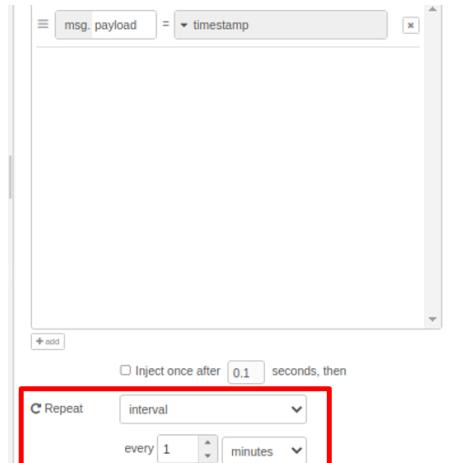
En este proyecto se implementa el envío de datos de forma local y en la nube, los servicios utilizados son: la base de datos MariaDB y las hojas de cálculo en Google Drive, mostrando las ventajas y desventajas de algunas formas de almacenamiento de información, gestionando la recepción de datos desde Node-RED hacia ambos servicios. A continuación, se muestran los bloques a configurar para establecer la transmisión de datos con estos servicios.

- Envío de datos a MariaDB

Los nodos por utilizar son:

- Nodo de inyección: con este nodo se establece el intervalo de tiempo para enviar los datos periódicamente hacia MariaDB. En el mensaje *payload* seleccionamos la opción *timestamp* y habilitamos la sección de intervalo para enviar una señal al nodo función que le indica cuando debe repetir su contenido según el tiempo especificado en este nodo.

Figura 58. **Configuración nodo *inject***



Fuente: elaboración propia, empleando Node-RED.

- Nodo función: acá podemos implementar el código que se encargue de adquirir los datos de las variables obtenidas por los nodos MODBUS y MQTT. Concatenando todo en un mensaje podemos incluir los comandos para insertar, borrar o modificar valores en la base de datos.

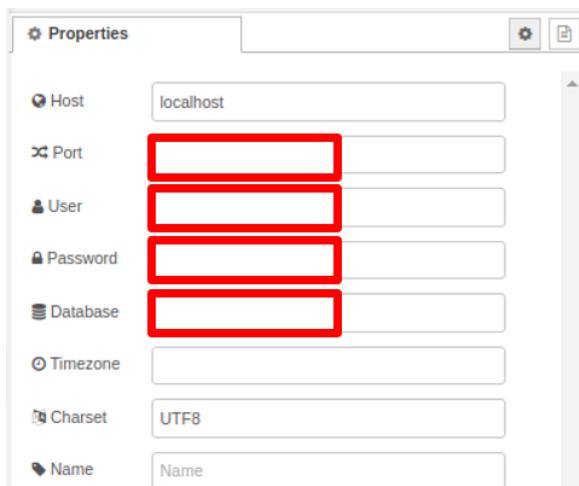
Figura 59. **Programación para insertar datos en MariaDB**

```
1 mensaje = "INSERT INTO Equipo1_T1 (T_L1,C_L1,P_L1,T_L2,C_L2,P_L2,T_L3,C_L3,P_L3)";  
2 mensaje = mensaje + "VALUES (";  
3 mensaje = mensaje + ""+flow.get("T_L1")+"','"+flow.get("C_L1")+"','"+flow.get("P_L1")+"";  
4 mensaje = mensaje + ",'" +flow.get("T_L2")+"','"+flow.get("C_L2")+"','"+flow.get("P_L2")+"";  
5 mensaje = mensaje + ",'" +flow.get("T_L3")+"','"+flow.get("C_L3")+"','"+flow.get("P_L3")+"";  
6 mensaje = mensaje + ")";  
7  
8 msg.topic = mensaje;  
9 return msg;
```

Fuente: elaboración propia, empleando Node-RED.

- Nodo mysql: para realizar la conexión entre la base de datos y Node-RED se configura el servidor con las credenciales de usuario y contraseña establecidas al instalar MariaDB, así como el *host* y número de puerto, en el resto de los parámetros escribimos la base de datos en la cual se encuentran las tablas que están destinadas para almacenar los campos de información. Si la configuración es correcta, al finalizar aparecerá un cuadro verde con la palabra OK.

Figura 60. **Configuración del nodo MySQL**



Fuente: elaboración propia, empleando Node-RED.

- Envío de datos a Google Drive

El primer bloque es el nodo inyección, el cual cumple con el mismo funcionamiento descrito anteriormente. Los nodos complementarios son:

- Nodo función: el código descrito obtiene cada una de las variables medidas y genera un diccionario que contiene toda la información.

Figura 61. **Programación para generar diccionario http**

```

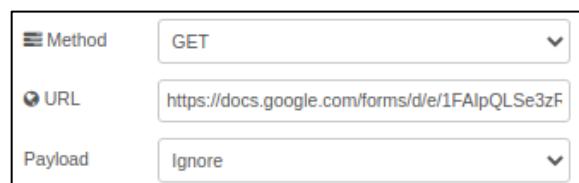
1 mensaje1 = flow.get("T_L1");
2 mensaje2 = flow.get("C_L1");
3 mensaje3 = flow.get("P_L1");
4 mensaje4 = flow.get("PR_L1");
5 mensaje5 = flow.get("PRI_L1");
6 mensaje6 = flow.get("PRC_L1");
7 mensaje7 = flow.get("PA_L1");
8 mensaje8 = flow.get("FP_L1");
9 mensaje9 = flow.get("T_L2");
10 mensaje10 = flow.get("C_L2");
11
12
13 - msg = {
14     payload1:mensaje1,
15     payload2:mensaje2,
16     payload3:mensaje3,
17     payload4:mensaje4,
18     payload5:mensaje5,
19     payload6:mensaje6,
20     payload7:mensaje7,
21     payload8:mensaje8,
22     payload9:mensaje9,
23     payload10:mensaje10
24
25 };
26 return msg;

```

Fuente: elaboración propia, empleando Node-RED.

- Nodo http *request*: este bloque permite enviar información a través de una URL que contiene todos los *payload* como datos de entrada.

Figura 62. **Configuración en nodo http request**



Fuente: elaboración propia, empleando Node-RED.

5.2.3.8. Configuración de base de datos en MariaDB

La creación de bases de datos y tablas conlleva una serie de pasos introduciendo comandos MySQL, por lo cual debemos ingresar a las configuraciones de MariaDB como un usuario administrador.

- Ingresar con el siguiente comando: `sudo mysql -u root -p -h localhost`.
- Crear una nueva base de datos: `create database nombre_base_datos;`
- Crear un usuario y contraseña: `create user 'usuario'@'localhost' identified by 'contraseña';`
- Conceder permisos de administración: `grant all privileges on nombre_base_datos.* to 'usuario'@'localhost';`
- Recargar las configuraciones: `flush privileges;`
- Crear tablas: `create table nombre_tabla (id int not null auto_increment primary key, fecha_hora timestamp default now(), variable_1 tipo_de_dato, variable_2 tipo_de_dato, ... variable_n tipo_de_dato);`

Estas credenciales son las que deben introducirse en los bloques de Node-RED, tanto el nombre de la base de datos, usuario, contraseña, nombre de tabla y las variables tienen que coincidir.

Comandos útiles para gestionar la base de datos: `show database`, `use nombre_base_datos`, `show tables`, `describe nombre_tabla`, `select*from nombre_tabla`.

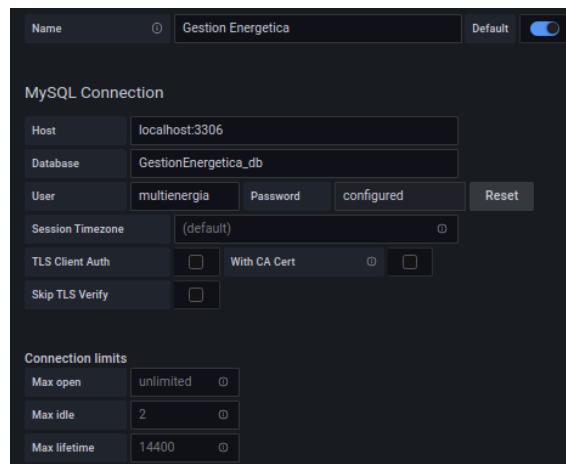
5.2.3.9. Configuraciones en Grafana

Siendo Grafana la aplicación empleada para generar la plataforma de monitoreo, esta requiere de ciertas configuraciones para obtener los datos almacenados y con base a ellos mostrar tableros que contengan gráficos, indicadores, entre otras visualizaciones. Grafana también permite gestionar la parte administrativa para crear las cuentas de los usuarios.

- Vincular base de datos

Accedemos a la configuración de Grafana y seleccionamos la opción para agregar una nueva fuente de datos y escogemos MySQL, en la cual se solicitan todos los parámetros de la base de datos creada en MariaDB. Para verificar que la comunicación se haya establecido correctamente presionamos *Save and Test*, lo que nos devolverá un mensaje de conexión establecida entre la base de datos y Grafana, de lo contrario nos indicará que ha ocurrido un error.

Figura 63. **Configuraciones MySQL en Grafana**

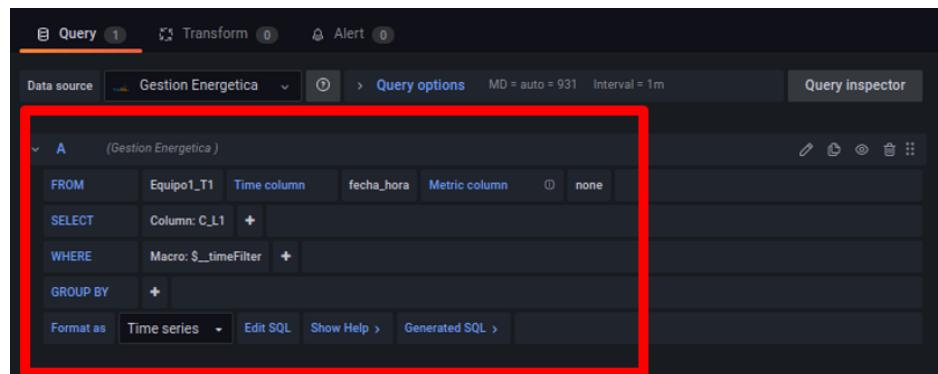


Fuente: elaboración propia, empleando Grafana.

- Creación de tableros y gráficos

Para crear un nuevo *dashboard* nos dirigimos a la opción de crear y seleccionamos iniciar con un panel vacío, en la siguiente ventana debemos especificar cual será la base de datos, tabla y columna de la cual Grafana mostrará una visualización en función a esos datos (es posible mostrar más de una variable en el mismo panel con el fin de observar diferencias gráficas, así como seleccionar datos de distintas fuentes creando una nueva *Query*).

Figura 64. **Selección de datos en panel Grafana**



Fuente: elaboración propia, empleando Grafana.

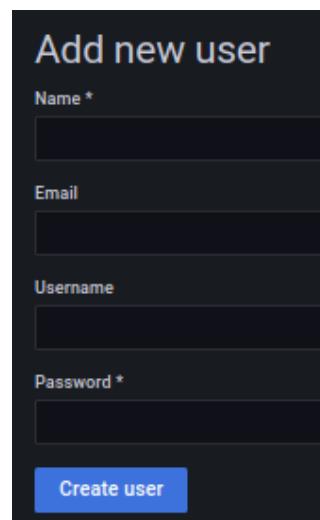
Las configuraciones del panel se encuentran al lado derecho, en esta sección podemos cambiar el tipo de visualización, configurar los títulos, establecer la unidad de la variable, asignar rangos para cada color que identifica el estado del proceso, entre otras funcionalidades propias de cada visualización.

Para guardar las modificaciones realizadas presionamos en el botón *Apply* y *Save*, de esta forma cada panel creado puede ser parte del *dashboard* que contiene múltiples paneles destinados a monitorear procesos en tiempo real.

- Configuración de usuarios

La implementación de usuarios en una plataforma de monitoreo es un requerimiento importante para conceder acceso a distintas personas encargadas de analizar los datos. Para gestionar a los usuarios en Grafana debemos ingresar a la administración del servidor, en esta sección encontramos la opción de crear un nuevo usuario y a continuación editamos los parámetros requeridos.

Figura 65. **Creación de nuevos usuarios en Grafana**



El formulario se titula 'Add new user'. Contiene los siguientes campos: 'Name *' (Nombre), 'Email' (Email), 'Username' (Nombre de usuario) y 'Password *' (Contraseña). A la derecha de cada campo hay un asterisco que indica que es un campo obligatorio. En la parte inferior del formulario hay un botón azul rectangular que dice 'Create user' (Crear usuario).

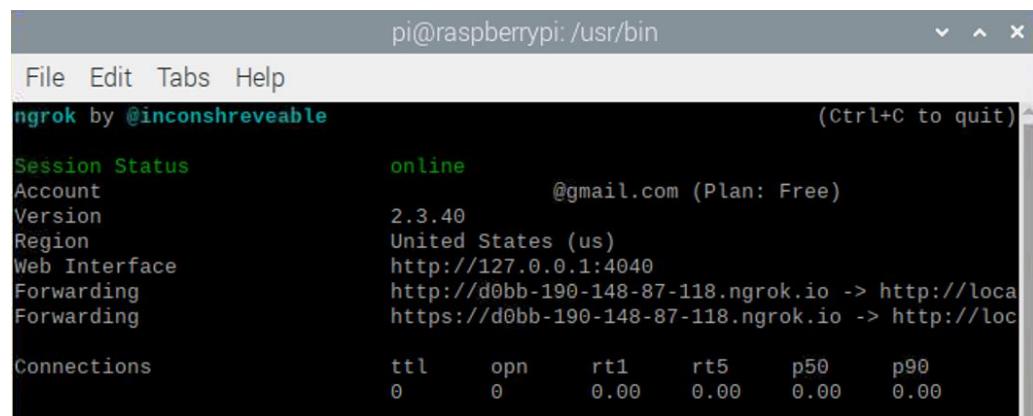
Fuente: elaboración propia, empleando Grafana.

5.2.3.10. Implementación de servidor ngrok

La versión gratuita de ngrok presenta algunas limitaciones de uso, por ejemplo: que solamente permite un proceso en línea con 4 túneles y 40 conexiones por minuto, sin embargo, esto es más que suficiente para comprobar el funcionamiento de la plataforma por medio de una demostración rápida utilizando los túneles de los protocolos HTTP/TCP.

Para iniciar el servicio, ingresamos a la carpeta en la cual esté alojado ngrok y ejecutamos el siguiente comando: `./ngrok.io http número_puerto` que origina la conexión con la aplicación en el puerto especificado, la terminal cambiará mostrando las direcciones URL con las cuales podemos acceder desde otra red. Es importante tener en consideración que, si esta terminal es cerrada entonces la conexión dejará de funcionar y será necesario iniciar una nueva.

Figura 66. **Terminal de servicio ngrok**



A terminal window titled "pi@raspberrypi: /usr/bin" showing the output of the ngrok command. The window has a menu bar with "File", "Edit", "Tabs", and "Help". The title bar says "ngrok by @inconshreveable" and "(Ctrl+C to quit)". The main content shows a table of session status:

Session	Status	Account	Version	Region	Web Interface	Forwarding	Forwarding
	online	@gmail.com (Plan: Free)	2.3.40	United States (us)	http://127.0.0.1:4040	http://d0bb-190-148-87-118.ngrok.io -> http://loc...	https://d0bb-190-148-87-118.ngrok.io -> http://loc...
Connections		ttl	open	rt1	rt5	p50	p90
	0	0	0.00	0.00	0.00	0.00	0.00

Fuente: elaboración propia, empleando ngrok.

5.2.4. **Configuraciones en tarjeta de desarrollo ESP32**

La ESP32 es compatible con múltiples lenguajes de programación y entornos de desarrollo integrado y considerando que Arduino es una de las plataformas más completas respecto a librerías, documentación y aportes realizados por la comunidad que optimizan y presentan soluciones a distintas problemáticas. Por estas razones programar en Arduino resulta ser ideal para desarrollar códigos en proyectos de rápida implementación y así realizar pruebas de funcionamiento en menor tiempo; queda a decisión del programador decidir cuál lenguaje o IDE utilizar.

5.2.4.1. Programación de sensores

El código final comprende la unión de los bloques de programación de cada sensor y funciones, no obstante, para un mejor entendimiento en el código, se describirá la programación independiente para cada bloque.

- Sensor DHT22: la programación de este sensor requiere incluir la librería DHT que permite hacer uso de funciones para crear objetos y lectura de valores. Es importante recordar que el intervalo para solicitar datos al sensor tiene que ser de 2 segundos como mínimo.

Figura 67. Programación de sensor DHT22

```
1 #include <DHT.h>
2 #define DHTTYPE DHT22
3 #define DHTPIN 23
4 DHT dht(DHTPIN, DHTTYPE, 22);
5 unsigned long contador, tiempo2s = 0;
6 void setup() {
7   dht.begin();
8 }
9 void loop() {
10   contador = millis();
11   if(contador - tiempo2s > 2000){
12     tiempo2s = contador;
13     float h = dht.readHumidity();
14     float t = dht.readTemperature();
15 }}
```

Fuente: elaboración propia, empleando Arduino IDE.

Se recomienda colocar un tercer parámetro cuando se crea el objeto, debido a que en algunos casos no es posible establecer la comunicación entre el sensor y la ESP32 si este valor que corresponde al modelo del sensor no es agregado.

- Sensor magnético: la forma en que se utiliza es para detección, por lo cual no requiere de librerías específicas, únicamente el pin seleccionado indicará si el sensor se encuentra en un estado alto o bajo.

Figura 68. **Programación de sensor magnético y conteo de aperturas**

```
1 const int buttonPin = 22;
2 int buttonPushCounter, buttonState, lastButtonState = 0;
3 unsigned long TiempoM, TiempoI, TiempoA, contador, tiempo;
4
5 void setup() {
6   pinMode(buttonPin, INPUT_PULLUP);
7 }
8
9 void loop() {
10   TiempoM = millis();
11   contador = millis();
12
13   if(contador - tiempo > 500){
14     tiempo = contador;
15     buttonState = digitalRead(buttonPin);
16
17     if (buttonState != lastButtonState) {
18       if (buttonState == LOW){
19         buttonPushCounter++;
20
21         while(buttonState == LOW){
22           buttonState = digitalRead(buttonPin);
23           TiempoI = (millis()-TiempoM)/1000;
24           sensorTemp();}
25       }
26       if(buttonState == HIGH) {
27         TiempoA = TiempoI + TiempoA;
28         TiempoI = 0;}
29     }
30     lastButtonState = buttonState;
31   }
32 }
```

Fuente: elaboración propia, empleando Arduino IDE.

El programa descrito anteriormente incluye líneas de código para registrar el conteo de apertura de la puerta, así como el tiempo que esta permanece abierta y el tiempo acumulado de las aperturas totalizadas. El objetivo principal es monitorear el estado de la puerta respecto al uso por parte de los empleados e identificar buenas prácticas a sugerir para la optimización del ingreso a las bodegas, evitando que la temperatura aumente porque la puerta esté abierta.

- Módulo MAX6675: la librería a utilizar es externa a las que incluye Arduino, para utilizarla debemos añadirla manualmente como un archivo ZIP (en la sección de Anexos encuentra el enlace para la descarga de la librería max6675.h). La programación del código requiere definir los tres pines

para establecer la comunicación, es recomendable investigar el *pinout* de la tarjeta de desarrollo a utilizar debido a que algunos pines suelen estar bloqueados o no pueden ser utilizados en determinadas funciones.

Figura 69. **Programación de sensor MAX6675**

```
1 #include "max6675.h"                                //importar la librería del módulo
2 int ktcS0 = 12;                                     //definir el pin serial output
3 int ktcCS = 14;                                     //definir el pin chip select
4 int ktcCLK = 27;                                    //definir el pin clock
5 MAX6675 ktc(ktcCLK, ktcCS, ktcS0);               //Creamos el objeto ktc
6
7 void setup() {                                       //Funciones para leer temperatura
8   Serial.begin(115200);
9   delay(500);
10 }
11
12 void loop() {
13   //Funciones para leer temperatura
14   Serial.println(ktc.readCelsius());
15   Serial.println(ktc.readFarenheit());
16   delay(1000);
17 }
```

Fuente: elaboración propia, empleando Arduino IDE.

5.2.4.2. Programación Arduino – MODBUS

La extensa compatibilidad que ofrece Arduino con gran variedad de módulos permite implementar sistemas complejos con el uso de librerías y bloques de código simples. La comunicación MODBUS/RS485 puede implementarse de dos formas en una tarjeta de desarrollo que soporte Arduino, configurándola en modo maestro o esclavo y en conjunto con el módulo RS485 puede ser parte de una arquitectura que utilice este protocolo.

Para instalar la librería ModbusRtu.h en la sección de Anexos puede encontrar el enlace de descarga. Esta librería incorpora algunos ejemplos de códigos, se recomienda el uso de los siguientes:

- Transmisor (maestro): el *sketch simple_master*, provee un código básico para establecer la comunicación con otros dispositivos MODBUS, depende del sistema a implementar la modificación de algunos parámetros o funciones. Las conexiones del módulo MAX485 son específicas para esta configuración y tipo de comunicación *semiduplex*.

Figura 70. **Arduino sketch simple master MODBUS**

```

17 #include <ModbusRtu.h>
18 uint16_t au16data[16];
19 uint8_t u8state;
20 Modbus master(0,Serial,0); // this is master and RS-232 or USB-FTDI
21 modbus_t telegram;
22 unsigned long u32wait;
23
24 void setup() {
25   Serial.begin( 19200 ); // baud-rate at 19200
26   master.start();
27   master.setTimeout( 2000 ); // if there is no answer in 2000 ms, roll over
28   u32wait = millis() + 1000;
29   u8state = 0;
30 }
31 void loop() {
32   switch( u8state ) {
33   case 0:
34     if (millis() > u32wait) u8state++; // wait state
35     break;
36   case 1:
37     telegram.u8id = 1; // slave address
38     telegram.u8fct = 3; // function code (this one is registers read)
39     telegram.u16RegAdd = 1; // start address in slave
40     telegram.u16CoilsNo = 4; // number of elements (coils or registers) to read
41     telegram.au16reg = au16data; // pointer to a memory array in the Arduino
42     master.query( telegram ); // send query (only once)
43     u8state++;
44     break;
45   case 2:
46     master.poll(); // check incoming messages
47     if (master.getState() == COM_IDLE) {
48       u8state = 0;
49       u32wait = millis() + 100; }
50     break;
51   }

```

Fuente: elaboración propia, empleando Arduino IDE.

- Receptor (esclavo): para esta configuración el *sketch simple_slave* contiene un bloque de código en el cual podemos configurar el vector que contiene los datos a ser enviados a través de distintas funciones.

Figura 71. **Arduino sketch simple slave MODBUS**

```

1 #include <ModbusRtu.h>
2
3 // data array for modbus network sharing
4 uint16_t au16data[16] = {
5   3, 1415, 9265, 4, 2, 7182, 28182, 8, 0, 0, 0, 0, 0, 0, 0, 1, -1 };
6
7 Modbus slave(1,Serial,0); // this is slave @1 and RS-232 or USB-FTDI
8
9 void setup() {
10   Serial.begin( 19200 ); // baud-rate at 19200
11   slave.start();
12 }
13
14 void loop() {
15   slave.poll( au16data, 16 );
16 }
```

Fuente: elaboración propia, empleando Arduino IDE.

5.2.4.3. Programación Arduino – MQTT

La programación utilizada para implementar el protocolo MQTT sobre la ESP32 consta de distintas funciones y bloques de código que en conjunto tienen la capacidad de enviar datos hacia el sistema central. Los bloques en los que fue dividido el programa permiten explicar la funcionalidad de cada segmento de código descritos a continuación:

- Configuraciones iniciales: en esta sección tenemos que incluir las librerías a utilizar, establecer los parámetros de red a la cual se conectará la ESP32 que está configurada en modo cliente y especificar la dirección del servidor o *broker* al que se suscribirá. Las variables dependen del tipo de aplicación a implementar, pudiendo agregar tantas como se requieran.

Figura 72. Programación inicial para ESP32 - MQTT

```
1 #include <WiFi.h>                                //librerias a incluir
2 #include <PubSubClient.h>
3 const char* ssid = "red_wifi";                    //configurar los parametros de red
4 const char* password = "contraseña";
5 const char* mqtt_server = "192.168.0.1";          //definir la IP del servidor
6 char dato_mqtt[5];                                //valor a enviar al broker
7 int dato;                                         //variable a medir
8 String messageData;                             //almacena los mensajes recibidos
9 WiFiClient esp32;                                //crear el objeto esp32
10 PubSubClient client(esp32);                     //definir como cliente
```

Fuente: elaboración propia, empleando Arduino IDE.

- Función para conectarse a la red WiFi: la ESP32 puede conectarse fácilmente con pocas líneas de código a una red WiFi, pasando a las funciones que incluye la librería los parámetros de las credenciales de la red, procedemos a comprobar el estado de la conexión de la ESP32 y cuando se haya establecido, el programa nos devolverá la dirección *IP* que se le haya asignado para identificarla dentro de la red.

Figura 73. Programación para conectar ESP32 a red WiFi

```
2 void config_wifi() {
3   WiFi.begin(ssid,password); //CONECTANDOSE A LA RED
4   while(WiFi.status() != WL_CONNECTED) {
5     delay(500);
6     Serial.print(".");
7   }
8   Serial.println(WiFi.localIP());}
```

Fuente: elaboración propia, empleando Arduino IDE.

- Función receptora de mensajes: la configuración de un cliente suscriptor para recibir mensajes desde el *broker* es recomendada para implementarla en la ESP32, ya que a través de este código podemos controlar

remotamente distintas funcionalidades que estén programadas. La función recibe como parámetros el *topic* y el mensaje enviado desde el servidor.

Figura 74. Programación de cliente receptor en la ESP32

```
26 void callback(String topic, byte* message, unsigned int length){  
27     Serial.print("Mensaje que llega del topic: ");  
28     Serial.print(topic);  
29     Serial.print(". Message: ");  
30  
31     for(int i = 0; i < length; i++){  
32         Serial.print((char)message[i]);  
33         messageData += (char)message[i];  
34     }  
35     if(messageData == "true"){digitalWrite(led,HIGH);}  
36     if(messageData == "false"){digitalWrite(led,LOW);}  
37 }
```

Fuente: elaboración propia, empleando Arduino IDE.

- Función para conectarse al *broker*: la conexión entre la ESP32 y el *broker* es iniciada por esta función, la cual verifica si el cliente ha establecido comunicación con el servidor y de lo contrario se mantiene intentándolo.

Figura 75. Programación para suscribir la ESP32 al broker

```
39 void reconnect(){  
40     while(!client.connected()){  
41         Serial.print("Iniciando conexión con Broker...");  
42         String clientId = "ESP32";  
43         if(client.connect(clientId.c_str())){  
44             Serial.println("Conectado");  
45             client.subscribe("esp32iot"); //topic para suscribir  
46         }  
47     else{  
48         Serial.print("Failed, rc = ");  
49         Serial.print(client.state());  
50         Serial.println(" esperando 3 segundos ...");  
51         delay(3000);  
52     }  
53 }  
54 }
```

Fuente: elaboración propia, empleando Arduino IDE.

- Funciones *setup* y *loop*: la conexión a la red WiFi debe ser ejecutada una vez, por eso es llamada en la función *setup*, en la línea siguiente establecemos el servidor y puerto de escucha especificado al instalar *mosquitto server* en la Raspberry Pi. En la función *loop* verificamos si la conexión con el *broker* ha sido establecida y para enviar los datos por el protocolo MQTT estos deben ser convertidos en una cadena de caracteres e indicar cuál será el *topic* destino.

Figura 76. **Programación para enviar datos desde la ESP32**

```

56 void setup() {
57   Serial.begin(115200);
58   config_wifi();
59   client.setServer(mqtt_server,1883);
60   client.setCallback(callback);
61 }
62
63 void loop() {
64   if(!client.connected()){reconnect();}
65   client.loop();
66
67   sprintf(dato_mqtt,"%3f",dato);
68   client.publish("topic/subtopic", dato_mqtt);
69 }

```

Fuente: elaboración propia, empleando Arduino IDE.

5.2.4.4. Programación Arduino – OTA

Al incluir los bloques de programación OTA en el código se facilita la reprogramación sin tener que conectarse con el cable USB, el único requerimiento es conectarse a la misma red WiFi desde el ordenador que enviará el código a través de un puerto de red. Es recomendable seccionar el código de nuestra aplicación del código OTA con el fin de no sobrecargar de líneas y funciones solamente una página, para lograr esto creamos nuevas pestañas que luego incorporamos al código principal.

- Bloque principal: esta pestaña incorpora las funciones *setup* y *loop*, por lo cual debemos instanciar el resto de pestañas en esta parte y programar el código que contiene las instrucciones de nuestra aplicación, de esa forma todas las funciones programadas estarán disponibles para ser llamadas desde cualquier línea en este bloque. Es importante mencionar que la primera carga del programa si requiere del cable USB para conexión.

Figura 77. **Pestaña principal en programación OTA**

```

OTA § ESP32_Utils.hpp § ESP32_Utils_OTA.hpp § config.h §
1 #include <WiFi.h>
2 #include <ArduinoOTA.h>
3 #include "ESP32_Utils.hpp"
4 #include "ESP32_Utils_OTA.hpp"
5
6 const char* ssid = "red_wifi";
7 const char* password = "contraseña";
8
9 void setup(){
10   Serial.begin(115200);
11   ConnectWiFi_STA();
12   InitOTA();
13 }
14
15 void loop(){
16   ArduinoOTA.handle();
17 }
18 }

```

Fuente: elaboración propia, empleando Arduino IDE.

- Conexiones a la red WiFi: la base para la reprogramación OTA es una conexión WiFi, ya que a través de la red es posible enviar el código actualizado. A continuación, se describe el código de dos funciones para conectarse a la red WiFi; la selección de una de estas dos queda a discreción del programador y del tipo de aplicación a implementar.

Figura 78. Código para conexión WiFi en OTA



```
OTA.h  ESP32_Utils.hpp  ESP32_Utils_OTA.hpp  config.h

1 void ConnectWiFi_STA(bool useStaticIP = false)
2 {
3     Serial.println("");
4     WiFi.mode(WIFI_STA);
5     WiFi.begin(ssid, password);
6     if(useStaticIP) WiFi.config(ip, gateway, subnet);
7     while (WiFi.status() != WL_CONNECTED)
8     {
9         delay(100);
10        Serial.print(".");
11    }
12
13    Serial.println("");
14    Serial.print("Iniciado STA:\t");
15    Serial.println(ssid);
16    Serial.print("IP address:\t");
17    Serial.println(WiFi.localIP());
18 }
19 void ConnectWiFi_AP(bool useStaticIP = false)
20 {
21     Serial.println("");
22     WiFi.mode(WIFI_AP);
23     while(!WiFi.softAP(ssid, password))
24     {
25         Serial.println(".");
26         delay(100);
27     }
28     if(useStaticIP) WiFi.softAPConfig(ip, gateway, subnet);
29
30     Serial.println("");
31     Serial.print("Iniciado AP:\t");
32     Serial.println(ssid);
33     Serial.print("IP address:\t");
34     Serial.println(WiFi.softAPIP());
35 }
```

Fuente: elaboración propia, empleando Arduino IDE.

- Funciones Arduino OTA: la librería OTA está incluida por defecto en la instalación de Arduino IDE, por lo cual incluye varios ejemplos de códigos para realizar pruebas de conexión. Es importante considerar que cuando se reprograme un nuevo código, este debe incluir el bloque de programación OTA de lo contrario se perderá la conexión.

Figura 79. Código de funciones OTA



```
OTA § ESP32_Utils.hpp § ESP32_Utils_OTA.hpp § config.h
1 void InitOTA() {
2   ArduinoOTA.onStart([]() {
3     String type;
4     if (ArduinoOTA.getCommand() == U_FLASH) {
5       type = "sketch";
6     } else { // U_SPIFFS
7       type = "filesystem";
8     }
9     // NOTE: if updating SPIFFS this would be the place to unmount SPIFFS using SPIFFS.end()
10    Serial.println("Start updating " + type);
11  });
12 ArduinoOTA.onEnd([]() {
13   Serial.println("\nEnd");
14 });
15 ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
16   Serial.printf("Progress: %u%\r", (progress / (total / 100)));
17 });
18 ArduinoOTA.onError([](ota_error_t error) {
19   Serial.printf("Error[%u]: ", error);
20   if (error == OTA_AUTH_ERROR) {
21     Serial.println("Auth Failed");
22   } else if (error == OTA_BEGIN_ERROR) {
23     Serial.println("Begin Failed");
24   } else if (error == OTA_CONNECT_ERROR) {
25     Serial.println("Connect Failed");
26   } else if (error == OTA_RECEIVE_ERROR) {
27     Serial.println("Receive Failed");
28   } else if (error == OTA_END_ERROR) {
29     Serial.println("End Failed");
30   }
31 });
32 ArduinoOTA.begin();
33 Serial.println("");
34 Serial.println("OTA iniciado");
35 }
```

Fuente: elaboración propia, empleando Arduino IDE.

6. FASE DE INVESTIGACIÓN, PLAN DE CONTINGENCIA CONTRA POSIBLES FALLAS O BAJAS DEL SISTEMA

6.1. Pérdidas de energía eléctrica en los dispositivos

Los cortes de energía comúnmente se originan por fallas en la red de distribución comercial que, por lo general, se producen en caídas de tensión, apagones o fallas permanentes. Estos factores entorpecen el funcionamiento de muchos dispositivos, equipos y maquinaria utilizado en la industria, debido a que al perder el suministro de energía de forma repentina los componentes internos tienden a deteriorarse más rápido.

La alimentación de los dispositivos utilizados en el sistema de monitoreo depende totalmente de la red de energía eléctrica, por lo cual, cuando sucede una falla estos se reinician y cierran todos los procesos que estaban ejecutando, generándose la probabilidad de que las configuraciones establecidas se hayan perdido o que algún programa no reinicie sus operaciones. Para mitigar esta problemática debe configurarse en cada proceso una serie de instrucciones que actúan cada vez que el dispositivo se enciende manualmente o por reinicio forzado, ordenando a cada uno de los procesos reiniciar sus actividades automáticamente evitamos tener que reanudar el sistema de forma manual.

Adicionalmente se puede añadir un UPS, el cual además de proteger a los dispositivos de las fallas en la red como picos o caídas de tensión, puede suministrar energía por un periodo prolongado de tiempo mientras se restablece el servicio de la red comercial. Las baterías son de utilidad cuando se requiere un monitoreo permanente sin pérdida de datos en los dispositivos.

6.2. Desconexión entre la red de internet y los dispositivos

Los proveedores de servicios de internet brindan una conexión para que los dispositivos puedan conectarse a internet, sin embargo, existen diversos factores por los cuales puede fallar la comunicación con la red y esto repercute en el mal funcionamiento del sistema. Algunos de estos problemas se dan a causa de un limitado ancho de banda contratado, cuando se alcanza el límite de un servicio de plan de datos prepago, debido a una caída de la red por parte del servidor o por una falla de comunicación directamente entre los dispositivos.

Para evitar perder los datos monitoreados, se plantea adecuar una alternativa con un sistema local que tenga la capacidad de almacenar información hasta que se restablezca el servicio de internet o que un técnico verifique y resuelva el problema, con el fin de que toda esa información almacenada de forma local por el dispositivo sea enviada posteriormente cuando la comunicación con el internet haya sido restaurada. Se recomienda lo siguiente:

La solución en la Raspberry Pi se presenta al instalar la base de datos MariaDB, ya que Node-RED no necesita de una conexión a internet y puede continuar almacenando la información en la base de datos.

La ESP32 puede ser adecuada en distintas situaciones para no perder la comunicación con la Raspberry Pi: la opción a implementar es una comunicación Bluetooth en el caso de que ambos dispositivos no se encuentren a una distancia superior del alcance permitido, por lo contrario, si los dispositivos se encuentran demasiado alejados uno del otro puede recurrir a agregar un módulo y memoria SD en la ESP32, lo que permite almacenar todos los datos censados en un archivo que será enviado posteriormente a la Raspberry Pi para actualizar la información del proceso monitoreado.

6.3. Sistema de ciberseguridad

Un sistema de ciberseguridad es implementado para defender a los ordenadores, servidores, dispositivos móviles, sistemas electrónicos, las redes y los datos de ataques maliciosos. Los sistemas de seguridad en informática pueden contener un alto grado de complejidad según el tipo de aplicación que se requiera escudar, en este caso los datos de monitoreo, el acceso a los dispositivos y servicios en la *web* son los puntos de importancia por proteger.

El envío de datos está relacionado directamente con el protocolo de red utilizado, para MQTT la seguridad que incluye es el transporte SSL/TLS y autenticación por usuario y contraseña o mediante certificado, no obstante, el implementar estas medidas significa una carga adicional al procesador del dispositivo. Es importante considerar el tipo de datos que estamos enviando y en base a esto decidir el nivel de seguridad necesario para evitar riesgos sin tener un impacto negativo en la eficiencia del sistema.

La autenticación por usuario y contraseña también puede ser implementada en los dispositivos, de esta forma nos aseguramos de que únicamente las personas con estas credenciales puedan acceder a las configuraciones. Una práctica común es cambiar el número de puerto por defecto en los servicios expuestos en la red o en casos que requieran mayor seguridad puede optar a generar certificados de seguridad para los servicios *web* ofrecidos a usuarios.

6.4. Mantenimiento preventivo a los dispositivos

Considerando el desgaste que sufren los dispositivos a través del tiempo y las condiciones ambientales del lugar en el que fueron instalados, el mantenimiento preventivo es un servicio por realizar en un periodo de por lo

menos dos veces al año, en el caso de que algún dispositivo o servicio no requiera de un mantenimiento correctivo previo a las fechas estipuladas, de lo contrario el sistema podrá garantizar un funcionamiento adecuado.

Las visitas a realizar comprenden una revisión exhaustiva de todos los dispositivos que conforman la topología, desde medir el nivel de voltaje suministrado por las fuentes de alimentación, verificar el funcionamiento de software en la ESP32 y en la Raspberry Pi, así como limpiar las terminales de conexión y liberar de partículas de polvo los conductos de circulación de aire en el *case*; sin olvidar comprobar que los datos enviados por los sensores sean correctos y que el cableado aún se mantenga en buenas condiciones.

Aplicando estas recomendaciones, podemos evitar fallas de bajas completas en el sistema, detectar si los dispositivos están funcionando óptimamente o si presentan algún desperfecto que requiera ser calibrado, reparado o sustituir el dispositivo por uno nuevo.

6.5. Condiciones ideales para la instalación de los dispositivos

La instalación de dispositivos en sectores industriales conlleva adecuar sistemas para mitigar las condiciones del lugar, por estas razones se recomienda implementar un gabinete o *case* en el cual se introduzcan las placas electrónicas para proteger del polvo, agua, luz solar y de otros factores que puedan dañarlas.

Los dispositivos que realizan procesamiento de datos tienden a elevar la temperatura de algunos componentes, por lo cual se les incorpora un disipador de aluminio para liberar de forma eficiente el calor transferido. De modo complementario, un sistema de ventiladores impide que la temperatura interna se eleve provocando sobrecalentamiento en los procesadores; la implementación

de este sistema es recomendado solamente para dispositivos que consumen muchos recursos de procesamiento o que realizan operaciones complejas, de esta forma se evita el deterioro de los componentes internos de las tarjetas electrónicas.

Figura 80. **Sistema disipador de calor en Raspberry Pi**



Fuente: Labists. Raspberry Pi 4 8GB RAM Starter Kit. Consultado el 15 de enero de 2022. Recuperado de <https://labists.com/products/raspberry-pi-4-8gb-ram-starter-kit-with-64gb>.

7. FASE DE ENSEÑANZA APRENDIZAJE

7.1. Demostración del funcionamiento del sistema al departamento de la empresa

La demostración del alcance que ha tenido el proyecto instalado en el punto de medición fue presentada a los gerentes de la empresa por medio de una reunión virtual en Microsoft Teams. En esta presentación se explicó la iniciativa de implementar este tipo de sistemas de monitoreo, a la vez se dieron a conocer cuáles fueron los dispositivos utilizados que conforman la arquitectura y las aplicaciones con los diversos servicios implementados. El informe de avances presentado en la reunión comprende la metodología del desarrollo del proyecto y la incorporación de los resultados obtenidos en una plataforma integral y escalable, a la cual los participantes tuvieron acceso con un usuario personal para observar el funcionamiento de este servicio desde dispositivos propios.

Figura 81. Presentación del sistema de monitoreo a la empresa



Fuente: elaboración propia, empleando Canvas.

El objetivo principal estaba enfocado en mostrar las bases para evaluar la posibilidad de desarrollar un servicio corporativo de este tipo y exhibir las soluciones que el IoT y la Industria 4.0 ofrecen al sector empresarial, asimismo se dieron a conocer algunas sugerencias para mejorar el sistema ya existente.

7.2. Propuestas para optimizar el consumo energético

Los datos obtenidos deben estudiarse para poder generar reportes que incluyan propuestas de ahorro energético o que muestren indicadores del comportamiento de los equipos monitoreados, con el propósito de conocer si la energía consumida es aprovechada de forma adecuada o si es necesario implementar otro sistema que optimice estas condiciones.

Considerando que a nivel industrial la carga más utilizada para distintos procesos es el motor eléctrico, a continuación se presentan algunas recomendaciones de los cálculos empleados para corregir problemas de un bajo factor de potencia, así como los dispositivos que optimizan el funcionamiento en los motores y por ende conlleva a ahorros en los pagos o a evitar multas por exceso de consumo o penalizaciones debido a no hacer un uso eficiente de la energía suministrada por la empresa eléctrica.

7.2.1. Cálculos para banco de capacitores

El sistema de monitoreo de energía tiene la capacidad para medir otras variables como: la potencia reactiva y el factor de potencia; realizando un análisis de estos datos se obtienen indicadores para conocer la eficiencia de los equipos monitoreados y en base a los resultados decidir si es conveniente agregar un banco de capacitores que suele ser la solución más común y utilizada en estos casos, sin embargo, solamente es aplicable en cargas lineales.

Obteniendo por medio del analizador de energía el valor de la potencia reactiva y del factor de potencia del sistema a estudiar, procedemos a sustituir los valores en la fórmula para realizar el cálculo del banco de capacitores.

Figura 82. **Fórmula de banco de capacitores**

Fórmula:	$C = \frac{Q}{2\pi f V^2}$	Donde:	C: capacitancia en faradios
			Q: potencia reactiva en VAR
			f: frecuencia de la red en Hertz
			V: tensión de la red en Voltios

Fuente: elaboración propia, empleando Word.

Para comprender de mejor forma la aplicación de la fórmula, se presenta a continuación un ejemplo práctico: asumiendo algunos datos de un sistema lineal que opera con un motor en vacío conectado a una red de 120 V a 60 Hz:

Figura 83. **Ejemplo de cálculo para banco de capacitores**

Datos:	Q = 542.8 VAR
	f = 60 Hz
	V = 120 V
Sustituyendo en la fórmula:	
$C = \frac{542.8}{2\pi * 60 * 120^2} = 99.98 \mu F$	

Fuente: elaboración propia, empleando Word.

Este análisis parte de la detección de un factor de potencia bajo que implica una potencia reactiva alta, siendo estos valores los que la empresa eléctrica

penaliza con multas a los clientes. Para mitigar este impacto en los sistemas, se requiere implementar el banco de capacitores que aumenta el factor de potencia entre valores de 0.9 a 1 lo cual disminuye la potencia reactiva.

7.2.2. Dispositivos controladores de potencia

Estos son dispositivos adicionales que pueden combinarse con los equipos que realizan los procesos industriales para la optimización del uso de la energía eléctrica. Los dispositivos o sistemas propuestos ajustan determinados parámetros de funcionamiento en el circuito eléctrico para evitar los transientes o la generación de otras alteraciones en la red.

- Banco de capacitores: es un grupo de capacitores conectados entre ellos dependiendo del valor de capacitancia requerida. Estos equipos regularmente son aplicados a sistemas eléctricos de baja, mediana y alta tensión con la finalidad de corregir el factor de potencia, mejorar el perfil de voltaje y en forma de complemento en filtros de armónicos; mayormente son utilizados en sistemas de arranque de motores o conexiones a cargas de alto consumo. Además, reducen las corrientes armónicas en la red, evitan pérdidas eléctricas y desgaste en los equipos.

Figura 84. **Capacitor industrial**



Fuente: Ceica. Capacitores. Consultado el 17 de enero de 2022. Recuperado de <https://www.ceica.com/product/capacitores/>.

- Arrancadores suaves: son utilizados para controlar generalmente motores trifásicos modificando el arranque y paro de una forma suave y lenta. Un motor de inducción interna tiende a demandar un consumo en el rango de 3 a 15 veces la corriente nominal en un arranque normal, lo cual, a largo plazo, daña el embobinado propio del motor y a la red de alimentación. Los arrancadores suaves controlan el voltaje y corriente que consume el motor hasta alcanzar el funcionamiento ideal, optimizando el par del motor se evita el pico de corriente en la demanda de arranque, con esto se logra aumentar la vida útil del motor y ahorros en el consumo de energía.

Figura 85. **Arrancador suave marca SIEMENS**

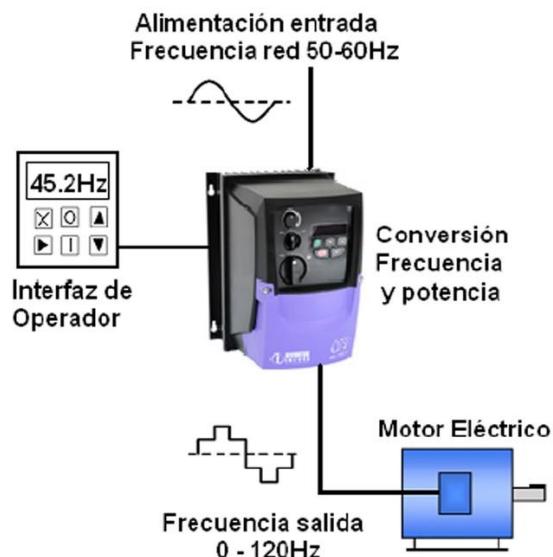


Fuente: NIVIHE. Arrancadores Suaves. Consultado el 17 de enero de 2022. Recuperado de <https://motores-electricos.com.ar/catalogo/arrancadores-suaves-siemens/arrancador-suave-sirius-3rw30-siemens/>.

- Variadores de frecuencia: son utilizados para regular la velocidad rotacional en motores de corriente alterna, esto es posible mediante la variación de frecuencia de la red eléctrica que alimenta al motor, de esta

forma se evita la pérdida de energía en forma de calor disipado; logrando optimizar el consumo eléctrico hasta en un 50 %, simplemente con la reducción del 20 % de la velocidad comparada con un arranque directo convencional. Otras de las ventajas son que alarga la vida útil del motor, disminuye el ruido generado por la fricción y permite recuperar la inversión en un periodo de 3 a 36 meses por la compra del dispositivo.

Figura 86. **Variador de frecuencia**



Fuente: ESCODA, Salvador. Variadores de Frecuencia reguladores de velocidad. Consultado el 17 de enero de 2022. Recuperado de <https://elblogdelinstalador.com/variadores-de-frecuencia-reguladores-de-velocidad-otra-de-alternativa-para-el-ahorro-energetico/>.

7.3. **Conferencias en los cursos de Electrónica 5 y 6, en la Facultad de Ingeniería USAC**

La comunidad estudiantil es uno de los pilares más importantes para el desarrollo del país, por lo cual se considera fundamental exponer las nuevas

tecnologías a los estudiantes para que ellos tengan el conocimiento y profundicen con su interés propio en los múltiples campos que aplican el IoT.

Estas presentaciones se impartieron en las respectivas salas virtuales en Google Meet de cada curso y para estas conferencias el enfoque se mantuvo orientado en la conectividad que el IoT proporciona a diversos sistemas en la industria y a los conocimientos básicos que debe poseer un desarrollador de este tipo de proyectos. Para concretar la explicación, se expuso un resumen acerca de la implementación del sistema de monitoreo desde sus inicios, su funcionamiento en la actualidad y proyecciones a futuro.

Figura 87. Presentación del sistema de monitoreo a la comunidad estudiantil



Fuente: elaboración propia, empleando Canvas.

8. ANÁLISIS DE RESULTADOS

Las funcionalidades de un sistema de monitoreo basado en el IoT y la reciente industria 4.0 comprenden una solución para las empresas que requieren conocer a profundidad los procesos industriales realizados y encontrar la forma de cómo mejorarlos, ya que al destinar recursos en este tipo de proyectos se espera a futuro valores que retribuyan los costos de implementación.

En esta sección se exponen los resultados obtenidos, describiendo precios y realizando comparaciones se entienden los beneficios de una cultura de mejoras continuas en base al monitoreo de procesos y análisis de datos.

8.1. Análisis financiero

Con el propósito de tener una expectativa del costo de dispositivos, pago de plataformas y la instalación en el sitio destinado a implementar un sistema de monitoreo, se presentan a continuación los presupuestos aproximados que pueden ser tomados como referencia para replicar este sistema a mayor escala con distintos puntos a monitorear y tener un parámetro importante en cálculos de inversión y tiempos de retorno en función a los ahorros proyectados.

Tabla XIV. Presupuesto de *hardware* para medición en un punto

Dispositivo	Valor	Cantidad	Costo
SISTEMA DE MONITOREO DE ENERGIA			
Kit Raspberry Pi 4 modelo B	Q 1 116,25	1	Q 1 116,25
Conversor RS485 a USB	Q 20,00	1	Q 20,00
Analizador de energía	Q 1 500,00	1	Q 1 500,00

Continuación de la tabla XIV.

Transformador de corriente	Q 400,00	3	Q 1 200,00
SISTEMA DE MONITOREO DE CAMARA FRIA			
ESP32	Q 125,00	1	Q 125,00
Sensores	Q 79,00	2	Q 158,00
PCB	Q 75,00	1	Q 75,00
Impresión 3D	Q 50,00	1	Q 50,00
SISTEMA DE MONITOREO DE ALTA TEMPERATURA			
ESP32	Q 125,00	1	Q 125,00
Módulo MAX6675	Q 64,00	4	Q 256,00
Termocupla	Q 125,00	4	Q 500,00
Impresión 3D	Q 50,00	1	Q 50,00
PCB	Q 75,00	1	Q 75,00
OTROS DISPOSITIVOS			
Cableado	Q 200,00	1	Q 200,00
Componentes extras	Q 100,00	1	Q 100,00
TOTAL =			Q 5 550,25

Fuente: elaboración propia, empleando Excel.

Los costos por dispositivos que conforman cada sistema pueden cambiar si se requiere monitorear más o menos procesos a la vez. Para el sistema de monitoreo de energía se plantean componentes para realizar la medición en un tablero o en un circuito específico, por lo que, si la aplicación lo requiere, debe presupuestar más analizadores y transformadores de corriente.

Es notable la diferencia en el precio del analizador de energía respecto a los demás componentes, por lo tanto, es recomendable considerar la reutilización de analizadores si la empresa ya dispone de estos, así también para los transformadores de corriente. De esta forma simplemente se realizan cambios de ubicación e instalación en los nuevos puntos a medir.

Tabla XV. **Presupuesto de servicios y software de monitoreo**

Descripción	Valor	Cantidad	Costo
Plan de datos mensual	Q 100,00	6	Q 600,00
PLATAFORMAS (plan gratuito)			
Grafana	Sin costo		
remote.it	Sin costo		
ngrok	Sin costo		
PLATAFORMAS (plan de pago mensual)			
Grafana Enterprise	Configurable		
Grafana Cloud	Q 378,00	1	Q 378,00
remote.it (professional)	Q 47,00	1	Q 47,00
ngrok (basic)	Q 39,00	1	Q 39,00

Fuente: elaboración propia, empleando Excel.

Para la contratación de servicios de plan de datos se sugiere realizar un análisis previo del consumo de datos por cada dispositivo que disponga de una conexión a internet, con el fin de poder ajustar de mejor forma un plan prepago de datos o una óptima conexión WiFi en el lugar.

Las plataformas disponen de funcionalidades para brindar al cliente dependiendo del tipo de plan contratado, en este caso no se hizo uso de servicios de paga ya que el proyecto constaba de un plan piloto para evaluar la viabilidad al utilizar estos servicios, sin embargo, se consideran los planes de pago mensual como referencia a la escalabilidad que puede precisar mayores funciones de las que una versión gratuita otorga.

A la vez, estos datos sirven como factores comparativos con otras plataformas que proveen servicios similares como: Azure IoT con Power BI, servicios IoT en AWS, IBM Wattson IoT Platform, Cayenne IoT entre otras.

Tabla XVI. **Presupuesto de instalación en un punto**

Descripción	Valor	Cantidad	Costo
Instalación	Q 2 500,00	1	Q 2 500,00
Viáticos	Q 1 500,00	1	Q 1 500,00
Desarrollo de <i>software</i>	Q 5 000,00	1	Q 5 000,00
TOTAL =			Q 9 000,00

Fuente: elaboración propia, empleando Excel.

La instalación de la solución en un punto comprende el desarrollo de programación, la puesta en marcha de todos los sistemas de monitoreo, las pruebas de funcionamiento, interconectividad entre los equipos y comunicaciones remotas; así como el funcionamiento de la plataforma y la integración de usuarios.

8.1.1. Comparación con principales alternativas

Los datos anteriores son indicadores que pueden ser utilizados para comparar con otras soluciones alternativas existentes en el mercado de las plataformas ya constituidas como servicios o si requiere contratar otro tipo de *software* o *hardware* para un desarrollo propio.

Es preciso enfatizar en el hecho de que, en las herramientas portátiles utilizadas para realizar auditorías energéticas no existe alguna con capacidades para analizar en conjunto las mediciones eléctricas y variables ambientales en una instalación, en contraste con la forma en la que se ha descrito e implementado en este trabajo. Por otro lado, muchos fabricantes y empresas de metrología ofrecen equipos y servicios de monitoreo, no obstante, algunos se

encuentran limitados por las conexiones permitidas únicamente entre dispositivos de la misma marca, lo cual restringe la incorporación de nuevos dispositivos de medición o control dentro del sistema central, además de poca flexibilidad en la plataforma de monitoreo.

Tabla XVII. Comparación de alternativas para sistemas de monitoreo

	Monitoreo basado en IoT	Instrumentos portátiles	Empresa de monitoreo
Fuentes de adquisición de datos	*Entradas digitales y analógicas. *Multiprotocolo. *Conectividad con variedad de sensores y módulos.	*Sensores. *Multímetros. *Analizador de redes. *Cámaras infrarrojas.	*Algunos protocolos. *Entradas limitadas. *Conectividad con sensores ya definidos.
Capacidad de registro y visualización	SI	SI	SI
Acceso remoto y mediciones en tiempo real	SI	NO	SI
Notificaciones de eventos (alertas, mensajes, etc.)	SI	NO	SI
Complejidad en implementación	Moderada	Sencilla	Realizada por la empresa.
Observaciones	*Amplia flexibilidad para incorporar nuevos sistemas. *Requiere tiempo de desarrollo.	*Escasa compatibilidad con otros equipos *Dispositivos embebidos.	*Corto tiempo de implementación *Dependencia para agregar nuevos sistemas.
Precio	Depende de la solución a implementar.	Precio único por compra.	Precio por contrato.

Fuente: elaboración propia, empleando Word.

Realizando una comparativa entre estas opciones y teniendo presente las diferencias entre sus capacidades, la decisión dependerá si la empresa que requiere implementar el sistema dispone de un departamento para el desarrollo de un sistema propio que puede ser adecuado a los requerimientos presentes y futuros o por lo contrario tendrá que depender de una compañía tercera para ejecutar dicho proyecto. Ambas opciones presentan similitudes en su funcionamiento, sin embargo, la diferencia principal radica en que una solución IoT conlleva al principio un tiempo para su desarrollo y planteamiento de los sistemas, mientras que al contratar un servicio de monitoreo la solución es casi inmediata respecto a tiempos de instalación y puesta en marcha de la plataforma.

Se considera viable el desarrollo de un sistema con *software* y *hardware* que se ajuste en su totalidad al tipo de solución a la cual se quiere llegar, de esta forma la escalabilidad del sistema, la implementación de nuevos dispositivos; añadir, modificar o quitar opciones en la plataforma de monitoreo serán responsabilidades por cumplir de acuerdo con requerimientos a largo plazo que surjan en decisiones corporativas.

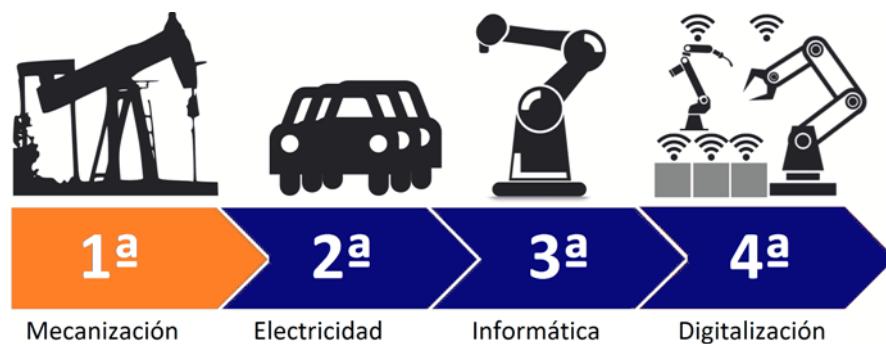
8.2. La Industria 4.0 y sus nuevas posibilidades

A lo largo del tiempo, la industria ha atravesado una serie de cambios debido a la innovación de la tecnología aplicada a los procesos que han beneficiado desde la primera revolución industrial ha aumentado la productividad y disminuido los costos de producción, y a partir de la implementación de dispositivos electrónicos y protocolos básicos de comunicación se ha hecho posible automatizar los procesos, incrementando aún más la productividad. Esto hace referencia a que, con cada avance de la tecnología, la industria evoluciona al emplear nuevas metodologías que ayudan a optimizar los recursos disponibles en una empresa.

En la actualidad, la cuarta revolución industrial es identificada por integrar la tecnología informática en los procesos de manufactura, con el fin de llevar a cabo la digitalización industrial, alcanzando mayores índices de flexibilidad, eficiencia y mejoras en los costos de producción. Esta revolución incorpora en la industria a los mayores desarrollos tecnológicos respecto a la conectividad de equipos, sistemas y personas; que son interconectados por medio del Internet de las Cosas y la amplia gama de dispositivos que facilitan la adquisición de datos y la interacción hombre-máquina.

De acuerdo con Forrester Big Data, en su artículo acerca del uso de los datos: las empresas basadas en datos, que aprovechan los conocimientos de toda su organización y los implementan están creciendo a un promedio de más del 30 % anual y están en camino de ganar 1 800 millones de dólares para 2021.⁴² Esto deja en evidencia la importancia de los datos para una empresa, dando lugar a la transformación digital de su modelo de negocios obteniendo relaciones aún más sostenibles y sinergias entre las distintas áreas que la conforman.

Figura 88. **Revoluciones industriales**



Fuente: Economipedia. Revolución Industrial. Consultado el 18 de enero de 2022. Recuperado de <https://economipedia.com/definiciones/primera-revolucion-industrial.html>.

⁴² FORRESTER. *Blogs & Podcasts*. Consultado el 18 de enero de 2022.

8.3. Aplicación IoT para la supervisión energética en ambientes industriales

El tiempo empleado en la realización del Ejercicio Profesional Supervisado ha consistido en el diseño, implementación, pruebas y desarrollo de las propiedades de un sistema basado en el IoT, el cual utiliza dispositivos tecnológicos de bajo coste destinados a formar parte de una aplicación de monitoreo energético ambiental en un sector industrializado.

Las actividades desempeñadas en la empresa pueden resumirse en: la instalación, configuración y programación de la Raspberry Pi 4 y ESP32, a la vez la integración de la plataforma de monitoreo constituida por diverso software, protocolos de comunicación y lenguajes de programación; estas dan origen a la combinación de dispositivos con funcionalidades en una aplicación IoT dedicada a la supervisión de distintos procesos monitoreados en una instalación.

El objetivo primordial de esta aplicación es permitir a las personas encargadas de las auditorias energéticas extraer las mediciones de un punto monitoreado de forma remota y posteriormente realizar el análisis de los datos obtenidos para la recomendación de buenas prácticas y medidas de ahorro.

Un sistema de monitoreo remoto se logra a partir de la conectividad que provee el internet y utilizando un *router* como distribuidor de señal WiFi, a la cual la Raspberry Pi se conecta y configura como el puente entre los analizadores de energía que no poseen una conexión directa a redes exteriores y, de esta forma, obtener las mediciones de consumos de potencia y energía. En complemento a este sistema la ESP32 proporciona la lectura y envío de datos de los sensores de temperatura, humedad y estado de puertas; formando parte de la misma arquitectura local con la capacidad de expandirse a nuevos procesos.

La información almacenada es la fuente de datos que utilizan las aplicaciones que se ejecutan en la Raspberry Pi para generar tableros de visualización gráfica del historial de las mediciones realizadas, además dispone de una conexión con hojas de cálculo en Google Drive para realizar análisis y gráficos propios.

Adicionalmente la Raspberry Pi dispone de otros servicios con los cuales el acceso remoto y compartir aplicaciones es posible para el ingreso de los usuarios desde un enlace *web* accediendo desde cualquier dispositivo conectado al internet. Con estos logros y resultados obtenidos es notorio el tipo de soluciones que se pueden obtener al aplicar los conceptos del IoT.

CONCLUSIONES

1. La aplicación del Internet de las Cosas en conjunto con dispositivos como la Raspberry Pi y ESP32 de bajo coste y alta flexibilidad de programación hacen posible el desarrollo de distintos sistemas de monitoreo capaces de operar en ambientes industriales
2. Los dispositivos con los que la empresa ya disponía fueron integrados a la arquitectura del sistema de monitoreo energético interconectado por medio de la Raspberry Pi y protocolos de comunicación.
3. Se analizaron los procesos realizados en la empresa, determinando que es factible implementar sistemas de monitoreo ambientales empleando la ESP32 para la lectura y envío de datos hacia un servidor central configurado en la Raspberry Pi.
4. La Raspberry Pi provee diversas ventajas en la capacidad de integrar múltiples aplicaciones IoT y facilidad en la conectividad a la red para dispositivos simples, además de que ejecuta servicios vinculados en la nube. Tomando en cuenta que este desarrollo debe ser realizado por una persona con conocimientos en el sistema operativo y el resto de los servicios y programas, es el único desafío encontrado para este sistema.
5. La plataforma de monitoreo representa una solución en la empresa con la cual es posible realizar análisis de los procesos monitoreados, y respecto a la optimización del consumo de energía fueron propuestos algunos de los dispositivos controladores de potencia que mejoran este factor.

RECOMENDACIONES

1. Fabricar manualmente las placas de circuito impresas es una buena práctica para asegurar las conexiones de buses entre dispositivos y sensores, no obstante, la fabricación industrial de PCB optimiza el diseño del circuito y mejora la presentación de este.
2. Recordar que el sistema de monitoreo central implementado consiste en un servidor local, sin embargo, es posible gestionar un servidor en la nube con la creación de una máquina virtual para establecer la comunicación remota con el resto de los subsistemas de monitoreo.
3. Tener en cuenta que los sistemas y servicios utilizados están ampliamente documentados en páginas *web* oficiales de los fabricantes y proveedores, por lo que son adecuados para el desarrollo de sistemas de monitoreo, sin olvidar que actualmente existen otras alternativas de servicios *web* por los que puede optar, así como otro tipo de dispositivos.
4. Evaluar que el sistema presentado tiene la capacidad para ser escalable e ir más allá, incluyendo nuevas funcionalidades, opciones para los usuarios y pudiendo implementar un sistema de control de cargas eléctricas, siendo soportado en la misma arquitectura de red.
5. Integrar sistemas del tipo IoT en los sectores industriales es cada vez más común, por lo que la investigación de los diferentes campos que aplican estos sistemas a sus procesos es adecuada para el estudiante o desarrollador que desea incursionar en la Industria 4.0.

6. Reconocer que los sensores destinados para un monitoreo constante en las cámaras congeladas deben ser de preferencia con un grado industrial, debido a la humedad en este ambiente las terminales de los módulos tienden a oxidarse, de otra forma el cableado debe estar recubierto completamente sin dejar expuestas las conexiones.

BIBLIOGRAFÍA

1. Aprendiendo Arduino. *Arquitecturas IoT*. [en línea]. <<https://aprendiendoarduino.wordpress.com/tag/modelos-de-capas-iot/>>. [Consulta: 5 de enero de 2022].
2. Aprendiendo Arduino. *MQTT*. [en línea]. <<https://aprendiendoarduino.wordpress.com/2018/11/19/mqtt/>>. [Consulta: 5 de enero de 2022].
3. ARROW. *Sensores de corriente*. [en línea]. <<https://www.arrow.com/es-mx/categories/sensors/current-sensors>>. [Consulta: 30 de diciembre de 2021].
4. AWS. *Amazon managed grafana*. [en línea]. <<https://aws.amazon.com/es/grafana/>>. [Consulta: 30 de diciembre de 2021].
5. AWS. *Microservicios*. [en línea]. <<https://aws.amazon.com/es/microservices/>>. [Consulta: 2 de enero de 2022].
6. BUDIA, Ernesto. *Modelo de auditoria energética en el sector industrial*. [en línea]. Trabajo de graduación de Ing. Industrial. Universidad Carlos III de Madrid. 2009. p. 6. <<https://e-archivo.uc3m.es/handle/10016/8175#preview>>. [Consulta: 22 de diciembre de 2021].

7. COMISIÓN NACIONAL DE ENERGÍA ELÉCTRICA. *Eficiencia energética*. [en línea]. <<https://www.cnee.gob.gt/wp/eficiencia-energetica/>>. [Consulta: 22 de diciembre de 2021].
8. CONCEPCIÓN, Ricardo. *Tarjetas de desarrollo (Episodio #8)*. [en línea]. <<https://www.rjconcepcion.com/podcast/tarjetas-de-desarrollo-episodio-8/>>. [Consulta: 27 de diciembre de 2021].
9. CRUZ MEDINA, Marie Chantelle. *Análisis arquitectónico de perfiles cortex-m y cortex-a en procesadores ARM y diseño de guía introductoria en su programación de bajo nivel con lenguaje ensamblador*. Trabajo de graduación de Ing. Electrónica. Universidad de San Carlos de Guatemala. 2019. p. 374.
10. DEL VAL ROMÁN, José Luis. *Industria 4.0: la transformación digital de la industria*. [en línea]. <<http://coddii.org/wp-content/uploads/2016/10/Informe-CODDII-Industria-4.0.pdf>>. [Consulta: 20 de diciembre de 2021].
11. Desarrollo web. *Protocolos de comunicaciones*. [en línea]. <<https://desarrolloweb.com/articulos/1617.php>>. [Consulta: 28 de diciembre de 2021].
12. Eclipse. *Eclipse mosquitto. An open source MQTT broker*. [en línea]. <<https://mosquitto.org/>>. [Consulta: 3 de enero de 2022].
13. Ecured. *Raspbian*. [en línea]. <<https://www.ecured.cu/Raspbian>>. [Consulta: 3 de enero de 2022].

14. EDS Robotics. *¿Qué es la integración de sistemas?*. [en línea]. <<https://www.edsrobotics.com/blog/integracion-de-sistemas-que-es/>>. [Consulta: 20 de diciembre de 2021].
15. ESCODA, Salvador. *Variadores de frecuencia reguladores de velocidad*. [en línea]. <<https://elblogdelinstalador.com/variadores-de-frecuencia-reguladores-de-velocidad-otra-de-alternativa-para-el-ahorro-energetico/>>. [Consulta: 17 de enero de 2022].
16. ESSS. *Conozca los pilares de la industria 4.0*. [en línea]. <<https://www.esss.co/es/blog/los-pilares-de-la-industria-4-0/>>. [Consulta: 20 de diciembre de 2021].
17. ESP32net. *The Internet of Things with ESP32*. [en línea]. <<http://esp32.net/>>. [Consulta: 27 de diciembre de 2021].
18. FORRESTER. *Blogs & Podcasts*. [en línea]. <<https://www.forrester.com/blogs/>>. [Consulta: 18 de enero de 2022].
19. GARCÍA, Joaquín. *Introducción a las SBC*. [en línea]. <<https://uytec.wordpress.com/2017/07/05/introduccion-a-las-sbc-computadoras-de-placa-reducida/>>. [Consulta: 23 de diciembre de 2021].
20. Grafana. *Grafana labs*. [en línea]. <<https://grafana.com/docs/>>. [Consulta: 30 de diciembre de 2021].

21. GONZALES, Oscar. *Comparativa y análisis completo de los módulos Wifi ESP8266 y ESP32.* [en línea]. <<https://blog.bricogEEK.com/noticias/electronica/comparativa-y-analisis-completo-de-los-modulos-wifi-esp8266-y-esp32>>. [Consulta: 6 de enero de 2022].
22. Hardware Libre. *ACS712: el módulo sensor de corriente.* [en línea]. <<https://www.hwlible.com/acs712>>. [Consulta: 30 de diciembre de 2021].
23. Hardware Libre. *GPIO: todo sobre las conexiones de la Raspberry Pi 4 y 3.* [en línea]. <<https://www.hwlible.com/gpio-raspberry-pi>>. [Consulta: 23 de diciembre de 2021].
24. Hard Zone. *Microcontroladores, arquitectura y usos de este tipo de CPU.* [en línea]. <<https://hardzone.es/reportajes/que-es/microcontroladores>>. [Consulta: 27 de diciembre de 2021].
25. HURTADO, José. *Introducción a las redes de comunicación industrial.* [en línea]. <http://www.infoplC.net/files/documentacion/comunicaciones/infoPLC_net_introduccic3b3n-a-las-redes-de-comunicac3b3n-industrial.pdf>. [Consulta: 28 de diciembre de 2021].
26. IBM. *Servicios web.* [en línea]. <<https://www.ibm.com/docs/es/was/9.0.5?topic=services-web>>. [Consulta: 2 de enero de 2022].

27. Ingertec. *UNE-EN 16247 Auditorias Energéticas*. [en línea]. <<https://ingertec.com/une-en-16247-auditorias-energeticas/>>. [Consulta: 3 de enero de 2022].
28. Ingtelecto. *Transformadores de corriente*. [en línea]. <<https://ingtelecto.com/transformador-de-corriente/>>. [Consulta: 30 de diciembre de 2021].
29. Instructables Circuits. *Medición de temperatura con RTD PT100, Transmisor 4-20 MA y Arduino*. [en línea]. <<https://www.instructables.com/Medicion-De-Temperatura-Con-RTD-PT100-Transmisor-4/>>. [Consulta: 30 de diciembre de 2021].
30. ISO. *ISO / IEC 30141: 2018 Internet de las Cosas (IoT): arquitectura de referencia*. [en línea]. <<https://www.iso.org/standard/65695.html>>. [Consulta: 3 de enero de 2022].
31. ISO. *ISO / IEC 27033-1: 2015 Tecnología de la información – Técnicas de seguridad – Seguridad de la red – Parte 1: Descripción general y conceptos*. [en línea]. <<https://www.iso.org/standard/63461.html>>. [Consulta: 3 de enero de 2022].
32. ISO Tools. *La Norma ISO 50001:2011 y la Gestión de la energía*. [en línea]. <<https://www.isotools.org/pdfs/monografico-ISO-50001.pdf>>. [Consulta: 3 de enero de 2022].
33. Logicbus. *Protocolos de comunicación industriales*. [en línea]. <<https://www.logicbus.com.mx/blog/protocolos-de-comunicacion-industriales/>>. [Consulta: 28 de diciembre de 2021].

34. MOROTE, José. *Auditorias energéticas*. [en línea]. <<https://ovacen.com/auditorias-energeticas-definicion-ambito-actuacion-normativa>>. [Consulta: 22 de diciembre de 2021].
35. Naylamp Mechatronics. *Conversor USB a RS485*. [en línea]. <<https://naylampmechatronics.com/conversores-ttl/230-conversor-usb-rs485.html>>. [Consulta: 29 de diciembre de 2021].
36. Ngrok. *Documentation*. [en línea]. <<https://ngrok.com/docs>>. [Consulta: 3 de enero de 2022].
37. Node-RED home. *Node-RED*. [en línea]. <<https://nodered.org/>>. [Consulta: 30 de diciembre de 2021].
38. PC components. *Características de la nueva Raspberry Pi 4 Modelo B+*. [en línea]. <<https://www.pccomponentes.com/caracteristicas-raspberry-pi-4>>. [Consulta: 23 de diciembre de 2021].
39. PCE-instruments. *Analizador de redes eléctricas*. [en línea]. <https://www.pce-instruments.com/espanol/instrumento-medida/medidor/analizador-de-redes-el_electricas-kat_162757.htm>. [Consulta: 30 de diciembre de 2021].
40. Programa fácil. *Guía introductoria a MQTT*. <<https://programarfácil.com/esp8266/mqtt-esp8266-raspberry-pi/>>. [Consulta: 5 de enero de 2022].

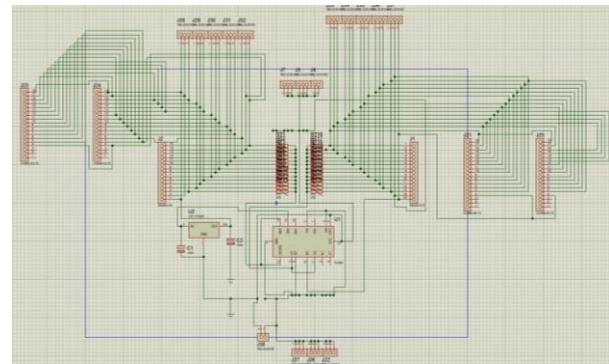
41. Proto Supplies. *Módulo de interfaz MAX485 TTL a RS-485*. [en línea]. <<https://protosupplies.com/product/max485-ttl-to-rs-485-interface-module/>>. [Consulta: 29 de diciembre de 2021].
42. Raspberry Pi. *Raspberry Pi Documentation*. [en línea]. <<https://www.raspberrypi.com/documentation/>>. [Consulta: 23 de diciembre de 2021].
43. REAL VNC. *All you need to know about VNC remote access technology*. [en línea]. <<https://discover.realvnc.com/what-is-vnc-remote-access-technology>>. [Consulta: 3 de enero de 2022].
44. Redes Zone. *Protocolos básicos en redes*. [en línea]. <<https://www.redeszone.net/tutoriales/internet/protocolos-basicos-redes>>. [Consulta: 28 de diciembre de 2021].
45. Remote.it. *Introducción*. [en línea]. <<https://docs.remote.it/introduction/get-started>>. [Consulta: 3 de enero de 2022].
46. Risoul. *Arrancadores suaves: funcionamiento, beneficios y aplicaciones*. [en línea]. <<https://www.risoul.com.mx/blog/arrancadores-suaves-funcionamiento-beneficios-y-aplicaciones>>. [Consulta: 17 de enero de 2022].
47. SANCHIS, Roberto; ROMERO, Julio; ARIÑO, Carlos. *Automatización industrial*. [en línea]. <<http://repositori.uji.es/xmlui/bitstream/handle/10234/24182/s31.pdf>>.

- f?sequence=6&isAllowed=y>. España: Universidad Jaume. 2010. p. 6. [Consulta: 21 de diciembre de 2021].
48. SCHALLENBERG, Julieta. *Energías renovables y eficiencia energética*. [en línea]. <<http://up-rid2.up.ac.pa:8080/xmlui/handle/123456789/1459>>. Instituto Tecnológico de Canarias. 2008. p. 122. [Consulta: 22 de diciembre de 2021].
49. Sensor GO. *¿Qué es la telemetría y cuáles son sus aplicaciones?*. [en línea]. <<https://sensorgo.mx/telemetria/>>. [Consulta: 22 de diciembre de 2021].
50. SUAREZ, Carmen. *Recursos de microcontroladores y familia de los PIC*. [en línea]. <<http://computointegradoits.blogspot.com/2012/03/recursos-de-microcontroladores-y.html>>. [Consulta: 27 de diciembre de 2021].
51. Tech Data. *Internet de las Cosas desde abajo: Dispositivos y controladores físicos*. [en línea]. <<https://blog.techdata.com/ts/latam/internet-de-las-cosas-desde-abajo-dispositivos-y-controladores-f%C3%ADSicos>>. [Consulta: 5 de enero de 2022].
52. Tecnobits. *Qué hace cada pin GPIO en Raspberry Pi 4*. [en línea]. <<https://tecnobits.net/que-hace-cada-pin-gpio-de-raspberry-pi-4/>>. [Consulta: 23 de diciembre de 2021].

53. The Engineering Projects. *Introduction to DHT22*. [en línea]. <<https://www.theengineeringprojects.com/2019/02/introduction-to-dht22.html>>. [Consulta: 29 de diciembre de 2021].
54. THIDO electrónica. *Módulo MAX485 convertidor TTL a RS485*. [en línea]. <<http://www.electronicathido.com/detallesProducto.php?id=U2c0RjI3Nk93cIBWWjhQSURoSkJqZz09>>. [Consulta: 29 de diciembre de 2021].
55. Unidad de Apoyo para el Aprendizaje. *Lenguajes de programación*. [en línea]. <https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html>. [Consulta: 27 de diciembre de 2021].
56. VARPER Technologies. *OTA (Over The-air Programming)*. [en línea]. <<https://www.varpertechnologies.com/ota-over-the-air-programming/?v=5bc574a47246>>. [Consulta: 3 de enero de 2022].
57. WEIS, Olga. *Virtual serial port*. [en línea]. <<https://www.virtual-serial-port.org/es/article/what-is-serial-port/rs232-vs-rs485.html>>. [Consulta: 2 de enero de 2022].

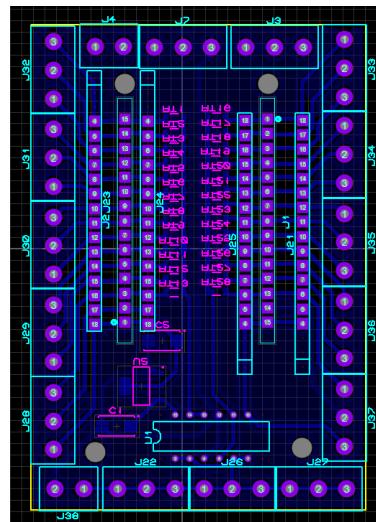
APÉNDICES

Apéndice 1. Esquemático de PCB para ESP32



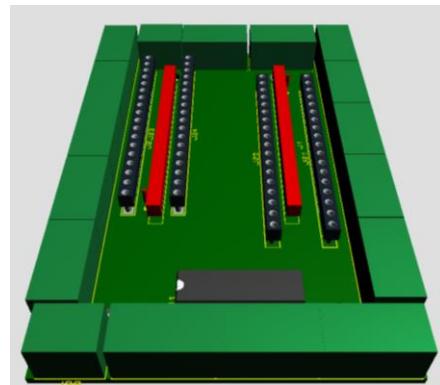
Fuente: elaboración propia, empleando Proteus.

Apéndice 2. Diseño de PCB para ESP32



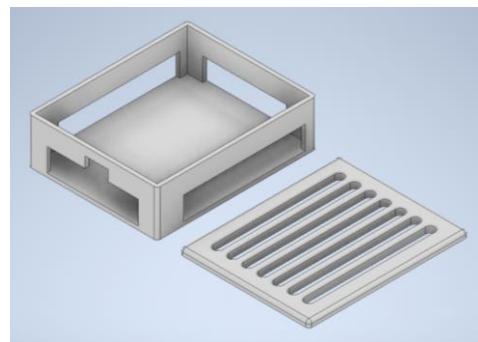
Fuente: elaboración propia, empleando Proteus.

Apéndice 3. **Visualización 3D de PCB para ESP32**



Fuente: elaboración propia, empleando Proteus.

Apéndice 4. **Diseño 3D de case para ESP32**



Fuente: elaboración propia, empleando Autodesk-Inventor.

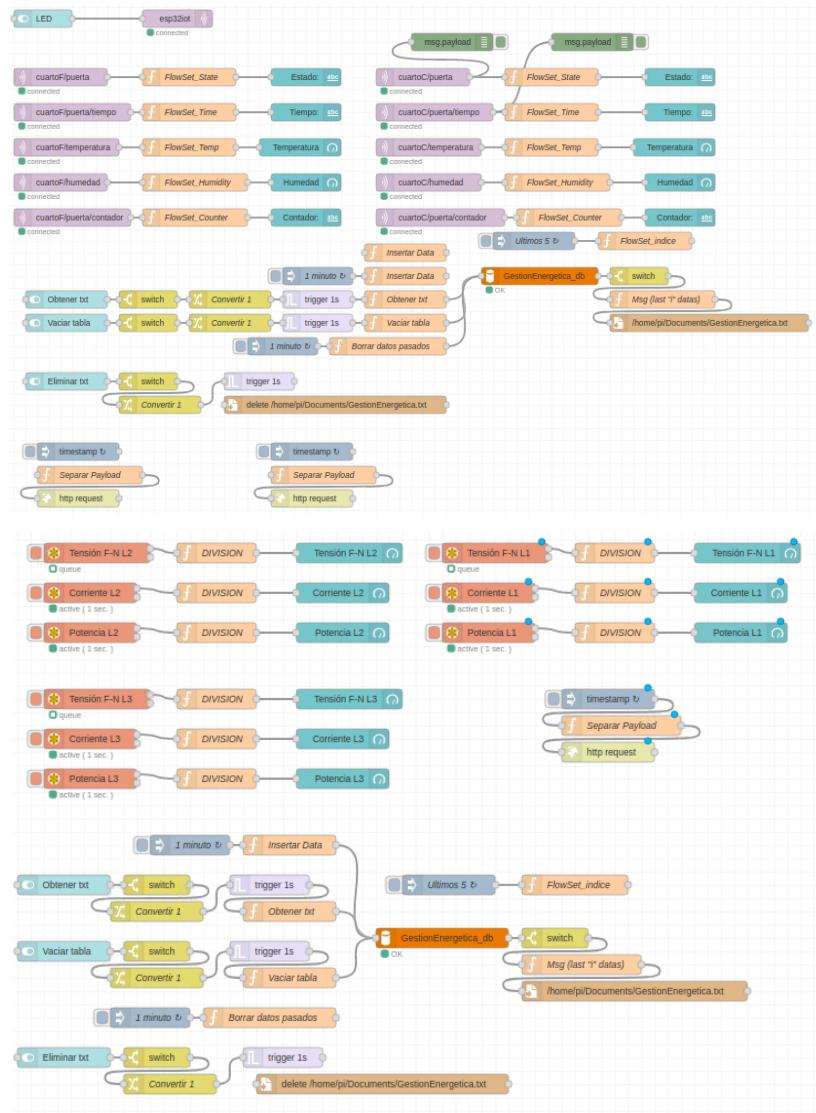
Apéndice 5. **Flujo de nodos para comunicación MQTT y MODBUS en Node-RED**

Los nodos por utilizar dependen de la finalidad que quiera lograr el desarrollador y los objetivos requeridos en el sistema. En esta sección se provee la vista general de los flujos utilizados para establecer la comunicación con

Continuación del apéndice 5.

distintos protocolos, los nodos para crear paneles y botones de control en el *dashboard*, así como la conectividad entre la base de datos, generación de archivos y comunicación con las hojas de cálculo en Google Drive.

- Flujo MQTT y MODBUS



Fuente: elaboración propia, empleando Node-RED.

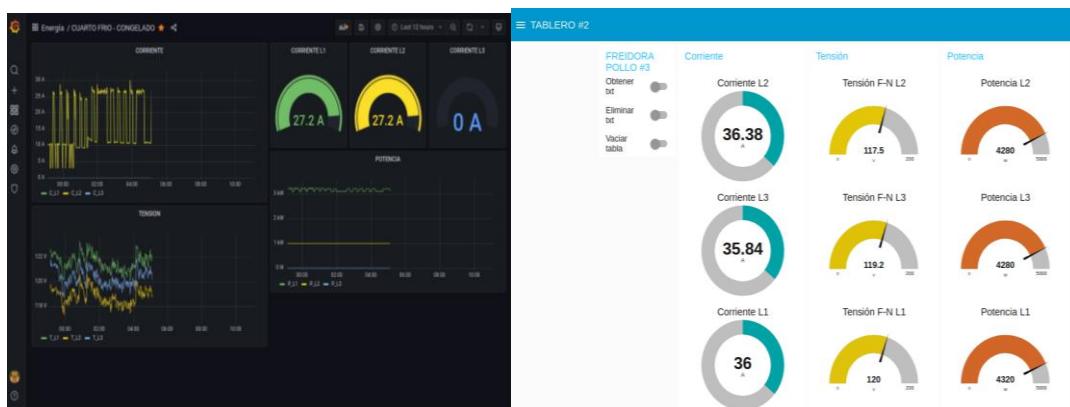
Apéndice 6. Plataformas de monitoreo

El proyecto implementado dispone de dos plataformas para el monitoreo de los datos censados, en ambos casos es posible generar tableros, relojes, botones de control para obtener reportes en distintos formatos de archivos.

Para ingresar a las visualizaciones de cada tablero puede acceder de forma local, conectándose a la misma red a la cual está conectada la Raspberry Pi y escribir en cualquier navegador la *IP* de la Raspberry Pi seguido de dos puntos y el número de puerto de cada servicio, como se muestra a continuación:

- Aplicación en Grafana: 0.0.0.0:3000
- *Dashboard* en Node-RED: 0.0.0.0:1880/ui

Sustituyendo la dirección de ceros por la dirección *IP* de la Raspberry Pi, así mismo puede ingresar a estas aplicaciones utilizando la conexión remota provista por remote.it y VNC, a la vez también puede utilizar la opción descrita en el capítulo 5, en la cual ngrok permite hacer público uno de estos servicios y acceder desde cualquier dispositivo que disponga del enlace generado.

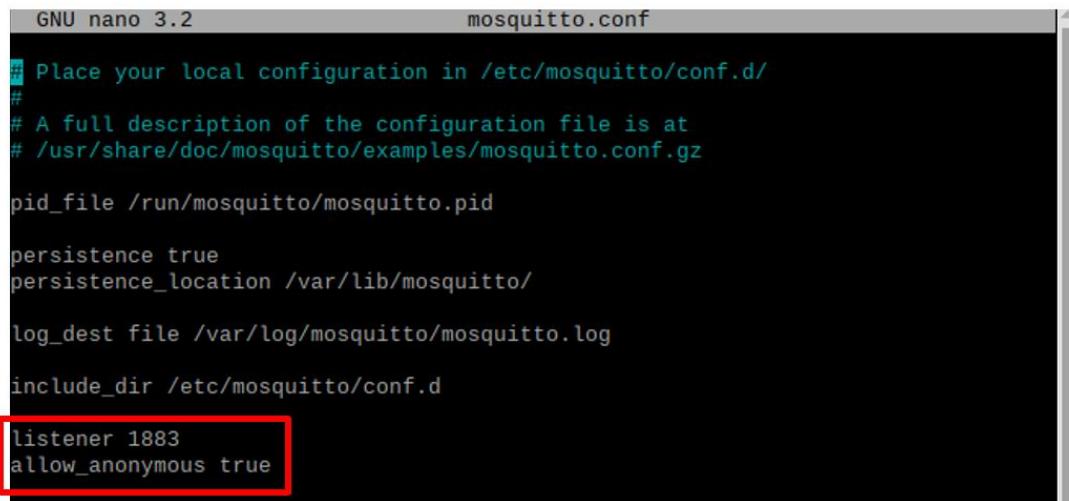


Fuente: elaboración propia, empleando Grafana y Node-RED.

Apéndice 7. Configuración de puerto de escucha en *broker* mosquitto

La conexión entre el *broker* y los dispositivos clientes está bloqueada por seguridad en la instalación por defecto, lo cual conlleva a denegar todas las conexiones que intenten establecerse con el servidor. Para habilitar esta opción debemos escribir un conjunto de comandos en el archivo de *mosquitto.conf* que se encuentra ubicado en la carpeta de instalación de *mosquitto*. El comando *sudo nano mosquitto.conf* concede los permisos como super usuario para poder modificar el archivo.

Las líneas por escribir son: *listener 1883* y *allow_anonymous true*, con esos comandos estamos definiendo que el puerto de escucha 1883 estará siempre disponible para las comunicaciones con dispositivos y redes externas; por último, guardamos la configuración pulsando las teclas *Ctrl+O* y *Ctrl+X* para salir.



```
GNU nano 3.2 mosquitto.conf
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.gz

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1883
allow_anonymous true
```

Fuente: elaboración propia, empleando Debian Buster.

Apéndice 8. Código cargado en la ESP32

La unión de los bloques de código descritos en el capítulo 5 es presentada en esta sección general que incluye la conectividad con sensores, el envío de datos por medio de MQTT y la comunicación OTA.

OTA.h	ESP32_Utils.hpp	ESP32_Utils OTA.hpp	config.h	OTA.h	ESP32_Utils.hpp	ESP32_Utils OTA.hpp	config.h
1 #include <WiFi.h>	38 //IP del servidor			1 #include <WiFi.h>	39 const char* mqtt_server = "192.168.1.120";		
2 #include <ArduinoOTA.h>	40 WiFiClient esp32;			2 #include <ArduinoOTA.h>	41 PubSubClient client(esp32);		
3 #include <DHT.h>	42 unsigned long tiemp, tiemp2, tdhtx, lastMsg = 0;			3 #include <DHT.h>	43 char msg[50];		
4 #include <PubSubClient.h>	44 //-----Lectura de datos que llegan despues de un publish			4 #include <PubSubClient.h>	45 void callback(String topic, byte* message, unsigned int length){		
5 #include "config.h" // Sustituir con datos de vuelo	46 Serial.print("Mensaje que llega del topic: ");			5 #include "config.h" // Sustituir con datos de vuelo	47 Serial.print(topic);		
6 #include "ESP32_Utils.hpp"	48 Serial.print(". Message: ");			6 #include "ESP32_Utils.hpp"	49 for(int i = 0; i < length; i++){		
7 #include "ESP32_Utils OTA.hpp"	50 Serial.print((char)message[i]);			7 #include "ESP32_Utils OTA.hpp"	51 messageData += (char)message[i];		
8 //-----SENSOR_1 DHT22-----	52 }			8 //-----SENSOR_1 DHT22-----	53 //Encender LED		
9 #define DHTTYPE DHT22	54 if(messageData == "true"){			9 #define DHTTYPE DHT22	55 digitalWrite(2,HIGH);		
10 #define DHTPIN 23	56 buttonPushCounter = 0;			10 #define DHTPIN 23	57 TiempoA = 0;		
11 DHT dht(DHTPIN, DHTTYPE, 22);	58 buttonPushCounter_2 = 0;			11 DHT dht(DHTPIN, DHTTYPE, 22);	59 buttonPushCounter_2 = 0;		
12 //-----SENSOR_2 DHT22-----	60 buttonPushCounter_2 = 0;			12 //-----SENSOR_2 DHT22-----	61 TiempoA_2 = 0;		
13 #define DHTPIN_2 21	62 buttonPushCounter_2 = 0;			13 #define DHTPIN_2 21	63 //-----Conexion a broker/Suscribir-----		
14 DHT dht_2(DHTPIN_2, DHTTYPE, 22);	64 void reconnect(){			14 DHT dht_2(DHTPIN_2, DHTTYPE, 22);	65 while(!client.connected()){		
15 //-----SENSOR PUERTA_1-----	66 Serial.print("Iniciandoconexion con Broker...");			15 //-----SENSOR PUERTA_1-----	67 String clienteId = "ESP32";		
16 const int buttonPin = 22;	68 if(client.connect(clienteId.c_str())){			16 const int buttonPin = 22;	69 Serial.println("Conectado");		
17 int buttonPushCounter = 0;	70 client.subscribe("esp32iot"); //topic para suscribir			17 int buttonPushCounter = 0;	71 }		
18 int buttonState = 0;	72 }			18 int buttonState = 0;	73 Serial.print("Failed, rc = ");		
19 int lastButtonState = 0;	74 Serial.print(client.state());			19 int lastButtonState = 0;			
20 unsigned long TiempoM, TiempoI, TiempoA;				20 unsigned long TiempoM, TiempoI, TiempoA;			
21 //-----SENSOR PUERTA_2-----				21 //-----SENSOR PUERTA_2-----			
22 const int buttonPin_2 = 19;				22 const int buttonPin_2 = 19;			
23 int buttonPushCounter_2 = 0;				23 int buttonPushCounter_2 = 0;			
24 int buttonState_2 = 0;				24 int buttonState_2 = 0;			
25 int lastButtonState_2 = 0;				25 int lastButtonState_2 = 0;			
26 unsigned long TiempoM_2, TiempoI_2, TiempoA_2;				26 unsigned long TiempoM_2, TiempoI_2, TiempoA_2;			
27 int led = 2;				27 int led = 2;			
28 char data_temp[12] = "";				28 char data_temp[12] = "";			
29 char data_humi[12] = "";				29 char data_humi[12] = "";			
30 char data_cont[12] = "";				30 char data_cont[12] = "";			
31 char data_cont_2[12] = "";				31 char data_cont_2[12] = "";			
32 char data_temp_2[12] = "";				32 char data_temp_2[12] = "";			
33 char data_humi_2[12] = "";				33 char data_humi_2[12] = "";			
34 char data_puerta[12] = "";				34 char data_puerta[12] = "";			
35 char data_puerta_2[12] = "";				35 char data_puerta_2[12] = "";			
36 String messageData;				36 String messageData;			
37				37			

Continuación del apéndice 8.

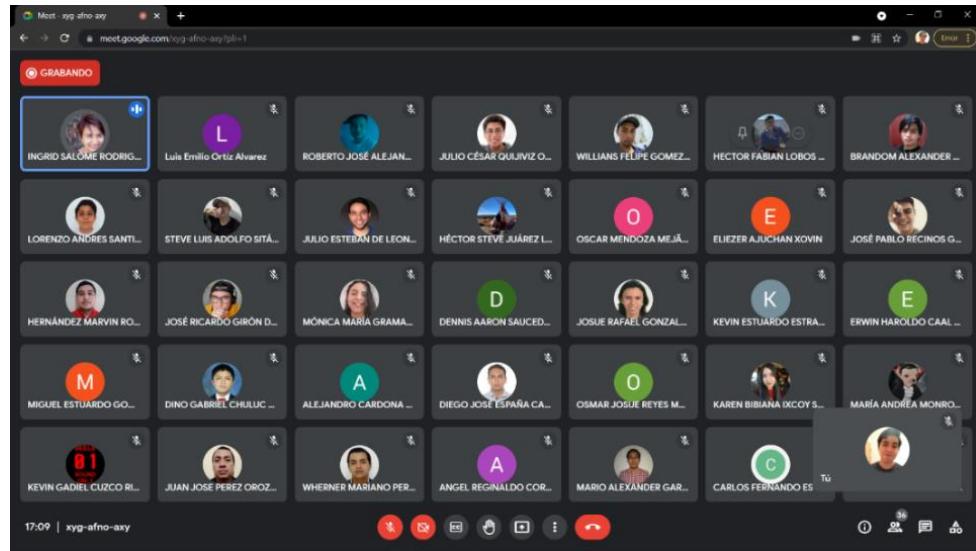
```

OTA\$  ESP32_Utils.hpp  ESP32_Utils_OTA.hpp  config.h
74     Serial.print(client.state());
75     Serial.println(" esperando 3 segundos ...");
76     delay(3000);
77   }
78 }
79
80 void setup() {
81   pinMode(2, OUTPUT);
82   pinMode(buttonPin, INPUT_PULLUP);
83   pinMode(buttonPin_2, INPUT_PULLUP);
84   Serial.begin(115200);
85   ConnectWiFi_STA();
86   InitOTA();
87   dht.begin();
88   client.setServer(mqtt_server, 1883);
89   client.setCallback(callback);
90 }
91 void loop(){
92   ArduinoOTA.handle();
93   if(!client.connected())reconnect();
94   client.loop();
95   sensorTemp();
96   TiempoM = millis();
97   TiempoM_2 = millis();
98   tiemp = millis();
99
100  if(tiemp - lastMsg > 500){
101    lastMsg = tiemp;
102    //-----Sensor de puerta_1-----//
103    buttonState = digitalRead(buttonPin);
104    if (buttonState != lastButtonState) {
105      if (buttonState == LOW) {
106        client.publish("cuartoF/puerta", "PuertaAbierta");
107        buttonPushCounter++;
108        while(buttonState == LOW){
109          buttonState = digitalRead(buttonPin);
110          TiempoI = (millis()-TiempoM)/1000;
111        }
112        delay(100);
113      }
114    }
115    if(buttonState == HIGH) {
116      client.publish("cuartoF/puerta", "PuertaCerrada");
117      sprintf(data_cont,"%5u", buttonPushCounter);
118      client.publish("cuartoF/puerta/contador", data_cont);
119      TiempoA = TiempoI + TiempoA;
120      sprintf(data_puerta,"%5u",TiempoA);
121      client.publish("cuartoF/puerta/tiempo", data_puerta);
122      TiempoI = 0;
123    }
124  }
125  lastButtonState = buttonState;
126  //-----Sensor de puerta_2-----//
127  buttonState_2 = digitalRead(buttonPin_2);
128  if (buttonState_2 != lastButtonState_2) {
129    if (buttonState_2 == LOW) {
130      client.publish("cuartoC/puerta", "PuertaAbierta");
131      buttonPushCounter_2++;
132      while(buttonState_2 == LOW){
133        buttonState_2 = digitalRead(buttonPin_2);
134        TiempoI_2 = (millis()-TiempoM_2)/1000;
135        sensorTemp();
136        delay(100);
137      }
138    }
139    if(buttonState_2 == HIGH) {
140      client.publish("cuartoC/puerta", "PuertaCerrada");
141      sprintf(data_cont_2,"%5u", buttonPushCounter_2);
142      client.publish("cuartoC/puerta/contador", data_cont_2);
143      TiempoA_2 = TiempoI_2 + TiempoA_2;
144      sprintf(data_puerta_2,"%5u",TiempoA_2);
145      client.publish("cuartoC/puerta/tiempo", data_puerta_2);
146      TiempoI_2 = 0;
147    }
148  }
149  lastButtonState_2 = buttonState_2;
150  delay(100);
151 }
152 messageData = "";
153 }
154
155 void sensorTemp(){
156   tiemp2 = millis();
157   if(tiemp2 - tdhtx > 2000){
158     tdhtx = tiemp2;
159     //enviando datos de humedad y temperatura
160     float h = dht.readHumidity();
161     float t = dht.readTemperature();
162     if(isnan(h) || isnan(t)){
163       Serial.println(F("Failed to read from DHT sensor"));
164       return;
165     }
166     sprintf(data_temp,"%3.2f",t);
167     client.publish("cuartoF/temperatura", data_temp);
168     sprintf(data_humi,"%3.2f",h);
169     client.publish("cuartoF/humedad", data_humi);
170     float h_2 = dht_2.readHumidity();
171     float t_2 = dht_2.readTemperature();
172     if(isnan(h_2) || isnan(t_2)){
173       Serial.println(F("Failed to read from DHT sensor 2"));
174       return;
175     }
176     sprintf(data_temp_2,"%3.2f",t_2);
177     client.publish("cuartoC/temperatura", data_temp_2);
178     sprintf(data_humi_2,"%3.2f",h_2);
179     client.publish("cuartoC/humedad", data_humi_2);
180   delay(100);
181 }
182 }

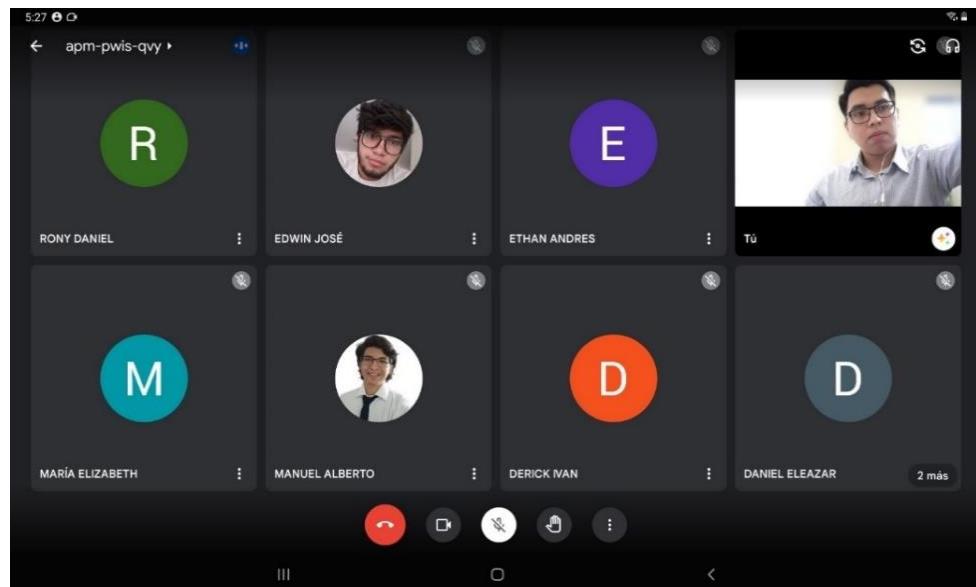
```

Fuente: elaboración propia, empleando Arduino IDE.

Apéndice 9. **Conferencia sobre IoT en los cursos de Electrónica 5 y 6 de la Facultad de Ingeniería - USAC**



Fuente: captura de pantalla, empleando Google Meet.

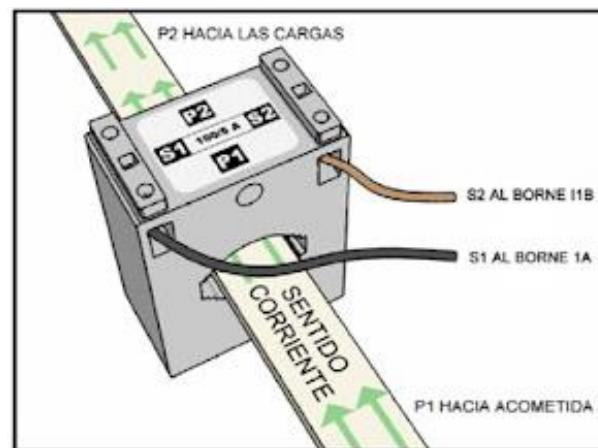
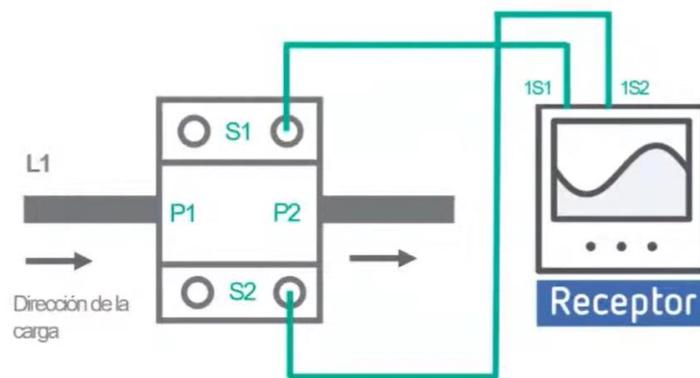


Fuente: captura de pantalla, empleando Google Meet.

ANEXOS

Anexo 1. Instalación de transformador de corriente

El transformador ha de seguir la secuencia P1 – P2 y S1 – S2 para ver la corriente como consumo. La conexión inversa llevaría a una lectura errónea viendo corriente negativa y afectando a los cálculos del receptor

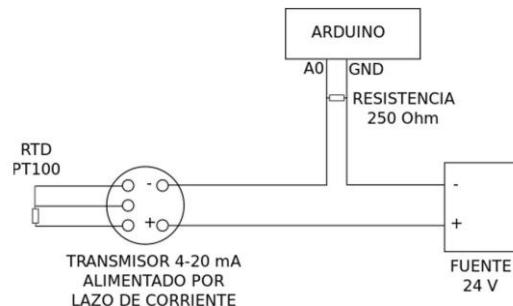


Fuente: Transformador de Intensidad /5. Consultado el 30 de diciembre de 2021.

Recuperado de <http://seinonnews.blogspot.com/2011/04/>.

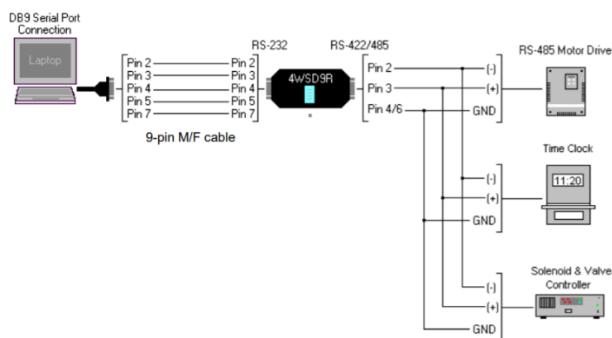
Anexo 2. Conexión de transmisor 4-20mA con Arduino

Generalmente, no se pueden conectar entradas directas de corriente en las tarjetas de desarrollo, por eso es necesario convertir la señal de corriente en voltaje. Esto se consigue agregando una resistencia entre las terminales de salida del transmisor, obteniendo un voltaje estandarizado que si puede ser leído por uno de los pines de la tarjeta.



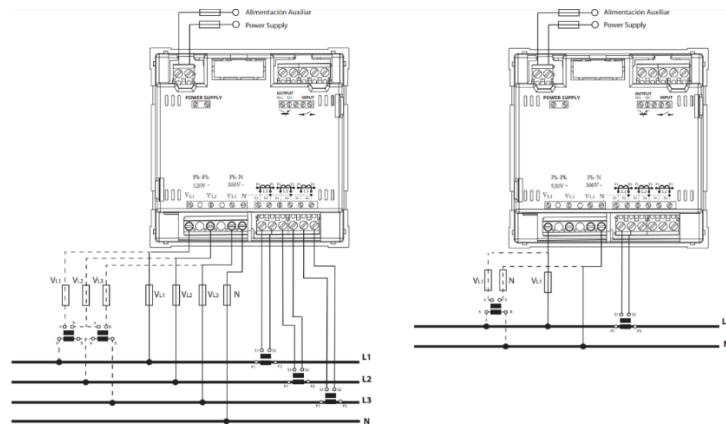
Fuente: Instructables. Medicion De Temperatura Con RTD PT100, Transmisor 4-20 MA Y Arduino. Consultado el 30 de diciembre de 2021. Recuperado de <https://www.instructables.com/Medicion-De-Temperatura-Con-RTD-PT100-Transmisor-4/>.

Anexo 2. Conexiones entre RS232 y RS485



Fuente: Micromax. Universal Converter 4WSD9R. Consultado el 2 de enero de 2022. Recuperado de <https://micromax.com.au/sites/default/files/4WSD9R.pdf>.

Anexo 3. Conexión trifásica y monofásica en CVM-C5



Fuente: Circutor. Analizador de redes, Multímetro multifunción para panel. Consultado el 6 de enero de 2022. Recuperado de http://circutor.es/docs/FT_CVM-C5_ES.pdf.

Para más detalles, puede consultar la página web CIRCUTOR, en la cual encontrará el manual y guías técnicas: <http://circutor.es/es/productos/medida-y-control/analizadores-de-redes-fijos/analizadores-de-redes/cvm-c5-itf-485-i-detail#documentaci%C3%B3n>.

Anexo 4. Mapa de memoria MODBUS (variables de energía)

Parámetro	Símbolo	Tarifa 1	Tarifa 2	Unidades
Energía activa	kWh III	3C-3D	6C-6D	Wh
Energía reactiva inductiva (kvarhL)	kvarhL III	3E-3F	6E-6F	varh
Energía reactiva capacitiva (kvarhC)	kvarhC III	40-41	70-71	varh
Energía aparente (kVAh)	kVAh III	5E-5F	72-73	kVAh
Energía activa generada	kWh III (-)	60-61	74-75	Wh
Energía reactiva inductiva generada	kvarhL III (-)	62-63	76-77	varh
Energía reactiva capacitativa generada	kvarhC III	64-65	78-79	varh
Energía aparente generada	kVAh III	66-67	7A-7B	VAh
Emisiones CO ₂	KgCO ₂	68-69	7C-7D	KgCO ₂ x 100000
Coste generada	\$	6A-6B	7E-7F	\$ x 100000
Horas por tarifa	Hours	80-81	82-83	seg

Fuente: Circutor. Manual de instrucciones CVM-C5. Consultado el 10 de enero de 2022. Recuperado de <http://docs.circutor.com/docs/M026B01-01.pdf>.

Anexo 5. Mapa de memoria MODBUS (variables de medida).

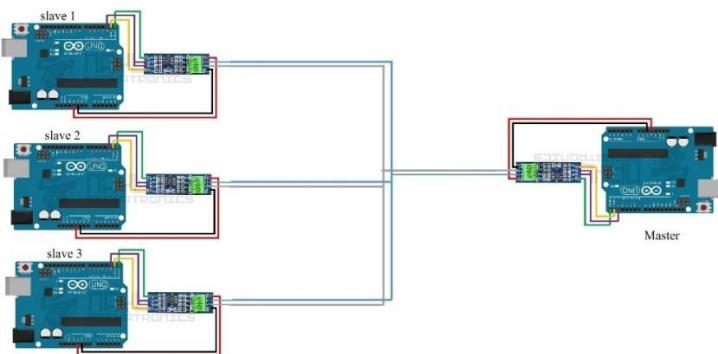
Parámetro	Simbolo	Instantáneo	Máximo	Mínimo	Unidades
Tensión fase-neutro L1	V 1	00-01	84-85	100-101	V x 10
Corriente L1	A 1	02-03	86-87	102-103	mA
Potencia activa L1	kW 1	04-05	88-89	104-105	W
Potencia reactiva L1	kvar 1	06-07	8A-8B	106-107	var
Potencia reactiva inductiva L1	kvarL1	17C-17D	18E-18F	1A0-1A1	var
Potencia reactiva capacitativa L1	kvarC1	182-183	194-195	1A6-1A7	var
Potencia aparente L1	kVAL1	188-189	19A-19B	1AC-1AD	VA
Factor de potencia L1	PF 1	08-09	8C-8D	108-109	x100
Tensión fase-neutro L2	V 2	0A-0B	8E-8F	10A-10B	V x 10
Corriente L2	A 2	0C-0D	90-91	10C-10D	mA
Potencia activa L2	kW 2	0E-0F	92-93	10E-10F	W
Potencia reactiva L2	kvar 2	10-11	94-95	110-111	var
Potencia reactiva inductiva L2	kvarL2	17E-17F	190-191	1A2-1A3	var
Potencia reactiva capacitativa L2	kvarC2	184-185	196-197	1A8-1A9	var
Potencia aparente L2	kVAL2	18A-18B	19C-19D	1AE-1AF	VA
Factor de potencia L2	PF 2	12-13	96-97	112-113	x100
Tensión fase-neutro L3	V 3	14-15	98-99	114-115	V x 10
Corriente L3	A 3	16-17	9A-9B	116-117	mA
Potencia activa L3	kW 3	18-19	9C-9D	118-119	W
Potencia reactiva L3	kvar 3	1A-1B	9E-9F	11A-11B	var
Potencia reactiva inductiva L3	kvarL3	180-181	192-193	1A4-1A5	var
Potencia reactiva capacitativa L3	kvarC3	186-187	198-199	1AA-1AB	var
Potencia aparente L3	kVAL3	18C-18D	19E-19F	1B0-1B1	VA
Factor de potencia L3	PF 3	1C-1D	A0-A1	11C-11D	x100
Potencia activa trifásica	kW III	1E-1F	A2-A3	11E-11F	W
Potencia inductiva trifásica	kvarL III	20-21	A4-A5	120-121	var
Potencia capacitiva trifásica	kvarL III	22-23	A6-A7	122-123	var
Cos φ trifásico	Cos φ III	24-25	A8-A9	124-125	x100
Factor de potencia trifásico	PF III	26-27	AA-AB	126-127	x100
Frecuencia	Hz	28-29	AC-AD	128-129	Hz x 10
Tensión L1 - L2	V12	2A-2B	AE-AF	12A-12B	V x 10
Tensión L2 - L3	V23	2C-2D	B0-B1	12C-12D	V x 10
Tensión L3 - L1	V31	2E-2F	B2-B3	12E-12F	V x 10
THD de tensión L1	%THDV1	30-31	B4-B5	130-131	% x 10
THD de tensión L2	%THDV2	32-33	B6-B7	132-133	% x 10
THD de tensión L3	%THDV3	34-35	B8-B9	134-135	% x 10
THD de corriente L1	%THDI1	36-37	BA-BB	136-137	% x 10
THD de corriente L2	%THDI2	38-39	BC-BD	138-139	% x 10
THD de corriente L3	%THDI3	3A-3B	BE-BF	13A-13B	% x 10
Potencia aparente trifásica	kVAIII	42-43	C6-C7	142-143	VA
Máxima demanda	Md(Pd)	44-45	C8-C9	-	W/VA/mA
Corriente trifásica (media)	I_AVG	46-47	CA-CB	146-147	mA
Máxima demanda I2	Md (Pd)	52-53	D6-D7	-	mA
Máxima demanda I3	Md (Pd)	54-55	D8-D9	-	mA
Tensión fase-fase (media)	VF-AVG	56-57	DA-DB	156-157	V x 10
Tensión fase-neutro (media)	VL-AVG	58-59	DC-DD	158-159	V x 10

Fuente: Circutor. Manual de instrucciones CVM-C5. Consulta: 10 de enero de 2022.

Recuperado de <http://docs.circutor.com/docs/M026B01-01.pdf>.

Anexo 6. Comunicación simplex entre Arduino utilizando MAX485

La documentación oficial de la librería minimalmodbus en la que puede encontrar distintas configuraciones y la descripción de las funciones que incluye la puede encontrar en: <https://minimalmodbus.readthedocs.io/en/stable/>.



Fuente: Arduino. Arduino master/slave communication using RS485. Consultado el 13 de enero de 2022. Recuperado de <https://arduino.stackexchange.com/questions/61839/arduino-master-slave-communication-using-rs485>.

Librerías para descargar: <https://github.com/smarmengol/Modbus-Master-Slave-for-Arduino/blob/master/ModbusRtu.h>.

Anexo 7. Recomendaciones para instalar el OS Raspbian

Para hacer uso de Raspbian se debe descargar el archivo binario de la página web y seleccionar la versión para montar la imagen del sistema operativo en una memoria SD de al menos 16GB y clase 10, por último, solamente resta esperar a que el programa notifique con un mensaje para retirar la memoria SD.

La descarga de cualquier imagen del OS la puede encontrar en: <https://www.raspberrypi.com/software/operating-systems/>.

Anexo 8. Enlaces para la descarga de los programas utilizados, librerías y documentación oficial

Los pasos para instalar Node-RED son descritos en el siguiente enlace: <https://nodered.org/docs/getting-started/>. También podrá encontrar los instaladores para distintos sistemas operativos como Windows y otras versiones de Linux, al igual que las opciones para instalarlo en algún servicio en la nube. Se recomienda la lectura de la documentación en: <https://nodered.org/docs/>.

Para gestionar la base de datos MariaDB puede dirigirse al siguiente enlace: https://mariadb.org/download/?t=mariadb&p=mariadb&r=10.6.5&os=windows&cpu=x86_64&pkg=msi&m=gigenet.

Grafana es un *software* multiplataforma que puede operar en muchos sistemas operativos como Windows, Linux, Mac o en un repositorio; la información puede encontrarla en: <https://grafana.com/grafana/download> acá podrá seleccionar la versión que quiere instalar.

Para descargar la versión de ngrok debe acceder a: <https://ngrok.com/download>.

La programación de la ESP32 desde Arduino IDE consiste en instalar las librerías y paquetes para que el gestor de tarjetas identifique las diferentes versiones de ESPRESSIF; en el siguiente enlace encontrará los pasos de instalación: <https://www.taloselectronics.com/blogs/tutoriales/programar-esp32-con-ide-arduino>.