



Universidad
Nacional
de Loja



CIEYT

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Diseño e implementación de un prototipo de sistema inteligente de iluminación y monitoreo en tiempo real de variables ambientales para la empresa de tecnología y servicios TECSERLED.

Trabajo de Titulación previo a optar por el Título de Ingeniero en Electrónica y Telecomunicaciones

AUTOR:

Jordy Carlos Carrión Lojan.

DIRECTORA:

Ing. Marianela del Cisne Carrión González. Mg. Sc.

LOJA – ECUADOR

2022

Certificación del trabajo de titulación

Loja 28 de marzo del 2022

Ing. Marianela del Cisne Carrión González. Mg. Sc.

DIRECTORA DEL TRABAJO DE TITULACIÓN

Certifico:

Que he revisado y orientado todo el proceso de la elaboración del trabajo de Titulación: **“Diseño e implementación de un prototipo de sistema inteligente de iluminación y monitoreo en tiempo real de variables ambientales para la empresa de tecnología y servicios TECSERLED”** de autoría del estudiante **Jordy Carlos Carrión Lojan**, previa a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.



Firmado electrónicamente por:
MARIANELA DEL
CISNE CARRION
GONZALEZ

Ing. Marianela del Cisne Carrión González. Mg. Sc.

DIRECTORA DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Jordy Carlos Carrión Lojan**, declaro ser autor del presente trabajo de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi trabajo de titulación en el Repositorio Digital Institucional – Biblioteca Virtual.



Firma:

Cédula de Identidad: 1104312747

Fecha: 30/08/2022

Correo electrónico: jordy.carrion@unl.edu.ec

Teléfono celular: 0959767813

Carta de autorización

Carta de autorización del trabajo de titulación por parte del autor para la consulta de producción parcial o total, y publicación electrónica de texto completo.

Yo, **Jordy Carlos Carrión Loján**, declaro ser autor del trabajo de titulación “**Diseño e implementación de un prototipo de sistema inteligente de iluminación y monitoreo en tiempo real de variables ambientales para la empresa de tecnología y servicios TECSERLED**”, como requisito para optar el título de **Ingeniero en Electrónica y Telecomunicaciones**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los treinta días del mes de agosto del dos mil veintidós.



Firma:

Autor: Jordy Carlos Carrión Loján

Cédula: 1104312747

Celular: 0959767813

Correo Electrónico: jordy.carrion@unl.edu.ec

Dirección: Loja –Miguel Hidalgo

DATOS COMPLEMENTARIOS:

Directora del trabajo de titulación: Ing. Marianela del Cisne Carrión González. Mg. Sc.

Dedicatoria

El presente trabajo está dedicado a mis padres Herman y Lorena quienes con su amor, paciencia y esfuerzo me han permitido llegar hasta aquí, siendo mi apoyo a lo largo de toda mi carrera universitaria y a lo largo de mi vida. A todas las personas especiales que me acompañaron en esta etapa con su apoyo incondicional, aportando a mi formación profesional como también humana.

Jordy Carlos Carrión Lejan

Agradecimiento

Agradezco a Dios por ser mi guía y acompañarme en el transcurso de mi vida, brindándome paciencia, sabiduría y salud para culminar con éxito mis metas propuestas.

Agradezco a la Universidad Nacional de Loja, por haberme permitido formarme en sus aulas, compartiendo ilusiones y anhelos. A toda la planta docente de la carrera de Ingeniería en Electrónica y Telecomunicaciones, en especial a mi directora de tesis Marianela Carrión, quien con su experiencia, conocimiento y motivación me oriento en la investigación.

Jordy Carlos Carrión Lejan

Índice de contenido

Portada	i
Certificación del trabajo de titulación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenido	vii
Índice de tablas	xii
Índice de figuras	xiii
Índice de anexos	xvii
1 Título	1
2 Resumen	2
2.1 Abstract	3
3 Introducción	4
3.1 Problema	4
3.2 Justificación	5
3.3 Objetivos	6
4 Marco teórico	7
4.1 Antecedentes investigativos	7
4.2 Fundamentación teórica	9
4.3 Sistema de iluminación inteligente	9
4.4 Luminotecnia	9
4.4.1 Magnitudes usadas en luminotecnia	10
4.5 Iluminación LED	11
4.6 Diodos emisores de luz	11
4.6.1 Principio de funcionamiento	11

4.6.2	Tipos de LED existentes	12
4.7	Controladores LED en el mercado actual (drivers LED)	13
4.8	Sistemas de iluminación LED inteligentes en el mercado actual	15
4.9	Regulación y control de luminarias LED con suministro AC	16
4.10	Dimerización de luminarias LED mediante atenuadores de control de fase..	16
4.10.1	Atenuadores de control de fase de borde delantero.....	17
4.10.2	Atenuadores de control de fase de borde posterior	20
4.11	Principios de factor de potencia aplicados a los sistemas de iluminación.....	22
4.12	Normativa de iluminación NTE INEN 2969-1.....	25
4.13	Automatización y control	26
4.13.1	Tipos de sistemas de control	26
4.13.2	Control de encendido y apagado (on/off).....	27
4.13.3	Controlador Proporcional, Integral y Derivativo (PID)	28
4.14	Dispositivos de hardware y software libre	31
4.15	Microcontroladores.....	31
4.16	Microprocesadores.....	32
4.17	Sensores y principio de transducción	34
4.17.1	Sensores inteligentes	35
4.17.2	Acondicionamiento de señales en sensores digitales	35
4.18	El internet de las cosas (IoT)	36
4.19	Protocolo de comunicación MQTT para redes WSN	37
4.19.1	MQTT y el modelo de publicación/suscripción	38
4.19.2	Características técnicas del protocolo de comunicación MQTT.....	39
4.19.3	Funcionamiento del protocolo MQTT	40
4.20	Herramientas de software libre para la gestión de la información	41
4.20.1	Python 3.....	41
4.20.2	Node-RED	41

4.20.3	Servidor LAMP	43
5	Metodología	45
5.1	Materiales	45
5.1.1	Microcontrolador NodeMCU ESP8266	46
5.1.2	Raspberry pi 3 Model B+	47
5.1.3	Dispositivos de medición y adquisición de datos.....	49
5.1.4	Módulo sensor de luminosidad BH1750.....	49
5.1.5	Sensor de temperatura y humedad relativa DTH22	51
5.1.6	Sensor de presencia PIR HC-SR501	53
5.1.7	Modulo PZEM-004T.....	54
5.1.8	Optoacoplador 4N35	56
5.1.9	MOSFET IRF840 CH-N	57
5.2	Metodología.....	59
5.2.1	Modalidad de la investigación.....	59
5.2.2	Técnicas e instrumentos de recolección de datos.....	59
5.2.3	Técnicas de análisis y procesamiento de la información	60
5.2.4	Desarrollo del proyecto	60
6	Resultados.....	63
6.1	Descripción general del sistema inteligente de iluminación LED.....	63
6.1.1	Adquisición de datos de los sensores	64
6.1.2	Proceso de acondicionamiento y procesamiento de la información	64
6.1.3	Sistema de control automático	64
6.1.4	Sistema de monitoreo, gestión y visualización	65
6.1.5	Diagrama de bloques del sistema de iluminación inteligente	66
6.2	Sistema de control de potencia basado en un atenuador de control de fase AC	66
6.2.1	Circuito detector de cruce por cero	67

6.2.2	Relé de estado sólido SSR.....	68
6.2.3	Simulación del circuito de control de potencia	71
6.2.4	Diseño electrónico.....	72
6.2.5	Diagrama del algoritmo de control de potencia	73
6.2.6	Prototipado del circuito de control de potencia.....	73
6.2.7	Formas de onda resultantes del circuito atenuador de potencia	74
6.2.8	Análisis del factor de potencia en el circuito de atenuación de iluminación	78
6.2.9	Desarrollo de PCB.....	81
6.3	Sistema de adquisición de datos de los sensores	82
6.3.1	Adquisición de datos de temperatura y humedad del sensor DHT22	83
6.3.2	Adquisición de datos de iluminancia del sensor BH1750.....	83
6.3.3	Adquisición de datos de consumo eléctrico del módulo PZEM-004T.....	83
6.3.4	Adquisición de datos del sensor de presencia PIR HC-SR501	83
6.3.5	Diagrama de cableado eléctrico	84
6.4	Sistema de control automático para la regulación de la iluminación	84
6.4.1	Punto de ajuste de iluminación.....	86
6.4.2	Identificación del modelo matemático de la planta de control.....	86
6.4.3	Cálculo de los parámetros de ajuste del controlador PID	92
6.4.4	Diagrama de flujo del algoritmo de control	97
6.4.5	Programación del sistema de control PID en el entorno ARDUINO.....	98
6.4.6	Pruebas y resultados del sistema de control de iluminación PID.....	98
6.5	Sistema de comunicación inalámbrico	101
6.5.1	Descripción del sistema de comunicación inalámbrico	103
6.5.2	Instalación y configuración del servidor local MQTT	105
6.6	Herramientas software open source para la gestión de la información	108
6.6.1	Configuración de la herramienta Node-RED	108

6.6.2	Instalación del servidor local LAMP	110
6.6.3	Creación de la base de datos local.....	111
6.7	Sistema de monitoreo, gestión y visualización de la información	112
6.7.1	Programación basada en flujo dentro de la herramienta Node-RED	113
6.7.2	Pruebas y resultados de la interfaz de usuario web.....	124
6.7.3	Plataforma de análisis de datos IoT.....	127
6.8	Implementación y validación del sistema de iluminación inteligente.....	130
6.8.1	Módulo de control de iluminación	131
6.8.2	Pruebas de funcionamiento del módulo de control de iluminación	133
6.9	Análisis de la reducción del consumo energético-económico.....	140
6.9.1	Ahorro energético mediante el sistema de detección de presencia	141
6.9.2	Ahorro energético mediante el sistema automático de control de potencia..	142
6.9.3	Ahorro energético-económico total del prototipo	146
7	Discusión	148
8	Conclusiones	151
9	Recomendaciones	153
10	Bibliografía	155
11	Anexos	162

Índice de tablas

Tabla 1 Resumen de magnitudes y unidades luminosas fundamentales	10
Tabla 2 Comparativa entre drivers LED internos y externos	14
Tabla 3 Productos inteligentes de iluminación LED en el mercado actual.....	15
Tabla 4 Formas de onda de voltaje y corriente en función al tipo de carga.....	24
Tabla 5 Niveles de iluminancia mantenida recomendables	25
Tabla 6 Clasificación de los sensores según los principios de transducción	34
Tabla 7 Clasificación de los sensores según la variable física a medir.....	35
Tabla 8 Características técnicas del protocolo de comunicación MQTT.....	39
Tabla 9 Materiales usados para la implementación del prototipo	45
Tabla 10 Estructura de la trama de datos del sensor DHT22	52
Tabla 11 Parámetros de funcionamiento del módulo PZEM-004T	55
Tabla 12 Secuencia de capítulos de investigación	60
Tabla 13 Parámetros eléctricos y FP medidos para una carga resistiva (Bombillo incandescente de 25W).....	78
Tabla 14 Parámetros eléctricos y FP medidos para una carga no lineal (Bombillo LED dimerizable).....	80
Tabla 15 Comparativa de los parámetros eléctricos medidos y proporcionados por el fabricante	81
Tabla 16 Efectos generales de los valores de ganancias K_p , K_i , K_d , en un sistema lazo cerrado	94
Tabla 17 Campos configurados en la tabla de la base de datos del sistema dentro de la plataforma phpMyAdmin.....	111
Tabla 18 Descripción de los nodos usados para la adquisición de datos	114
Tabla 19 Descripción de los nodos usados para el envío de datos.....	115
Tabla 20 Descripción de los nodos usados para la monitorización de los datos ambientales	116
Tabla 21 Descripción de los nodos usados para la gestión del consumo energético	123
Tabla 22 Ahorro de Wh en función del sensor PIR	145
Tabla 23 Ahorro de Wh en función de la atenuación de la luminaria.....	145
Tabla 24 Ahorro energético-económico total del prototipo	146

Índice de figuras

Figura 1 Diagrama de energía de un diodo LED	12
Figura 2 Tipos de Leds en el mercado actual.....	13
Figura 3 Driver LED externo	14
Figura 4 Representación de foco LED con driver interno	14
Figura 5 Forma de onda de corriente del atenuador TRIAC capturada (carga incandescente de 75 W).....	18
Figura 6 Diagrama de circuito atenuador de corte de fase de dos cables basado en TRIAC... 18	
Figura 7 Formas de onda y fase recortada usando atenuador de TRIAC.....	19
Figura 8 Módulo dimmer TRIAC y su aplicación junto a un Arduino Uno	20
Figura 9 Atenuador de corte de fase basado en MOSFET de 3 cables	22
Figura 10 Forma de onda de corriente del atenuador de borde posterior capturada	22
Figura 11 Diagrama de bloques de un sistema en lazo abierto	27
Figura 12 Diagrama de bloques de un sistema en lazo cerrado	27
Figura 13 Diagrama de bloques de un controlador on/off	28
Figura 14 Diagrama de bloques de un controlador PID.....	30
Figura 15 Estructura genérica de un microcontrolador	32
Figura 16 Aspecto interno de un microprocesador básico	33
Figura 17 Diagrama de bloques de un sensor inteligente.....	35
Figura 18 Modelo de comunicación publicador/suscriptor basado en temas.....	38
Figura 19 Redes de sensores inalámbricos integrados con comunicación publicador/suscriptor	39
Figura 20 Interfaz gráfica de usuario en Node-RED.....	43
Figura 21 Servidor LAMP.....	44
Figura 22 NodeMCU ESP8266 V.2 y distribución de pines.....	47
Figura 23 Raspberry pi 3 Model B+.....	49
Figura 24 Diagrama de bloques del módulo BH1750.....	50
Figura 25 Módulo sensor de luminosidad BH1750	51
Figura 26 Protocolo de comunicación del sensor DHT22	52
Figura 27 Sensor DHT22	53
Figura 28 Sensor PIR HC-SR501.....	54
Figura 29 Diagrama de bloques funcional de un módulo PZEM-004T-100A.....	55
Figura 30 Diagrama de cableado eléctrico para el módulo PZEM-004T-100A	56

Figura 31 Optoacoplador 4N35 y diagrama esquemático	57
Figura 32 Funcionamiento básico de un optoacoplador 4N35.....	57
Figura 33 Transistor MOSFET IRF840	58
Figura 34 Diagrama de bloques del sistema de control de potencia	67
Figura 35 Circuito detector de cruce por cero	68
Figura 36 Esquema básico de un relé de estado sólido SSR.....	70
Figura 37 Circuito relé de estado sólido SSR basado en MOSFET.....	70
Figura 38 Circuito electrónico de control de potencia	71
Figura 39 Gráficas de las formas de onda de salida.....	72
Figura 40 Diseño electrónico del prototipo inteligente de iluminación.....	73
Figura 41 Prototipado y pruebas de funcionamiento del circuito de control de potencia.....	74
Figura 42 Señal de entrada AC y pulsos generados por el circuito detector de cruce por cero	75
Figura 43 Información de los pulsos de detección de cruce por cero	75
Figura 44 Señal de control PWM (Ciclo de trabajo de 13.25 %) de salida del microcontrolador	76
Figura 45 Señal de control PWM (Ciclo de trabajo de 60.24 %) de salida del microcontrolador	76
Figura 46 Señal de control PWM (Ciclo de trabajo de 93 %) de salida del microcontrolador	76
Figura 47 Señal de voltaje de la lampara con un nivel bajo de encendido	77
Figura 48 Señal de voltaje de la lampara con un nivel medio de encendido	77
Figura 49 Señal de voltaje de la lampara con un nivel máximo de encendido	77
Figura 50 Diagrama de cableado eléctrico para la obtención de los datos de los sensores	84
Figura 51 Diagrama de bloques general del sistema de control automático de iluminación	85
Figura 52 Diagrama de bloques resumido del sistema de control digital	85
Figura 53 Respuesta de la planta a una excitación de entrada escalón	88
Figura 54 Importación de datos a la aplicación de identificación de sistemas	88
Figura 55 Preprocesamiento de los datos para la identificación del sistema	89
Figura 56 Estimación de modelos para el sistema	89
Figura 57 Aproximación de los modelos al comportamiento real del sistema	90
Figura 58 Hoja de trabajo de la aplicación System Identification	90
Figura 59 Respuesta a una entrada escalón y diagrama de estabilidad respectivamente.....	91
Figura 60 Código implementado para analizar la respuesta del sistema P2U	91
Figura 61 Obtención del lugar geométrico de las raíces	92
Figura 62 Lugar geométrico de las raíces en el plano.....	92

Figura 63 Respuesta del sistema a una entrada escalón e impulso unitario respectivamente ..	92
Figura 64 Diagrama de bloques del controlador PID.....	94
Figura 65 Código implementado para la obtención de los parámetros del controlador PID ...	94
Figura 66 Obtención de los parámetros del controlador PID de forma predeterminada.....	95
Figura 67 Respuesta al escalón del sistema de control PID de base y parámetros de rendimiento	95
Figura 68 Respuesta al escalón del sistema de control PID sintonizado y parámetros de rendimiento.....	96
Figura 69 Rechazo a una perturbación de entrada	96
Figura 70 Simulación de los bloques del sistema de control PID	97
Figura 71 Comparativa del sistema de control PID	97
Figura 72 Respuesta del controlador PID (Setpoint fijo y sensor PIR activado).....	99
Figura 73 Respuesta del controlador PID (Setpoint variable y sensor PIR activado).....	100
Figura 74 Respuesta del controlador PID (Setpoint fijo y sensor PIR activado/desactivado)	101
Figura 75 Respuesta del controlador PID (Setpoint fijo, sensor PIR activado, variaciones de luminosidad).....	101
Figura 76 Topología del sistema de comunicación inalámbrico.....	103
Figura 77 Esquema general del funcionamiento del protocolo MQTT	105
Figura 78 Terminal Linux, visualización de las contraseñas cifradas MQTT	106
Figura 79 Terminal Linux, edición del fichero de configuración MQTT	107
Figura 80 Terminal Linux, visualización del estado del bróker Mosquitto	107
Figura 81 Terminales Linux, funcionamiento del proceso de autenticación con el bróker MQTT	108
Figura 82 Terminal Linux, generación de contraseña cifrada Node-RED.....	109
Figura 83 Terminal Linux, edición del fichero de configuración Node-RED	109
Figura 84 Proceso de login configurado dentro de Node-RED	109
Figura 85 Proceso de login de la interfaz phpMyAdmin	111
Figura 86 Interfaz de phpMyAdmin, estructura de la tabla de datos	112
Figura 87 Interfaz Node-RED; flujo para la adquisición de datos.....	113
Figura 88 Configuración de las propiedades del nodo mqtt in	114
Figura 89 Interfaz Node-RED; flujo para el envío de datos	115
Figura 90 Interfaz Node-RED; flujo para el monitoreo de los datos ambientales	116
Figura 91 Interfaz Node-RED; cálculo del tiempo de encendido de la luminaria	119
Figura 92 Interfaz Node-RED; obtención de los parámetros eléctricos de la luminaria.....	119

Figura 93 Interfaz Node-RED; escritura en la base de datos del sistema	120
Figura 94 Interfaz Node-RED; obtención de la fecha en el Date Picker para realización de consultas	120
Figura 95 Interfaz Node-RED; obtención de la duración de encendido de la luminaria	121
Figura 96 Interfaz Node-RED; obtención de los valores promedio de potencia activa y factor de potencia.....	122
Figura 97 Interfaz Node-RED; cálculo del ahorro energético-económico	122
Figura 98 Configuración de las propiedades del nodo mysql	123
Figura 99 Interfaz de usuario web; Datos ambientales	125
Figura 100 Reporte Gmail del estado de la temperatura actual	126
Figura 101 Interfaz de usuario web; Control de Cargas.....	126
Figura 102 Interfaz de usuario web; Datos Energía	127
Figura 103 Reporte Gmail del estado de consumo energético.....	127
Figura 104 Configuración de los canales en la plataforma ThingSpeak	129
Figura 105 Obtención de API Key dentro de ThingSpeak.....	130
Figura 106 Construcción y montaje del hardware del prototipo	132
Figura 107 Módulo de pruebas del prototipo de sistema inteligente de iluminación	133
Figura 108 Respuesta del sistema de iluminación en horas nocturnas; Configuración de baja iluminación.....	134
Figura 109 Respuesta del sistema de iluminación en horas nocturnas; Nivel establecido en 100 Lx	135
Figura 110 Respuesta del sistema de iluminación en horas nocturnas; Nivel establecido en 300 Lx	136
Figura 111 Respuesta del sistema de iluminación en horas nocturnas; Nivel establecido en 500 Lx	137
Figura 112 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 100 Lx	138
Figura 113 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 300 Lx	138
Figura 114 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 500 Lx	140
Figura 115 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 800 Lx	140
Figura 116 Ahorro energético de Wh, en función del sensor PIR	142

Figura 117 Ahorro energético en función del control de potencia de la luminaria.....	144
Figura 118 Ahorro energético total por la implementación del prototipo.....	146
Figura 119 Interfaz Datos energía, visualización del ahorro energético.....	147
Figura 120 Análisis del consumo energético para un día entero de funcionamiento del prototipo	147

Índice de anexos

Anexo 1 Diagrama de bloques general del prototipo de sistema inteligente de iluminación	162
Anexo 2 Diagrama de flujo del algoritmo de control de potencia del circuito atenuador	163
Anexo 3 Hoja de características de funcionamiento del bombillo LED Toledo A60 10W DIM	164
Anexo 4 Diseño PCB del prototipo de sistema de iluminación inteligente	165
Anexo 5 Simulación 3D del diseño de placa PCB	165
Anexo 6 Resultado final de la construcción de las placas PCB y circuito electrónico	166
Anexo 7 Diagrama de flujo para la adquisición de datos del sensor DHT22	167
Anexo 8 Diagrama de flujo para la adquisición de datos del sensor BH1750.....	168
Anexo 9 Diagrama de flujo para la adquisición de datos del módulo PZEM-004T	169
Anexo 10 Código para la captación de datos del sensor DHT22, BH1750 y PZEM004Tv3	170
Anexo 11 Diagrama del algoritmo de control inteligente de iluminación	174
Anexo 12 Hoja de características del foco LED A60 Maviju.....	175
Anexo 13 Código implementado para el sistema de control PID y la comunicación con el bróker MQTT.....	175
Anexo 14 Código implementado para la lectura (DHT22, PZEM004T) y la comunicación con el bróker MQTT	183
Anexo 15 Código implementado para el envío de datos a la plataforma ThingSpeak	188
Anexo 16 Visualización de los canales del sistema de iluminación inteligente en ThingSpeak	192
Anexo 17 Certificado de traducción.....	193

1 Título

Diseño e implementación de un prototipo de sistema inteligente de iluminación y monitoreo en tiempo real de variables ambientales para la empresa de tecnología y servicios TECSERLED

2 Resumen

La eficiencia energética es uno de los campos de estudio de mayor interés actual, en tanto que los sistemas de iluminación presentan un abanico de posibilidades para el desarrollo de tecnologías y prototipos que permitan maximizar el ahorro energético y la reducción del gasto económico dentro del sector residencial. El objetivo del trabajo es reducir el consumo de energía mediante el diseño e implementación de un prototipo de sistema inteligente de iluminación que integra tecnologías de sensores para el aprovechamiento máximo de la luz natural, como también permitir realizar el monitoreo en tiempo real de diversas variables ambientales para la efectuación de acciones programadas. El prototipo construido consta de un circuito de control de potencia basado en un atenuador de control de fase de borde posterior, con un microcontrolador NodeMCU ESP8266 como unidad de procesamiento, que adicionalmente, permite regular el nivel de atenuación de la luminaria mediante la implementación de un algoritmo de control automático apoyado en un controlador PID, que toma en consideración la ocupación del área como también el nivel de iluminancia requerido en luxes. Desde una perspectiva más general, el sistema se basa en una topología inalámbrica WSN con el uso de la tecnología WiFi y el protocolo MQTT para el intercambio de información, lo cual supone una comunicación centrada en datos, otorgando al sistema un alto grado de flexibilidad y escalabilidad. Por su parte, el sistema de monitoreo-control y gestión de la información, fue implementado en el ordenador central del sistema (Raspberry Pi), permitiendo almacenar el servidor local MQTT, la herramienta Node-RED para la programación de las operaciones y un servidor LAMP para la gestión de la base de datos local del sistema. Los resultados experimentales demuestran la robustez del sistema propuesto y su capacidad para ahorrar energía, así pues, el acceso y gestión de la información tanto del consumo-ahorro energético como también la reducción del gasto económico, obtenida de la implementación y las pruebas de funcionamiento realizadas, se puede analizar de manera simple mediante su visualización gráfica en la interfaz de usuario web desarrollada.

Palabras claves: eficiencia energética, iluminación inteligente, controlador PID, WSN, protocolo MQTT, Node-RED

2.1 Abstract

Energy efficiency is one of the area of study of a greatest interest, while lighting systems present a range of possibilities for the development of technologies and prototypes that allow maximize energy savings and to reduce the economic spending within the residential sector. The objective of the work is to reduce the energy consumption through the design and implementation of a prototype of an intelligent lighting system that integrates sensor technologies for the maximum use of natural light, in addition to allowing real-time monitoring of various environmental variables to carry out programmed actions. The constructed prototype consists of a power control circuit based on a trailing edge phase control dimmer, with a NodeMCU ESP8266 microcontroller as a processing unit, which additionally allows to regulate the dimming level of the luminaire by implementing an automatic control algorithm supported by a PID controller, which takes into account the occupation of the area as well as the required illuminance level in luxes. From a more general perspective, the system is based on a WSN wireless topology with the use of WiFi technology and the MQTT protocol for information exchange, which means data-centric communication, which gives the system a high degree of flexibility and scalability. For its part, the monitoring-control and information management system was implemented in the system's central computer (Raspberry Pi), allowing the storage of the local MQTT server, the Node-RED tool for programming operations and a server LAMP for the management of the local database of the system. The experimental results demonstrate the robustness of the proposed system and its energy saving capacity, therefore, the access and management of the information on both energy consumption-saving and economic cost reduction, obtained from the implementation and operation tests carried out, can be analyzed in a simple way through its graphic visualization in the developed web user interface.

Keywords: energy efficiency, smart lighting, PID controller, WSN, MQTT protocol, Node-RED

3 Introducción

La iluminación a nivel mundial representa el 20% de la energía generada, lo cual produce más dióxido de carbono que toda la industria de la aviación (BBC News, 2020). En consecuencia, hoy en día las tecnologías de control y gestión de la iluminación son un tema de gran interés, ya que tienen como objetivo la creación de ciudades energéticamente sostenibles.

En el Ecuador la eficiencia energética es un concepto que se fortalece en el país, y se concibe como la conjunción de acciones tendientes a consumir la menor cantidad posible de energía para realizar un proceso determinado, promoviendo el uso de fuentes renovables (CONELEC, 2013, pág. 40). Dentro del sector residencial existen numerosas estrategias, tales como el aumento de la eficiencia energética en usos térmicos y eléctricos específicos, por ejemplo, el uso de luminarias de tipo LED (*Light-Emitting Diode*, en inglés). Como resultado, el plan de Eficiencia Energética para el Ecuador estimula el desarrollo de la domótica para controlar y automatizar la gestión de la demanda de energéticos de manera eficiente, esto permite realizar el control del funcionamiento de la iluminación, climatización, agua caliente sanitaria, electrodomésticos, seguridad, comunicación, entre otros.

En concordancia con (Manotoa, 2018) en Ecuador los sistemas de iluminación LED podrían reducir el consumo eléctrico en un 50% adicional al obtenido con el programa de sustitución por focos ahorradores. Además de esto, tienen características muy esenciales, pues emiten hasta 90% menos CO₂ que las lámparas incandescentes, fluorescentes y halógenas, contando así con un equilibrio ecológico y ahorrativo debido a su funcionamiento de bajo consumo eléctrico y una vida útil mucho más duradera (EL UNIVERSO, 2018).

3.1 Problema

En lo concerniente a las residencias en el Ecuador, estas en su mayoría usan sistemas de iluminación convencionales, es decir, sistemas manuales para su encendido y apagado, sin tener ningún control sobre el consumo ni las horas de encendido, ocasionando un gasto energético innecesario reflejado en la cuota mensual de consumo de energía. Si a esto sumamos el uso de dispositivos ineficientes como lámparas incandescentes o fluorescentes actualmente aún en funcionamiento, resulta evidente la importancia del presente trabajo. Si bien una solución para reducir el consumo es el uso de luminarias tipo LED con tecnología más eficiente, su consumo se puede mejorar considerablemente con el uso de un sistema inteligente de iluminación para su funcionamiento controlado, esto a su vez impone el uso de dispositivos eléctricos con mayor tecnología para el control de potencia y lámparas LED dimerizables que permitan su atenuación

de forma correcta preservando su vida útil. En este sentido, los controladores LED del mercado actual aun suponen altos costos de inversión, lo cual limita la adaptación de estos sistemas de control en usuarios de pequeñas o medianas empresas Pymes, por lo tanto, resulta necesario la creación de un mecanismo alternativo que nos permita hacer de nuestra luminaria LED (bajo vataje) un sistema inteligente, amigable y económico.

3.2 Justificación

Evidentemente, la automatización y control dentro de una residencia permiten de una forma descentralizada y/o remota apagar, encender y regular mecanismos o sistemas cuya finalidad es la reducción del consumo energético y el confort del usuario. Por tal motivo, en este trabajo se propone un producto de sistema inteligente de iluminación basado en el uso de tecnología LED como fuente luminosa, el cual permite el monitoreo en tiempo real de variables ambientales y la ejecución automática de los procesos de encendido, apagado y regulación de intensidad luminosa, específicamente utilizando dispositivos electrónicos (*open source hardware*), y el software correspondiente (*open source software*) para su programación. Sobre todo, que el prototipo permita brindar confort en paralelo a la reducción del consumo energético dentro de los ambientes donde se instale esta tecnología.

A fin de cumplir con este objetivo, se realizó un estudio retrospectivo, analizando algunas técnicas electrónicas para el control de luminarias LED, que permitió generar una solución de bajo costo, conformando de esta manera un circuito electrónico basado en un atenuador de control de fase, es decir funciona cortando partes del voltaje de CA, permitiendo una reducción del consumo de energía en paralelo con el nivel de luz reducido. Desde una perspectiva más general, el sistema desarrollado está basado en el uso de sensores y actuadores, que en conjunto, permiten la obtención de un sistema de control que regula la iluminación necesaria de acuerdo a las condiciones del ambiente, en tanto que monitoriza en tiempo real diversas variables ambientales (temperatura, calor, humedad, iluminación) para la predicción de tendencias a partir de datos de IoT recopilados que permitan en una futura etapa la implementación de actuadores con el fin de controlar tales variables ambientales en entornos *indoor*.

En particular, para el desarrollo del proyecto se contempla diferentes etapas en las que se incluye el desarrollo electrónico y del sistema de control automático de iluminación el cual nos permitirá conseguir los objetivos planteados con la precisión que amerita nuestra necesidad tanto en el ámbito doméstico como también empresarial, en base a la Norma Técnica Ecuatoriana de “Iluminación de los lugares de trabajo en interiores” NTE INEN 2669-1. De

igual forma se incluye la etapa de adquisición de datos, el desarrollo del sistema de comunicación inalámbrico, la monitorización, y la visualización del consumo energético del sistema mediante una interfaz de usuario web almacenada en el ordenador central que trabaja en colaboración con un servidor de base de datos local, los dos alojados en una Raspberry Pi 3. En última etapa se evalúa la reducción del consumo energético mediante la implementación en el área de estudio del sistema inteligente desarrollado.

Por todo lo anteriormente expresado, el proyecto es factible ya que presenta una solución tecnológica para reducir el consumo de energía eléctrica en la empresa de estudio TECSERLED, contando con los conocimientos necesarios, la tecnología y las diferentes fuentes bibliográficas para el cumplimiento del proyecto.

3.3 Objetivos

Con la finalidad de realizar un correcto desarrollo del presente proyecto de titulación se han establecido los siguientes objetivos:

Objetivo General

- Diseñar e implementar un prototipo de sistema inteligente de iluminación y monitoreo en tiempo real de variables ambientales en la empresa TECSERLED, bajo la utilización de software y hardware abierto.

Objetivos Específicos

- Emplear las tecnologías de microcontroladores y sistemas de desarrollo embebido *open source*, en un sistema inteligente de iluminación.
- Diseñar e implementar un sistema que permita controlar automáticamente la intensidad de iluminación.
- Diseñar e implementar un sistema de monitoreo en tiempo real *indoor*, de humedad, temperatura, luminosidad, y plasmar la información en una aplicación web.
- Analizar el ahorro energético y reducción del gasto económico por la implementación del sistema de control automático de iluminación.

4 Marco teórico

El presente apartado es un resumen del estado del arte con el que se cuenta en referencia a los temas relacionados con el trabajo de titulación, dado que es importante establecer una breve base teórica, en la cual se detallan conceptos fundamentales relacionados con la iluminación inteligente, la tecnología LED, las diferentes formas de control y regulación de su intensidad luminosa, así como conceptos de microcontroladores que son el centro de procesamiento de datos de todo sistema embebido y otros dispositivos hardware como sensores que permiten la adquisición de datos para su posterior análisis.

4.1 Antecedentes investigativos

La innovación y los avances tecnológicos han permitido el desarrollo de sistemas de control en diversos campos de la ciencia permitiendo el progreso de dispositivos industriales como también domésticos, en estos últimos se destaca su incorporación en distintos artefactos que facilitan la vida cotidiana del hombre. Como se afirmará luego “un sistema de control está compuesto por un grupo de elementos que en conjunto permiten un control absoluto de un proceso determinado y tiene como objetivo completar de manera efectiva las tareas y asignaciones para las cuales fue programado” (Alejandro, 2021).

En el campo de la eficiencia energética, los sistemas de control desempeñan un papel crucial para la optimización, y reducción de costos, así pues, en base a esta línea de investigación se han implementado algunos proyectos relacionados al uso eficiente de la energía, en el contexto de las luminarias se tiene por ejemplo, el trabajo desarrollado por (Huamán, 2017) con el tema “CONTROL INTELIGENTE DE SISTEMAS DE ILUMINACIÓN EN EDIFICIOS” el cual obtiene como resultado un sistema embebido con árboles de decisión como sistema de adaptación de la iluminación al usuario y como controlador del nivel mínimo de iluminación un control PID adaptativo, permitiendo de esta manera un control inteligente de la iluminación dentro de cualquier ambiente interior. Así mismo como punto adicional, se ha de mencionar el siguiente trabajo desarrollado por (Outeiriño, 2020) con el título “TECNOLOGÍAS FOTÓNICAS Y MODELOS MATEMÁTICOS PARA ALUMBRADO INTELIGENTE Y EFICIENCIA ENERGÉTICA” el cual da una introducción al concepto de iluminación inteligente con luminarias LED en el contexto del Internet de las Cosas o IoT (*Internet of Things*, en inglés) y también concluye que la iluminación LED inteligente representa una enorme oportunidad tanto de I+D+i como de negocio.

Hay que hacer notar que las investigaciones actuales buscan el diseño de sistemas inteligentes de iluminación basados en redes de sensores inalámbricos distribuidos (WSN o *Wireless Sensor Network*, en inglés), aquí se incluyen sensores PIR o Doppler de microondas para la detección de presencia, y sensores de luz ambiental para el aprovechamiento máximo de la luz natural. En relación a la idea anterior se tiene por ejemplo el siguiente artículo científico en la base de datos del MPDI (*Multidisciplinary Digital Publishing Institute*, en inglés) desarrollado por (Yusi, Chen, & Jingfeng, 2020) con el tema “DESIGN AND APPLICATION OF A SMART LIGHTING SYSTEM BASED ON DISTRIBUTED WIRELESS SENSOR NETWORKS” donde uno de sus componentes más importantes es el análisis de factores tales como la densidad de flujo de personas y la iluminación deseada, dando como resultado ahorros de energía de alrededor del 40%.

Por otra parte, han existido avances en el desarrollo de dispositivos electrónicos para el control de potencia de la iluminación, en base a esta línea de investigación existen diferentes artículos científicos en la base de datos de la IEEE (*Institute of Electrical and Electronics Engineers*, en inglés). Por ejemplo, el trabajo propuesto por (Kim, Jung, Ryu, & Baek, 2011) con el tema “A SIMPLE DIMMER USING A MOSFET FOR AC DRIVEN LAMP” presenta un atenuador PWM que utiliza dos MOSFET para el accionamiento de una lámpara incandescente, obteniendo un amplio rango de atenuación y libre de parpadeo, haciendo alusión que el atenuador PWM propuesto puede producir una onda sinusoidal y no causa problemas de armónicos. Otro artículo de la IEEE relacionado al desarrollo de circuitos atenuadores fue el desarrollado por (SangCheol, Koo, & Gun-Woo, 2015) con el título “DIMMING-FEEDBACK CONTROL METHOD FOR TRIAC DIMMABLE LED DRIVERS” el cual propone un método de control de retroalimentación de atenuación que tiene una ganancia de bucle constante en CC para controladores LED atenuables, en resumen, el circuito atenuador TRIAC controla la tensión de entrada RMS (*Root Mean Square*, en inglés) cortando el ángulo de fase de la tensión, de esta manera el circuito controla la corriente de salida del LED y por ende la potencia enviada a las lámparas.

En respuesta al reto de mejorar la tecnología para el aprovechamiento energético, la UNL ha propuesto una línea de I+D+I a través del proyecto SMART CAMPUS, esto mediante la implementación de tecnologías y fabricación de prototipos electrónicos. En el contexto de los sistemas de iluminación actualmente existe un primer prototipo de “Luminaria Inteligente”, cuyo funcionamiento está dado por un conjunto de dispositivos hardware y sensores que permiten el control de la luminaria con suministro AC (corriente alterna). En concordancia a

los objetivos de este proyecto por parte de la UNL, con el desarrollo del presente trabajo se da un paso más de carácter innovador en el desarrollo de un sistema inteligente de iluminación.

4.2 Fundamentación teórica

4.3 Sistema de iluminación inteligente

Un sistema de iluminación inteligente tiene un gran impacto en el ahorro de energía, la comodidad, la salud y la seguridad del usuario.

En comparación con el sistema de iluminación tradicional basado en el encendido/apagado manual, un sistema de iluminación inteligente aprovecha los avances en los diodos emisores de luz (LED) y las tecnologías de la información y las comunicaciones (TIC) como redes inalámbricas y sensores para crear una iluminación eficiente y saludable, usando información diversa como la presencia de ocupantes y la luz del día disponible.

Del mismo modo se puede lograr un mayor ahorro de energía, optimizando los niveles de atenuación de los LED con el objetivo de minimizar el consumo de energía de todos los LED sujetos a las limitaciones de iluminación. Los sistemas de iluminación en aplicación real se enfrentan a tres desafíos principales: primero la gran cantidad de luminarias que se deben controlar, segundo que se requiere un equilibrio entre el rendimiento del sistema y el costo del ciclo de vida, estos costos incluyen los gastos de diseño, fabricación y mantenimiento, y tercero que los sistemas de iluminación inalámbricos deben hacer frente al gran tráfico de comunicaciones, lo cual puede provocar mayores retrasos. (Yusi, Chen, & Jingfeng, 2020)

4.4 Luminotecnia

“La luminotecnia es la ciencia que estudia las distintas formas de producción de luz, así como su control y aplicación, es decir, es el arte de la iluminación con luz artificial para fines específicos” (González, 2019).

El objetivo de la luminotecnia es decidir los niveles óptimos de iluminación para una instalación concreta. En luminotecnia algunas de las lámparas utilizadas para iluminación son las incandescentes, fluorescentes, halógenas y lámparas LED, siendo éstas últimas las más eficientes como se afirmará en párrafos siguientes.

4.4.1 Magnitudes usadas en luminotecnia

Las magnitudes y unidades que se emplean para cuantificar y establecer las características más idóneas de las fuentes de luz para aplicaciones en luminotecnia se detallan a continuación.

Tabla 1 Resumen de magnitudes y unidades luminosas fundamentales

Magnitud	Símbolo	Unidad	Definición de la unidad	Relación
Flujo Luminoso	Φ	Lumen [lm]	Flujo emitido en un ángulo sólido unidad por una fuente con una intensidad luminosa de una candela	$\Phi = I * W$
Rendimiento Luminoso	η	Lumen por vatio [lm/W]	Flujo luminoso emitido por unidad de potencia eléctrica	$\eta = \frac{\Phi}{W}$
Cantidad de luz	Q	Lumen por Segundo [lms]	Flujo luminoso emitido por unidad de tiempo	$Q = \Phi * t$
Intensidad luminosa	I	Candela [cd]	1/60 de la intensidad luminosa por cm^2 del “cuerpo negro” a la temperatura de fusión del platino (2046°K)	$I = \frac{\Phi}{W}$
Iluminancia	E	Lux [lx]	Flujo Luminoso de un lumen que recibe una superficie de $1 m^2$	$E = \frac{\Phi}{S}$
Luminancia	L	Candela por m^2 [cd/m^2]	Intensidad luminosa de una candela por unidad de superficie	$L = \frac{I}{S}$

Nota. Adaptado de (Guanín, 2015)

4.5 Iluminación LED

Son sistemas que utilizan diodos emisores de luz LED (*Lihgt Emitting Diode* en inglés), como la principal fuente de iluminación gracias a las características lumínicas que poseen.

La emisión de luz es el resultado del proceso electrónico que produce la corriente eléctrica al pasar por una unión semiconductora, presentando valores significativos de eficiencia energética y vida útil. Las nuevas lámparas LED generan gran expectativa en cuanto a las prestaciones que ofrecen, además, permiten la interacción de dispositivos electrónicos para su control y correcto funcionamiento (Gago & Fraile, 2012).

4.6 Diodos emisores de luz

El diodo LED es un dispositivo tipo semiconductor que emite luz cuando este está polarizado directamente. Este tipo de diodos emplean el mecanismo de electroluminiscencia para emitir luz. Entre algunas de sus particularidades es que no generan luz ultravioleta (UV) ni infrarroja, tampoco poseen partes sensibles al movimiento como los filamentos de las lámparas incandescentes. Por otra parte, el tiempo de vida que permite esta tecnología es de alrededor de entre 15,000 a 40,000 horas, muy por encima del resto de tecnologías de iluminación. (Gago & Fraile, 2012)

4.6.1 Principio de funcionamiento

Los diodos están contruidos por dos tipos de materiales llamados tipo N y tipo P, estos se encuentran unidos a través de una junta llamada barrera o unión. Los materiales semiconductores puros que llevan impurezas en mínimas cantidades han atravesado un proceso llamado dopaje. Dependiendo del tipo de dopaje que se hace al material semiconductor, este puede presentar dos tipos:

- Material Tipo N, tiene electrones libres (exceso de electrones).
- Material Tipo P, tiene huecos libres (ausencia o falta de electrones).

Al unir los dos materiales parte de los electrones de N se trasladan a P, de la misma forma los huecos del cristal tipo P pasan a N. En esta unión se crea una franja llamada zona de transición. Cuando se aplica una diferencia de potencial con polaridad positiva al lado P y polaridad negativa al lado N, la zona de transición se vuelve estrecha permitiendo que el flujo de electrones circule libremente a través del diodo, el sentido de la corriente será desde P hacia N. En el caso de polarizar inversamente, la banda de transición se vuelve mucho más ancha e

impide el flujo de electrones a través de la misma, por lo cual, el diodo se comporta como un interruptor abierto. (Luis & Danny, 2015)

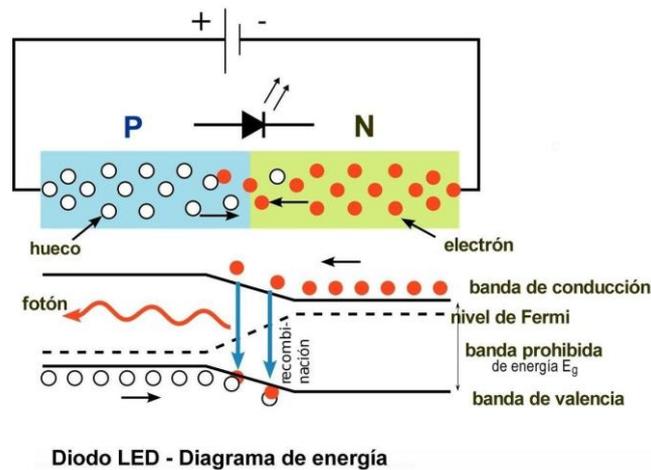


Figura 1 Diagrama de energía de un diodo LED

Nota. La imagen representa la polarización correcta del LED para permitir fluir la corriente de P a N. Adaptado de WikiFisica2013, CC BY-SA 3.0, vía Wikimedia Commons.

4.6.2 Tipos de LED existentes

Se puede hacer una clasificación de los distintos tipos de Leds disponibles en el mercado, basándose en su potencia nominal o encapsulado (Miguel, 2018), en base a esto se realiza la siguiente clasificación:

- **LED DIP o Dual In-Line Package:** Fueron los primeros chips y los más básicos, su diseño consta de un diodo dentro de una carcasa de plástico duro transparente, del que sobresalen dos pines de conexión en paralelo. Posee una baja eficiencia lumínica (4 Lúmenes por LED). Su uso es más extendido como indicador luminoso en diferentes productos electrónicos.
- **LED SMD o Surface Mounted Diode:** como su nombre lo indica el diodo está montado sobre una superficie con un encapsulado de resina semirrígida que se suelda al circuito de forma superficial. Son mucho más pequeños y eficientes que los originales chips DIP. Permiten la inclusión de 3 diodos en el mismo chip, así se consigue reproducir luz blanca la cual es la unión de los 3 colores primarios, rojo, verde y azul, o a su vez una gama de hasta 16 millones de combinaciones de colores RGB. Su alta eficiencia les permite producir entre 60 y 80 lm/W. Una de sus desventajas es que no son recomendables para funcionar prolongadamente debido a la temperatura que se genera en la unión del chip a la placa.

- **LED COB o Chip on Board:** su diseño se basa en un conjunto de Leds agrupados en serie o paralelo, reduciendo hasta un 20% el coste de fabricación respecto a los chips SMD. Ofrecen un rendimiento lumínico superior a 120 lm/W, siendo muy usados en focos de potencia de hasta 50 W. Además, su diseño permite una disipación de calor eficiente, y una buena durabilidad debido a su capacidad para soportar los cambios de tensión en la corriente eléctrica.
- **Micro LED:** estos chips han permitido la generación de nuevas tecnologías de televisores de pantallas planas, en la que cada pixel está formado por su propia fuente de luz. A diferencia de OLED, Micro LED utiliza material inorgánico y permite pantallas de mayor brillo, mejor contraste, menor tiempo de respuesta (latencia) y mayor eficiencia energética.



Figura 2 Tipos de Leds en el mercado actual.

Nota. LED DIP, LED SMD, LED COB respectivamente. Adaptado de (EfectoLED Blog, 2017)

4.7 Controladores LED en el mercado actual (drivers LED)

Los controladores LED del mercado actual permiten suministrar y adaptar la energía necesaria para el correcto funcionamiento de una lámpara LED.

Los Leds trabajan con corriente continua (CC), por ende, si se requiere que la lámpara funcione en una instalación de corriente alterna (AC) el driver es el encargado de realizar esta conversión de AC-DC, además de proporcionar la cantidad de corriente requerida y a su vez disminuir el voltaje a las necesidades de la luminaria, garantizando su correcto desempeño. (LEDBOX, 2019)

Con respecto al proceso de regulación y control, algunos drivers LED poseen entradas adicionales que permiten modificar la intensidad luminosa, como un sistema PWM (modulación por ancho de impulso) que permite variar rápidamente el suministro de energía entregado al LED, lo que permite una atenuación del mismo. Las características de los Leds pueden cambiar considerablemente con la temperatura, que como resultado puede ser perjudicial, a menos que se controle cuidadosamente. (ARROW Electronics, 2021)

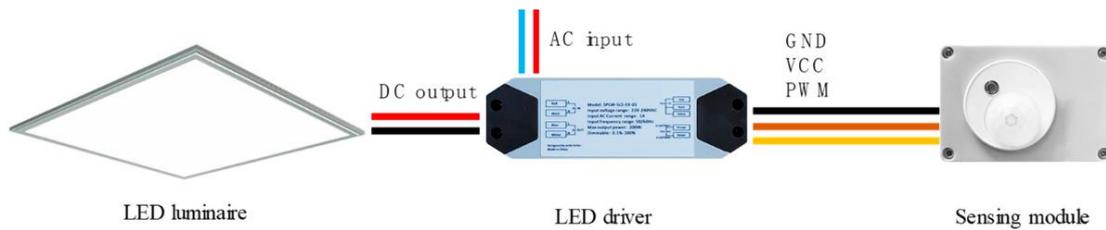


Figura 3 Driver LED externo

Nota. La imagen representa el diagrama de conexión de un driver LED externo, y su aplicación para la regulación de iluminación. Adaptado de (Yusi, Chen, & Jingfeng, 2020)

Como puede inferirse el propósito de un driver LED es salvaguardar la eficiencia energética mediante el aprovechamiento total de la energía eléctrica. Este aprovechamiento se mide gracias al valor de factor de potencia (PF o *Power Factor* en inglés), por consiguiente, cuanto más se acerque este valor a la unidad mayor será la energía aprovechada en la conversión (LEDBOX Blog, 2017).

Sucede pues, que en la mayoría de hogares se utilizan lámparas domésticas LED donde el driver se encuentra instalado internamente en la lampara, usualmente son de menor tamaño y potencia, pero poseen algunas similitudes y diferencias con respecto a las lámparas de driver externo.

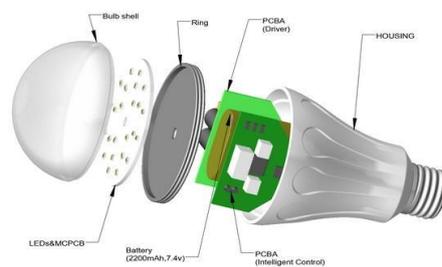


Figura 4 Representación de foco LED con driver interno

Nota. Adaptado de ONULEC, (<https://www.onulec.com/blog/noticias-sector-electrico/234-como-funcionan-las-bombillas-led>)

En la Tabla 2 se hace una breve comparativa que permite analizar qué tipo de lámpara se adapta mejor a la instalación eléctrica, y permite decantarse por un tipo de driver por sobre el otro.

Tabla 2 Comparativa entre drivers LED internos y externos

	Drivers Internos	Drivers Externos
Ventajas	-Instalación más rápida y fácil por cualquier usuario.	-Productos de mayor potencia y luminosidad.

	<ul style="list-style-type: none"> -Solución todo en uno, sin necesidad de espacio externo en la luminaria. -Precio ligeramente inferior. 	<ul style="list-style-type: none"> -En caso de falla, se puede sustituir el driver, salvaguardando la luminaria. -Luminarias de tamaño reducido. -Mayor vida útil para todos los componentes. -Más versatilidad en cuanto a mantenimiento. -Mayor fiabilidad. -Su diseño permite la disipación del calor hacia el exterior, manteniendo integra la luminaria.
Desventajas	<ul style="list-style-type: none"> -Requiere un espacio mayor en el interior del bloque óptico de la luminaria. -Si hay exceso de calor, un fusible térmico (PTC) interrumpirá el funcionamiento para salvar la integridad de la bombilla. -Luminarias de menor potencia y luminosidad. -Cuando el driver llega al final de su vida útil, hay que sustituir la luminaria. 	<ul style="list-style-type: none"> -Necesita de un compartimiento adicional para la instalación del driver externo. -La instalación debe ser realizada por personal cualificado. -Precio más elevado. -Ocupa espacio externo fuera de la luminaria. -Menor oferta en el mercado LED.

Nota. Adaptado de (LEDBOX Blog, 2017)

4.8 Sistemas de iluminación LED inteligentes en el mercado actual

Dentro del mercado actual existen muchos productos de iluminación que permiten un funcionamiento inteligente del mismo, esto mediante la reducción del nivel de luz por ajuste personalizado o automático, la incorporación de sensores de presencia para su encendido o apagado e incorporación de aplicaciones instaladas en un *smartphone* o *tablet* para su control. Para ejemplificar tales características se presenta algunas empresas de todo el mundo que han desarrollado productos de iluminación LED inteligentes, como son los siguientes:

Tabla 3 Productos inteligentes de iluminación LED en el mercado actual.

Empresa	Sistema de Iluminación Inteligente
Philips	Ofrece tres tipos diferentes de bombillas LED regulables: atenuación estándar con un interruptor de atenuación, atenuación SceneSwitch con su interruptor de luz existente e iluminación inteligente que se atenúa con una aplicación instalada en un Smartphone.
Hunt Dimming	Posee un controlador de iluminación Simplicity ® PhotoSlide™ con encendido momentáneo que combina la regulación manual y el control de

	aprovechamiento de la luz natural para la iluminación LED en un único dispositivo de montaje en pared. Este controlador deslizante permite a los usuarios conmutar o atenuar la iluminación LED de 120V o 277V.
Helia de Soraa	Es un sistema de iluminación para el hogar, la bombilla inteligente Helia de Soraa posee un modo nocturno eficiente para tener un sueño perfecto. La bombilla utiliza Leds violetas en lugar de la luz azul. El centro de la lámpara viene con un módulo Smart Snap, este detecta si el espacio está ocupado o vacante, también ajusta el brillo según el progreso del día. Además, posee un sistema Cloud Connect para hacer que este bulto inteligente se conecte con otros sistemas IoT.
Xiaomi	La lámpara de techo inteligente Xiaomi está equipada con 72 Leds de un índice de reproducción de color hasta 95 Ra, con regulación de intensidad. Al ser una lámpara inteligente la luz se puede controlar por teléfono celular a través de WiFi o Bluetooth. Además, incluye sensores de presencia de humanos, y puede ser instalada en dormitorios, salas de estudio, comedores, hoteles, etc.
Lidl	Posee un modelo de lámpara con una luz de tipo LED de 10 vatios de potencia que integra un sensor de movimiento con cobertura de 360°, además puede montarse tanto en interiores como en exteriores gracias a que cuenta con certificación IP44, posee hasta seis niveles de luz ajustables en el tiempo, pero no posee una regulación automática.

4.9 Regulación y control de luminarias LED con suministro AC

La regulación de luminarias LED se hace posible a través de una técnica llamada dimerización, en la cual se varía la potencia entregada a la luminaria LED y por ende el nivel de luminosidad entre otros parámetros. Una característica de los drivers modernos es que existen fabricantes que permiten la dimerización de sus lámparas LED mediante el uso de atenuadores de control de fase, esta característica se denota en las especificaciones técnicas de la lámpara LED y es de vital importancia para su compatibilidad con la técnica de dimerización, a la vez que permitirá una mayor eficiencia energética, preservando la vida útil de la lámpara. Aquí la regulación se realiza conectando el regulador entre la línea de alimentación y la luminaria.

4.10 Dimerización de luminarias LED mediante atenuadores de control de fase

Un hecho importante dentro de la atenuación de la iluminación es que los atenuadores electrónicos modernos permiten una reducción en el consumo de energía en paralelo con el nivel de luz reducido, y esto presenta una valiosa oportunidad de ahorro de energía. Un atenuador de iluminación funciona esencialmente cortando partes del voltaje de CA, esto permite que solo partes de la forma de onda pasen a la lámpara. El brillo de la lámpara está determinado por la potencia que se le transfiere, por lo que cuanto más se corta la forma de onda, más se atenúa. (Won-Seok, 2013)

Los reguladores de control de fase (o corte de fase) disponibles son "borde posterior" y "borde delantero", se diferencian principalmente en los componentes y costo para su fabricación, además funcionan de diferentes maneras, lo que finalmente afecta su compatibilidad con ciertas luminarias. (XIAMEN HUZZDA , 2019)

Casi todos los reguladores fabricados en el mundo antes del año 2010 fueron TRIAC, por eso, durante mucho tiempo, los controladores LED no funcionaron bien conectados a estos reguladores. Sin embargo, los principales fabricantes de controladores LED han trabajado para que ambos tipos de reguladores de control de fase sean útiles para la regulación de los Leds. (LEDBOX, 2019)

4.10.1 Atenuadores de control de fase de borde delantero

La atenuación de borde delantero realiza un corte de fase en el encendido de la onda, el atenuador de vanguardia se usa tradicionalmente para cargas inductivas, además tiene una carga mínima relativamente alta que a menudo restringe su uso con lámparas LED o CFL eficientes (XIAMEN HUZZDA , 2019). Según (Won-Seok, 2013) trabajador de Fairchild Semiconductor:

Los atenuadores de borde delantero basados en un TRIAC poseen algunas desventajas que conducen a una alta interferencia electromagnética EMI (*Electromagnetic Interference*, en inglés) y bombillas ruidosas. Esto debido a que en ciertas partes de la forma de onda se genera un aumento muy rápido del voltaje lo que a su vez provoca un flujo de corriente instantáneo extremadamente alto (pico de corriente) durante el encendido de la lámpara, provocando un mal funcionamiento cuando la carga electrónica es no resistiva. Un TRIAC es un dispositivo semiconductor que permite el paso de corriente AC mediante una señal de disparo en cada semiciclo de la señal AC.

Para ejemplificar el problema antes mencionado, la siguiente Figura 5 muestra como el dispositivo TRIAC se enciende en un ángulo de fase seleccionado durante el medio ciclo de la señal de entrada. En este caso particular se puede analizar que el tiempo de subida de la entrada de la red es increíblemente rápido, provocando un pico de corriente durante el encendido de la lámpara. Esto ejerce una enorme presión sobre el atenuador y sobre los componentes electrónicos de la lámpara (no resistiva), lo que a su vez repercute en la vida útil de la misma.

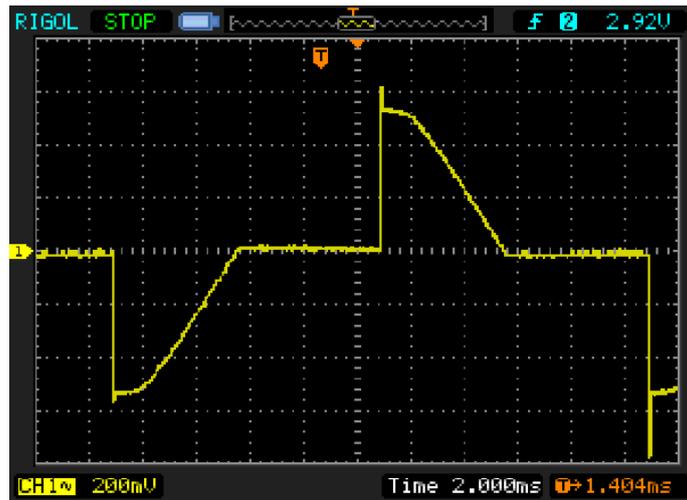


Figura 5 Forma de onda de corriente del atenuador TRIAC capturada (carga incandescente de 75 W).

Nota. Adaptado de (Elliott, Lighting Dimmers, 2017)

La Figura 6 muestra el diseño y simulación de un atenuador comercial de borde delantero, el cual se basa en un circuito de dos etapas, el detector de cruce por cero y el control de TRIAC. De este modo, mediante el ajuste del ángulo de disparo del TRIAC se permite manipular la potencia entregada a la carga, dicho en otras palabras, al variar el RMS que es el valor eficaz de voltaje que se entrega a la carga, se varía la potencia entregada. (AX, Desarrollos, 2013)

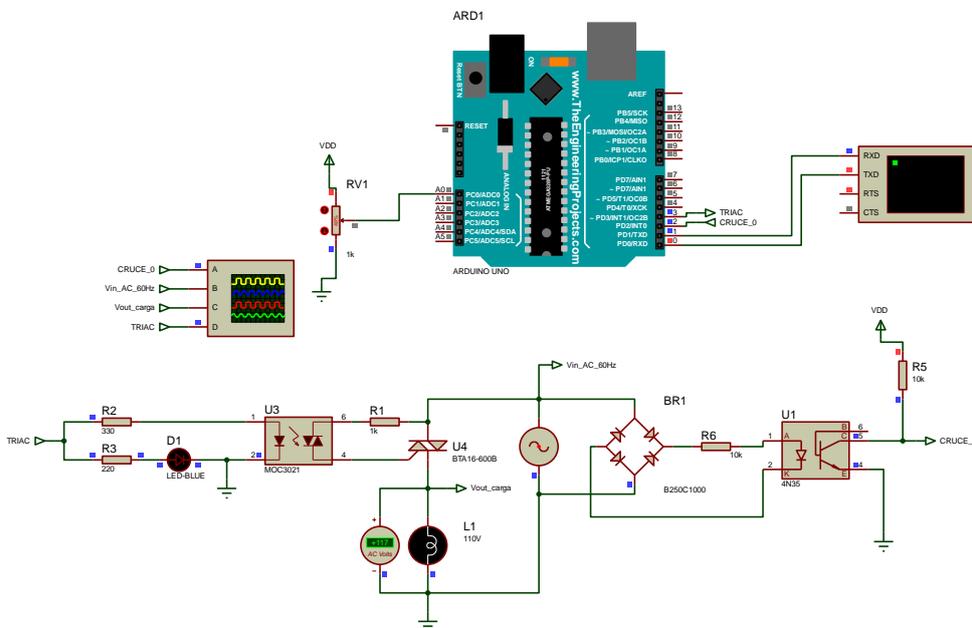


Figura 6 Diagrama de circuito atenuador de corte de fase de dos cables basado en TRIAC

Nota. Simulación realizada en Proteus 8 Professional

La Figura 7 muestra la forma de onda resultante al variar la potencia aplicada a la carga y la Figura 8 muestra un ejemplo de un circuito comercial utilizado para el control de fase directo.

“Los atenuadores de vanguardia (basados en TRIAC) son más baratos y simples que los de borde posterior y se usaron inicialmente para atenuar bombillas incandescentes y halógenas o los transformadores magnéticos de alambre, también permiten variar la velocidad en motores AC como ventiladores, la potencia de un calentador resistivo y más”. (XIAMEN HUIZDA , 2019)

Según (Elliott, 2017) las cargas inductivas (como los transformadores de núcleo de hierro convencionales o los motores de ventiladores eléctricos comunes) son bastante seguras con atenuadores de vanguardia, porque la inductancia limita el tiempo de subida de la corriente a valores seguros. De igual forma afirma que las lámparas incandescentes son ideales para este método de control, no obstante, las lámparas CFL o LED no regulables nunca deben conectarse a un circuito atenuador de este tipo, incluso si el atenuador está ajustado al máximo, ya que la corriente consumida por el circuito de la lámpara puede aumentar drásticamente (5 veces o más) ocasionando interferencias y reduciendo la vida útil de los componentes electrónicos de la lámpara.

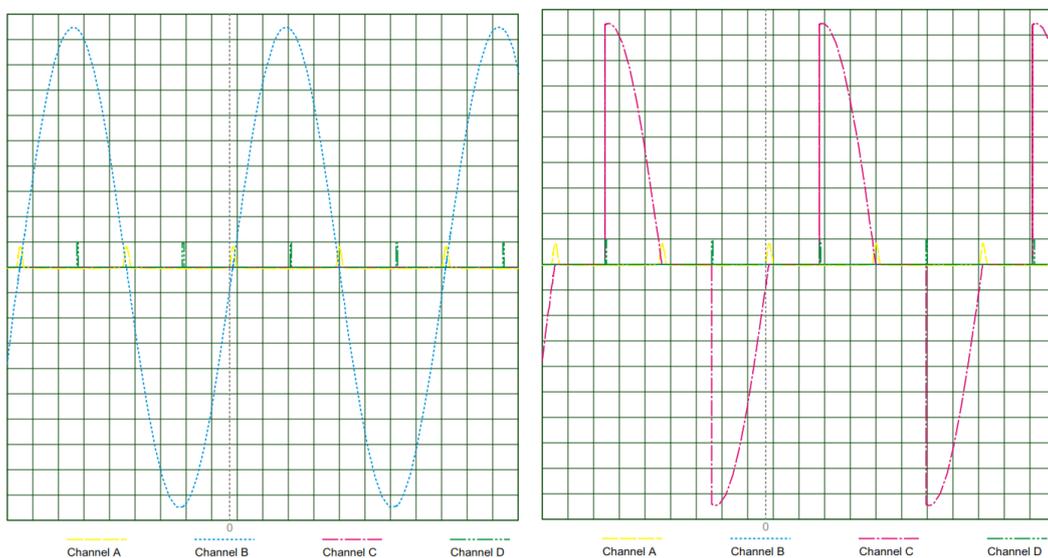


Figura 7 Formas de onda y fase recortada usando atenuador de TRIAC.

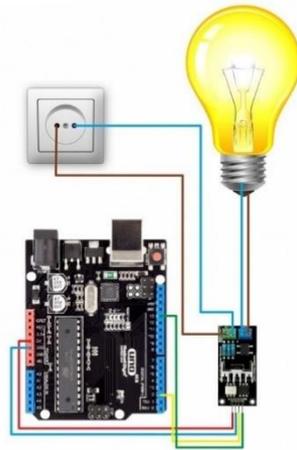


Figura 8 Módulo dimmer TRIAC y su aplicación junto a un Arduino Uno

Nota. Adaptado de MaxElectrónica (<https://maxelectronica.cl/prototipo/625-modulo-dimmer-ac-optoacoplado-1ch-bta16600b-logica-control-33-5v.html>)

Cabe destacar que el circuito atenuador TRIAC controla el ancho de pulso de voltaje de CA entregado a la carga apagando el TRIAC cuando su corriente de retención está por debajo del nivel de umbral mínimo, esto ocurre cerca del cruce por cero de CA. De este modo cuando la corriente cae por debajo del mínimo para el TRIAC, se apaga. Esto es un problema para las lámparas LED de bajo vataje en particular, porque consumen relativamente poca corriente, por lo que se alcanza un punto en el ciclo de atenuación en el que la carga no consume suficiente corriente para que el TRIAC permanezca encendido, como resultado el TRIAC puede apagarse casi inmediatamente después de activarse, y la lámpara parpadeará o se apagará por completo. (Elliot, 2015)

4.10.2 Atenuadores de control de fase de borde posterior

Este tipo de atenuador generalmente utiliza un circuito más complejo, debido a que el dispositivo de conmutación debe encenderse apenas la onda de CA pasa por cero (Cruce por cero) y posteriormente apagarse en cualquier instante de tiempo de la forma de onda restante usando algún dispositivo de control. Para lograr este funcionamiento, se requiere del control absoluto de la conmutación, lo cual no es permitido con el uso de dispositivos TRIAC debido a la corriente de retención. En consecuencia, se usan dispositivos MOSFET (*metal-oxide-semiconductor field-effect transistor*, por sus siglas en inglés) o IGBT (*insulated gate bipolar transistor*, por sus siglas en inglés) ya que no requieren corriente de control, son resistentes y confiables. Por lo general los MOSFET se caracterizan por una conmutación más rápida pero una mayor pérdida de conducción en comparación con los IGBT, especialmente en un rango de corriente alto. (Won-Seok, 2013)

Esta característica de corte de fase en el apagado de la onda, causa menos interferencias en comparación con la atenuación de borde delantero. En concordancia con (Won-Seok, 2013), al usar IGBT o MOSFET para controlar las luces, la atenuación puede comenzar en el borde posterior del ciclo de energía como se muestra en la Figura 10, lo cual permite un encendido y apagado suave, eliminando así los rápidos tiempos de subida que producen altos picos de corriente dañinos para el atenuador o la carga electrónica, de esta manera permite prolongar la vida útil de las lámparas.

En este contexto, una bombilla halógena puede modelarse como una carga resistiva, pero una bombilla LED que viene con condensadores, inductores y controladores LED, posee una carga más compleja. Por lo tanto, su regulación mediante un atenuador TRIAC como el analizado anteriormente puede provocar un mal funcionamiento o afectar la vida útil de la lámpara LED. Debido a que un atenuador de borde posterior es adecuado para cargas capacitivas y adicionalmente posee una carga mínima más baja que los atenuadores de borde delantero, lo convierte en una mejor opción para atenuar circuitos de iluminación de baja potencia de tamaño modesto (p.ej. Luminarias LED). (XIAMEN HUZZDA , 2019)

La siguiente Figura 9 muestra un diagrama de bloques que ejemplifica las diferentes partes que componen un circuito atenuador de borde posterior de tres cables (activo, carga y neutro), el cual permite su funcionamiento independientemente de la carga, ya que los MOSFET se encienden mediante una señal de control, independientemente de la corriente consumida por la carga (Elliot, 2015). El circuito se puede dividir en 4 partes: la primera el detector de cruce por cero, el cual normalmente se compone por un circuito optoacoplador, segundo una fuente de alimentación con salida CC, la tercera se compone de un circuito relé de estado sólido SSR (*Solid State Relay*, por sus siglas en inglés) para la conmutación de los MOSFET y finalmente un circuito de control que permite variar el tiempo de encendido de los dispositivos MOSFET y en consecuencia controlar la potencia entregada a la carga.

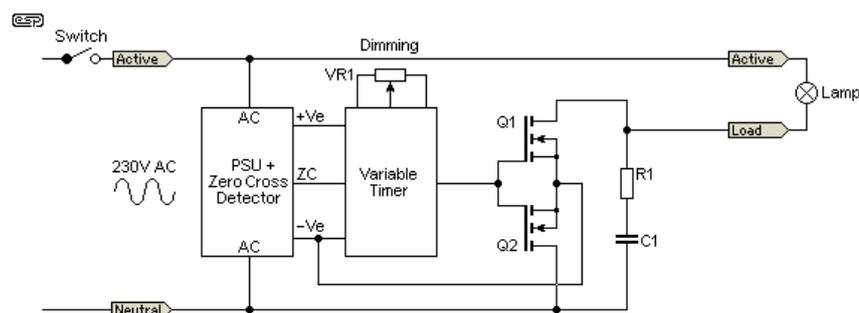


Figura 9 Atenuador de corte de fase basado en MOSFET de 3 cables

Nota. Adaptado de (Elliot, 2015)

Del mismo modo, para ejemplificar el funcionamiento de un atenuador de borde posterior la siguiente Figura 10 muestra como el dispositivo MOSFET se enciende en un ángulo de fase seleccionado durante el medio ciclo de la señal de entrada. Se puede inferir que en este caso el voltaje de salida aumenta lentamente y se apaga de manera rápida, por lo cual se eliminan los picos de corriente masivos que causaban los atenuadores de vanguardia. Por ende, según (Elliott, 2017) algunas lámparas CFL o LED regulables funcionan perfectamente bien con este tipo de atenuador, no obstante, nunca debe usarse para cargas de motor o transformadores con núcleo de hierro.

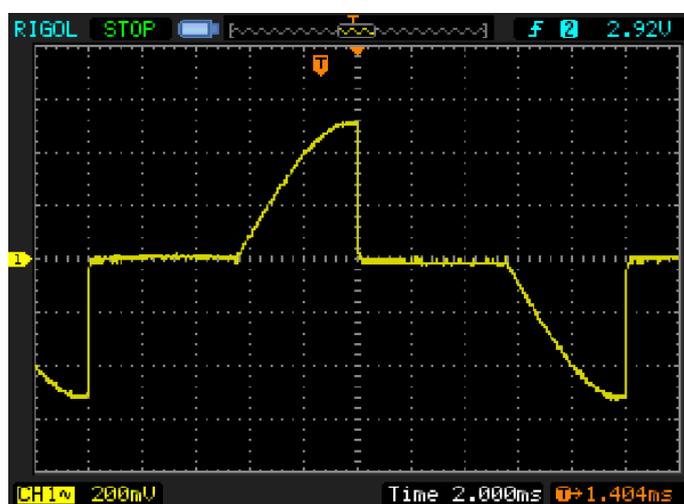


Figura 10 Forma de onda de corriente del atenuador de borde posterior capturada

Nota. Adaptado de (Elliott, Lighting Dimmers, 2017)

4.11 Principios de factor de potencia aplicados a los sistemas de iluminación

El factor de potencia o FP en adelante, es un indicador del correcto aprovechamiento de la energía eléctrica dentro de una instalación. La potencia eléctrica se define como la capacidad de una máquina eléctrica de producir o demandar energía en un instante de tiempo determinado, existen tres tipos diferentes de potencias dentro de un circuito, tales como:

- Potencia activa (P): su unidad de medida es el vatio [W], también llamada potencia media o real, es aquella que consume cualquier carga puramente resistiva y le permite realizar un proceso de transformación de la energía eléctrica en trabajo. Se calcula como:

$$P = V \times I \times \cos\varphi = W \quad (1)$$

El $\cos\phi$ también se conoce como "Factor de Potencia", es el ángulo de desfase entre el voltaje V y la corriente I.

- Potencia reactiva (Q): su unidad de medida es el voltamperio reactivo [VAr], representa la capacidad de las bobinas y condensadores de almacenar energía eléctrica en su interior para generar campos magnéticos o eléctricos respectivamente, pero no se transforma en trabajo efectivo, sino que fluctúa por la red entre el generador y los receptores (Proyecto987). Esta potencia no supone un consumo en los receptores, aunque sí hay que generarla y transportarla hasta los equipos, generando el desgaste de la red. Se calcula como:

$$Q = V \times I \text{ seno } \phi = VAr \quad (2)$$

- Potencia aparente (S): su unidad de medida es el voltamperio [VA], es la potencia total consumida por la carga y es el producto de los valores eficaces de tensión e intensidad. Se obtiene como la suma vectorial de las potencias activa y reactiva y representa la ocupación total de las instalaciones debida a la conexión del receptor. Se calcula como:

$$S = V_{RMS} \times I_{RMS} = \sqrt{P^2 + Q^2} = VA \quad (3)$$

Donde VA es el producto de los valores efectivos de tensión y corriente en la carga, y $P=\text{Re}(S)$, $Q=\text{Im}(S)$.

Como se ha afirmado antes el FP es una medida de la eficacia con la que un sistema utiliza la energía eléctrica, de forma que este valor varía de cero a uno, cuanto más bajo es este valor el sistema demandará más corriente para conseguir la misma cantidad de energía útil y realizar el mismo trabajo. El $\cos\phi$ permite una forma fácil de determinar este factor, pero en la actualidad las cargas no lineales no permiten calcular este valor usando el método tradicional, de modo que una mejor aproximación del FP consiste en calcular la siguiente expresión:

$$\text{Factor de potencia (FP)} = \frac{\text{potencia real (P)}}{\text{potencia aparente (S)}} \quad (4)$$

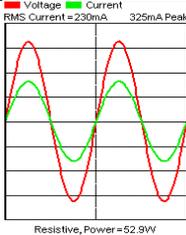
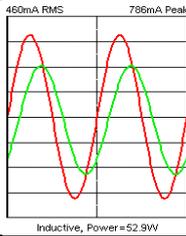
En el Ecuador con el fin de regular y mantener una buena calidad del servicio eléctrico de distribución se estableció el límite mínimo del factor de potencia en 0,92, por lo que valores por debajo de este número están sujetos a penalizaciones (CONELEC, 2001). Si bien la facturación para un inmueble residencial está dada por la energía activa (medida en kWh/mes), la energía reactiva se mide para controlar el FP promedio del usuario, de aquí la importancia de mantener una buena eficiencia eléctrica en las instalaciones.

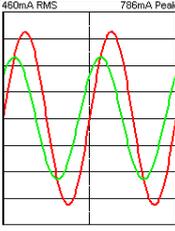
En otro orden de ideas, el FP para una lámpara incandescente se aproxima a la unidad, es decir la potencia aparente y la activa son iguales, por otra parte, el de una lámpara fluorescente compacta CFL está por debajo del de una incandescente, alrededor de 0,5 para lámparas sin corrección (Durán, 2004), adicionalmente se incluye a los sistemas de iluminación de descarga (fluorescentes, halogenuros metálicos, vapor de mercurio, etc.) los cuales han sido cargas 'problemáticas' en el pasado con un factor de potencia retardado (inductivo) ya que crean una forma de onda de corriente no lineal con pocas capacidades de corrección (Elliott, 2020). De aquí la importancia en la innovación tecnológica en los productos de iluminación, que permita mejorar el desempeño y eficiencia de la energía eléctrica en paralelo con la calidad de la luminaria.

En este sentido, dentro de los sistemas de iluminación LED existen lámparas que poseen la desventaja de tener un pobre FP lo que puede causar contaminación en el sistema de cableado electrónico, y la red eléctrica. Si bien existen productos de marcas premium donde el factor de potencia de las luces LED está entre 0,9 y 1, también los hay aquellos en los que este valor está muy por debajo de los valores recomendados, sobre todo en lámparas económicas de bajo vataje. Por tal motivo la elección de bombillas LED con un buen factor de potencia permite reducir la potencia reactiva dentro de la factura eléctrica evitando así costos por sanciones, de igual forma un buen FP asegura una alta eficiencia de la luminaria, mayor durabilidad y rentabilidad económica.

La siguiente Tabla 4 muestra el comportamiento de la corriente que puede tener una carga electrónica ya sea resistiva como un foco incandescente como también una carga no lineal formada por: fuentes de alimentación, lámparas CFL, iluminación LED, entre otras.

Tabla 4 Formas de onda de voltaje y corriente en función al tipo de carga

Tipo	Comportamiento de la corriente	Formas de Onda
Unidad	La corriente y el voltaje están en fase y tienen formas de onda idénticas (cargas resistivas)	
Retraso	La corriente ocurre después del voltaje, causada por cargas inductivas (motores, transformadores)	

Adelanto	La corriente ocurre antes que el voltaje, causada por cargas capacitivas (poco común)	
No lineal	El voltaje y la corriente están en fase, pero tienen diferentes formas de onda (muchas cargas electrónicas)	

4.12 Normativa de iluminación NTE INEN 2969-1

Una buena iluminación, además de ser un factor de seguridad, productividad y de rendimiento en el trabajo, mejora el confort visual y hace más agradable y acogedora la vida. (NEC, 2013)

Esta norma tiene como objeto establecer un conjunto de recomendaciones para una buena práctica de iluminación en lugares de trabajo en interiores en términos de cantidad y calidad de iluminación (NTE INEN 2969-1, 2015). Los valores dados por la norma son válidos para condiciones visuales normales y tienen en cuenta factores como:

- Aspectos psico-fisiológicos tales como el confort visual y el bienestar.
- Requisitos para tareas visuales.
- Ergonomía visual.
- Experiencia práctica.
- Contribución a la seguridad funcional.
- Economía.

La siguiente Tabla 5 muestra algunos valores de los requisitos de iluminación, en particular se establece el valor de iluminancia mantenida $\overline{E_M}$ en la superficie del área interior, tarea o actividad.

Tabla 5 Niveles de iluminancia mantenida recomendables

TIPO DE INTERIOR, TAREA, ACTIVIDAD	NIVEL DE ILUMINANCIA MANTENIDA $\overline{E_M}$
------------------------------------	---

Zonas de tráfico dentro de edificio	
Áreas de circulación (pasillos, corredores, etc.)	100 luxes
Escaleras, escaleras automáticas, cintas transportadoras, ascensores, montacargas.	100 luxes
Áreas generales dentro de edificios	
Vestuarios, salas de lavado, cuartos de baño, servicios	200 luxes
Almacenes y cuarto de almacén	100 luxes
Oficinas	
Salas de conferencias y reuniones	500 luxes
Mostrador de recepción y áreas de venta	300 luxes
Lugares de pública concurrencia	
Salas de ensayos	300 luxes
Área de lectura (Bibliotecas)	500 luxes
Áreas de aparcamiento	75 luxes
Establecimientos educativos	
Aulas, aulas de tutoría	300 luxes
Auditórium, sala de lectura	500 luxes
Escaleras	150 luxes
Salas de deportes, gimnasios, piscinas	300 luxes

Nota. Adaptado de (NTE INEN 2969-1, 2015)

4.13 Automatización y control

Un proceso automatizado se compone de dispositivos hardware y el software correspondiente, que, en conjunto, permiten ejercer un control y mantenimiento de sus parámetros a un valor deseado de operación, a fin de completar de forma adecuada la tarea para la cual fue previamente programada, en el caso del presente trabajo la variable a controlar es el nivel de iluminación medido en luxes.

En términos técnicos un sistema de control está compuesto de entradas o señales actuantes, y las salidas o variables controladas, en donde el objetivo principal es el completo control de las variables de salida mediante las entradas y los elementos del sistema de control (Kuo, 1996). Las principales ventajas que proporciona un sistema de control son: reducción de costos, aumento de la calidad, reducción de errores, ahorro de tiempo y mejor administración de recursos (Huamán, 2017).

4.13.1 Tipos de sistemas de control

- **Sistema en Lazo Abierto:** Es aquel sistema de control cuya salida no tiene efecto sobre la acción de control, es decir, la salida del proceso no se compara con la entrada de referencia. Cada entrada corresponderá a una operación diferente sobre la señal de salida (Manotoa, 2018). Se recomienda utilizar este modelo en sistemas donde hay la menor

cantidad posible de perturbaciones externas que no se detectan y pueden interferir con el proceso.

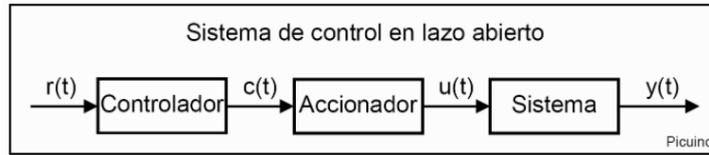


Figura 11 Diagrama de bloques de un sistema en lazo abierto

Nota. Adaptado de (Picuino, 2021)

- **Sistema en Lazo Cerrado:** un sistema retroalimentado compara el valor real de la variable de proceso y el valor deseado para controlar el sistema. La salida del proceso es utilizada para regular la amplitud de su entrada (Manotoa, 2018).

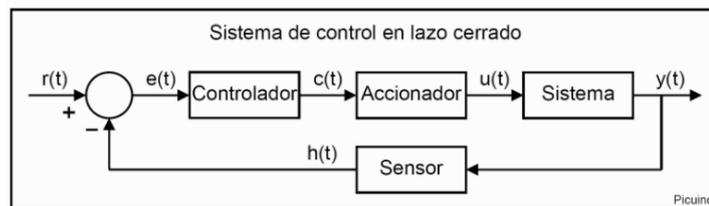


Figura 12 Diagrama de bloques de un sistema en lazo cerrado

Nota. Adaptado de (Picuino, 2021)

4.13.2 Control de encendido y apagado (on/off)

Esta es una forma de control simple y barata, razón por la cual su uso es extendido en sistemas de control tanto industriales como domésticos (Ogata, 2010, pág. 22). Este tipo de controladores se puede encontrar en sistemas basados en relés fotoeléctricos o temporizadores. Suponiendo que la señal de salida del controlador es $u(t)$ y que la señal de error es $e(t)$, la acción de control se representa mediante la siguiente función:

$$u(t) = U_1, \quad \text{para } e(t) > 0$$

$$u(t) = U_2, \quad \text{para } e(t) < 0$$

Donde U_1 representaría el valor máximo y U_2 tendría un valor mínimo o cero. El diagrama de bloques de un controlador *on/off* se representa de la siguiente manera.

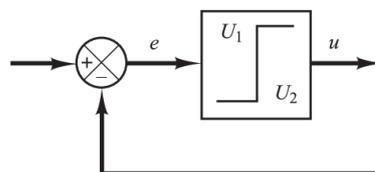


Figura 13 Diagrama de bloques de un controlador on/off

Nota. Adaptado de (Ogata, 2010)

Este modo de control es aún utilizado en sistemas de iluminación comerciales, sin embargo, no proporciona un control inteligente más allá del encendido y apagado de la lámpara. Por lo tanto, para maximizar el control de la luminaria como su regulación de intensidad luminosa, es necesario una combinación con sistemas híbridos de alta eficiencia que asocian la luz natural del medio ambiente. Esto permite una realimentación de la señal captada por el sensor de luminosidad, determinar la señal de error en base a una señal de referencia y posteriormente realizar los cálculos para el ajuste del proceso.

4.13.3 Controlador Proporcional, Integral y Derivativo (PID)

4.13.3.1 Acción de control proporcional P.

Para un controlador con acción de control proporcional, la relación entre la salida del controlador $u(t)$ y la señal de error $e(t)$ es:

$$u(t) = K_p e(t) \quad (5)$$

o bien, en cantidades transformadas por el método de Laplace,

$$\frac{U(s)}{E(s)} = K_p \quad (6)$$

donde K_p es la ganancia proporcional del proceso. El objetivo del control proporcional es aproximar el error en estado estacionario a cero, modificando el valor de K_p . Con un valor K_p alto el error de estado estacionario se reduce rápidamente, sin embargo, esto hace que la respuesta sea más oscilatoria. Por el otro lado, un valor K_p bajo obtiene una respuesta lenta ante variaciones del punto de referencia.

4.13.3.2 Acción de control proporcional-integral PI.

En un controlador con acción de control integral, el valor de la salida del controlador $u(t)$ se cambia a una razón proporcional a la señal de error $e(t)$ (Ogata, 2010, pág. 23). Es decir,

$$\frac{du(t)}{dt} = K_i e(t) \quad (7)$$

o bien,

$$u(t) = K_i \int_0^t e(t) dt \quad (8)$$

La función de transferencia del controlador integral es:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (9)$$

El objetivo del control integral es calcular la integral de la señal de error para disminuir y eliminar el error en estado estacionario provocado por perturbaciones exteriores y las cuales no se pueden corregir por el control proporcional. En otras palabras, este control actúa cuando hay una desviación entre la variable de entrada y el punto de referencia, integrando esta desviación en el tiempo y haciéndola reducir.

Con un control proporcional es necesario que exista una señal de error para obtener una acción de control distinta de cero, sin embargo, con la acción de control integral, un error pequeño positivo siempre nos dará una acción de control creciente, y si fuera negativo la señal de control será decreciente (Mazzone, 2002). En consecuencia, esto permite contrarrestar el error en régimen permanente y eliminar el off-set.

El conjunto de estos dos controladores forma un controlador PI el cual posee las ventajas de ambos, permitiendo llegar al punto de referencia de forma rápida y eliminar el error en estado estacionario al integrar el área bajo la curva del error presente en el proceso. La acción de control de un controlador proporcional-integral (PI) se define mediante:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (10)$$

La función de transferencia del controlador PI es:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s}\right) \quad (11)$$

donde T_i es el tiempo integral, y ajusta la acción integral, a menor valor de T_i el controlador integrará más rápido el error.

4.13.3.3 Acción de control proporcional-integral-derivativa PID.

La acción de control derivativa actúa sobre la derivada del error para obtener la tangente en el punto actual, lo cual permite analizar las acciones futuras del controlador. Esta acción depende mucho de la inercia del sistema, es decir, de la velocidad de actuación del sistema. Por lo que una acción de control derivativa grande nos aleja de la predicción del error.

El objetivo de la acción derivativa es generar una acción de control proporcional al cambio de rango del error. Esto tiende a tener un efecto estabilizante, pero por lo general genera actuaciones de control grandes, por lo que su uso no es adecuado en sistemas con señales de

error mal sintonizadas, ya que puede producir otra señal de error amplificada. La acción de control derivativa a la salida del controlador se rige por la siguiente ecuación:

$$u(t) = K_d \frac{de(t)}{dt} \quad (12)$$

O también,

$$u(t) = K_p T_d \frac{de(t)}{dt} \quad (13)$$

donde T_d es el tiempo derivativo. Aunque el control derivativo no afecta en forma directa al error en estado estacionario, añade amortiguamiento al sistema y por tanto, permite un valor más grande que la ganancia K , lo cual provoca una mejora en la precisión en estado estable (Mazzone, 2002).

La acción de control proporcional-integral-derivativa combina las ventajas de cada una de ellas, permitiendo obtener la siguiente ecuación de un controlador PID:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (14)$$

Y su función de transferencia resulta en:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (15)$$

De este modo se resume el diagrama de bloques de este controlador en la siguiente Figura 14:

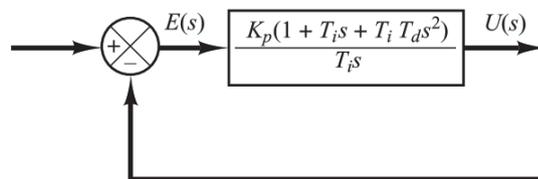


Figura 14 Diagrama de bloques de un controlador PID

Nota. Adaptado de (Ogata, 2010)

4.14 Dispositivos de hardware y software libre

El *software* libre es un asunto de libertad, no de precio. En términos generales, significa que los usuarios son libres de usar el *software*, como también de copiarlo y redistribuirlo, con o sin modificaciones. El termino *hardware* libre es aquel cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita (Stallman, 2015).

4.15 Microcontroladores

Un microcontrolador en adelante MCU, es un dispositivo electrónico capaz de llevar a cabo procesos lógicos o tareas a alta velocidad, usualmente es el componente principal de una aplicación embebida. Para realizar el trabajo es necesario que los procesos o instrucciones sean programados en un lenguaje entendible por el usuario, y posteriormente introducidos en el chip a través de un programador. Un MCU a diferencia de un microprocesador, posee todas sus partes y funcionalidades integradas en un único circuito integrado reducido (Aguayo, 2004). Dentro de la estructura interna de un microcontrolador se pueden distinguir las siguientes partes:

- Procesador: es la parte fundamental que se encarga de direccionar la memoria de instrucciones, así como de la ejecución de la operación que implica la instrucción.
- Memoria: integrada en el chip, se compone de la memoria volátil RAM (*Random Access Memory*) para guardar las variables y los datos, y la memoria no volátil ROM (*Read-Only Memory*) encargada de contener el programa de instrucciones. Los microcontroladores modernos incluyen memorias FLASH (no volátil) las cuales son más rápidas y de mayor capacidad que una memoria EEPROM (*Electrical Erasable Programmable Read Only Memory*), al igual que esta permite la escritura/borrado eléctricamente desde el propio grabador, bajo el control programado de un PC y dentro del propio circuito.
- Puertos de entrada/salida: a través de los pines del MCU, estos puertos permiten añadir dispositivos externos y proporcionar soporte a las señales de entrada, salida y control.
- Reloj principal: es un circuito oscilador que genera una onda cuadrada de alta frecuencia para la sincronización de todas las operaciones del sistema.

Todas las partes hardware detalladas anteriormente permiten proporcionar al usuario recursos específicos que caracterizan a la mayoría de los microcontroladores, entre algunos de

estos componentes destacan: temporizadores (Timers), estado de reposo (Sleep mode), conversores A/D o D/A, comparadores analógicos, modulación PWM (*Pulse Wide Modulation*), puertos de E/S digitales, entre otros.

Puesto a la gran capacidad de control y programación que permiten los microcontroladores, se ha visto ampliada su integración en varios productos y artefactos de la vida cotidiana. Como resultado se aprecia un aumentando en las prestaciones de los productos, así como reducción de su tamaño, coste, consumo y confiabilidad. Actualmente se puede encontrar microcontroladores en multitud de sistemas como hornos microondas, frigoríficos, lavadoras, y aplicaciones como robótica, proyectos IoT, o tarjetas de aprendizaje como Arduino por mencionar algunos.

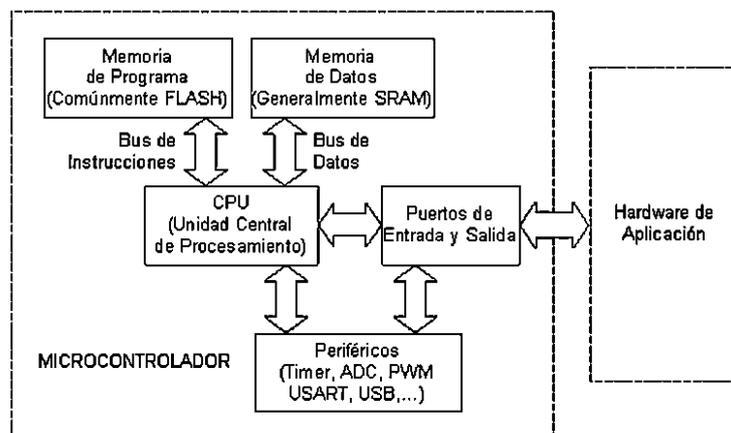


Figura 15 Estructura genérica de un microcontrolador

Nota. Adaptado de (Aguayo, 2004)

4.16 Microprocesadores

Un microprocesador también llamado CPU es el circuito central de un sistema electrónico, es donde se lleva a cabo las operaciones lógicas y aritméticas. Gracias a su estructura interna permite manejar de diversas formas una serie de datos binarios, otorgando una gran flexibilidad que permite aplicar exactamente el mismo microprocesador en una computadora pequeña, en una consola de videojuegos, en el control de un automóvil, en el panel de un proceso industrial, etc. (Reynada, 2012)

Los microprocesadores son los antecesores de los microcontroladores, y se diferencian principalmente en las partes que lo componen. Por ejemplo, un microcontrolador contiene un microprocesador y un microprocesador al menos incluye: una ALU, memoria y una unidad de control. De la Figura 16 se analiza la estructura interna de un microprocesador básico, la cual se compone de algunas partes como:

- La unidad aritmética lógica o (ALU): es la parte donde el procesador lleva a cabo las decisiones y las operaciones matemáticas. Esta unidad es la que determina la potencia de cálculo del dispositivo. Se conecta al resto del dispositivo por medio de un bus de comunicación interno, entendiéndose por “bus” a los mecanismos de comunicación que posee un procesador para conectarse con el resto del chipset o los dispositivos periféricos (Uriarte, 2020).
- Registros: son pequeños bloques de memoria interna donde se almacenan temporalmente los bits y bytes (banderas de estado) que usa la ALU para sus cálculos. De igual manera se conectan al bus interno para el intercambio de información.
- Bus de control: Se encarga de expedir o introducir instrucciones al microprocesador. Por ejemplo, aquí se encuentran las líneas que determinan si un dato se va a leer o grabar en la memoria RAM.
- Bus de datos: es la puerta de entrada y salida de los bits que se procesarán dentro del microprocesador, en otras palabras, es donde circula toda la información con la que trabaja el microprocesador, tanto la que entra como la que sale.
- Bus de direcciones: sirve para que el microprocesador pueda leer su memoria externa, o para determinar exactamente qué dispositivo externo se utilizará en un momento dado.

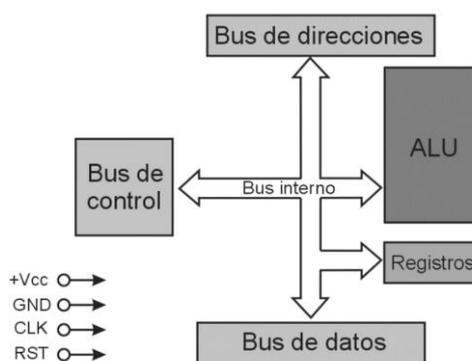


Figura 16 Aspecto interno de un microprocesador básico

Nota. Adaptado de (Reynada, 2012)

Para que el microprocesador pueda funcionar al igual que un MCU necesita de una fuente de poder externa, y también de una señal de reloj que sirva como referencia para la ejecución de todas las operaciones internas (Reynada, 2012). Es importante aclarar que el uso de los microprocesadores o procesadores no está ligado netamente a las computadoras

personales, ya que su flexibilidad ha permitido que estos invadan todos los rincones del hogar, hoy en día se puede encontrar estos dispositivos en diferentes artefactos como: televisores, equipos de sonido, y electrodomésticos varios.

4.17 Sensores y principio de transducción

La representación de un parámetro de algún fenómeno físico, mediante el cual se describa su comportamiento, se conoce como variable física. Tanto las variables físicas como las magnitudes físicas están presentes dentro de un sistema de procesamiento de información. Un sistema de ingeniería está compuesto de un sensor, un procesador y una unidad actuadora. Cuantificar una variable física es el primer paso dentro de un sistema o proceso, el instrumento que realiza esta función es el sensor. Para su funcionamiento, tanto los sensores como actuadores se basan en el llamado principio de transducción. (Corona, Abarca, & Carreño, 2014).

Un transductor es un dispositivo capaz de convertir una variable física (señal analógica), en otra cuyo dominio es diferente (señal eléctrica), generalmente este forma parte de un sensor o actuador. Por otra parte, un sensor es el dispositivo final que proporciona una salida útil para ser usada como variable de entrada a un sistema de procesamiento (p.ej. un microcontrolador) y el actuador se encarga de ejecutar una acción concreta en base a la decisión ejecutada por este sistema de procesamiento.

Como se mencionó anteriormente, un transductor se basa en un principio de transducción, es decir, un tipo de fenómeno físico que permite la transformación de la energía. La siguiente Tabla 6 muestra una clasificación de los sensores según los principios de transducción.

Tabla 6 Clasificación de los sensores según los principios de transducción

Clasificación de los sensores según los principios de transducción	Piezo-resistivo
	Capacitivo
	Piezoeléctrico
	Ultrasónico
	Magnético
	Termoeléctrico
	Fotoeléctrico
Químico	

Se puede hacer una clasificación más común de los sensores en base a la variable física a medir, por ejemplo:

Tabla 7 Clasificación de los sensores según la variable física a medir

Clasificación de los sensores según la variable física a medir	De posición, velocidad y aceleración
	De nivel y proximidad
	De humedad y temperatura
	De fuerza y deformación
	De flujo y presión
	De color, luz y visión
	De gas y pH
	Biométricos
De corriente y voltaje	

4.17.1 Sensores inteligentes

El avance de la electrónica y la tendencia a la minimización de los costos ha permitido el diseño de sensores en placas reducidas, esto mediante la integración de todas las funciones en un solo elemento y por ende la reducción en el número de componentes discretos. Por ejemplo, en la actualidad se puede ver un sensor digital en un solo chip o tarjeta embebida. Un sensor inteligente incluye memoria, convertidores A/D o D/A, protocolos de comunicación, inclusive algunos llevan parte del sistema de control en un pequeño microcontrolador. La siguiente Figura 17 muestra las diferentes partes que componen un sensor inteligente en la actualidad.

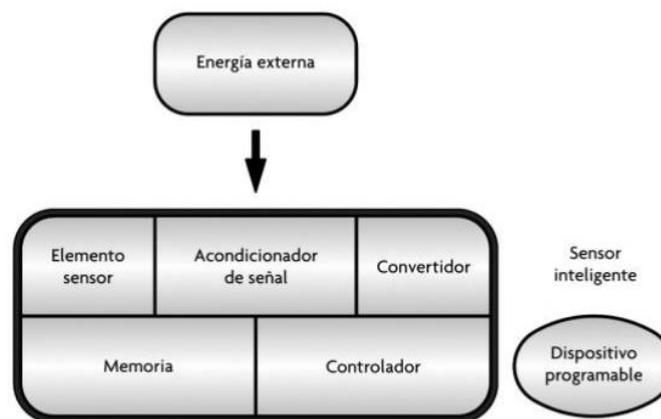


Figura 17 Diagrama de bloques de un sensor inteligente

Nota. Adaptado de (Corona, Abarca, & Carreño, 2014)

4.17.2 Acondicionamiento de señales en sensores digitales

Es imprescindible el uso de acondicionadores de señales que permitan la adecuación de la señal analógica captada por un sensor, previo a su envío final en forma digital a un dispositivo de procesamiento y control. Para esto la señal debe pasar por un conjunto de etapas que se detallan a continuación:

- Filtrado: etapa donde se desprecian los componentes no deseados de una señal, por ejemplo, picos de ruido, o frecuencias no deseadas.
- Amplificación: etapa donde la señal es amplificada en términos de magnitud, por ejemplo, una señal de 5-10mV pasa a ser una de 0-5V, esto gracias a un factor de amplificación.
- Linealización: etapa que permite convertir una señal no lineal a otra de comportamiento lineal.
- Aislamiento eléctrico: mediante una desconexión física, impide que las señales no deseadas de entrada pasen a la salida. Para esto la señal de entrada y salida están separadas por una conversión óptica o magnética.
- Excitación: es la aplicación de corriente o tensión al sensor para producir una variación proporcional a la magnitud a medir.
- Conversión: a través de circuitos convertidores, permite traducir la señal analógica a un formato digital para que sea entendida por un sistema de procesamiento y control.

4.18 El internet de las cosas (IoT)

El internet de las cosas o IoT del término anglosajón “*Internet of Things*”, es una red de comunicación de sensores, actuadores y dispositivos que interactúan entre sí mediante plataformas basados en protocolos estándar de comunicación que permiten el intercambio de datos e información del entorno físico, ejecutando procesos que desencadenan acciones y crean servicios con o sin la intervención humana. (ACADEMIA, 2020)

La integración del internet de las cosas con la nube, *Cloud-IoT*, permite transferir tareas de almacenamiento y procesamiento de datos para la ejecución de múltiples aplicaciones, además de permitir la monitorización, control y gestión de un sistema en general. (Manotoa, 2018)

El IoT consta de tres elementos principales que permiten desarrollar una internet generalizada:

- Hardware: Compuesto por sensores y hardware de comunicación embebido.

- Middleware: Corresponde a sistemas de almacenamiento y herramientas de análisis de datos.
- Presentación: Son interfaces que permiten entender y visualizar la información.

Asimismo, se pueden integrar nuevas herramientas que pueden ser utilizadas en diferentes plataformas y aplicaciones en un amplio campo de la ciencia y la tecnología. En base a un artículo publicado por (Greengard, 2021) en la revista de tendencias de ciencia y tecnología aplicada JASTT (*Journal Of Applied Science And Technology Trends*, por sus siglas en inglés) este manifiesta que en la actualidad el concepto de IoT se utiliza en la informática, el Big data, las ciudades inteligentes, las redes de carga móviles, la monitorización inteligente, los sistemas de energía inteligentes, el suministro de agua, la seguridad medioambiental, el comercio minorista inteligente, la cadena de suministro inteligente y las compras en línea. Cabe destacar que el IoT intenta comunicarse con aparatos autónomos para crear información relacionada sin la mediación humana, haciendo énfasis a que el IoT se produce cuando los objetos cotidianos y reconocibles se vinculan a microchips.

4.19 Protocolo de comunicación MQTT para redes WSN

Las redes WSN (*Wireless Sensors and Actuator Networks*, en inglés) son redes de dispositivos hardware que incorporan sensores y actuadores distribuidos dentro de un área para la ejecución de una tarea programada, que puede incluir el monitoreo y recolección de información acerca de las variables físicas dentro del área (p.ej. temperatura, presión, luminosidad) como también información del estado de un proceso determinado para su control y regulación. A su vez la información recolectada es enviada inalámbricamente a un nodo central o computadora central cuyo objetivo es analizar, almacenar y procesar la información. En una red WSN la comunicación bidireccional permite al nodo central enviar señales de control para modificar el comportamiento de los dispositivos y actuar sobre el ambiente efectuando alguna acción programada. El área de aplicación de estas redes está ligada con la eficiencia energética, entornos de alta seguridad, sensores ambientales, automatización, domótica entre otros.

En este trabajo, el protocolo de comunicación inalámbrico que se estableció para la comunicación entre la placa NodeMCU (que recibe los datos de los sensores inteligentes), con el ordenador central (Raspberry Pi 3) es el MQTT. MQTT son las siglas de *MQ Telemetry Transport*, un protocolo de mensajería asíncrona basado en publicación/suscripción que supone una forma de comunicación centrada en datos, el cual es de gran aplicación dentro de las redes

WSAN. Una comunicación centrada en datos es aquella en la que la información se entrega a los consumidores no en función de sus direcciones de red, sino en función de sus contenidos e intereses, esto supone una alta escalabilidad y soporte de una topología de aplicación dinámica, ya que permite agregar nuevas fuentes de datos/consumidores o reemplazar módulos existentes dañados (Hunkeler, 2008).

4.19.1 MQTT y el modelo de publicación/suscripción

El funcionamiento de este protocolo se basa en un servicio de mensajería con el modelo de publicador/suscriptor, donde los clientes intercambian datos con un servidor centralizado o bróker. Los publicadores son los componentes que producen la información (p.ej. dispositivos incrustados con sensores inteligentes), y los suscriptores son los que acceden a la información enviada al ordenador central. En este sentido, el bróker es la entidad que asegura que la información llegue de los publicadores a los suscriptores dentro de la red.

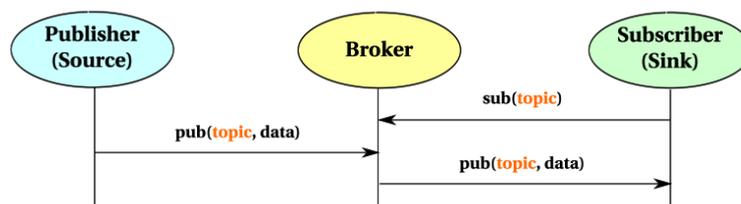


Figura 18 Modelo de comunicación publicador/suscriptor basado en temas

Nota. Adaptado de (Hunkeler, 2008)

La Figura 18 muestra un modelo MQTT basado en temas (en adelante llamado tópicos), el cual es un modelo ampliamente usado en redes WSAN, ya que permite a los publicadores enviar datos referentes a un tema, (p.ej. temperatura/área/1) y del otro lado permite a uno o varios suscriptores acceder a información de un tema en particular que puede ser información de otro sensor (p.ej. temperatura/área/1) o información de control para el accionamiento de algún actuador (p.ej. activar/ventilador/área/1). La siguiente Figura 19 ejemplifica la arquitectura de una red de sensores inalámbricos que usan el protocolo MQTT para su comunicación, aquí la función principal de las puertas de enlace es proporcionar a los dispositivos acceso al bróker, y este a su vez se ubica en el centro de la red debido a su mayor rendimiento en cuanto a ancho de banda y capacidad de procesamiento, de modo que todos los demás componentes se conectan a él y se comunican entre sí mediante el servicio publicador/suscriptor en un determinado tema de interés. Es importante destacar que el uso de este sistema publicador/suscriptor permite a los desarrolladores de aplicaciones ocultar la

complejidad de la red subyacente y aprovechar el tiempo en el diseño de la propia aplicación (Hunkeler, 2008).

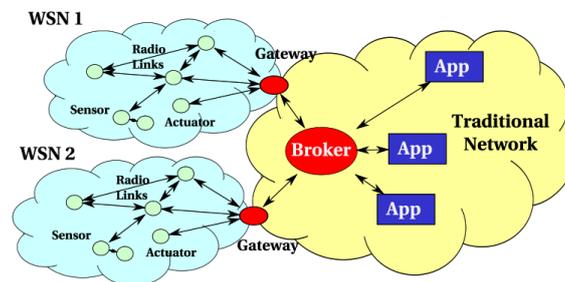


Figura 19 Redes de sensores inalámbricos integradas con comunicación publicador/suscriptor

Nota. Adaptado de (Hunkeler, 2008)

4.19.2 Características técnicas del protocolo de comunicación MQTT

La siguiente Tabla 8 resume las características principales que permitieron seleccionar este protocolo para la comunicación inalámbrica del prototipo con el servidor local almacenado en una Raspberry Pi 3. Entre sus características más importantes destaca su bajo consumo de energía, bajo procesamiento de cómputo, y bajo ancho de banda, que permite obtener una mejor eficiencia en ambientes IoT, incluso desplazando a otros protocolos de comunicación como HTTP (*Hypertext Transfer Protocol*, en inglés), AMQP (*Advanced Message Queuing Protocol*, en inglés) y XMPP (*Extensible Messaging and Presence Protocol*, en inglés) (Yuan, 2017). MQTT se ejecuta sobre la pila TCP/IP y gracias a su simplicidad posee una cabecera muy pequeña de 2-5 bytes, lo cual comparado contra HTTP es mucho menor que sus 137 bytes (Vicmar, 2020). MQTT también puede ejecutarse en SSL/TLS, que es un protocolo para garantizar que toda la comunicación de datos entre dispositivos esté cifrada y segura, además el protocolo establece tres niveles que definen la calidad del servicio QoS, y por ende mayor o menor sobrecarga al sistema, estos tres niveles se resumen a continuación:

- QoS 0: El mensaje se envía una única vez, en caso de fallo puede que algún mensaje no se entregue.
- QoS 1: El mensaje se envía hasta que se garantiza la entrega, en caso de fallo, el suscriptor puede recibir algún mensaje duplicado.
- QoS 2: Se garantiza que cada mensaje se entrega al suscriptor, y únicamente una vez.

Tabla 8 Características técnicas del protocolo de comunicación MQTT

Protocolo de comunicación MQTT				
Seguridad	Puertos	Calidad de	Estructura trama	Características

	utilizados	servicio		
-Permite conexiones TCP/IP simples o conexiones seguras SSL/TLS para mensajes confidenciales.	-1883: MQTT sin cifrar. -8883: MQTT encriptado. -8884: MQTT cifrado con certificación del cliente. -8080: MQTT sobre WebSockets sin cifrar. -8081: MQTT sobre WebSockets encriptado.	Tres niveles de QoS seleccionables.	- Cabecera fija: ocupa 2 a 5 bytes. - Cabecera variable: contiene información adicional 0-Y bytes. - Contenido: Es el contenido real del mensaje, (limitada a 256MB de información) en implementaciones reales el máximo es de 2 a 4 kB.	-Sencillez y ligereza. -Empleado por dispositivos de escasa potencia. -Utiliza un menor consumo de energía. -Requiere un ancho de banda mínimo. -Flexibilidad en la seguridad y calidad del servicio. -Implementaciones en código abierto. -Estándar para aplicaciones IoT.

4.19.3 Funcionamiento del protocolo MQTT

Como se mencionó anteriormente un mensaje MQTT consta de una cabecera fija de 2 bytes, una cabecera variable y el contenido o *payload* que no siempre están presentes. En este protocolo cada mensaje tiene un comando y carga útil de datos, por lo que existen diferentes tipos de mensajes dependiendo del comando utilizado (p.ej. un mensaje CONNECT o un mensaje SUBSCRIBE).

En primer lugar, el cliente se conecta al intermediario enviando un mensaje CONNECT. El mensaje CONNECT solicita establecer una conexión desde el cliente al intermediario, a su vez este mensaje posee algunos parámetros como: nombre de usuario, contraseña, lastWillTopic, lastWillQos o el keepAlive que permite al cliente mantener la conexión activa con el intermediario. Posteriormente el cliente recibirá un mensaje CONNACK del intermediario, que actúa como acuse de recibo de la conexión, además, mediante el parámetro sessionPresent indica si la conexión ya tiene una sesión persistente, o utiliza el parámetro returnCode para identificar algún error. Una vez establecida la conexión el cliente puede enviar un mensaje SUBSCRIBE al intermediario para indicar que recibirá mensajes de un tema de interés, de igual forma el mensaje contiene algunos parámetros que son de petición como el QoS (3 niveles de calidad de servicio) y el parámetro tema al cual se suscribe el cliente. Justo después de suscribirse el cliente a un tema, el intermediario devuelve un mensaje SUBACK que actúa como acuse de recibo de suscripción, este mensaje viene con uno o varios parámetros returnCode que definen el nivel de éxito correspondiente al QoS (0-2) o indicando un error. Aunque en otro sentido, el cliente también puede publicar mensajes al intermediario, para lo cual envía un mensaje PUBLISH, con parámetros como el topicName (definiendo el tema),

gos, retainFlag, y carga útil con la información correspondiente al tema. En otro orden de ideas, el cliente tiene la responsabilidad de enviar un mensaje dentro de cada intervalo Keep Alive; en caso de que no haya un mensaje relacionado con datos, el cliente envía un mensaje PINGREQ, que el intermediario confirmará con un PINGRESP, esta operación no tiene otra función que la de mantener una conexión viva y asegurarse de que la conexión TCP no ha sido cerrada por un *gateway* o enrutador. Finalmente existen otros comandos como el UNSUBSCRIBE que envía el cliente para darse de baja a un tema, o el DISCONNECT que indica que el cliente se ha desconectado.

4.20 Herramientas de software libre para la gestión de la información

4.20.1 Python 3

Python es un lenguaje de alto nivel, interpretado y multipropósito, y uno de los más empleados en el desarrollo de software.

Un lenguaje interpretado a diferencia de uno compilado indica que este requiere de un intérprete que traduzca el código línea por línea al momento de la ejecución, mientras que el otro utiliza un compilador para poder traducir todo el código al lenguaje de maquina y poder ejecutar el programa. (Victoria, 2017)

Python es *open source*, cualquiera puede contribuir a su desarrollo y divulgación, incluso su interprete se distribuye de forma gratuita para diferentes plataformas, además, puede ser empleado en cualquier sistema operativo como: Windows, Mac OS X, Linux. Este lenguaje permite desarrollar *software* para diferentes aplicaciones científicas, comunicaciones de red, aplicaciones con interfaz gráfica GUI, creación de juegos, aplicaciones web, entre otros. (Fernandez, 2013)

Una particularidad de este lenguaje es su simplicidad de sintaxis, facilitando el aprendizaje del lenguaje, por ejemplo, no es necesario la creación de tipos de variables ya que Python las fija al momento de su asignación. Otra característica es que incluye una serie de estructuras de datos de alto nivel como listas, tuplas, diccionarios y funciones incluidas en su librería estándar que cubre la mayoría de las necesidades básicas de un programador.

4.20.2 Node-RED

Para el presente trabajo se usará la plataforma Node-RED instalada en la Raspberry Pi, la cual es una herramienta de desarrollo basado en flujo de código abierto para la integración de dispositivos hardware IoT, interfaces de programación de aplicaciones o APIs (*Application Programming Interfaces*, en inglés) y servicios en línea desarrollados por el grupo IBM

Emerging Technology (Heath, 2014). Node-RED es una herramienta gratuita basada en JavaScript creada en la plataforma Node.js, que proporciona un editor de flujo visual para permitir a los desarrolladores conectar bloques de código predefinidos, conocidos como "nodos", y realizar una tarea específica. Al pasar mensajes entre diferentes nodos, en una combinación de nodos de entrada, nodos de proceso y nodos de salida, se forma un flujo o subflujo que permite realizar una tarea programada. Una particularidad de esta herramienta es que se ejecuta fácilmente en la nube, en un PC, como también en el Raspberry Pi OS donde ya viene preinstalada. Entre algunas de sus características más importantes se detallan las siguientes:

- Edición de flujo basado en navegador: esto facilita la interconexión de flujos utilizando la amplia gama de nodos de la paleta, además permite crear funciones de JavaScript dentro de un editor de texto enriquecido (Rodger, 2016).
- Utiliza el formato JSON: el formato de notación de objetos de JavaScript permite un formato ligero de intercambio de datos, el cual es independiente de cualquier lenguaje de programación y que resulta sencillo de leer y escribir para los programadores y simple de interpretar y generar para las máquinas. Es comúnmente utilizado por las API web para devolver datos.
- Basado en Node.js: Esto lo hace ideal para ejecutarse en el borde de la red en hardware de bajo costo como Raspberry Pi, así como en la nube, y aprovechar al máximo su modelo de E/S sin bloqueo y controlado por eventos. Este modelo lo hace adecuado para crear aplicaciones en tiempo real con uso intensivo de datos que se ejecutan en dispositivos distribuidos (Heath, 2014).

En la siguiente Figura 20 se aprecia la interfaz gráfica de usuario de Node-RED y cómo la plataforma divide los sistemas en sus partes constituyentes. Aquí cada uno de los nodos es una representación visual de un bloque de código JavaScript diseñado para realizar una tarea específica, que puede ir desde mostrar un mensaje en pantalla como también unir servicios web y hardware para un proceso de automatización.

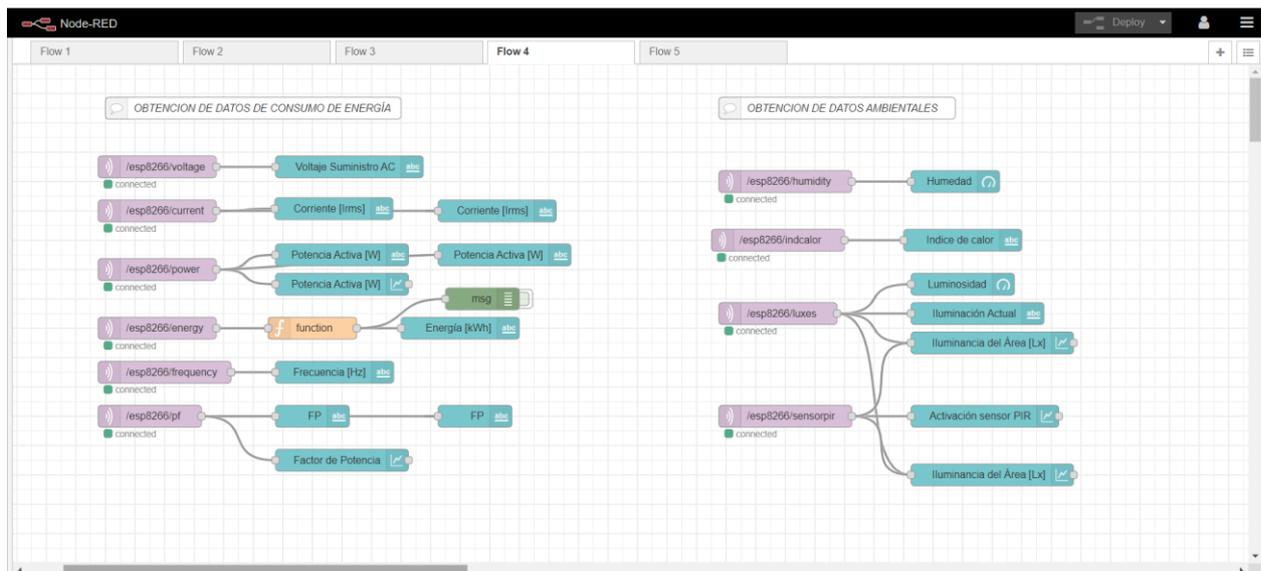


Figura 20 Interfaz gráfica de usuario en Node-RED

4.20.3 Servidor LAMP

Para el desarrollo del presente trabajo, se optó por la implementación de un servidor LAMP alojado en el ordenador central (Raspberry Pi 3) que permitirá agrupar distintas herramientas software y conformar una plataforma que corre desde el lado del servidor, para el proceso de almacenamiento y gestión de la información proveniente del prototipo.

En este sentido, resulta conveniente desarrollar servidores de información mediante aplicaciones de código abierto y gratuitas, como lo es el conocido servidor LAMP (LAMP: Linux, Apache, MySQL y PHP/PERL/Python). Un servidor LAMP es un paquete de software que contiene un ambiente de desarrollo completo de sitios web como también un conjunto de herramientas para implementar el diseño de un sistema de base de datos distribuida, que, además, permite la posibilidad de acceder de forma segura a la información, y su administración, definiendo distintos niveles de control. En general, el termino LAMP hace referencia a cualquier combinación de herramientas *open source* para el desarrollo web e interfaz de base de datos como (Ipiña, 2005):

- Linux podría remplazarse por OpenBSD.
- MySQL por PostGreSQL o MariaDB.
- La P, podría traducirse en PHP, Perl, Python o Ruby



Figura 21 Servidor LAMP

Nota. Adaptado de (YMANT, 2021)

4.20.3.1 Linux

Linux es el sistema operativo base de la plataforma, y puede ser cualquier tipo de sistema GNU/Linux, donde podrán ejecutarse el resto de los componentes del servidor. Linux es de código abierto, multiplataforma, multiusuario y multitarea, utiliza una Licencia Pública General (GPL) de GNU, lo cual implica que todo el mundo puede ejecutar, estudiar, compartir y modificar el sistema de software (Bigelow, 2021). Los sistemas operativos GNU/Linux se encuentran normalmente en forma de compendios conocidos como distribuciones a las cuales se les han añadido selecciones de aplicaciones y programas para usuarios con diferentes necesidades. Debido a su estructura, una ventaja de Linux es que es estable para muchas cargas de trabajo altamente disponibles y confiables, especialmente en servidores de aplicaciones y centros de datos (Red Hat, 2018).

4.20.3.2 Apache

Apache es un servidor HTTP para diferentes sistemas operativos como: Unix, GNU/Linux, Windows, Mac etc. Siendo así un servidor web gratuito, potente, fácil de manejar y de configurar, sobre todo en distribuciones Linux donde forma parte de los repositorios de cada distribución. Apache permite alojar y despachar páginas web dinámicas y estáticas en la computadora donde se tenga instalado este servidor, con soporte para lenguajes como Perl, PHP y Python. Adicionalmente Apache integra funcionalidades para brindar seguridad al software gracias a los módulos de autorización y autenticación y funciones de control de acceso, además de un soporte para cifrado por certificados SSL/TLS (Reyes, 2006).

4.20.3.3 MySQL

MySQL es el gestor de base de datos de distribución gratuita más extendido en la actualidad, y es software libre licenciado bajo GNU/GPL (Reyes, 2006). Esta herramienta permite organizar los datos de forma segura por medio de tablas, índices y registros, para permitir consultas a clientes locales o remotos que solicitan esta información. MySQL se

caracteriza por la ejecución de tareas en simultáneo tanto lectura como escritura y ofrece gran velocidad de acceso a los datos, además, soporta múltiples motores de almacenamiento como MyISAM e INNODB. La funcionalidad del sistema permite que un cliente acceda a dicha información y realice diferentes tareas dependiendo del nivel de privilegios que posea gracias a una gestión de usuarios y contraseñas. En resumen, MySQL ofrece un rendimiento bueno, es flexible y fácil de implementar en cualquier sistema operativo, posee una integración perfecta con PHP, control de acceso a usuarios y al igual que MariaDB son soluciones estandarizadas para el 90% de las bases de datos pequeñas o medianas (Borgues, 2019).

4.20.3.4 phpMyAdmin

phpMyAdmin es una herramienta de software libre escrita en PHP, la cual a través de una interfaz web intuitiva permite crear, modificar, borrar o realizar consultas en nuestra base de datos, es decir, esta herramienta está destinada a manejar la administración de MySQL en la Web. Adicionalmente phpMyAdmin admite una amplia gama de operaciones en MySQL y MariaDB, tales como: operaciones de configuración del servidor, administración de cuentas de usuario y privilegios, administración de las bases de datos, realización de instrucciones SQL, importación y exportación de datos en diferentes formatos (CSV, SQL, XML entre otros), como también la creación de gráficos de su base de datos para visualización (García, 2021). En definitiva, phpMyAdmin permitirá administrar la base de datos almacenada en el ordenador central (Raspberry Pi 3), mediante la generación de sentencias SQL personalizadas, además de gestionar operaciones más complejas como la creación de las tablas, modificación de su estructura, y trabajos de búsqueda, inserción o borrado de los registros que contendrán la información del sistema propuesto.

5 Metodología

5.1 Materiales

En el presente apartado se realiza una lista de los materiales usados a lo largo del desarrollo del trabajo, para posteriormente analizar las características técnicas de los componentes principales, que en conjunto permitirán comprender el funcionamiento del circuito eléctrico diseñado para el control inteligente de iluminación.

Tabla 9 Materiales usados para la implementación del prototipo

Material	Unidades	Precio Unidad USD	Costo Total USD
NodeMCU ESP8266	2	9,76	19,52
Raspberry pi 3 Model B+	1	60,00	60,00
Sensor BH1750	1	8,88	8,88

Sensor PIR HC-SR501	1	4,00	4,00
Sensor DHT22	1	10,00	10,00
Módulo PZEM004Tv3	1	30,00	30,00
Regulador voltaje L7805CV	1	3,00	3,00
Regulador 100-240 V AC, 50-60 Hz a 12V/3W DC HI-LINK- HLK-5M12	1	8,25	8,25
MOSFET IRF840 CH-N	2	2,63	5,26
Transistor NPN 2N3904	1	0,80	0,80
Optoacoplador 4N35/NTE3041	2	1,60	3,20
Puente rectificador W08M	1	0,40	0,40
Fusible vidrio 3 ^a 250V europeo	1	0,17	0,17
Porta fusible horizontal PCB	1	0,67	0,67
Resistencia 10K Ω 2W	4	0,06	0,25
Resistencia 270 Ω 1/2W	2	0,07	0,14
Capacitor Electrolítico 0.1 μ F 50V	2	0,13	0,26
Capacitor Electrolítico 0.33 μ F 50V	2	0,14	0,28
Capacitor Electrolítico 2.2 μ F 50V	2	0,17	0,34
Conector espadín macho 40 Pines (Pitch 2.54mm)	2	0,67	1,34
Conector espadín hembra 40 Pines (Pitch 2.54mm)	4	0,76	3,03
Placas PCB	5	-	50,00
Cable de timbre 2x23 AWG (metros)	2	0,26	0,52
Cable Gemelo SPT Calibre 2X12 (metros)	4	0,40	1,60
Borneras PCB	4	0,15	0,60
Focos Maviju A60 DIM	4	3,50	12,00
Boquillas E27	4	0,50	2,00
Total			\$ 226,51

5.1.1 Microcontrolador NodeMCU ESP8266

La placa de desarrollo NodeMCU ESP8266 viene con el módulo ESP-12E que contiene el chip ESP8266 con el microprocesador Tensilica Xtensa LX106 RISC de 32 bits. Este dispositivo funciona a una frecuencia de reloj ajustable de 80 MHz a 160 MHz. NodeMCU tiene 128 KB de RAM y 4 MB de memoria Flash para almacenar datos y programas. Su alta potencia de procesamiento con funciones integradas de Wi-Fi y Deep Sleep Operating lo hacen ideal para proyectos de IoT (Parihar, 2019).

En cuanto a las capacidades de conectividad WiFi, es compatible con el estándar IEEE 802.11 b/g/n, además de soportar seguridad WEP, WPA y WPA2. El ESP8266 permite una conexión inalámbrica a una red local o a Internet, esto permite una gran cantidad de posibilidades, como poder conectar o desconectar electrodomésticos (usando un relé) u otro tipo de sistemas mecánicos de nuestro hogar para domotizar la casa y controlarla por Internet desde nuestro smartphone o cualquier ordenador conectado desde cualquier parte (Hardwarelibre, 2019). La placa cuenta con los siguientes pines para trabajar.

- 9 pines digitales numerados del D0 al D8 con I2C y SPI.
- 1 pin analógico numerado A0.
- 3 pines de 3,3V.
- 1 pin de 5V.
- 4 pines de tierra GND.

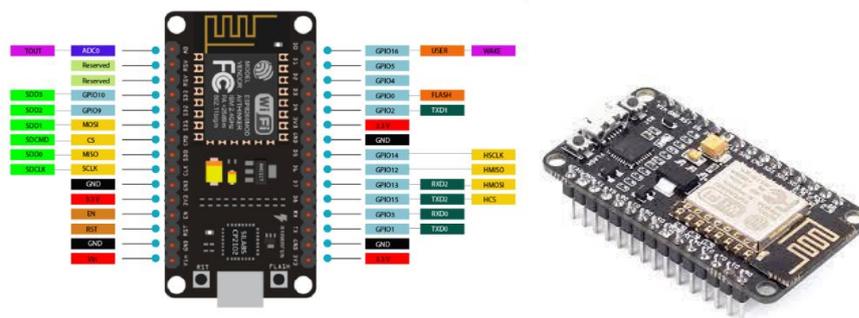


Figura 22 NodeMCU ESP8266 V.2 y distribución de pines

Nota. Adaptado de Escuela Técnica Virtual ETI,2021 (<https://sites.google.com/a/educ.ar/eti/project-updates/introalnodemculolinv3esp8266>)

El uso de este dispositivo dentro del prototipo, tiene la finalidad de ser la unidad principal de procesamiento y acondicionamiento de las señales provenientes de los sensores inteligentes, así como también implementar el algoritmo de control de potencia, la regulación automática de intensidad de iluminación y el envío de la información a un servidor local alojado en el ordenador central (Raspberry Pi), por medio de la comunicación inalámbrica WiFi y el uso del protocolo de comunicación MQTT.

5.1.2 Raspberry pi 3 Model B+

El módulo Raspberry Pi 3 B+ es un equivalente en miniatura de un ordenador personal completo. El punto más importante del dispositivo es el procesador de cuatro núcleos Broadcom

BC2837 ARM-8 de 64 bits con los gráficos integrados Cortex-A53. La memoria operativa en el modelo 3 B+ es de 1GB y se utiliza una tarjeta MicroSD como disco duro, que se inserta en el lector del dispositivo. La tarjeta de memoria puede instalar el Sistema operativo Raspbian el cual es un software libre y de código abierto basado en el núcleo de Linux, este a su vez alberga internamente el lenguaje de programación Python el cual es interpretativo, orientado a objetos y su codificación en Python se puede realizar mediante la consola de comandos o en el editor de texto Thonny incorporado por defecto (Sałuch, Tokarski, & Adamczewski, 2018). Entre algunas de sus características más relevantes resaltan las siguientes:

- Procesador: 1,2 GHz.
- Número de núcleos: 4 Quad Core.
- RAM: 1 GB.
- Memoria: MicroSD externa.
- GPIO: 40 (Pines de entrada/salida de propósito general).
- Número de puestos USB: 4.
- HDMI: 1.4 Standard.
- Comunicación: Ethernet, Wi-Fi, Bluetooth.
- Protocolos: UART, I2C, SPI.

Debido a su gran potencia de procesamiento, como también a su tamaño reducido y bajo costo, esta placa es comúnmente utilizada en proyectos IoT e Industria conectada, también en sistemas de control, para procesamiento de grandes cantidades de datos y aplicaciones como servidores de archivos web. El uso de este dispositivo cumple con la finalidad de ser el ordenador central del sistema para la recepción de información proveniente de los sensores inteligentes y enviada por los clientes MCU, esto por medio de la comunicación inalámbrica WiFi y el protocolo de comunicación MQTT, a su vez, permitirá el almacenamiento de información valiosa en un servidor de base de datos local, como también ejecutar una interfaz gráfica de usuario GUI para el monitoreo, control y gestión de la información del prototipo.

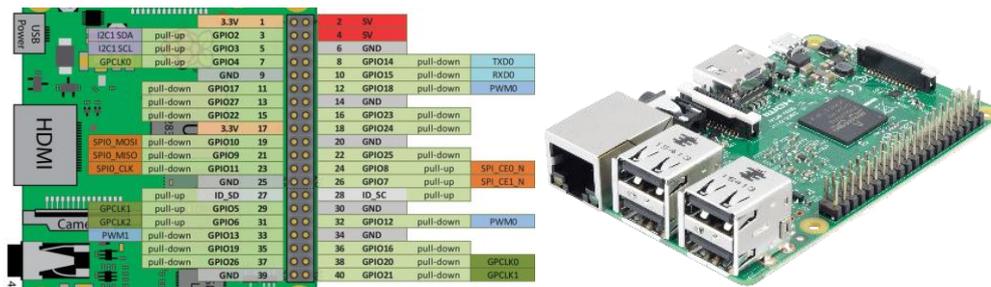


Figura 23 Raspberry pi 3 Model B+

Nota. Adaptado de (Hardwarelibre, 2019)

5.1.3 Dispositivos de medición y adquisición de datos

En el siguiente apartado se describen las características principales de los sensores incorporados en el diseño del sistema inteligente de iluminación. La elección del tipo de sensor a utilizar, fue producto del análisis bibliográfico y de las experiencias descritas por otros autores. De esta forma se analizó la mejor alternativa, en base a su disponibilidad en el mercado, precio y compatibilidad con las placas de desarrollo utilizadas en el presente trabajo.

5.1.4 Módulo sensor de luminosidad BH1750

El BH1750 es un sensor de luz ambiental de bajo costo, bajo consumo de energía y con un amplio rango de detección y resolución. Su alta resolución permite captar un rango de valores de entre (1 - 65535 lx) lo cual está muy por encima de nuestro rango de trabajo. En primera fase el sensor utiliza un fotodiodo para detectar la luz y esta a su vez se convierte en una salida de voltaje con la ayuda de un OP-AMP de integración. Seguidamente, el ADC incorporado proporciona los datos digitales usando 16 bits de resolución. Por último, mediante la ayuda del oscilador interno se transmite la información usando la interfaz I2C incorporada. La lógica interna del BH1750 evita la necesidad de cálculos complicados, ya que genera directamente datos digitales significativos en luxes (Lx). Entre algunas otras características de su hoja de datos en (ROHM SEMICONDUCTOR, 2009) se incluyen las siguientes:

- Comunicación mediante interfaz de bus I2C.
- Oscilador interno (320 kHz). Es el CLK para la lógica interna.
- Baja corriente de consumo por la función de apagado.
- Función de rechazo de ruido de la luz (50Hz / 60Hz).
- Permite medir fuentes de luz proveniente de lámparas como: incandescentes, fluorescentes, halógenas, LED y luz solar.

- La influencia de los infrarrojos es muy pequeña.
- Permite 2 tipos de dirección esclava I2C: ADDR=HIGH:0X5C y ADDR=LOW:0X23C

El diagrama de bloques del sensor se resume de la siguiente manera, con sus partes detalladas anteriormente:

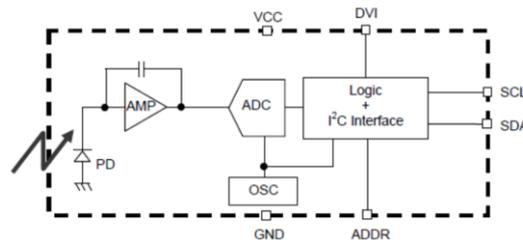


Figura 24 Diagrama de bloques del módulo BH1750

Nota. Adaptado de (ROHM SEMICONDUCTOR, 2009)

Los modos de medición del sensor se dividen en dos grupos; mediciones continuas (*Continuously Mode*) y únicas (*One Time Mode*). En modo continuo, el sensor mide continuamente el valor de luminosidad y en modo de una única vez el sensor realiza una medición y luego pasa automáticamente en modo de apagado. Cada modo tiene tres precisiones diferentes dependiendo de la resolución requerida y el tiempo usado para la medición, estos son:

- Modo de baja resolución: (resolución de 4 lx, tiempo de medición de 16 ms)
- Modo de alta resolución: (resolución de 1 lx, tiempo de medición de 120 ms)
- Modo de alta resolución 2 - (resolución de 0,5 lx, tiempo de medición de 120 ms)

De forma predeterminada, la biblioteca BH1750.begin () usa el modo continuo de alta resolución, pero puede cambiarse a cualquier otro modo mediante software. Hay que destacar que las especificaciones del *Datasheet* recomiendan el uso del modo de alta resolución debido a que en este tiempo de medición el sensor rechaza de mejor manera algún tipo de ruido (incluido el ruido de 50Hz / 60Hz).

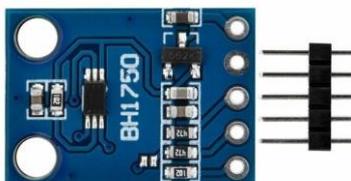


Figura 25 Módulo sensor de luminosidad BH1750

Nota. Adaptado de Andreas, GNU 1.3, vía Wiki Laub-Home.

5.1.5 Sensor de temperatura y humedad relativa DHT22

El sensor DHT22 fue escogido por su alta precisión y facilidad de uso, este sensor está compuesto en dos partes, un sensor de humedad capacitivo y un termistor, también consta de un circuito integrado en el interior, que hace la conversión de analógico a digital y envía en una única señal los datos con la temperatura y la humedad para su lectura en el MCU. (ABC PROYECTOS ELECTRÓNICOS, 2014)

Entre algunas de las características de este sensor DHT22 se abordan las siguientes:

- Alimentación de 3,3V a 5VDC
- Corriente máxima 2,5mA durante la conversión
- Lectura de humedad con un +/- 2% a 5% de precisión
- Lectura de temperatura con un +/- 0,5°C de precisión
- Capaz de medir humedad de 0% a 100%
- Capaz de medir temperatura de -40°C a 125°C
- Dimensiones: 15,1mm x 25mm x 7,7mm

Como desventaja se incluye que este sensor al utilizar un único pin digital de salida, utiliza un protocolo de comunicación que lo hace lento en cierto aspecto, ya que no se puede consultar más de una vez cada segundo. El protocolo de comunicación se puede resumir de (Aosong Electronics) en el siguiente orden:

1. Cuando el MCU envía la señal de inicio, el DHT22 cambia del modo de bajo consumo al modo de operación. El estado libre del bus de datos es de alto nivel de tensión.
2. Cuando la comunicación entre el MCU y el DHT22 comienza, el programa del MCU transformará el nivel de tensión del bus de datos de alto a bajo nivel y este proceso debe

durar al menos 1ms para asegurar que el DHT22 pueda detectar la señal del MCU, entonces el MCU esperará 20-40µs para la respuesta del DHT22.

3. El DHT22 envía la señal de respuesta al MCU. Cuando el DHT22 detecta la señal de inicio, envía una señal de bajo nivel de voltaje y esta señal dura 80µs como señal de respuesta, entonces el programa del DHT22 transforma el nivel de voltaje del bus de datos de bajo a alto nivel y dura 80µs para la preparación del DHT22 para enviar datos.
4. El DHT22 envía datos al MCU. Cuando el DHT22 envía datos al MCU, la transmisión de cada bit comienza con un nivel de baja tensión que dura 50µs, la longitud de la señal de alta tensión siguiente decide si el bit es "1" o "0", por ejemplo, 70 µs de longitud de tensión significa 1bit de datos "1" y 26-28 µs de longitud de tensión significa 1 bit de datos "0".

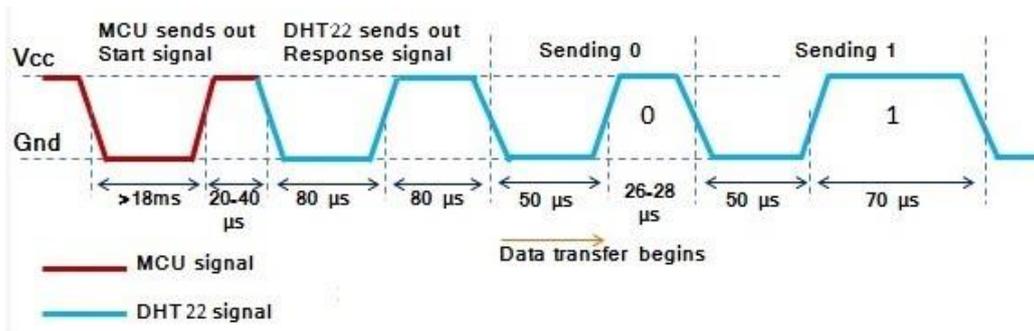


Figura 26 Protocolo de comunicación del sensor DHT22

Nota. Adaptado de (Gomez, 2016)

Por otra parte, la estructura de la trama de datos proporcionada por este sensor tiene un tamaño total de 40 bits, los cuales conforman las partes enteras y decimales de las dos variables. Si la transmisión de datos es correcta, la suma de comprobación debe ser los últimos 8 bits de "8-bit integral RH + 8-bit decimal RH + 8-bit integral T + 8-bit decimal T" (Aosong Electronics). La siguiente Tabla 10 detalla cómo se subdivide esta trama.

Tabla 10 Estructura de la trama de datos del sensor DHT22

40 bits (5 Bytes)				
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
8-bit RH	integral RH	8-bit decimal	8-bit integral T	8-bit decimal T
				8-bit Checksum

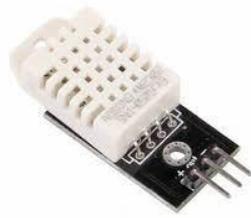


Figura 27 Sensor DHT22

Nota. Adaptado de (Gomez, 2016)

5.1.6 Sensor de presencia PIR HC-SR501

Con el objetivo de realizar el control inteligente de la iluminación resulta crucial detectar la presencia de ocupantes en el área de estudio, por lo tanto, se optó por usar un sensor infrarrojo pasivo PIR (*Passive Infrared*) HC-SR501, debido a su rapidez, rentabilidad y precisión de detección. Especialmente este sensor es capaz de detectar longitudes de onda de aproximadamente 10 micrones que es la longitud de onda máxima de la energía térmica emitida por los humanos y su salida está representada por cero o uno correspondiente al estado desocupado o al estado ocupado respectivamente.

El sensor incluye un lente de Fresnel el cual es un encapsulado semiesférico cuyo objetivo es permitir el paso de la radiación infrarroja en el rango de los 8 a 14 micrones, con un ángulo de apertura de 110°. Otra finalidad de este encapsulado es concentrar la energía en la superficie del sensor PIR, permitiendo una mayor sensibilidad del dispositivo (Punto Flotante S.A., 2017). Entre algunas de sus especificaciones técnicas más relevantes se incluye:

- Voltaje de alimentación: de 5 a 20 V_{DC}.
- Consumo promedio: <1 mA.
- Rango de distancia de 3 a 7 metros ajustable.
- Posee 2 potenciómetros para ajustar de rango de detección y el tiempo de alarma activa.
- Jumper para configurar la salida de alarma en modo mono disparo o disparo repetitivo.
- Salida de alarma de movimiento con ajuste de tiempo entre 3 segundos a 5 minutos.
- Salida de alarma activa con nivel alto de 3,3 volts.
- Tiempo de inicialización: después de alimentar el módulo HC-SR05, debe transcurrir 1 minuto antes de que inicie su operación normal.

- Tiempo de salida inactiva: cada vez que la salida pase de activa a inactiva, permanecerá en ese estado inactiva los siguientes 3 segundos. Cualquier evento que ocurre durante este lapso de tiempo es ignorado.
- Temperatura de operación: -15° a $+70^{\circ}$ C.
- Dimensiones: 3,2 x 2,4 x 1,8 cm

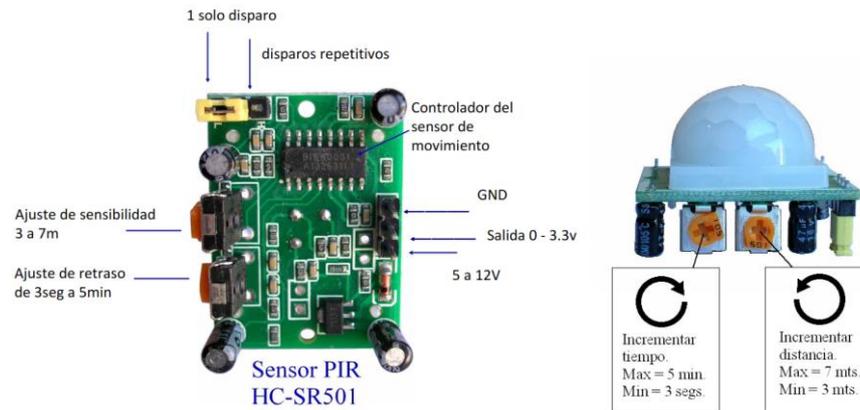


Figura 28 Sensor PIR HC-SR501

Nota. Adaptado de (Punto Flotante S.A., 2017)

5.1.7 Módulo PZEM-004T

Este módulo permite obtener un medidor no invasivo para aplicaciones de medición del consumo eléctrico residencial o para proyectos básicos de análisis de consumo a baja escala, además, es perfecto para su integración con aplicaciones IoT usando Arduino / ESP8266 / Raspberry Pi como plataformas de código abierto. Entre sus principales funciones se tiene: medir voltaje en CA, corriente, potencia activa, frecuencia, factor de potencia y la energía activa que toma una carga conectada a una línea monofásica de 110 / 220V como por ejemplo una estufa, nevera, motor, lampara, etc. (InnovatorsGuru, 2019)

Entre algunas otras funcionalidades del módulo destaca la alarma de sobrecarga la cual se activa cuando la potencia activa medida supera el umbral establecido, además la capacidad de acumular y mantener el uso de energía activa hasta 9999,99 kWh para después poder restablecerse a 0 mediante *software*. Para la comunicación este módulo utiliza en su capa física la interfaz de comunicación UART a RS485, lo cual proporciona una solución robusta para la transmisión de datos. De esta manera mediante una comunicación serial pasiva TTL puede conectarse a un microcontrolador (p.ej. Arduino Uno) usando los pines RX y TX y una fuente de alimentación externa. Adicionalmente la información obtenida puede ser enviada a la PC

usando un adaptador TTL a USB, que permite una solución rápida para la visualización de los parámetros eléctricos, ajuste y calibración del módulo. Los principales parámetros eléctricos de trabajo de este módulo se plasman en la Tabla 11 como se ve a continuación.

Tabla 11 Parámetros de funcionamiento del módulo PZEM-004T

Voltaje	Factor de potencia	Frecuencia
-Rango de medición: 80 ~ 260 V	-Rango de medición: 0,00 ~ 1,00	-Rango de medición: 45 Hz ~ 65 Hz
-Resolución: 0,10 V	-Resolución: 0,01	-Resolución: 0,10 Hz
-Precisión de medición: 0,5%	-Precisión de medición: 1%	-Precisión de medición: 0,5%
Corriente	Potencia activa	Energía activa
-Rango de medición: 0 ~ 10A (PZEM-004T-10A) 0 ~ 100A (PZEM-004T-100A)	-Rango de medición: 0 ~ 2,3 kW (PZEM-004T-10A) 0 ~ 23 kW (PZEM-004T-100A)	-Rango de medición: 0 ~ 9999,99 kWh
-Corriente de medida inicial: 0,01A (PZEM-004T-10A) 0,02A (PZEM-004T-100A)	-Potencia de medida inicial: 0,4 W	-Resolución: 1 Wh
-Resolución: 0,001A	-Resolución: 0,1 W	-Precisión de medición: 0,5%
-Precisión de medición: 0,5%	-Precisión de medición: 0,5%	

Nota. Adaptado de (InnovatorsGuru, 2019)

La Figura 29 muestra el diagrama de bloques funcional de este módulo, el bloque principal lo conforma el sistema de medición, a su vez este se conecta al bloque de alimentación AC, al bloque de la señal de medición y finalmente al bloque de interfaz TTL que permite su comunicación, no sin antes pasar por el bloque de aislamiento (utiliza optoacopladores), el cual permite un aislamiento galvánico del módulo con el resto de dispositivos externos de procesamiento y control.

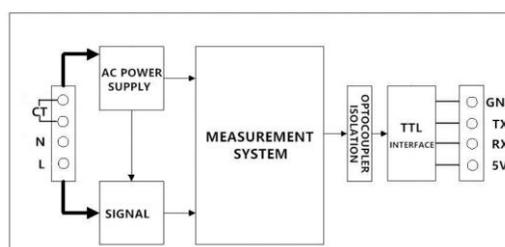


Figura 29 Diagrama de bloques funcional de un módulo PZEM-004T-100A

Nota. Adaptado de (InnovatorsGuru, 2019)

Para ejemplificar el funcionamiento de este módulo, la Figura 30 muestra el diagrama de cableado eléctrico. Como se puede observar el sistema permite una conexión simple y una medición no invasiva en el circuito eléctrico, además de poseer un tamaño apropiado para su incorporación en proyectos de poco espacio o encapsulados pequeños. En este sentido, este

módulo medidor permitirá realizar una lectura precisa de la potencia activa consumida por la luminaria LED en función del proceso de atenuación de la intensidad luminosa, a su vez, permitirá analizar el factor de potencia producto del proceso de dimerización, con el objeto de analizar el desempeño del prototipo.

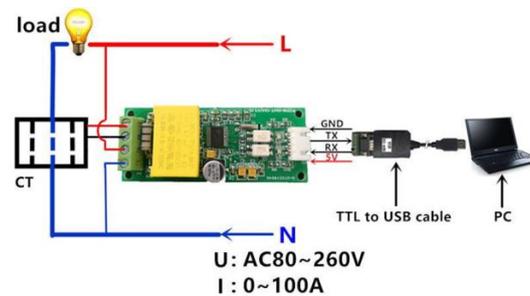


Figura 30 Diagrama de cableado eléctrico para el módulo PZEM-004T-100A

Nota. Adaptado de (InnovatorsGuru, 2019)

5.1.8 Optoacoplador 4N35

Un optoacoplador, también llamado aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor o fototriac (Malvino, 2000). Este dispositivo optoacoplador consiste en un diodo emisor de infrarrojos de arseniuro de galio acoplado ópticamente a un detector fototransistor monolítico de silicio. De este modo se combinan en un solo dispositivo encapsulado de 6 pines que por lo general es del tipo DIP. Entre las principales funciones que permiten estos dispositivos y las cuales serán implementadas dentro del prototipo elaborado, se incluyen las siguientes:

- Circuitos de conmutación de propósito general.
- Sistemas de interconexión y acoplamiento de diferentes potenciales e impedancias.
- Crear puentes entre señales de corriente alterna y circuitos lógicos que utilicen un voltaje menor o igual a 5V.
- Control en etapas de potencia.
- Circuitos de monitorización y detección.
- Relés de estado sólido.

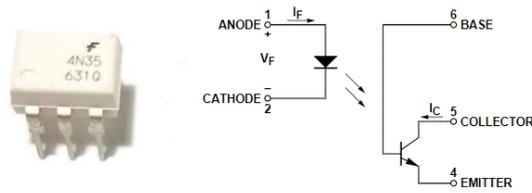


Figura 31 Optoacoplador 4N35 y diagrama esquemático

Nota. Adaptado de (Avago Technologies, 2007)

La Figura 32 presenta el diagrama esquemático de este optoacoplador, el cual posee algunas características eléctricas importantes como:

- Tensión de aislamiento: 5,3 kV
- Corriente directa máxima I_F : 50 mA
- Voltaje de colector emisor $V_{(br) ceo}$: 80 V
- Temperatura de operación: -55°C (mínima), 100°C (máxima)

El funcionamiento de este optoacoplador consiste en inyectar al ánodo por medio de la resistencia R_D un pulso de voltaje (*Input*), de modo que el LED sature el fototransistor al otro lado, lo cual genera a su vez una corriente en R_L , así el voltaje de salida (*Output*) será de cero volts y cuando no exista voltaje a la entrada del optoacoplador el voltaje de salida será igual a V_{CC} . Así pues, este dispositivo permitirá la elaboración del bloque detector de cruce por cero de la señal de suministro AC, además de permitir el control del circuito relé, proporcionando un aislamiento galvánico para el control de la etapa de potencia.

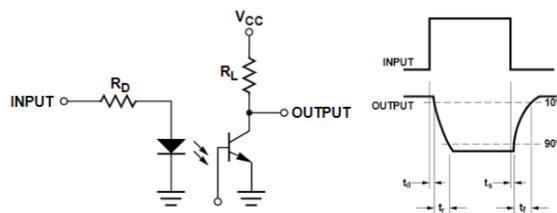


Figura 32 Funcionamiento básico de un optoacoplador 4N35

Nota. Adaptado de (Avago Technologies, 2007)

5.1.9 MOSFET IRF840 CH-N

El dispositivo de conmutación seleccionado para el control y regulación de la potencia suministrada a la lámpara LED para su atenuación fue el MOSFET IRF840. El Transistor IRF840 de efecto de campo metal-óxido-semiconductor o MOSFET (*Metal-oxide-*

semiconductor Field-effect transistor, en inglés) es un transistor utilizado para amplificar o conmutar señales electrónicas. Este transistor MOSFET permite una conmutación rápida, baja resistencia de encendido y rentabilidad, lo cual contribuye a su amplia aceptación en toda la industria microelectrónica, ya sea en circuitos analógicos o digitales (UNIT Electronics, 2016).

Entre sus principales características eléctricas destacan:

- Disipación total del dispositivo (P_d): 125 W
- Tensión drenaje-fuente (V_{ds}): 500 V
- Tensión compuerta-fuente (V_{gs}): 20 V
- Corriente continua de drenaje (I_d): 8 A
- Temperatura operativa máxima (T_j): 150 °C
- Tensión umbral compuerta-fuente $V_{gs(th)}$: 4 V
- Carga de compuerta (Q_g): 63 nC
- Conductancia de drenaje-sustrato (C_d): 1500 pF
- Resistencia drenaje-fuente $R_{DS(on)}$: 0,85 Ohm
- Empaquetado / Estuche: TO220

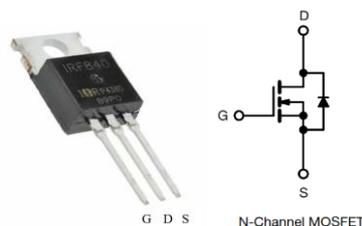


Figura 33 Transistor MOSFET IRF840

Nota. Adaptado de (UNIT Electronics, 2021)

La Figura 33 muestra el esquema de este dispositivo. Su funcionamiento consiste en un interruptor que se activa por tensión en el terminal llamado puerta (*gate*), permitiendo el paso de la corriente I_{DS} entre el terminal llamado fuente (*source*) y el terminal llamado drenador (*drain*). De esta manera al aplicar tensión conduce y cuando no hay tensión en la puerta no conduce. Este dispositivo MOSFET de potencia está diseñado para aplicaciones como reguladores de conmutación, convertidores de conmutación, controladores de motores, controladores de relés y controladores de transistores de conmutación bipolares de alta potencia

que requieren alta velocidad y baja potencia de accionamiento de la puerta (FAIRCHILD SEMICONDUCTOR, 2002).

5.2 Metodología

5.2.1 Modalidad de la investigación

En esta tesis de investigación aplicada, se ha seguido una metodología científica, basada en los conocimientos teóricos de distintas ramas de la ingeniería como la eléctrica, electrónica, energías renovables y sistemas. Así pues, se utilizó los conocimientos existentes para la creación de un prototipo electrónico en respuesta al problema de la baja eficiencia energética de los sistemas de iluminación tradicionales. En este sentido, la modalidad de investigación se puede subdividir en los siguientes puntos, tales como:

- Investigación bibliográfica: permitió la exploración y recopilación de información sobre temas de interés, desde una perspectiva estructurada y profesional, para lo cual se analizó diversas fuentes como: libros, informes técnicos, artículos científicos, trabajos de tesis relacionados, sitios web, entre otros.
- Investigación experimental: tuvo como función el análisis y desarrollo de diversas pruebas dentro del laboratorio de electrónica de la empresa TECSERLED, para la evolución del prototipo propuesto. Además, para el correcto desempeño del prototipo se realizó diversas pruebas que involucró el ajuste de componentes a nivel de hardware como también cambios a nivel software, que en conjunto permitió la optimización del funcionamiento del prototipo.
- Investigación empírica: a partir de las pruebas de funcionamiento y la observación de las mediciones tomadas de los parámetros eléctricos del prototipo, se pudo realizar diversas conclusiones que enfatizan el correcto funcionamiento del sistema, y la reducción del consumo energético.

5.2.2 Técnicas e instrumentos de recolección de datos

En el desarrollo del trabajo y para la recolección de información se aplicó la técnica de observación documental, para lo cual se obtuvo y registró ordenadamente la información de libros, artículos científicos, trabajos de tesis relacionados, entre otros documentos de bases de datos confiables. Esta técnica permitió obtener información de gran ayuda, como manuales de programación, hojas de datos de componentes electrónicos, patentes, libros de análisis matemático, como también la búsqueda de información puntual mediante foros o sitios web.

5.2.3 Técnicas de análisis y procesamiento de la información

Para el procesamiento de la información recopilada durante el trabajo se utilizó un enfoque cuantitativo de investigación. Eso quiere decir que, para la aplicación de esta técnica en la investigación se procedió a realizar lo siguiente:

- Análisis del contenido de los documentos bibliográficos relacionados a los sistemas de iluminación LED, tipos de sistemas de control y regulación de luminarias, así como información específica de los componentes hardware para la implementación del prototipo.
- Estudio y análisis de diversas propuestas de solución que permitan el desarrollo de un sistema de iluminación más eficiente y de bajo coste económico.
- Determinación en base a los resultados obtenidos de la mejor propuesta de solución para el sistema de control y regulación de potencia de una luminaria LED dimerizable.
- Desarrollo de pruebas a partir de métodos experimentales del funcionamiento del sistema de control de iluminación, recopilación de los resultados obtenidos de las pruebas, y análisis cuantitativo de los datos obtenidos. Este análisis permitió en segunda instancia demostrar la validez y confiabilidad de los resultados obtenidos, utilizando instrumentos que han demostrado ser válidos y confiables en estudios previos.

5.2.4 Desarrollo del proyecto

Por su parte, el desarrollo del presente trabajo de titulación se subdivide en el siguiente orden o marco metodológico:

Tabla 12 Secuencia de capítulos de investigación

Secuencia de capítulos de investigación	
1	Sistema de control de potencia basado en un circuito atenuador de control de fase AC
2	Sistema de adquisición de datos de los sensores
3	Sistema de control automático para la regulación de la iluminación
4	Sistema de comunicación inalámbrico
5	Herramientas software <i>open source</i> para el almacenamiento y gestión de la información
6	Sistema de monitoreo, gestión y visualización de la información
7	Implementación y validación del prototipo de sistema inteligente de iluminación
8	Análisis de la reducción del consumo energético por la implementación del prototipo

Por otro lado, la metodología se ha estructurado en las siguientes tareas:

Tarea 1. Sistema de control de potencia basado en un circuito atenuador de control de fase AC

- Estudio de circuitos atenuadores de control de fase AC.
- Diseño y simulación del circuito de control de potencia.
- Selección de componentes y diseño electrónico.
- Programación en Arduino IDE del algoritmo para el sistema de control de potencia.
- Selección de lámpara LED dimerizable y especificaciones técnicas.
- Prototipado y análisis de las formas de onda resultantes.
- Análisis del factor de potencia en el circuito de atenuación de iluminación.
- Desarrollo de placas de circuito impreso PCB.

Tarea 2. Sistema de adquisición de datos de los sensores

- Estudio del módulo sensor BH1750 para la captación del nivel de luminosidad.
- Estudio del sensor de temperatura y humedad relativa DHT22.
- Estudio del módulo sensor de presencia PIR HC-SR501.
- Estudio del dispositivo PZEM004T para la medición de parámetros eléctricos (voltaje, corriente, potencia activa, frecuencia, factor de potencia y consumo de energía).
- Diagrama y cableado eléctrico para la correcta adquisición de los datos de los sensores.
- Programación en la plataforma Arduino IDE y adquisición de datos de los sensores.

Tarea 3. Sistema de control automático para la regulación de la iluminación

- Estudio de los tipos de sistemas de control.
- Estudio del controlador proporcional, integral, derivativo PID.
- Identificación del modelo matemático de la planta de control, mediante herramientas de software matemático.
- Obtención de los parámetros del controlador PID y simulación del comportamiento de la planta.
- Programación en la plataforma Arduino IDE del algoritmo de control automático para la regulación de la iluminación.
- Realización de pruebas y validación del algoritmo de control automático.

Tarea 4. Sistema de comunicación inalámbrico

- Instalación y configuración del sistema operativo Raspbian en el ordenador central Raspberry Pi.

- Estudio del protocolo de comunicación MQTT para redes inalámbricas WSN.
- Instalación y configuración del servidor local MQTT en el ordenador central del sistema.
- Programación en la plataforma Arduino IDE para la comunicación de los clientes NodeMCU ESP8266 con el bróker MQTT.
- Pruebas de transmisión y recepción de información proveniente de los sensores entre el dispositivo NodeMCU y el ordenador central del sistema.

Tarea 5. Herramientas software *open source* para el almacenamiento y gestión de la información

- Configuración de la herramienta Node-RED desde la terminal de Linux.
- Instalación del servidor local LAMP en el ordenador central.
- Creación y configuración de la base de datos local para el funcionamiento del prototipo.
- Realización de pruebas y gestión de la base de datos.

Tarea 6. Sistema de monitoreo, gestión y visualización de la información

- Estudio de la interfaz de programación de aplicaciones de Node-RED.
- Estudio de los nodos y programación de distintos flujos de programa en Node-RED para la elaboración de la interfaz de usuario web.
- Realización de pruebas de transmisión y almacenamiento de datos en el servidor.
- Pruebas e implementación del sistema de monitoreo en tiempo real de variables ambientales mediante el uso de la interfaz de usuario web.
- Estudio de la plataforma Cloud Iot ThingSpeak.
- Programación en Python y envío de datos del sistema a la plataforma ThingSpeak.

Tarea 7. Implementación y validación del prototipo de sistema inteligente de iluminación

- Construcción y montaje del hardware del prototipo.
- Construcción de un módulo de pruebas para el acoplamiento del sistema de iluminación.
- Implementación de los módulos individuales contruidos para el control de la iluminación.
- Realización de pruebas, detección y corrección de errores del prototipo para su validación.

Tarea 8. Análisis de la reducción del consumo energético por la implementación del prototipo

- Realización de pruebas, recopilación de información y almacenamiento de los parámetros de funcionamiento del prototipo en la base de datos local.
- Análisis cuantitativo de los datos obtenidos y utilización de técnicas estadísticas.
- Análisis del ahorro energético en función del sistema de detección de presencia.
- Análisis del ahorro energético en función del sistema automático de control de potencia.
- Validación de los resultados obtenidos mediante la interfaz de usuario web.
- Evaluación del ahorro energético y reducción del gasto económico mediante la implementación del sistema de iluminación inteligente.

6 Resultados

6.1 Descripción general del sistema inteligente de iluminación LED

El desarrollo del prototipo se dividió en diferentes fases de diseño, así como de elección de dispositivos hardware que, en conjunto, permitió establecer la topología general del sistema de iluminación inteligente. En este sentido, el objetivo de la propuesta consistió en desarrollar un prototipo que permita encender/apagar las luminarias en respuesta a la presencia de personas, regular la potencia entregada a las luminarias en base a las condiciones de luminosidad del ambiente, así como también realizar paralelamente la monitorización de las condiciones ambientales del área y la gestión energética del sistema de iluminación a través del envío de esta información a una plataforma de control y visualización almacenada en una Raspberry pi 3, permitiendo así la capacidad de adquirir la información necesaria para validar la optimización energética dentro del área de estudio.

Como resultado el prototipo final se separa en dos partes fundamentales: el control de iluminación que se realiza por medio de un circuito electrónico en placa PCB y la gestión de la información a través de una interfaz de usuario web alojada en el ordenador central. Para una mejor descripción de las dos partes, puede apreciarse el diagrama general del sistema (ver Anexo 1), el cual a su vez se puede dividir en cuatro bloques principales para el objeto de estudio, tales como:

- La adquisición de datos de los sensores inteligentes.
- El proceso de acondicionamiento y procesamiento de la información.
- El sistema de control y regulación automático de la luminaria.
- El sistema de monitoreo, gestión y visualización.

6.1.1 Adquisición de datos de los sensores

El bloque de adquisición de datos permite obtener la información necesaria para el desarrollo de un sistema inteligente de iluminación. En esta parte se utilizó el sensor luxómetro BH1750 para la obtención del nivel de iluminancia del área de estudio (medida en luxes), que, junto a la de la señal de presencia de personas obtenida por el sensor PIR, permite obtener la información necesaria para implementar un algoritmo de control automático basado en un sistema en lazo cerrado que analiza tanto la presencia de personas como también el nivel de iluminación requerido y establecido en luxes en base a la norma NTE INEN 2969-1 (ver apartado 4.12), para posteriormente por medio del algoritmo enviar una señal de ajuste a la planta cuya finalidad es regular la potencia entregada a la luminaria. Adicionalmente esta etapa conlleva la adquisición de los datos provenientes del sensor DHT22 que permite obtener información precisa de la temperatura y humedad para su análisis y envío a la interfaz de usuario web. Por último, con la ayuda del módulo PZEM004T se obtiene información valiosa relacionada con el consumo eléctrico de la luminaria dentro del área de funcionamiento, la cual, a través de su visualización en una interfaz gráfica permite analizar la optimización energética lograda por el prototipo.

6.1.2 Proceso de acondicionamiento y procesamiento de la información

El proceso de acondicionamiento y procesamiento de la información se lleva a cabo tanto en los mismos sensores inteligentes como en el MCU que procesa las tramas de datos enviadas por los sensores, esto con el objetivo de procesar el algoritmo de control inteligente, como también acondicionar la información para su envío al ordenador central del sistema por medio del uso de la tecnología de comunicación inalámbrica WiFi. En última instancia el ordenador central (Raspberry Pi) procesa y almacena la información para permitir su visualización al usuario y realizar diferentes operaciones de gestión como también de cálculo del ahorro energético-económico.

6.1.3 Sistema de control automático

El sistema de control automático es la base fundamental del prototipo electrónico y está dividido a su vez en dos partes; la parte de software de control automático mediante un algoritmo implementado en el MCU con los datos provenientes de los sensores (luxómetro y PIR) y la parte del circuito electrónico (*dimmer*) que permite la regulación de la intensidad luminosa, y en otras palabras la potencia aplicada a la carga (lámpara LED), en base a la señal de control suministrada por el algoritmo. Para explicar el funcionamiento del sistema de control inteligente de iluminación, se puede resumir de la siguiente manera: Inicialmente el MCU

realiza la lectura de los sensores y determina si el nivel de iluminancia en el área está por debajo de un valor máximo establecido como rango de operación, en base a este valor medido determina si enciende o no la lámpara LED, y con ello el algoritmo de control inteligente para su regulación automática. Al activar el algoritmo de control automático este verificará constantemente el estado del sensor PIR, de modo que, la presencia de personas permite al algoritmo mantener el nivel de iluminancia en el valor de referencia establecido en luxes (en adelante llamado *Setpoint*) y la ausencia de personas permite realizar un apagado suave de la lámpara hasta que se detecte un nuevo evento de activación del sensor PIR, y con ello realizar un encendido adecuado de la luminaria.

6.1.4 Sistema de monitoreo, gestión y visualización

La información enviada por el MCU mediante una comunicación inalámbrica WiFi, posteriormente es procesada y almacenada en una base de datos local, para permitir su visualización en una interfaz de usuario web (phpMyAdmin) alojada en el ordenador central (Raspberry Pi 3) para su correspondiente administración. Por otra parte, mediante la programación en la herramienta Node-RED integrada en el mismo ordenador central se estructura y programa el sistema de monitoreo y control. Este sistema permite al usuario por medio de una interfaz gráfica GUI (*Graphic User Interface*, en inglés) el manejo del sistema de iluminación inteligente, ya sea de forma automática mediante el ajuste del nivel de iluminación deseado, como también un control manual del sistema, a fin de cumplir estas características se realizó una comunicación bidireccional para el envío y recepción de información entre el dispositivo NodeMCU y el ordenador central con el uso del protocolo de comunicación MQTT analizado posteriormente. Aunado a esto, el sistema permite acceder a la información del consumo energético en cualquier instante gracias a la obtención de reportes de consumo programados y accesibles por medio de la interfaz gráfica para su visualización.

Es importante mencionar que todo el prototipo electrónico a su vez cuenta con dos tipos de suministros eléctricos. La parte de suministro eléctrico de corriente continua en baja tensión permite alimentar a los dispositivos sensores y microcontroladores, por otra parte, el suministro eléctrico en corriente alterna de bajo voltaje (120 V_{AC}) permite alimentar las fuentes convertidoras, así como también permite el funcionamiento del circuito atenuador *dimmer* para la regulación de las lámparas LED. En este sentido fue necesario realizar un aislamiento galvánico para el acoplamiento de diferentes potenciales al prototipo como se detallará más adelante.

6.1.5 Diagrama de bloques del sistema de iluminación inteligente

El diagrama de bloques elaborado (ver Anexo 1) responde a la descripción general del sistema inteligente de iluminación LED, el cual como se mencionó anteriormente está dividido en cuatro conjuntos de bloques, que permiten la funcionalidad del sistema y el cumplimiento de los objetivos abordados en el presente trabajo, los cuales son:

- Adquisición de datos de los sensores
- Proceso de acondicionamiento y procesamiento de la información
- Sistema de control automático
- Sistema de monitoreo, gestión y visualización

6.2 Sistema de control de potencia basado en un atenuador de control de fase AC

Este apartado tiene la finalidad de describir el circuito electrónico diseñado para realizar el control de potencia otorgada a la luminaria y por ende la regulación de su intensidad luminosa. En base al estudio realizado de los distintos tipos de dispositivos que permiten la regulación y control de luminarias con suministro AC (ver apartado 4.9), se determinó que el diseño del circuito electrónico responda a un atenuador de control de fase de borde posterior (ver apartado 4.10.2), ya que sus características son las más adecuadas para su aplicación con luminarias LED. En este sentido se puede dividir el circuito electrónico atenuador en diferentes bloques que permiten analizar su funcionamiento y las señales que pasan por cada uno de ellos, estos son:

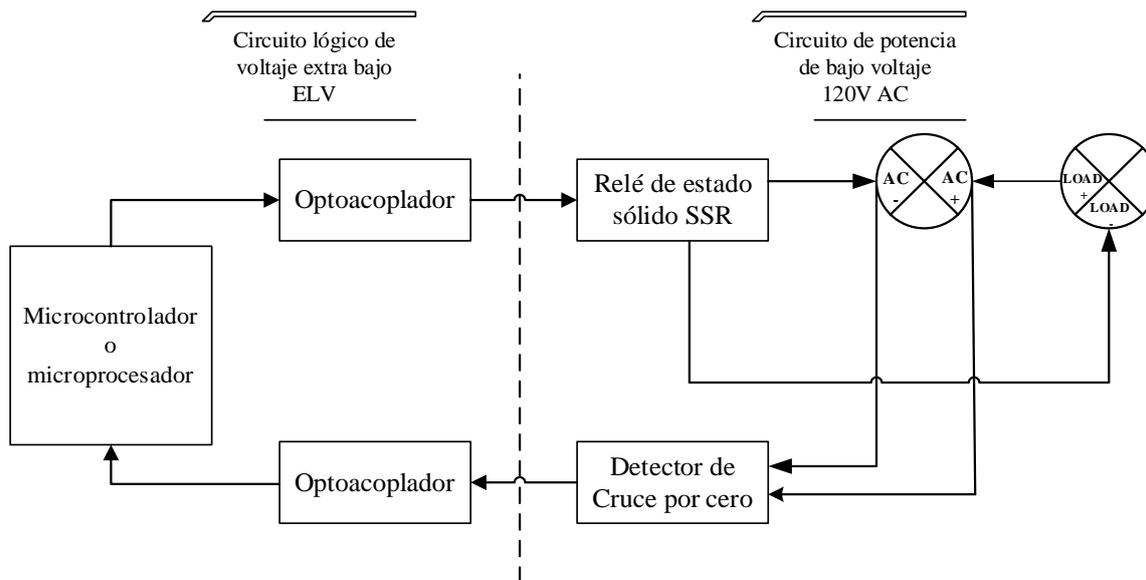


Figura 34 Diagrama de bloques del sistema de control de potencia

De la Figura 34 se observan 3 bloques principales separados entre ellos por un aislamiento galvánico el cual se logra con el uso de optoacopladores, el primer bloque está conformado por un circuito detector de cruce por cero, que tiene la finalidad de proporcionar una señal de información de entrada al MCU, este a su vez genera una señal de salida que permite controlar un circuito relé para su activación o desactivación gracias a la ayuda intermedia de otro dispositivo optoacoplador. Los apartados siguientes tienen la finalidad de realizar una explicación detallada de los bloques y los dispositivos electrónicos necesarios que permiten el funcionamiento del circuito electrónico (*dimmer*) para la regulación de la intensidad luminosa.

6.2.1 Circuito detector de cruce por cero

Este circuito eléctrico permite detectar el instante de tiempo en el cual el voltaje de la red tiene un valor de 0 volts, de esta manera el circuito emite un pulso de valor positivo que posteriormente es leído por el MCU para generar una interrupción externa y emitir la señal de control al circuito de potencia. Para este propósito se utilizó la configuración mostrada en la Figura 35 en donde el puente rectificador tiene la función de permitir el paso de las señales de voltaje a la entrada del optoacoplador (pin 1), tanto en alternancia positiva como en alternancia negativa, de este modo permite polarizar constantemente el ánodo del LED del optoacoplador, en consecuencia, el LED satura el fototransistor del otro lado, lo cual genera a su vez una corriente en la resistencia de *pull-up* (R3), así el voltaje de salida será de cero volts. Por el contrario, cuando se detecta un cruce por cero, la señal de entrada al LED es de 0 volts, el LED no se polariza y el otro lado del optoacoplador actúa como un circuito abierto, generando una

señal de voltaje a la salida. La parte derecha de la Figura 35 muestra la simulación del circuito implementado, los pulsos de salida generados serán leídos posteriormente por el MCU mediante sus pines de entrada GPIO.

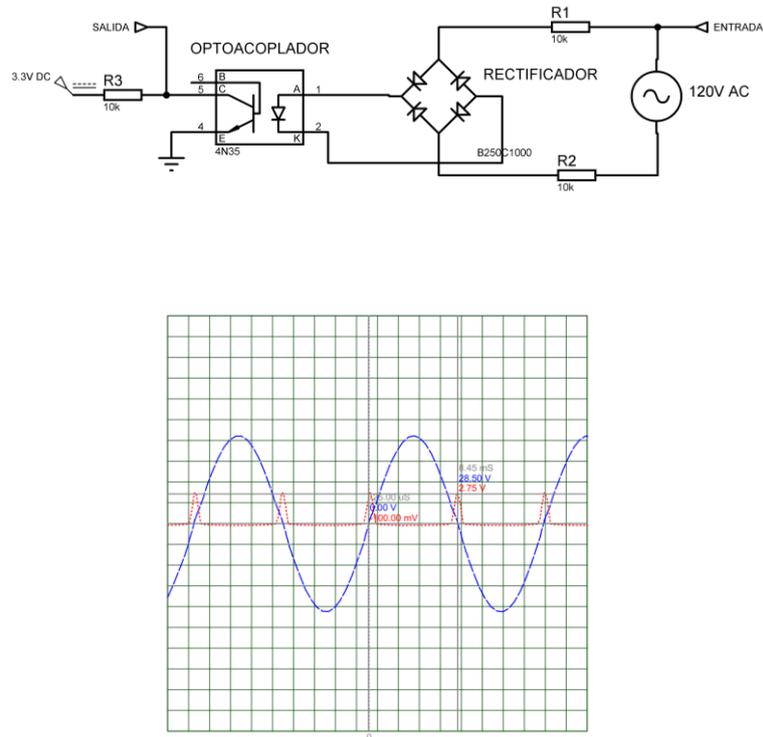


Figura 35 Circuito detector de cruce por cero

Nota. Simulación realizada en Proteus 8 Professional

6.2.2 Relé de estado sólido SSR

Un relé de estado sólido SSR (*Solid State Relay*, por sus siglas en inglés) es un dispositivo electrónico de conmutación que se enciende o apaga cuando se aplica una tensión externa a sus terminales de control. Un SSR tiene la misma función que un relé electromecánico, pero sin partes móviles, además permite velocidades de conmutación más rápidas y no tienen contactos físicos que se desgasten, otra ventaja es que necesita menos corriente y tensión para controlar cargas AC de bajo voltaje (Texas Instruments Incorporated, 2016). Para su funcionamiento los SSR utilizan dispositivos semiconductores de potencia como tiristores o transistores, que permiten conmutar corrientes de hasta 100 A, otra particularidad es que un SSR puede estar diseñado para controlar una carga de tensión o corriente del tipo alterna o continua.

Según (Elliott, 2019) la ventaja de la implementación de un SSR MOSFET es la capacidad de controlar la velocidad de conmutación para minimizar la EMI (Interferencia electromagnética) a la vez de reducir la caída de voltaje en comparación con un relé basado en TRIAC o SCR, permitiendo una gestión eficiente de la energía. Otra ventaja es que un relé MOSFET no tiene problemas con la corriente de retención, la cual está presente en los dispositivos TRIAC y SCR, esto permite al relé MOSFET controlar miliamperios y amperios con facilidad.

El esquema general de un SSR AC se puede observar en la Figura 36, se utilizan dos MOSFET de conmutación de canal N, con sus fuentes y puertas unidas, la señal y la carga están conectadas a cada uno de los terminales de drenaje, no importa cuál es cuál, porque el interruptor es simétrico. Para mostrar el funcionamiento de conmutación del SSR MOSFET se analiza primero el lado izquierdo de la Figura 36, aquí el uso de los dos transistores conectados entre sí permite que fluyan corrientes positivas y negativas durante el tiempo de encendido, esto utilizando el diodo de cuerpo de uno de los MOSFET para la corriente positiva y el otro para la corriente negativa. Durante el tiempo de apagado, los diodos de cuerpo bloquean el flujo de corriente porque el diodo de cuerpo superior e inferior se polarizan inversamente, como se muestra en el lado derecho de la Figura 36 (Texas Instruments Incorporated, 2016) . Para una mejor explicación del funcionamiento se analiza la Figura 37, la activación del MOSFET consiste en aplicar un voltaje entre los terminales puerta y fuente, lo cual para los MOSFET utilizados, se conducirá completamente cuando el voltaje V_{GS} exceda el voltaje de umbral $V_{GS(th)}$, aunque siempre es adecuado proporcionar un controlador de puerta de 10-12 V, esto permite reducir la resistencia del canal, una mayor concentración de cargas libres en el canal y en consecuencia permitir una mayor corriente I_{DS} , en este caso se dice que el transistor está en conducción ($V_{GS} > V_{GS(th)}$). Cuando los MOSFET están conduciendo, la energía fluye desde el activo a través de la carga y luego de regreso al neutro a través de Q1 y Q2 en serie, además, la conexión de la fuente es necesaria para proporcionar una referencia para el voltaje de la puerta y como una ruta de retorno para la fuente de alimentación de regreso al neutro (a través del diodo interno en Q2). Por el contrario, cuando no existe voltaje V_{GS} en la entrada de control, los MOSFET están apagados ($V_{GS} < V_{GS(th)}$) y no fluye la corriente I_{DS} por ninguno de los dos transistores ya que ambos se comportan como interruptores abiertos, en este caso se sobreentiende que los transistores están en la región de corte.

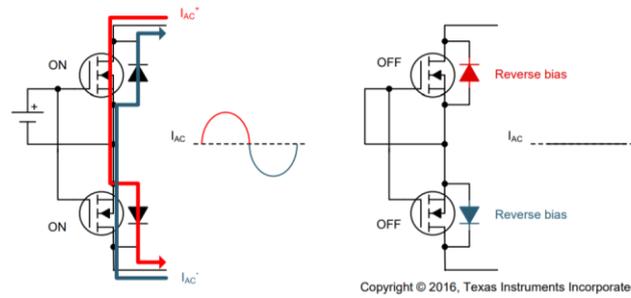


Figura 36 Esquema básico de un relé de estado sólido SSR

Nota. Adaptado de (Texas Instruments Incorporated, 2016)

Ahora bien, el circuito de control generalmente está aislado del circuito SSR de conmutación, adicionalmente se requiere una alimentación aislada para el circuito de activación de puerta, es decir, para activar los MOSFET se necesita una fuente de corriente continua que se active o desactive según sea necesario encender o apagar el relé. En este sentido la función de conmutación puede ser implementada mediante optoaisladores (LED + fototransistor). Para ejemplificar este funcionamiento se observa la parte izquierda de la Figura 37, aquí la señal de control (entrada al optoacoplador) se acopla al SSR mediante un aislamiento galvánico entre ambos circuitos lo cual es producido gracias al acoplamiento óptico, de modo que la señal de voltaje de control enciende un LED interno que ilumina y activa el fototransistor, esto a su vez genera una corriente en la resistencia de *pull-down* (R6) que permite generar el voltaje V_{GS} para la activación de los MOSFET y el encendido de la carga, no obstante, cuando la entrada al optoacoplador está en un nivel bajo, el fototransistor del otro lado no se activa, no se genera corriente y la caída de tensión en la resistencia es prácticamente 0V, por lo tanto los transistores estarán apagados.

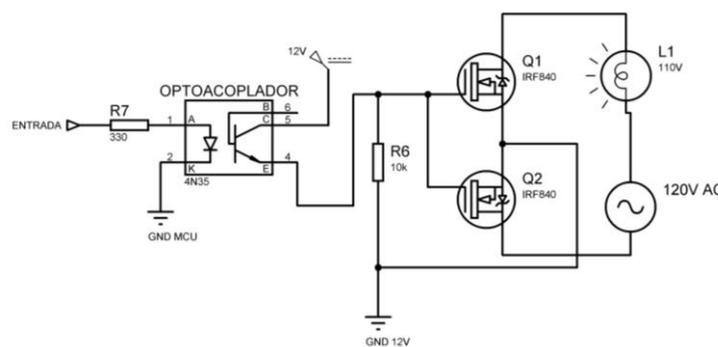


Figura 37 Circuito relé de estado sólido SSR basado en MOSFET

Es importante destacar que la etapa de alimentación de CC para la activación de las puertas esta proporcionada por una fuente de alimentación independiente (es decir un transformador alimentado por la red por separado), aquí la CC aplicada a las puertas MOSFET

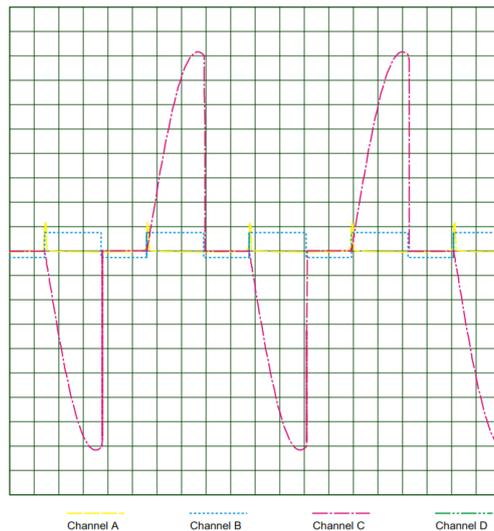


Figura 39 Gráficas de las formas de onda de salida

Nota. Canal A (detección de cruce por cero), Canal B (señal de control PWM), Canal C (Forma de onda de voltaje recortada en fase).

6.2.4 Diseño electrónico

En la Figura 40 se observa el diseño electrónico del prototipo de sistema de control inteligente de iluminación, el cual consta del dispositivo microcontrolador, el circuito detector de cruce por cero y el relé de estado sólido SSR. El MCU se cablea directamente con los sensores a través de la regleta de pines representada por J3, la cual también dispone de pines para su alimentación. El bloque conformado por el transistor NPN (Q3) hace la función de inversor, es decir, la señal de salida (voltaje) es la inversa de la señal de entrada, esta funcionalidad permite eliminar encendidos accidentales de la lámpara LED debido a pulsos indeseados al momento de arrancar (*boot*) el microcontrolador. Los capacitores C1, C2, C3 permiten suavizar las ondulaciones y mantener la estabilidad del regulador L7805CV evitando las oscilaciones de voltaje a la salida del regulador y en otras palabras mejorando la respuesta transitoria.

Cabe destacar que para la alimentación del circuito se utiliza una fuente HI-LINK HLK-5M12 la cual permite suministrar el voltaje necesario ($12V_{DC}$) para la activación de los MOSFET, así como también alimentar mediante un regulador de voltaje L7805CV al MCU y al resto de dispositivos que trabajan en corriente continua en voltaje extra bajo ELV (sensores inteligentes, modulo medidor de energía eléctrica).

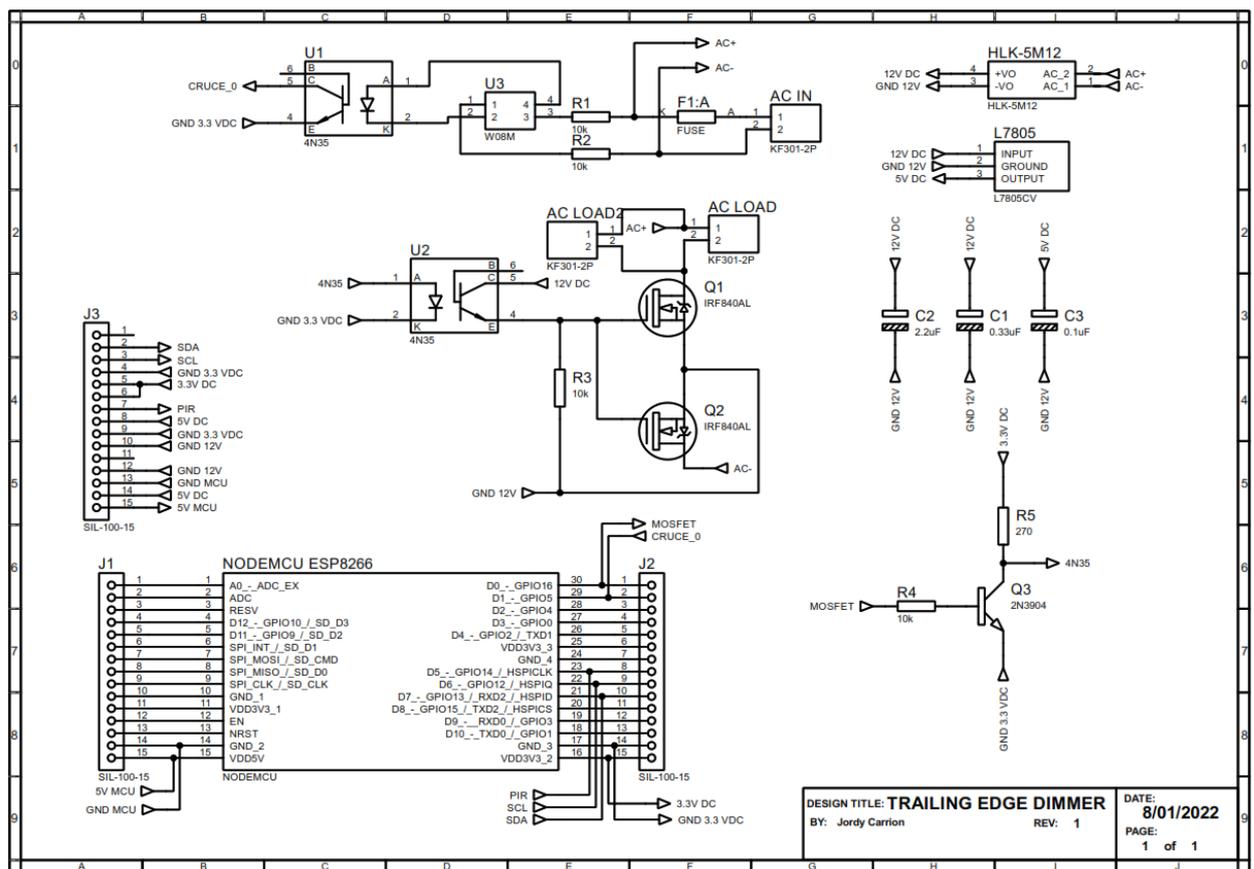


Figura 40 Diseño electrónico del prototipo inteligente de iluminación

6.2.5 Diagrama del algoritmo de control de potencia

El presente apartado describe el proceso o flujo del algoritmo implementado para la regulación de potencia del circuito electrónico *dimmer*. El algoritmo implementado para esta función consta de dos señales principales: una señal de entrada correspondiente a la señal de detección de cruce por cero y una señal de control de salida (PWM), esta última determina el ciclo de trabajo para la activación del relé SSR, lo cual permite variar el nivel de potencia entregada a la carga. Es importante destacar que, para fines de pruebas y experimentación, el ajuste del ciclo de trabajo de la señal PWM se realizó con el uso de un potenciómetro conectado al pin A0 correspondiente al convertidor digital del MCU. Esto debido a que posteriormente el ajuste de la señal de control tendrá un comportamiento automático, gracias a la implementación de un algoritmo de control inteligente de iluminación. De esta manera el diagrama de flujo mostrado en (ver Anexo 2) describe el flujo del algoritmo implementado.

6.2.6 Prototipado del circuito de control de potencia

Una vez definido el diseño del circuito electrónico encargado del control de potencia, basado en un atenuador de control de fase de borde posterior, se procedió a realizar el prototipado del circuito y las pruebas de funcionamiento. En consecuencia, luego de muchas

pruebas experimentales dentro del laboratorio, y el ajuste de las variables del algoritmo de control de potencia, se logró establecer los parámetros de funcionamiento óptimos para el correcto desempeño del circuito atenuador de iluminación. En este sentido, la siguiente Figura 41 ejemplifica el desarrollo de las pruebas realizadas en la fase de prototipado del circuito. Cabe mencionar que el control de la atenuación es implementado mediante el uso de un potenciómetro y el convertidor ADC del NodeMCU, además, se consideró en primera instancia el uso de una carga puramente resistiva (foco incandescente), para una mejor visualización de la señal de voltaje obtenida en la carga.

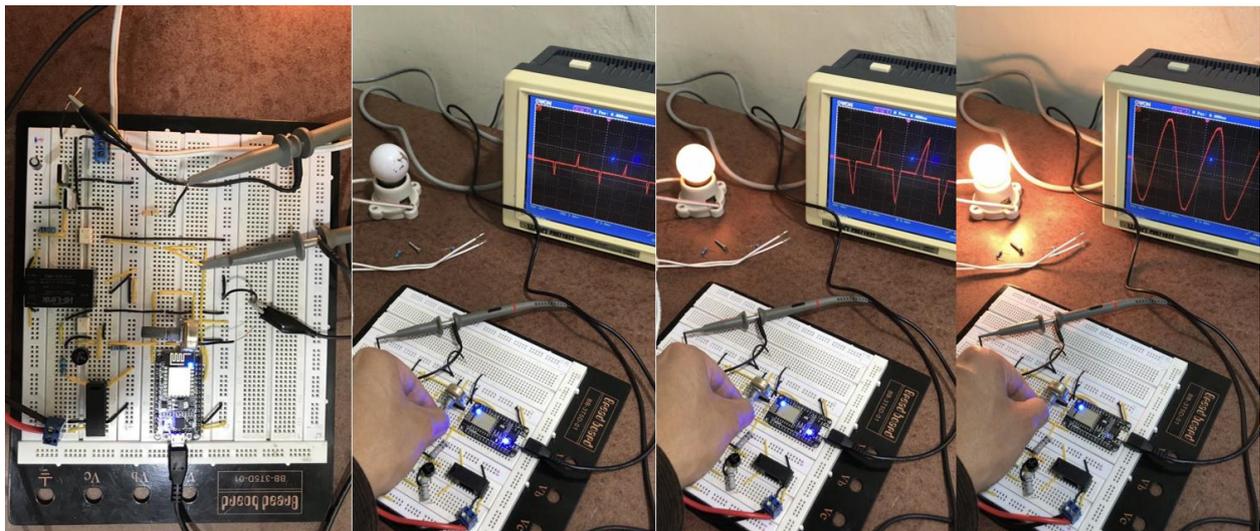


Figura 41 Prototipado y pruebas de funcionamiento del circuito de control de potencia

6.2.7 Formas de onda resultantes del circuito atenuador de potencia

Para mostrar el funcionamiento del prototipo se procedió a captar y visualizar las diferentes señales de salida de los bloques analizados en el circuito electrónico (ver apartado 6.2.3). Para este propósito se usó un osciloscopio digital de la marca OWON PDS71027, y el software correspondiente, que en conjunto permite obtener una interfaz gráfica en la PC para la obtención de los datos provenientes del osciloscopio y su visualización detallada. De esta forma se analiza cada una de las señales obtenidas en las diferentes partes del prototipo electrónico.

- a) Señal de salida del circuito detector de cruce por cero.

La Figura 42 muestra la señal AC de entrada al circuito detector de cruce por cero, y los pulsos generados a la salida del optoacoplador 4N35 (pin 5), los resultados se asemejan con precisión a los valores simulados (ver apartado 6.2.3). La información generada por el osciloscopio de la señal de la Figura 43 determina que los pulsos se generan a una frecuencia

de 120 Hz y poseen una amplitud de 1,88 V, esta señal de entrada al microcontrolador permite generar los tiempos de activación para el circuito relé SSR.

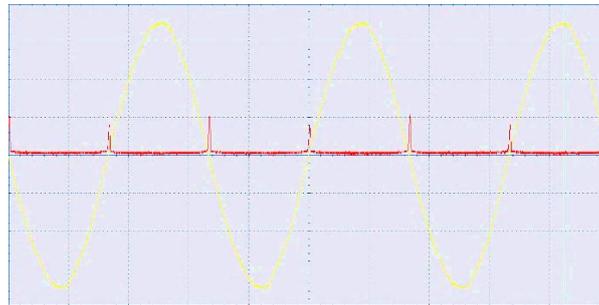


Figura 42 Señal de entrada AC y pulsos generados por el circuito detector de cruce por cero

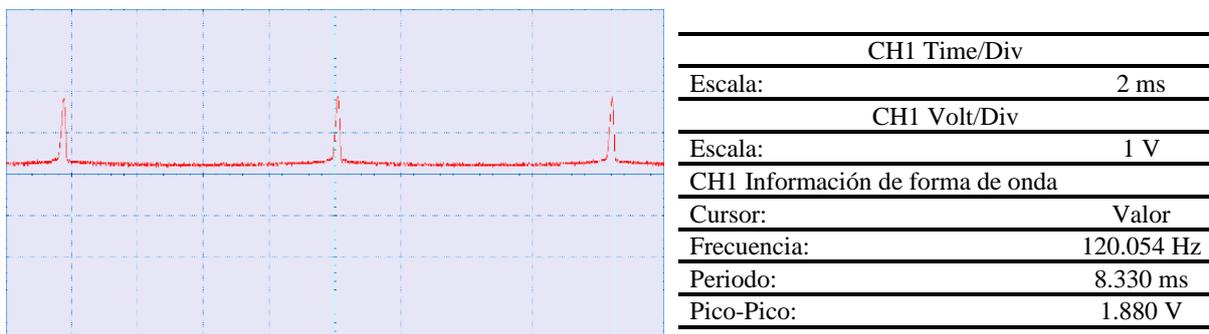


Figura 43 Información de los pulsos de detección de cruce por cero

b) Señal de salida de control del microcontrolador.

La Figura 44 muestra la señal de salida del microcontrolador, la cual es generada al detectarse la señal de entrada de cruce por cero, el microcontrolador realiza una interrupción e inmediatamente genera la señal PWM con un ciclo de trabajo cuyo valor está calculado en base al algoritmo de control, el límite de duración de esta señal está establecido por el periodo/2 de la señal AC de alimentación que es de 8,333 milisegundos. Al variar el ciclo de trabajo de la señal de salida, se controla que parte de la onda senoidal estará activa y de esta forma se regula la cantidad de energía entregada a la carga. Las gráficas siguientes muestran como varía la señal de control, otorgando así diferentes ciclos de trabajo. Cabe destacar que las señales u ondas cuadradas obtenidas en el osciloscopio se asemejan a una respuesta ideal, estas ondas cuadradas serán las encargadas de alimentar el diodo LED del optoacoplador y permitir la conmutación del circuito relé SSR.

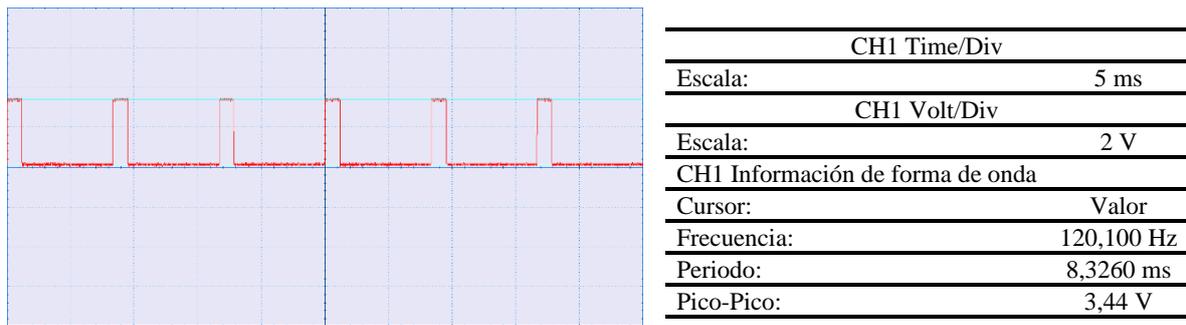


Figura 44 Señal de control PWM (Ciclo de trabajo de 13.25 %) de salida del microcontrolador

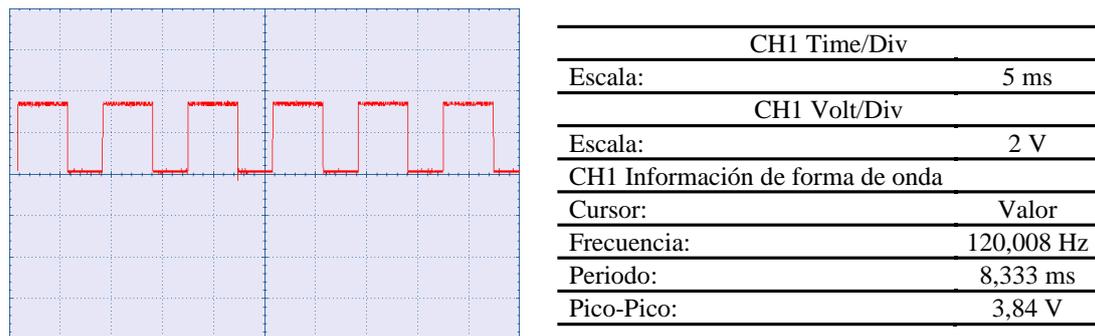


Figura 45 Señal de control PWM (Ciclo de trabajo de 60.24 %) de salida del microcontrolador

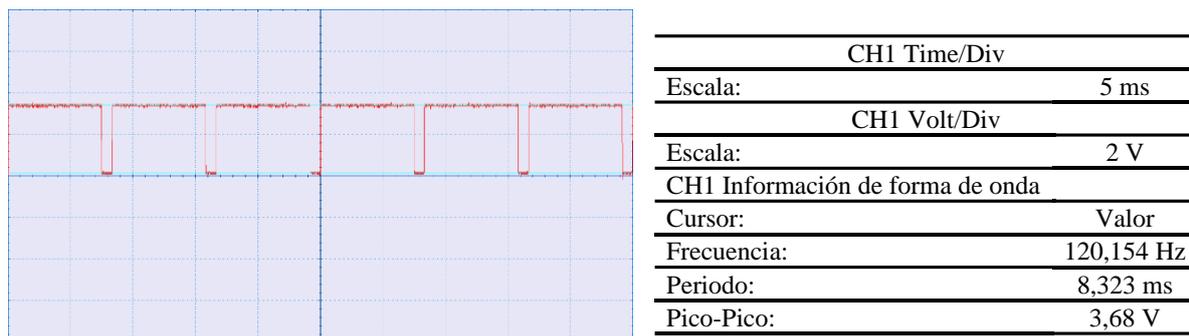
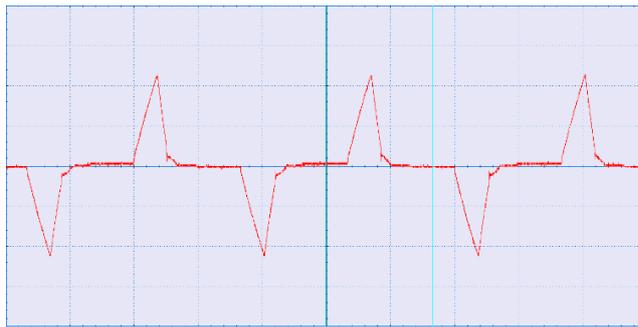


Figura 46 Señal de control PWM (Ciclo de trabajo de 93 %) de salida del microcontrolador

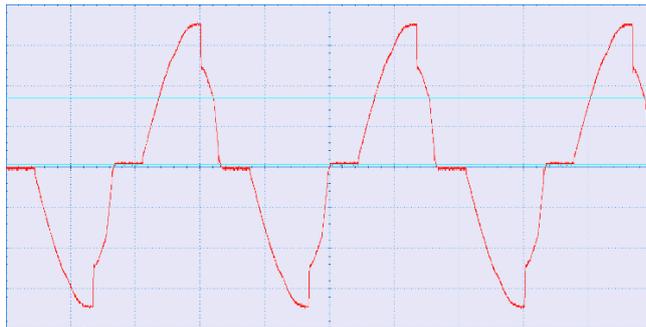
c) Señal de salida del circuito atenuador de control de fase.

En última etapa se analiza la forma de onda correspondiente al voltaje de la carga. La carga que se utilizó para el efecto de esta prueba corresponde a un foco LED de la marca SYLVANIA A60 DIM, el cual posee un consumo de 10 W a 120 V_{AC} y tiene la capacidad de ser regulable o dimerizable (ver Anexo 3). En este contexto se procede a analizar las gráficas resultantes, para diferentes niveles de atenuación.



CH1 Time/Div	
Escala:	5 ms
CH1 Volt/Div	
Escala:	50.0 V
CH1 Información de forma de onda	
Cursor:	Valor
Frecuencia:	59,952 Hz
Periodo:	16,680 ms
Pico-Pico:	226,00 V

Figura 47 Señal de voltaje de la lampara con un nivel bajo de encendido



CH1 Time/Div	
Escala:	5 ms
CH1 Volt/Div	
Escala:	50.0 V
CH1 Información de forma de onda	
Cursor:	Valor
Frecuencia:	60,010 Hz
Periodo:	16,664 ms
Pico-Pico:	352,00 V

Figura 48 Señal de voltaje de la lampara con un nivel medio de encendido



CH1 Time/Div	
Escala:	5 ms
CH1 Volt/Div	
Escala:	50.0 V
CH1 Información de forma de onda	
Cursor:	Valor
Frecuencia:	60,010 Hz
Periodo:	16,656 ms
Pico-Pico:	352,00 V

Figura 49 Señal de voltaje de la lampara con un nivel máximo de encendido

Los resultados obtenidos permiten concluir que el comportamiento del sistema es favorable y el circuito electrónico *dimmer* es compatible con luminarias LED dimerizables, permitiendo su atenuación sin ningún tipo de ruido ni parpadeo. En cada caso la señal de control PWM se enciende en el punto de cruce por cero lo que permite activar el circuito relé SSR en el momento adecuado, esto permite que el sistema se actualice constantemente y no existan problemas de temporización. En configuraciones de baja atenuación el temporizador termina muy rápidamente (alrededor de 1 ms), y a medida que se extiende el ciclo de trabajo de la señal de control, el tiempo de duración se aproxima a la totalidad del periodo/2 de la señal de alimentación, lo cual permite que pase toda la forma de onda de la red antes de que los MOSFET apaguen la corriente en la carga.

6.2.8 Análisis del factor de potencia en el circuito de atenuación de iluminación

Este apartado tiene la finalidad de analizar técnicamente el nivel de eficacia con la que el sistema de iluminación inteligente utiliza la energía eléctrica dentro de una instalación doméstica. Como se mencionó en (ver apartado 4.7) un foco led al incluir un dispositivo driver, se asemeja más a una carga no lineal, en donde el voltaje aplicado es una onda sinusoidal (nominal) de la red, pero la corriente consumida por la carga no lo es.

Para esto se analiza los valores de la potencia real (vatios), así como los valores de voltaje y corriente RMS obtenidos del medidor PZEM004T, que permiten calcular la potencia aparente S . De esta forma a partir de los datos obtenidos se procede a verificar los valores del FP proporcionados por el módulo, con los valores calculados mediante las ecuaciones analizadas anteriormente (ver apartado 4.11), ya que esta aproximación supone una forma más fácil y correcta de medición para cargas no lineales, que calcular el $\cos\phi$, que es el coseno del ángulo de desfase entre V y I . En este sentido, un factor de potencia cercano a uno, indicará que el voltaje y la corriente están en fase (iguales formas de onda) y que la potencia real será igual a la potencia aparente proporcionada por la red de distribución eléctrica.

A fin de cumplir con este propósito se toma en consideración dos casos particulares, el primer caso analiza el FP para una carga resistiva conformada por una lámpara incandescente con un consumo especificado por el fabricante de 25 W a 120 V_{AC}, el segundo caso corresponde a una carga compleja conformada por un foco LED de la serie A60 regulable del fabricante Maviju, el cual posee una disipación de 9 W con voltajes de suministro entre 110-240V AC, un FP especificado de 0,9 y cuya hoja de datos se aprecia en el Anexo 12.

De esta manera se analiza el FP en función del ciclo de trabajo del circuito atenuador, lo cual permite variar la intensidad luminosa de ambas cargas desde un valor mínimo a máximo. Por consiguiente, la siguiente Tabla 13 muestra los parámetros obtenidos del medidor y el FP correspondiente a la carga resistiva.

Tabla 13 Parámetros eléctricos y FP medidos para una carga resistiva (Bombillo incandescente de 25W)

Parámetros eléctricos medidos en la carga resistiva								
PWM [% Ciclo Trabajo]	Voltaje [V]	Corriente [A]	Potencia [W]	Energía [kWh]	Frecuencia [Hz]	FP	FP Calc.	% Error
13	22,10	0,08	1,70	0,003	60,00	1,00	0,96	3,85
16	29,60	0,09	2,70	0,003	60,00	1,00	1,01	1,35
18	31,00	0,09	2,90	0,003	59,90	1,00	1,04	3,94

22	33,60	0,10	3,30	0,003	59,90	1,00	0,98	1,79
26	43,90	0,12	5,00	0,003	59,90	0,99	0,95	4,13
30	49,40	0,12	6,10	0,003	59,90	1,00	1,03	2,90
34	60,60	0,14	8,40	0,003	59,90	1,00	0,99	0,99
40	65,80	0,15	9,50	0,003	59,90	0,99	0,96	2,78
43	77,90	0,16	12,50	0,003	59,90	0,99	1,00	1,30
49	81,70	0,17	13,50	0,003	59,90	1,00	0,97	2,80
55	91,80	0,18	16,20	0,003	59,90	0,99	0,98	0,97
58	98,20	0,19	18,10	0,003	60,00	0,99	0,97	2,01
63	105,30	0,19	20,20	0,003	60,00	0,99	1,01	1,98
68	111,70	0,20	22,20	0,004	60,00	1,00	0,99	0,63
73	114,30	0,20	23,00	0,004	60,00	1,00	1,01	0,61
78	117,80	0,21	24,20	0,004	60,00	1,00	0,98	2,17
82	119,00	0,21	24,60	0,004	59,90	1,00	0,98	1,56
87	120,10	0,21	24,90	0,004	60,00	1,00	0,99	1,27
92	120,50	0,21	25,00	0,004	60,00	1,00	0,99	1,21
98	120,70	0,21	25,10	0,004	59,90	0,99	0,99	0,03
100	120,60	0,21	25,10	0,004	60,00	1,00	0,99	0,89

Los datos fueron tomados desde un ciclo de trabajo del 13% ya que para valores inferiores el voltaje disminuía por debajo del rango de lectura del medidor. De la Tabla 13 se observa que para un 13% de ciclo de trabajo la potencia activa o real proporcionada por el medidor es de 1,7 W, y los valores efectivos de tensión y corriente en la carga son de 22,1 V y 0,08 A respectivamente, esto nos da una potencia aparente de:

$$S = V_{RMS} \times I_{RMS} = 22,1 * 0,08 = 1,768 VA$$

Calculando el FP se tiene que:

$$\text{Factor de potencia (FP)} = \frac{\text{potencia real (P)}}{\text{potencia aparente (S)}} = \frac{1,7 W}{1,768 VA} = 0,96$$

Calculando el error porcentual se tiene:

$$E_P = \left| \frac{V_V - V_A}{V_V} \right| * 100$$

Donde:

- E_P es el error porcentual.
- V_V es el valor verdadero (Valor medidor).
- V_A es el valor aproximado (Valor calculado).

$$E_P = \left| \frac{1 - 0,96}{1} \right| * 100 \approx 4\%$$

Este valor de FP calculado teóricamente se asemeja al valor proporcionado por el medidor, lo cual permite concluir la exactitud de los cálculos, de igual manera se observa que el valor de la potencia P cuando la tensión de la red es de 120 V_{AC} se aproxima con exactitud a 25 W de consumo proporcionado por el fabricante.

De manera similar se procede a realizar el análisis para el segundo tipo de carga, se puede inferir de la Tabla 14 que los datos fueron tomados del mismo modo a partir de un ciclo de trabajo del 13%, esto debido a que para valores inferiores el voltaje disminuía por debajo del rango de lectura del medidor.

Tabla 14 Parámetros eléctricos y FP medidos para una carga no lineal (Bombillo LED dimerizable)

Parámetros eléctricos medidos en el circuito de atenuación de iluminación								
PWM [% Ciclo Trabajo]	Voltaje [V]	Corriente [A]	Potencia [W]	Energía [kWh]	Frecuencia [Hz]	FP	FP Calc.	% Error
13	47,5	0,02	0,50	0,005	59,9	0,44	0,53	5,26
16	51,6	0,03	1,00	0,005	60,0	0,61	0,65	5,90
19	54,9	0,03	1,20	0,005	60,0	0,64	0,73	12,09
23	60,6	0,04	1,80	0,005	60,0	0,72	0,74	3,14
31	73,9	0,05	3,20	0,005	60,0	0,80	0,87	8,25
42	91,1	0,06	4,80	0,005	59,9	0,85	0,88	3,31
52	104,7	0,07	6,10	0,005	59,9	0,87	0,83	4,33
61	114,9	0,07	7,20	0,005	59,9	0,88	0,90	1,73
67	117,2	0,07	7,60	0,005	60,0	0,90	0,93	2,93
74	119,2	0,08	8,10	0,005	60,0	0,90	0,85	5,62
80	119,9	0,08	8,30	0,005	60,0	0,90	0,87	3,86
85	120,3	0,08	8,40	0,005	59,9	0,91	0,87	4,09
90	119,4	0,08	8,40	0,005	59,9	0,91	0,88	3,36
95	119,4	0,08	8,40	0,005	59,9	0,90	0,88	2,29
100	119,5	0,08	8,40	0,005	60,0	0,90	0,88	2,37

De modo similar, analizando la primera línea de la Tabla 14 el valor de la potencia activa es de 0,5 W, y los valores efectivos de tensión y corriente en la carga son de 47,5 V y 0,02 A respectivamente, esto nos da una potencia aparente de:

$$S = V_{RMS} \times I_{RMS} = 47,5 \times 0,02 = 0,95 \text{ VA}$$

Calculando el FP se tiene que:

$$\text{Factor de potencia (FP)} = \frac{\text{potencia real (P)}}{\text{potencia aparente (S)}} = \frac{0,5 \text{ W}}{0,95 \text{ VA}} = 0,53$$

Por otra parte, cuando el ciclo de trabajo del circuito atenuador está al 100 %, se observa que la carga se estabiliza, la potencia activa P es de 8,4 W y los valores efectivos de tensión y corriente en la carga son de 119,5 V y 0,08 A respectivamente, lo que a su vez proporciona una potencia aparente S y un FP de:

$$S = V_{RMS} \times I_{RMS} = 119,5 \times 0,08 = 9,56 \text{ VA}$$

$$\text{Factor de potencia (FP)} = \frac{\text{potencia real (P)}}{\text{potencia aparente (S)}} = \frac{8,4 \text{ W}}{9,56 \text{ VA}} = 0,88$$

De esta forma se realiza la siguiente Tabla 15 que permite contrastar los resultados obtenidos de las pruebas con los proporcionados por el fabricante para este segundo caso, permitiendo concluir que los cálculos y resultados obtenidos son válidos.

Tabla 15 Comparativa de los parámetros eléctricos medidos y proporcionados por el fabricante

Parámetros eléctricos del fabricante		Parámetros medidos	
Consumo con un voltaje de 120 V _{AC}	9W	Consumo con un voltaje de 120 V _{AC}	8,4 W
Factor de potencia	0,9	Factor de potencia	0,9
Corriente de entrada a 120 V _{AC}	-	Corriente de entrada a 120 V _{AC}	80 mA

Como nota adicional de las pruebas realizadas, se puede concluir que, para una carga no lineal el circuito atenuador de corte de fase disminuye el FP, especialmente en configuraciones bajas de atenuación, no obstante, en configuraciones de atenuación por encima de la mitad del ciclo de trabajo el FP se mantiene dentro de un rango que no supone un deterioro significativo en relación a su valor nominal. En este sentido, la elección de un producto de iluminación LED de mayor calidad tiene más probabilidad de poseer un mejor FP, lo cual es crucial para el desempeño del sistema, por lo que no solo consume menos corriente, sino que también presenta una forma de onda de corriente amigable con la red de suministro.

Otro punto importante que se observa, es la reducción de la potencia activa P demandada por la carga y en consecuencia la disminución de la energía activa medida en kWh, que a fin de cuentas es lo que se acostumbra a ver en la factura mensual.

6.2.9 Desarrollo de PCB

Para el proceso de diseño de las placas de circuito impreso PCB, se usó el software Proteus 8 Professional, el cual posee un gran número de herramientas, y funcionalidades específicas de alto nivel para el diseño de placas de múltiples capas de cobre. Este software

permite un gestor de reglas del diseño personalizado, enrutado manual o automático, generación de superficies de disipación, una gestión simplificada del trazado y modificación manual de pistas, aunado a esto incorpora una amplia librería de encapsulados de componentes de agujero pasante que incluye la mayoría de los integrados más comunes, transistores, diodos, conectores, etc. Cabe destacar que el uso de este software permite una extensión de ficheros de salida para la generación de archivos Gerber/Excellon con formato RS274X, este formato ofrece una representación completa de la placa de circuito impreso PCB con información para producción de la placa final al fabricante. En este contexto se desarrolló el diseño de la placa para el prototipo cuyo esquema electrónico se aprecia en (ver Figura 40) y el diseño final corresponde a una placa PCB de dos caras de cobre (*Top Copper* y *Bottom Copper*) cuyo PCB layout se aprecia detalladamente en el Anexo 4.

Posteriormente gracias a la herramienta de visualización 3D se pudo comprobar el resultado final, antes de enviar a producción las placas, esto permitió la verificación y detección de posibles errores para su corrección, y con ello ahorrar posibles gastos adicionales al no seleccionar el componente con el encapsulado adecuado para el montaje. De este modo se observa el resultado final de las placas mediante su simulación 3D vista en el Anexo 5. Cabe destacar que, para el proceso de fabricación se optó por el fabricante JLCPCB, así pues, se envió el archivo Gerber generado junto a información adicional de las placas (p.ej. material de base, número de placas, color, espesor, entre otros) para completar el pedido. Una vez obtenidas las placas PCB se procedió a realizar el montaje y soldadura de los componentes de forma cuidadosa, para seguidamente realizar una exhaustiva comprobación del trabajo realizado con ayuda de un multímetro digital. Finalmente se visualiza el resultado final del circuito elaborado en el Anexo 6.

6.3 Sistema de adquisición de datos de los sensores

El desarrollo del prototipo de sistema inteligente de iluminación requiere del conocimiento de diferentes variables físicas tales como temperatura, humedad, luz, corriente/voltaje. La adquisición de esta información para su procesamiento dentro del algoritmo automático de control, permite en última instancia activar y regular un actuador o proceso. En este contexto, el bloque de adquisición de datos conformado por un conjunto de sensores inteligentes permite el acondicionamiento de las señales para su tratamiento y envío al dispositivo de procesamiento seleccionado. De tal forma los sensores utilizados para la captación de las variables antes mencionadas, fueron los sensores DHT22, BH1750 y el medidor PZEM-004T respectivamente. Este apartado presenta una explicación detallada de los

pasos y comandos utilizados dentro del entorno de programación ARDUINO IDE para la adquisición correcta de los datos, cabe destacar que para las pruebas realizadas se usó la placa NodeMCU (ver apartado 5.1.1) la cual es compatible con la programación en el entorno de programación antes mencionado.

6.3.1 Adquisición de datos de temperatura y humedad del sensor DHT22

El proceso de adquisición de los datos provenientes del sensor DHT22 se puede analizar en (ver Anexo 7) el cual muestra el diagrama de flujo para la obtención correcta de las variables físicas de temperatura y humedad para su posterior visualización por medio del monitor serie dentro de la plataforma de ARDUINO.

6.3.2 Adquisición de datos de iluminancia del sensor BH1750

El proceso de adquisición de los datos provenientes del sensor BH1750 en referencia al nivel de iluminación se puede analizar en (ver Anexo 8), el cual mediante un diagrama de flujo representa la esquematización gráfica del algoritmo implementado y los pasos a seguir para extraer la información del sensor luxómetro, misma que será de gran utilidad para la elaboración del algoritmo de control inteligente de iluminación.

6.3.3 Adquisición de datos de consumo eléctrico del módulo PZEM-004T

El proceso de adquisición de los datos provenientes del medidor eléctrico PZEM-004T, es similar a la obtención de la información por parte de los sensores inteligentes, ya que este módulo a través de su interfaz TTL permite una comunicación tipo Modbus-RTU para la transmisión de los parámetros eléctricos al dispositivo de procesamiento MCU. De igual forma se presenta el diagrama de flujo (ver Anexo 9) que permite una comprensión general del proceso de obtención de esta información, la cual permitirá analizar el consumo de energía del prototipo de iluminación inteligente en apartados posteriores.

6.3.4 Adquisición de datos del sensor de presencia PIR HC-SR501

La obtención de los datos de este sensor permite analizar la presencia y ausencia de personas dentro del área de trabajo e implementación del prototipo. Esta información es crucial en el desarrollo del algoritmo de control inteligente, ya que permite apagar/encender el suministro de energía a la luminaria y en consecuencia reducir el consumo energético de la misma. La obtención de esta información corresponde a una entrada lógica de dos niveles alto/bajo, la cual es leída por el MCU a través de un pin digital, este valor conmuta el estado de una variable booleana true/false para su posterior análisis.

6.3.5 Diagrama de cableado eléctrico

Con la finalidad de analizar detalladamente el proceso de obtención de los datos provenientes de los sensores inteligentes, así como también la adquisición de los parámetros del medidor, se presenta el diagrama de cableado eléctrico utilizado en el desarrollo de estas pruebas. De esta manera se incluye el código para la adquisición de los datos y cuyos resultados obtenidos de la ejecución del programa se pueden observar detalladamente (ver Anexo 10).

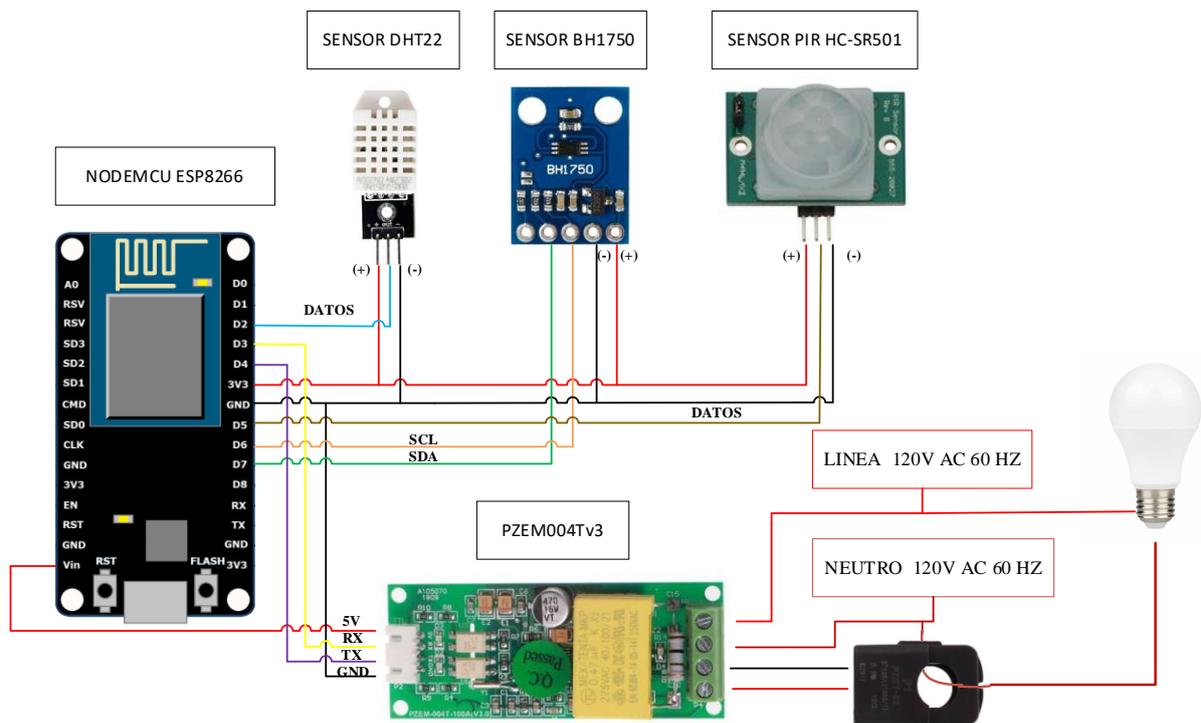


Figura 50 Diagrama de cableado eléctrico para la obtención de los datos de los sensores

6.4 Sistema de control automático para la regulación de la iluminación

Con el fin de diseñar un sistema de iluminación inteligente, se optó por el diseño de un sistema de control en lazo cerrado. Según (Ogata, 2010) la realimentación al controlador de la señal de error de actuación, que es la diferencia entre la señal de referencia y la señal de realimentación, permite reducir rápidamente el error y llevar la salida del sistema a un valor deseado. Esto permite garantizar un buen desempeño del proceso automatizado mediante un control robusto de variables como temperatura, luminosidad, caudal etc.

Para el diseño del sistema de control digital se partió del análisis de la estructura básica del sistema electrónico desarrollado, el cual está compuesto de un conjunto de: fuentes, sensores inteligentes, actuadores, y el sistema electrónico detallado anteriormente. En consecuencia, se ejemplifica el sistema de control digital en el siguiente diagrama de bloques que muestra el diseño general del prototipo.

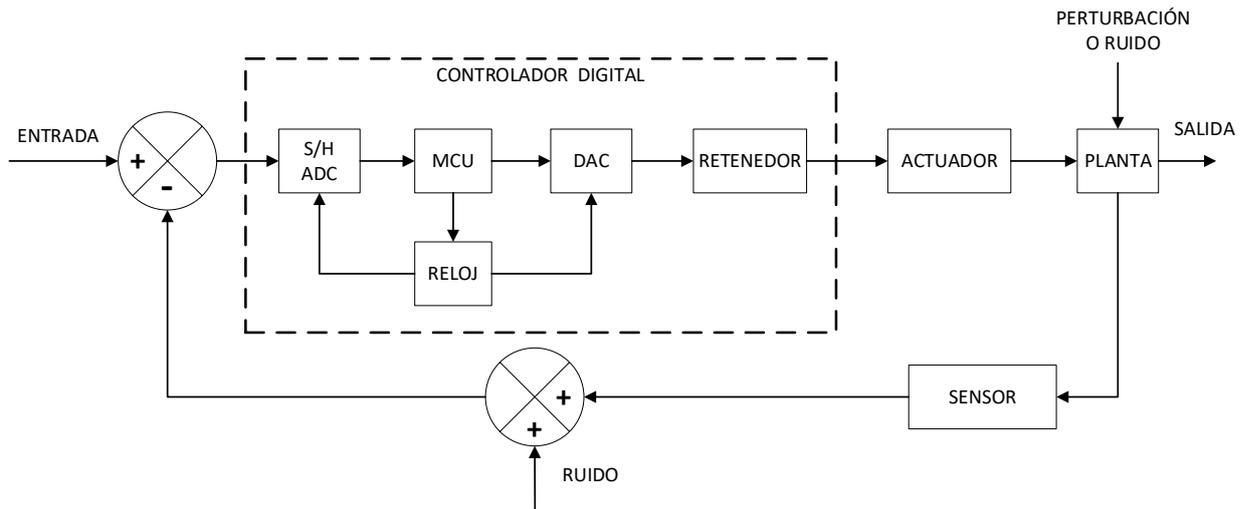


Figura 51 Diagrama de bloques general del sistema de control automático de iluminación

La operación del controlador es proporcionada por la señal de reloj del MCU, en algunas partes del sistema pasan señales digitales como lo es a la salida del convertidor digital del sensor luxómetro, dicha señal establece un valor digital medido en luxes y es comparado con la señal de entrada o valor de referencia también medido en luxes para calcular la señal de error y posteriormente por medio de un algoritmo calcular los parámetros de ajuste del controlador digital dentro del MCU. En otras partes resaltan señales analógicas como a la salida del MCU, el cual proporciona una señal PWM que permite modificar el ciclo de trabajo de una señal periódica y de esta manera controlar la dinámica de la planta y en consecuencia la cantidad de energía suministrada a la carga. También se observa la presencia de señales de ruido en el proceso, las cuales representan interferencias en la captación de luz por el sensor, así como también el ruido de cuantificación producto de la conversión A/D.

Se puede resumir el diagrama de bloques anterior y obtener otro diagrama equivalente para el sistema de control digital, de manera que permita descomponer las variables de entrada y salida de cada bloque del proceso con la finalidad de analizar el comportamiento del sistema.

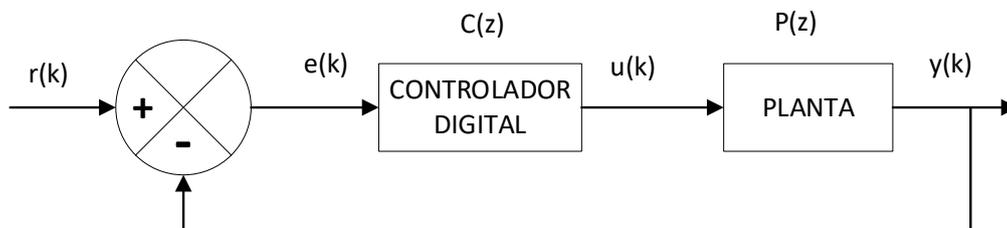


Figura 52 Diagrama de bloques resumido del sistema de control digital

Donde:

- $y(k)$ es la variable de salida (Iluminancia), la cual es leída por el sensor luxómetro.
- $r(k)$ es la señal de referencia (luxes) de la misma naturaleza que $y(k)$.
- $e(k)$ es la señal de error, dada por $e(k) = r(k) - y(k)$.
- $C(z)$ es el controlador PID discreto.
- $u(k)$ es la señal de control a la salida del controlador digital.
- $P(z)$ es la planta o proceso que realiza la regulación de la iluminancia de la zona.

Hay que destacar que las funciones del controlador y la planta están dadas en función de la variable compleja z ya que se hace referencia al dominio discreto.

6.4.1 Punto de ajuste de iluminación

El punto de ajuste o punto de referencia es la señal externa que recibe el controlador, esta señal establece el valor deseado a alcanzar, y es de la misma naturaleza que la señal realimentada que proporciona el sensor. Para este trabajo dicha señal representa la iluminancia de la zona de trabajo y se mide en Lux [lx] la cual representa el flujo luminoso de un lumen que recibe una superficie de 1 m^2 .

Para establecer el punto de ajuste de nuestro controlador se tomó como referencia la Normativa de iluminación de lugares de trabajo en interiores NTE INEN 2969-1 (ver apartado 4.12) la cual establece los valores adecuados para mantener términos de cantidad y calidad de iluminación en condiciones visuales normales.

6.4.2 Identificación del modelo matemático de la planta de control

Es indispensable identificar el modelo matemático de la planta de regulación de luminosidad a controlar para posteriormente calcular los parámetros de ajuste del controlador PID. Este modelo permite al diseñador realizar y validar mediante simulación el ajuste de los parámetros del controlador que permiten obtener una respuesta que satisfaga las especificaciones de diseño (López, 2010).

Para esto se debe obtener la función de transferencia de la planta, es decir una función que relacione su respuesta en base a una señal de entrada. En este contexto, debido a la complejidad de la planta para desarrollar un modelo fenomenológico basado en leyes físico-químicas que permitan obtener una ecuación diferencial de su comportamiento, se optó por otro

método de caracterización basado en un mecanismo experimental que permite obtener una curva de reacción de la planta, y con ello aplicar técnicas de identificación de sistemas por medio de esta curva de respuesta.

Según (López, 2010) el proceso de identificación es un proceso iterativo que se puede resumir en los siguientes pasos:

- Obtención de datos de entrada - salida.
- Tratamiento previo de los datos registrados.
- Elección de la estructura del modelo.
- Obtención de los parámetros del modelo.
- Validación del modelo.

6.4.2.1 Generación de entrada escalón y adquisición de datos.

Para determinar la curva de reacción se utiliza una señal de excitación en forma de escalón generada mediante la programación en Arduino. Esta señal de excitación consiste en aplicar a la planta una señal PWM del 100% que genere a su salida la máxima intensidad luminosa, de modo que el sensor luxómetro analice el cambio de iluminancia con un tiempo de muestreo de 160 mili segundos. De este modo se guardan los datos de salida obtenidos por el sensor, en base a la señal de excitación de entrada para su análisis pertinente.

En la Figura 53 se aprecia la respuesta del sensor graficada con los datos adquiridos en la plataforma Arduino, se puede inferir que no existe ningún retardo en la respuesta del sistema, y la salida del sensor luxómetro es proporcional a la señal de excitación o lo que es lo mismo al valor de ciclo de trabajo de la señal PWM que regula la intensidad luminosa de la lampara LED.

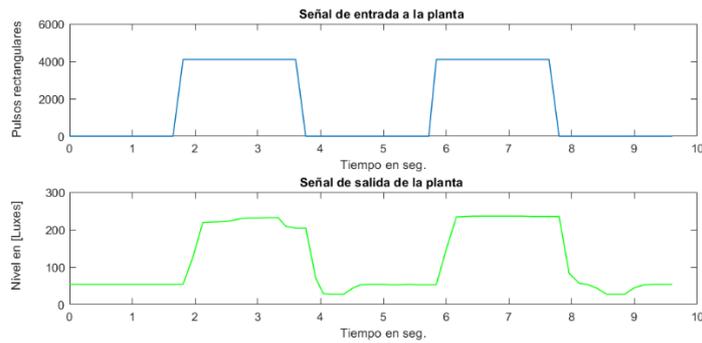


Figura 53 Respuesta de la planta a una excitación de entrada escalón

6.4.2.2 Análisis en plataforma de software matemático.

Una vez obtenidos los datos de entrada y salida de la planta en el dominio del tiempo, se procede a agregar estos valores en una aplicación para la identificación de sistemas. Para esto se importan los datos en entrada y salida de la planta, y se configura el tiempo de inicio igual a 0 segundos y el tiempo de muestreo de 0,16 segundos.

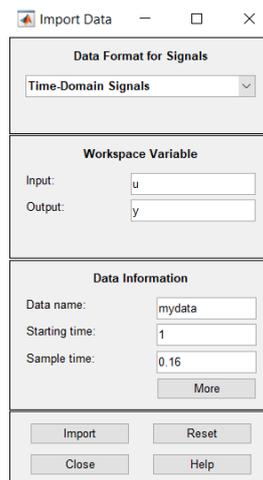


Figura 54 Importación de datos a la aplicación de identificación de sistemas

Con los valores ingresados al sistema se procede a realizar el preprocesamiento de los datos. Para esto se realiza la división del conjunto de datos en dos partes, esto se realiza seleccionando dos rangos de valores usados para el cálculo de la función de transferencia mediante la operación “*select range*”. Con el conjunto dividido en dos partes, un conjunto de ellos pertenecerá a los datos de trabajo “*working data*” y el otro conjunto será usado para la validación de los datos “*validation data*” dentro de la plataforma de trabajo.

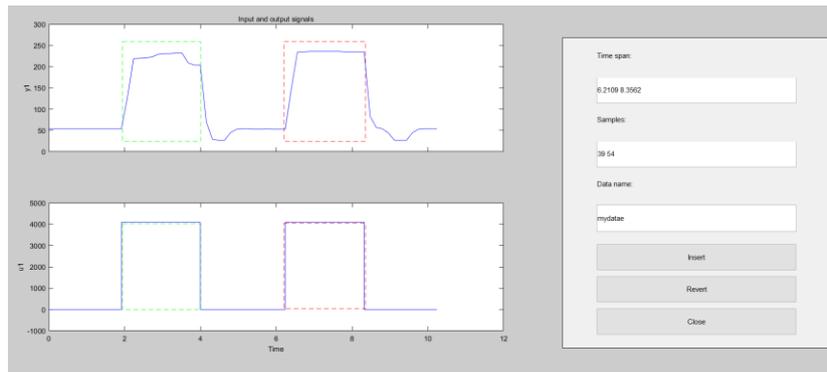


Figura 55 Preprocesamiento de los datos para la identificación del sistema

Con esta operación se procede a realizar la estimación de dos modelos diferentes mediante el menú desplegable “Estimate” y la función “Process Models”.

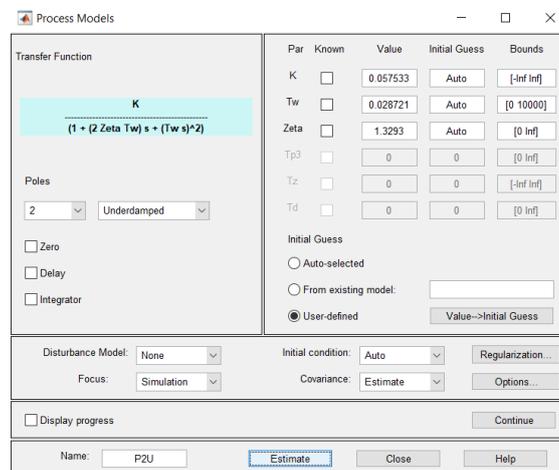


Figura 56 Estimación de modelos para el sistema

Como resultado se distinguen dos modelos obtenidos en el tablero. Estos modelos denotados como P1 para el sistema de primer orden y como P2U para el sistema de segundo orden permiten analizar sus propiedades mediante diferentes representaciones gráficas dentro de la casilla de “Model views”. Así pues, se obtuvo una representación gráfica de los dos modelos simulados, de los cuales el P2U obtuvo un mayor porcentaje de aproximación al comportamiento real de la planta denotado con la línea negra.

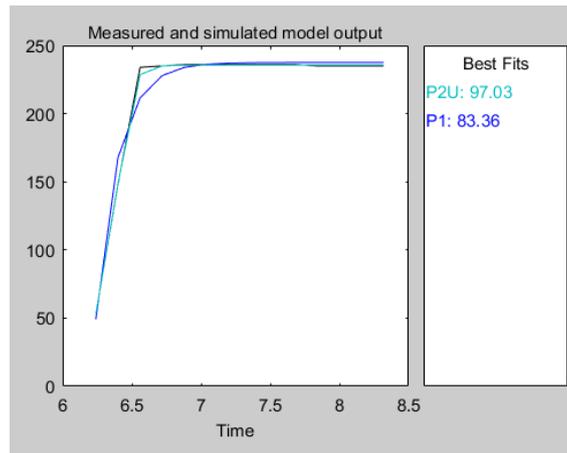


Figura 57 Aproximación de los modelos al comportamiento real del sistema

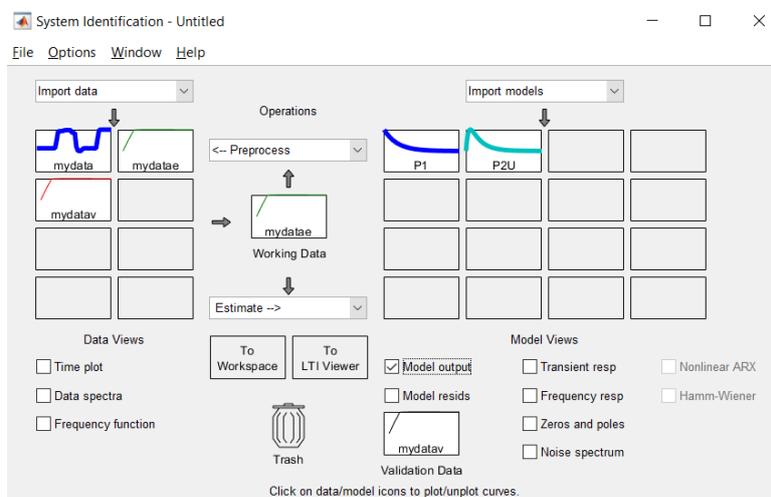


Figura 58 Hoja de trabajo de la aplicación System Identification

Con esto se obtuvo la representación matemática de la función de transferencia de la planta con una exactitud del 97 %, expresada de la siguiente manera:

$$G(s) = \frac{K_p}{1+2*\text{Zeta}*T_w*s+(T_w*s)^2} \tag{16}$$

Donde:

- $K_p = 0,057533 \pm 0,00021607$
- $T_w = 0,028721 \pm 136,42$
- $\text{Zeta} = 1,3293 \pm 4161,8$

Así pues, se observa que el factor de amortiguamiento zeta está por encima de la unidad y la planta posee un comportamiento sobre amortiguado, es decir, el sistema no posee oscilaciones en su respuesta, además el diagrama de estabilidad poseerá dos polos reales.

Para analizar las propiedades del sistema se hace uso de las gráficas proporcionadas dentro del panel “Model view”. En la siguiente Figura 59 se analiza la respuesta ante una entrada escalón a la planta y el diagrama de polos para posteriormente realizar su validación mediante una simulación del mismo, y comparar su respuesta con la respuesta real del sistema.

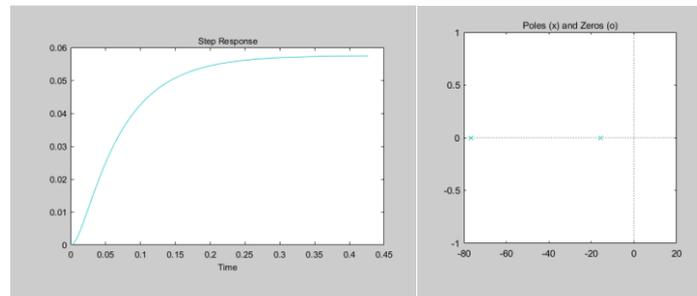


Figura 59 Respuesta a una entrada escalón y diagrama de estabilidad respectivamente

6.4.2.3 Validación del modelo matemático de la planta.

Para validar el modelo anterior, se realiza una simulación del mismo en un algoritmo implementado, para lo cual se establece la función de transferencia obtenida en términos adecuados, esto es:

$$G(s) = \frac{K_p}{1 + 2\zeta W_n s + (W_n s)^2} = \frac{0,057533}{1 + 2(1,3293 \times 0,028721)s + (0,028721s)^2}$$

$$= \frac{0,057533}{0,000824895 s^2 + 0,07635765 s + 1}$$

Con la función de transferencia obtenida, se ingresa las siguientes líneas de código y a continuación se observa los resultados generados.

```
G2= tf([0.0575],[0.0008248895,0.0763576506, 1])
K2=dcgain(G2)
rlocus (G2)
polos=pole(G2)
pzmap(G2)
title('Lugar geométrico de las raíces del sistema P2U')
step(G2)
title('Respuesta al escalón del sistema P2U')
xlabel('Tiempo en seg.')
ylabel('Amplitud')
impulse(G2)
title('Respuesta al Impulso del sistema P2U')
xlabel('Tiempo en seg.')
ylabel('Amplitud')
```

Figura 60 Código implementado para analizar la respuesta del sistema P2U

Resultados:

```

G2 =
      0.0575
-----
0.0008249 s^2 + 0.07636 s + 1
Continuous-time transfer function.
K2 =
      0.0575
polos =
      -76.7776
      -15.7896

```

Figura 61 Obtención del lugar geométrico de las raíces

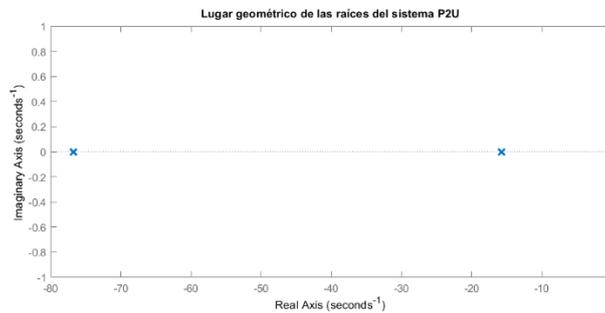


Figura 62 Lugar geométrico de las raíces en el plano

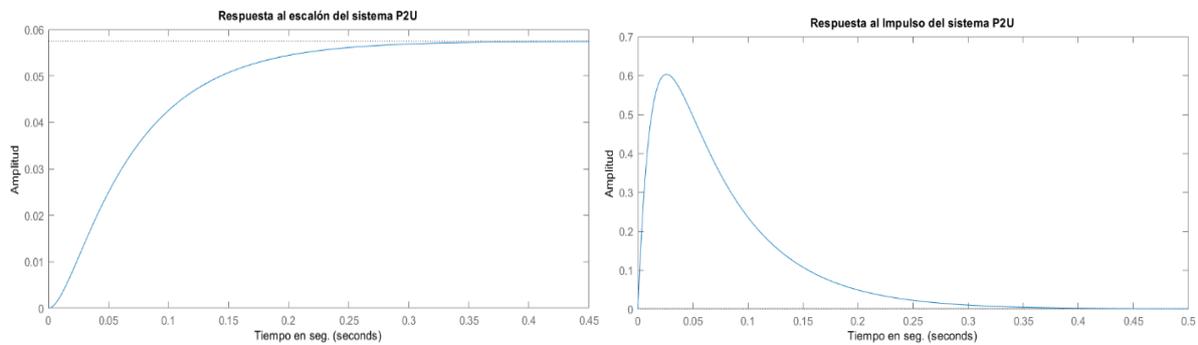


Figura 63 Respuesta del sistema a una entrada escalón e impulso unitario respectivamente

De los resultados obtenidos se concluye que el modelo matemático para la función de transferencia de la planta es correcto, ya que aproxima con gran precisión al comportamiento real del sistema, además de verificar la estabilidad del mismo.

6.4.3 Cálculo de los parámetros de ajuste del controlador PID

Con el objetivo de desarrollar un controlador que se adapte de forma adecuada a cualquier ambiente y condiciones de trabajo, se optó por el desarrollo de un controlador PID (ver apartado 4.13.3). Según (Abu-Khalaf, Chen, & Turevskiy, 2009)

El ajuste de un controlador PID consiste en encontrar de forma segura y sistemática el conjunto de ganancias que garantiza el mejor rendimiento del sistema de control para una tarea específica. Existen diversas maneras de encontrar tales ganancias, como el ajuste mediante métodos manuales, o basados en reglas. Los métodos manuales consumen tiempo y pueden causar daños en el hardware del sistema, y por el otro lado los métodos basados en reglas poseen la limitación de que no todos son compatibles con los modelos de plantas, por ejemplo, plantas inestables o de alto orden.

En este trabajo se optó por calcular los parámetros de ajuste del controlador a partir de una simulación del comportamiento del sistema mediante la ayuda de software matemático y herramientas para la sintonización de controladores PID. Esta aplicación permite una sintonización automática del PID, además proporciona un conjunto de herramientas basado en una interfaz gráfica que permite examinar el rendimiento del controlador en el dominio del tiempo y frecuencia. El proceso de sintonización permite extraer las ganancias del controlador PID para una planta dada, además permite reducir el sobre impulso o mejorar el rechazo a una perturbación en la entrada de la planta.

Par el cálculo de los parámetros de ajuste, se estableció algunas condiciones de diseño como un tiempo de establecimiento de 3,5 segundos, de manera que la lampara LED no genere un cambio abrupto de intensidad luminosa, es decir, posea una rampa de atenuación suave de encendido y apagado generando así un impacto visual agradable al usuario. Otra de las características requeridas del controlador es que no posea ningún tipo de sobre impulso ya que esto generaría un mal funcionamiento de la lampara pudiendo incidir al parpadeo de la misma. En la siguiente Figura 64 se muestra el diagrama de bloques del controlador digital PID, en donde el objetivo es determinar las ganancias K_p , K_i , K_d del controlador, que permita la regulación adecuada (intensidad luminosa) de la lampara LED a la salida de la planta, en base a un nivel de referencia establecido en luxes a la entrada del sistema.

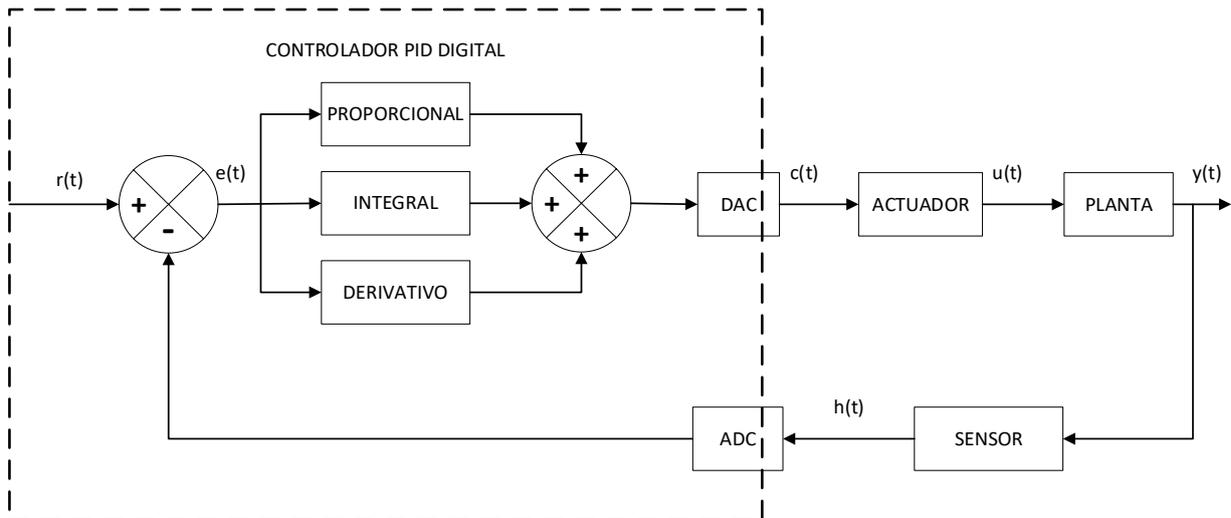


Figura 64 Diagrama de bloques del controlador PID

6.4.3.1 Sintonización del controlador PID.

Para el proceso de sintonización del controlador PID es importante tener en cuenta los efectos generales que ocasiona cada parámetro del controlador en un sistema de lazo cerrado. Esto permite escoger una combinación de ganancias que cumpla con los parámetros de diseño y a la vez maximice la eficiencia del controlador. De este modo los efectos de cambiar estos valores de ganancia se pueden resumir en la siguiente Tabla 16.

Tabla 16 Efectos generales de los valores de ganancias K_p , K_i , K_d , en un sistema lazo cerrado

Parámetro	Tiempo-Subida (<i>Rise-Time</i>)	Sobre-Impulso (<i>Overshoot</i>)	Tiempo- Estabilización (<i>Settling Time</i>)	Error en estado estacionario
K_p	Decrece	Aumenta	Cambio pequeño	Decrece
K_i	Decrece	Aumenta	Aumenta	Decrece
K_d	Cambio pequeño	Decrece	Decrece	No cambia

Para el proceso de sintonización del controlador PID se ejecuta las siguientes líneas de código y se analiza los resultados generados.

```
G2= tf([0.0575],[0.0008248895,0.0763576506, 1])
K2=dcgain(G2)
gc2= pidtune(G2,'PID')
pidTuner(G2,gc2)
```

Figura 65 Código implementado para la obtención de los parámetros del controlador PID

Resultados:

gc2 =

$$K_p + K_i * \frac{1}{s} + K_d * s$$

with $K_p = 33.6$, $K_i = 712$, $K_d = 0.347$

Continuous-time PID controller in parallel form.

Figura 66 Obtención de los parámetros del controlador PID de forma predeterminada

Hay que destacar que el algoritmo de ajuste automático elige las ganancias del PID para equilibrar el rendimiento (tiempo de respuesta, ancho de banda) y la robustez (márgenes de estabilidad) de forma predeterminada para un margen de fase de 60 grados. Obteniendo de esta manera la Figura 67 de respuesta a una entrada escalón, y la tabla de rendimiento del controlador en lazo cerrado.

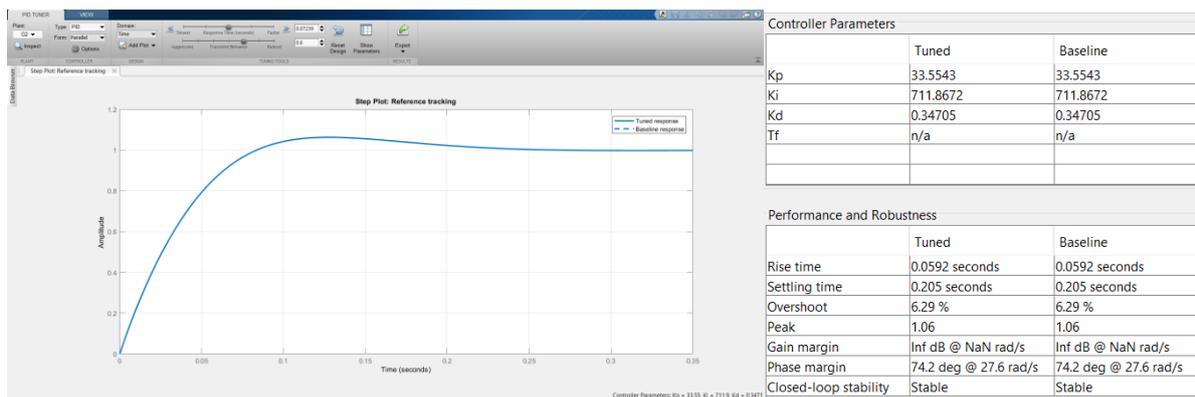


Figura 67 Respuesta al escalón del sistema de control PID de base y parámetros de rendimiento

De esta manera se procede a sintonizar los parámetros del controlador, adaptando la respuesta al escalón en base a las condiciones de diseño requeridas (Settling Time=3,5 s, Overshoot=0%), obteniendo la siguiente Figura 68 de resultado.

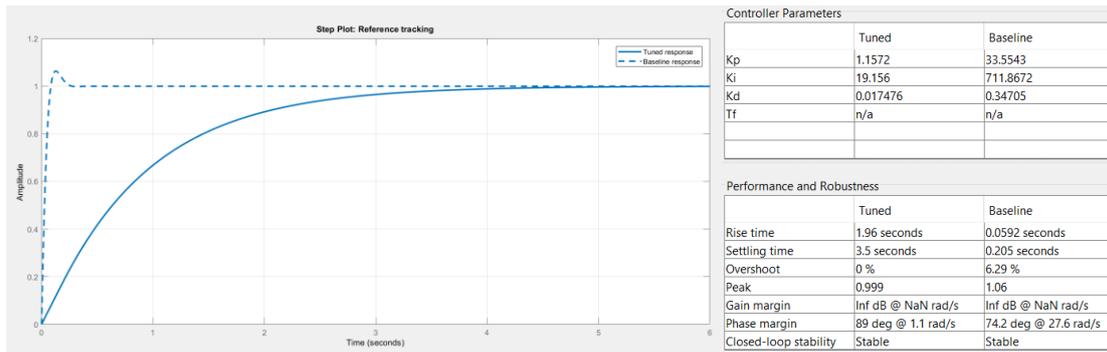


Figura 68 Respuesta al escalón del sistema de control PID sintonizado y parámetros de rendimiento

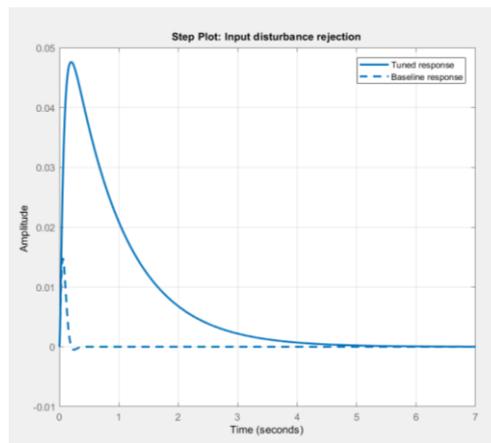


Figura 69 Rechazo a una perturbación de entrada

Mediante la Figura 69 se analiza que el tiempo de establecimiento de rechazo a las perturbaciones de entrada se ha incrementado considerablemente, lo que indica que el sistema rechazará perturbaciones de cortos periodos de tiempo sin ocasionar grandes cambios en su señal de control, lo cual es favorable para nuestro sistema ya que permitirá rechazar, por ejemplo, sombras incidentes en el sensor por cortos periodos de tiempo. Así pues, se determinó los parámetros de ajuste del controlador dados por: $K_p=1$, $K_i=20$, $K_d=0,01$, mismos que serán integrados en el algoritmo de control PID del sistema de iluminación inteligente.

6.4.3.2 Simulación del sistema de control de iluminación PID.

Para ejemplificar el comportamiento general del circuito se realizó la simulación del sistema completo con el uso de software matemático. El diagrama de bloques obtenido se presenta en la Figura 70, donde se hace la comparativa del funcionamiento del sistema para los dos conjuntos de parámetros PID analizados anteriormente. De esta manera se establece una entrada escalón de 0 a 500 luxes que actúa como señal de referencia o *Setpoint* dentro del bloque del controlador PID, adicionalmente se agrega un bloque de ruido a la salida de la planta, que simula las variaciones en las lecturas del sensor luxómetro para el cálculo de la señal de error.

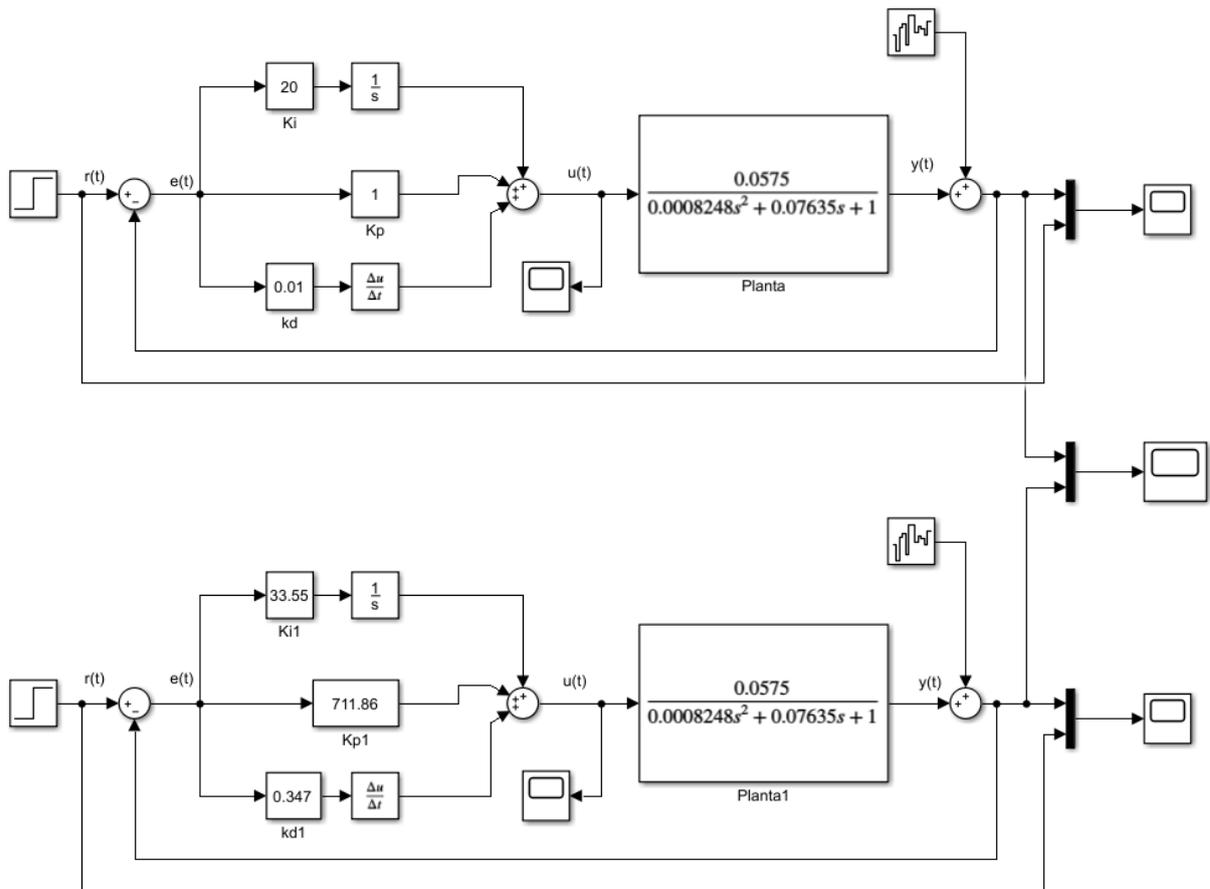


Figura 70 Simulación de los bloques del sistema de control PID

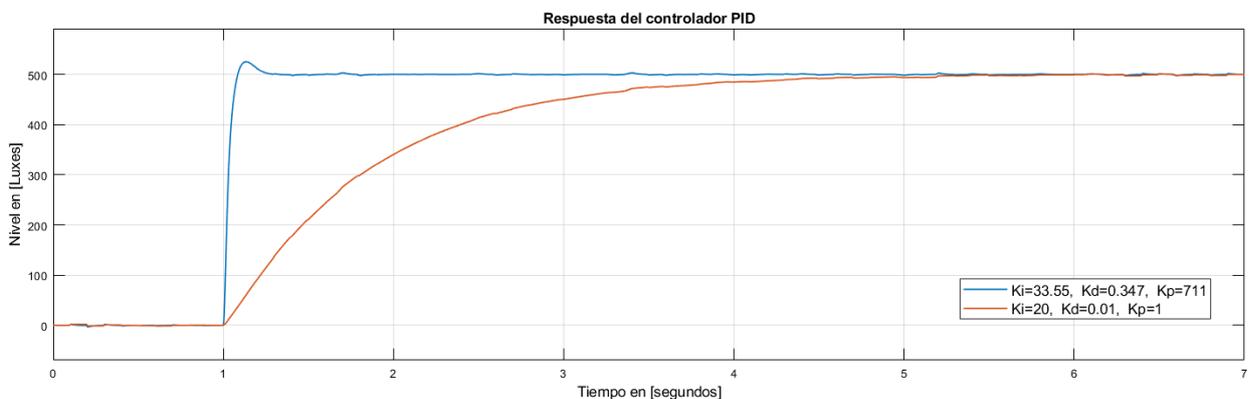


Figura 71 Comparativa del sistema de control PID

6.4.4 Diagrama de flujo del algoritmo de control

Este diagrama fue diseñado desde el punto de vista del sistema embebido, en donde el diagrama (ver Anexo 11) muestra la actualización de los parámetros conforme los bloques condicionales verifican el estado de verdadero o falso en base a la lectura de los sensores. Como resultado el primer bloque condicional verifica que el nivel de luxes de la zona de trabajo este por debajo de un valor máximo de luxes, caso contrario el sistema de control de iluminación

permanecerá apagado. El segundo bloque condicional verifica el estado del sensor PIR, de manera que permite realizar dos funciones diferentes, tales como: un encendido suave de la lampara LED para su posterior regulación automática mediante el algoritmo de control PID y un apagado suave al detectar el estado apagado del sensor PIR.

6.4.5 Programación del sistema de control PID en el entorno ARDUINO

Para la programación del algoritmo de control PID se optó por usar la librería <PID_v1.h> de (Beauregard, 2011), la cual posee numerosas ventajas además de solucionar los problemas que abordan los controladores PID discretos. Por lo consiguiente se enumera algunas de las ventajas que conlleva el uso de esta librería, tales como:

- Proporciona un control PID de nivel industrial, para su aplicación en cualquier proceso de automatización y control digital.
- Permite evaluar el algoritmo de control en intervalos regulares, mediante el comando *SetSampleTime ()*.
- Permite ahorra recursos de procesamiento para el cálculo de otras funciones.
- Elimina el problema de la derivada del error cuando se producen perturbaciones indeseadas en la señal de realimentación. Lo cual provocaba señales de control erróneas de gran amplitud.
- Permite ajustar los parámetros de sintonización del controlador PID mientras se ejecuta el algoritmo, sin afectar el rendimiento del sistema.
- Permite eliminar el efecto *Windup* que surge cuando la señal de error crece por un largo tiempo (acción integral), provocando retrasos en la señal de control. Esto mediante el comando *SetOutputLimits ()*.
- Permite ajustar la dirección de la señal de control, ya sea de forma directa (ascendentemente) o de forma inversa (descendentemente), mediante la función *SetControllerDirection ()*.

En este sentido, se desarrolló el algoritmo de control PID para el sistema inteligente de iluminación basado en el diagrama de flujo que se muestra en (ver Anexo 11).

6.4.6 Pruebas y resultados del sistema de control de iluminación PID

El siguiente apartado tiene la finalidad de analizar el comportamiento del sistema de control de iluminación PID en varios escenarios diferentes. Para esto se ajusta los parámetros

de control con los obtenidos del estudio relacionado a la sintonización del controlador PID (ver apartado 6.4.3.1) y el algoritmo de control elaborado. Cabe considerar por otra parte que, para cumplir con este propósito se tomaron en cuenta algunas consideraciones como: Las pruebas fueron realizadas con poca presencia de luz de modo que el circuito tenga total control sobre la luminosidad del área de estudio, además para fines de visualización la señal de control de la planta $u(t)$ fue acondicionada para mantener un rango entre 0 a 1000, al igual que la señal de lectura del sensor PIR con un valor de 0 o 100 (apagado/encendido). De esta manera se realiza el análisis del circuito en varios escenarios diferentes, tales como:

a) Iluminación ambiental constante, *Setpoint* constante, PIR activado.

De la Figura 72 desprende que el circuito de control de iluminación real posee una respuesta muy similar al comportamiento del circuito simulado y analizado (ver apartado 6.4.3.2). De la gráfica se puede inferir que el tiempo de establecimiento es de aproximadamente 5 segundos para pasar de un valor de (15 a 500) luxes aproximadamente, también se concluye que no existe ningún tipo de sobre impulso en la respuesta y la señal de iluminancia llega al *setpoint* sin ningún tipo de error en estado estacionario. Cabe destacar que se observa un retardo en la señal de iluminancia (lectura del sensor luxómetro) en referencia a la señal $u(t)$ del controlador PID, esto es debido a que la lampara LED no emite luz significativa por debajo de un valor de voltaje aplicado, o lo mismo que por debajo de un valor de la señal de control $u(t)$, en consecuencia, esta señal continúa creciendo apresuradamente hasta que la señal de iluminancia alcanza el punto de referencia.

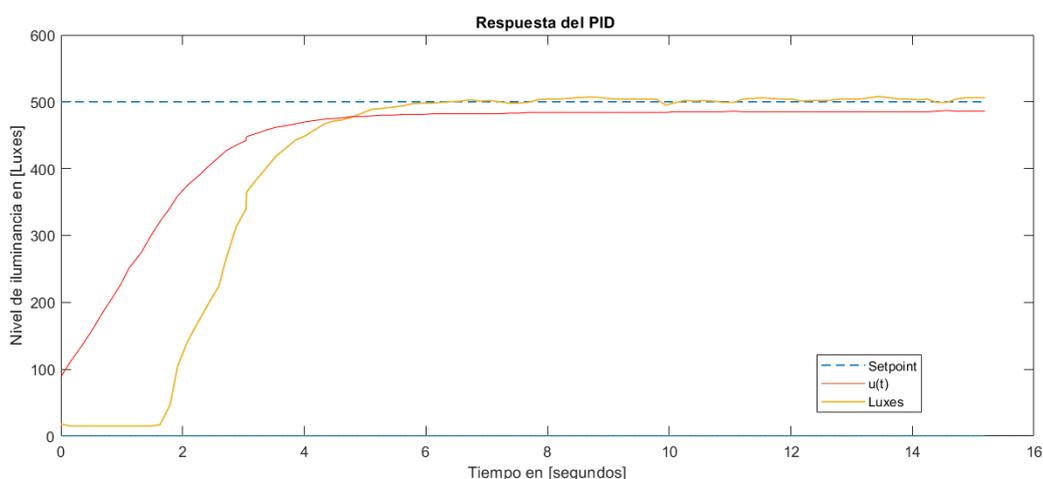


Figura 72 Respuesta del controlador PID (*Setpoint* fijo y sensor PIR activado)

b) Iluminación ambiental constante, *Setpoint* variable, PIR activado.

La Figura 73 muestra el comportamiento del controlador PID cuando la señal de referencia cambia en el tiempo, como resultado se puede señalar que el sistema posee un buen desempeño al cambio de *setpoint* sin afectar el rendimiento general del sistema. Este comportamiento es adecuado ya que permite establecer el valor de iluminación referencial por parte del usuario mientras el sistema se encuentra en funcionamiento, lo cual permite un control remoto del sistema de iluminación por medio de una interfaz de usuario como se verá más adelante.

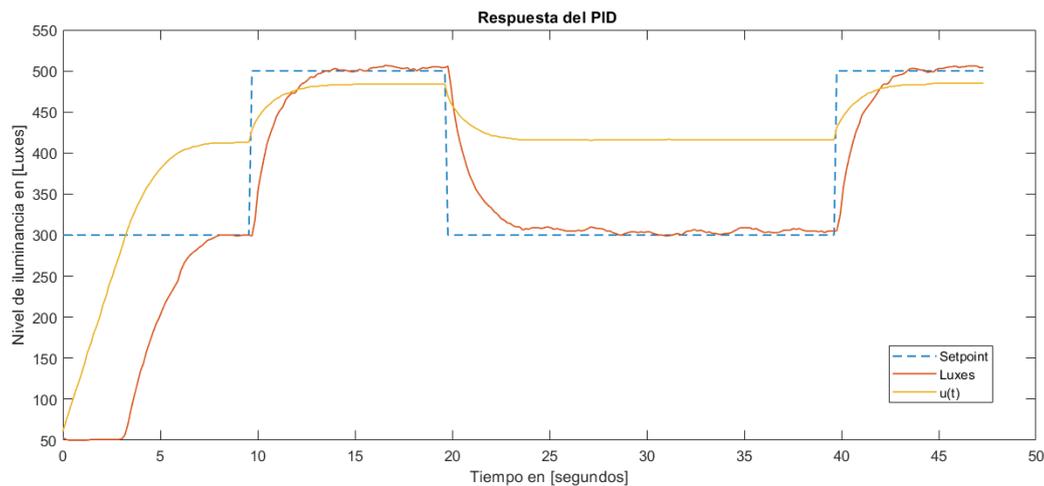


Figura 73 Respuesta del controlador PID (*Setpoint* variable y sensor PIR activado)

- c) Iluminación ambiental constante, *Setpoint* constante, con cambios de estado del sensor PIR.

La Figura 74 muestra el comportamiento del controlador PID cuando el sensor PIR se encuentra en funcionamiento. Para el desarrollo de esta prueba se configuró la salida del sensor PIR HC-SR501 de modo que la salida de alarma en modo disparo repetitivo posea un tiempo de activación de aproximadamente 8 segundos al detectar la presencia de personas. De la gráfica se concluye que la respuesta del controlador PID a la ausencia de personas genera una atenuación suave de la lámpara LED hasta que se genera un nuevo evento de activación del sensor PIR. Esta función de rampa de atenuación suave en la respuesta de control $u(t)$ permite que la lámpara LED se apague en un intervalo de tiempo controlado de aproximadamente 3 segundos lo cual refleja una percepción visual agradable. Además, tan pronto se detecta la presencia de personas, el controlador PID se activa nuevamente llevando la señal de iluminación al valor deseado o *setpoint*.

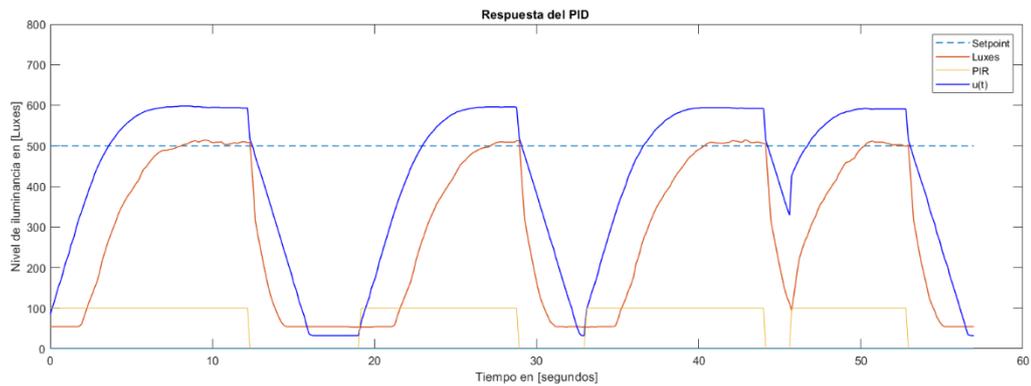


Figura 74 Respuesta del controlador PID (Setpoint fijo y sensor PIR activado/desactivado)

d) Iluminación ambiental variable, Setpoint constante, PIR activado.

La Figura 75 muestra el comportamiento del controlador PID cuando se generan cambios abruptos en la lectura del sensor luxómetro, para esto se realizó la obstrucción de la luz que es captada por el sensor durante cortos intervalos de tiempo y se analiza la respuesta $u(t)$ del controlador. De la gráfica se puede inferir que el controlador PID no genera grandes cambios en la señal de control $u(t)$ de la planta, a menos que la duración de la obstrucción de la luz sea prolongada, esto incide a un buen funcionamiento de la luminaria y evita los parpadeos indeseados. Se comprueba que el controlador PID posee una buena respuesta a las interferencias que pueden ser causadas por sombras en el sensor luxómetro, así como haces de luz incidentes en el mismo.

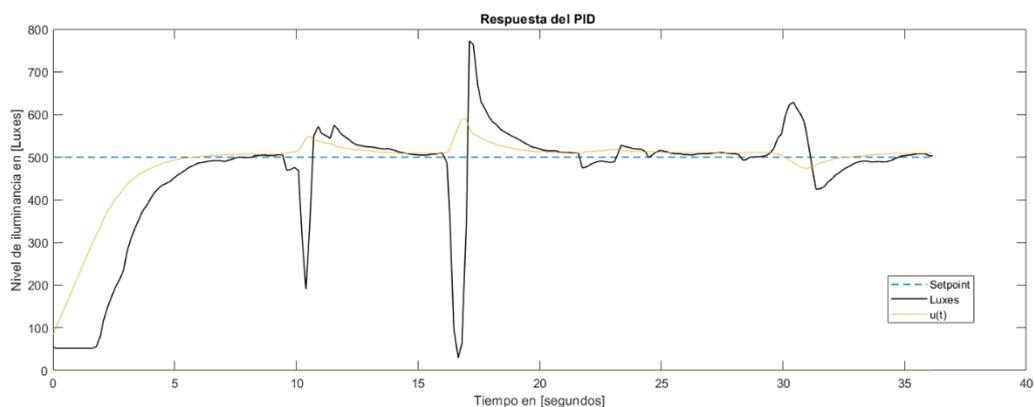


Figura 75 Respuesta del controlador PID (Setpoint fijo, sensor PIR activado, variaciones de luminosidad)

Nota. En el intervalo de 7 a 20 segundos se obstruye la luz captada por el sensor por cortos periodos de tiempo, y en el segundo 30 se inyecta un haz de luz externo al sensor luxómetro.

6.5 Sistema de comunicación inalámbrico

El presente apartado describe el sistema de comunicación inalámbrico implementado para la comunicación bidireccional entre el dispositivo NodeMCU y el ordenador central

(Raspberry Pi 3). El MCU luego de que obtiene las señales de los sensores inteligentes, procesa y acondiciona las tramas para su envío por medio de la tecnología IEEE 802.11 Wi-Fi y el uso del estándar para aplicaciones IoT denominado MQTT, ampliamente utilizado en redes inalámbricas de sensores y actuadores (WSAN) (ver apartado 4.19). De este modo el ordenador central recibe toda la información proveniente del prototipo para su almacenamiento y elaboración de la interfaz de usuario web mediante la herramienta Node-RED, que permite al usuario el control remoto de la luminaria mediante esta interfaz alojada en el servidor local (Raspberry Pi 3). Posteriormente los parámetros de funcionamiento definidos por el usuario son transmitidos mediante el envío de una señal de control a la placa NodeMCU para el ajuste del nivel de referencia dentro del sistema de control y regulación de la luminaria. Cabe destacar que el ordenador central requiere de una conexión a Internet para el funcionamiento general del prototipo, la cual puede ser de forma inalámbrica o cableada. La puerta de enlace (Enrutador) o GW (*Gateway*, en inglés) cumple la función de permitir el acceso de los usuarios a la información contenida en el servidor local, así como también permitir la comunicación MQTT de los clientes (publicadores y suscriptores) al servidor local con el bróker privado Mosquitto alojado en la Raspberry Pi. En este sentido, la puerta de enlace permitirá interconectar múltiples redes WSAN para la recopilación de la información del entorno y su posterior transferencia a las aplicaciones, además, su comportamiento es el de una puerta de enlace transparente, es decir, para cada cliente MQTT conectado, un GW transparente configura y mantiene una conexión MQTT con el intermediario (bróker), por lo tanto, habrá tantas conexiones MQTT entre el GW y el intermediario como clientes MQTT conectados al GW.

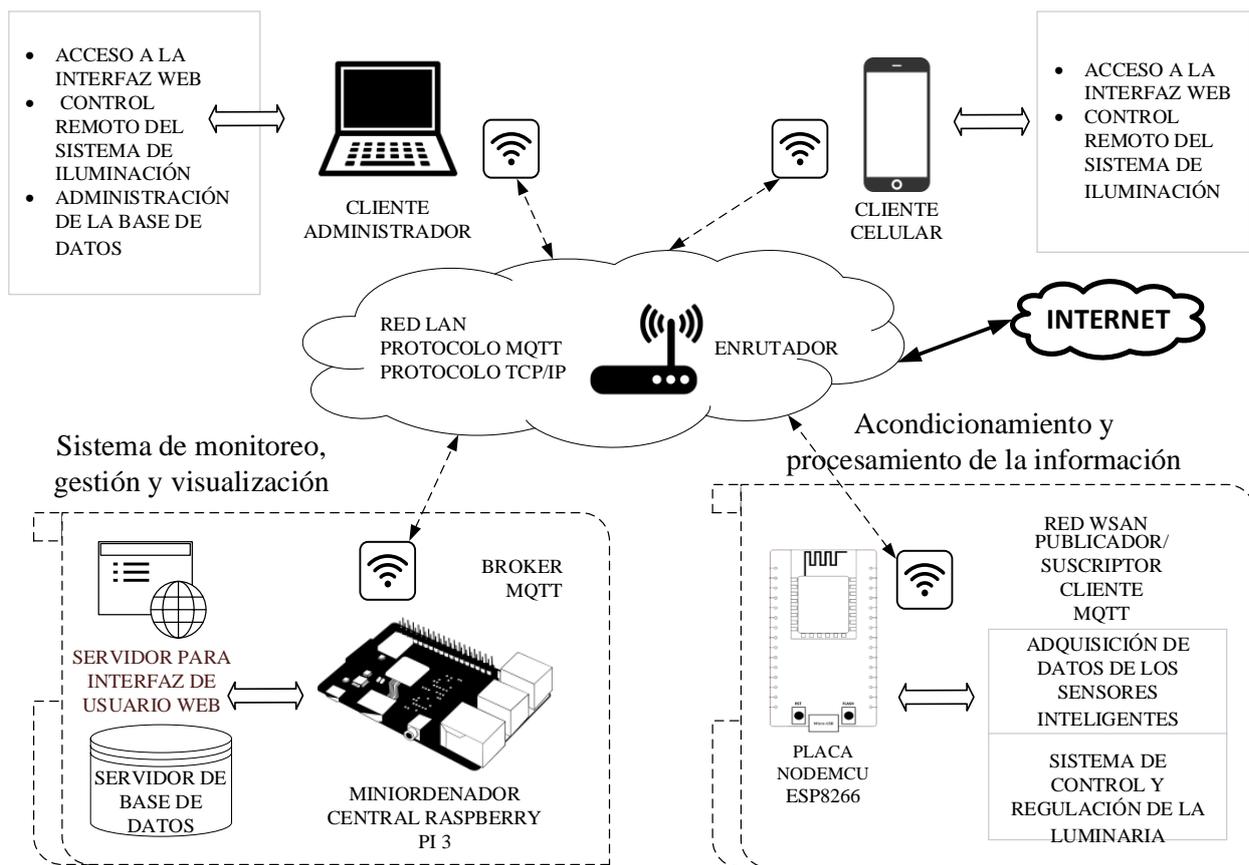


Figura 76 Topología del sistema de comunicación inalámbrico

6.5.1 Descripción del sistema de comunicación inalámbrico

La comunicación inalámbrica del sistema hace uso de la tecnología Wi-Fi incorporada tanto en la placa NodeMCU ESP8266 como en el ordenador central Raspberry Pi 3, y la cual es compatible con la pila de protocolos TCP/IP para su comunicación. Por lo tanto, el protocolo que permite la comunicación entre ambos dispositivos es el MQTT, este es un protocolo pensado para IoT que está al mismo nivel que HTTP o CoAP (Capa de aplicación del modelo TCP/IP). Para entender el funcionamiento del sistema es necesario establecer los componentes básicos que conforman la topología del protocolo MQTT, de este modo se describe los roles que desempeña cada componente en el prototipo de sistema de iluminación inteligente, esto es:

- El bróker: es el servidor alojado en la Raspberry Pi que distribuye la información a los clientes interesados conectados al servidor. Al recibir un mensaje, éste debe buscar y encontrar todos los dispositivos que poseen una suscripción a ese tema y enviarles dicha información.
- El cliente: es el dispositivo NodeMCU que se conecta al bróker y publica o recibe información de forma constante.

- Tópico: es el nombre del tema al cual se suscribe un cliente para enviar o recibir información por medio del bróker (p.ej. /esp8266/luxes).
- Publicador: el cliente NodeMCU envía información al bróker para distribuirla a los clientes interesados en base al nombre del tema. La información corresponde a los datos provenientes de los sensores inteligentes, con datos de: temperatura, humedad, luminosidad, voltaje, corriente, potencia activa, frecuencia, consumo de energía y factor de potencia.
- Suscriptor: el cliente NodeMCU recibe información del bróker correspondiente a información de control para el ajuste del funcionamiento del prototipo. Esta información es distribuida en varios temas que permite su funcionamiento, por ejemplo, el mensaje enviado por el bróker y recibido por el cliente suscriptor, con el tópic (/esp8266/output5) establece un nivel de iluminancia dentro del área de 300 luxes.
- QoS: la calidad del servicio establecida para las conexiones entre el bróker y los clientes se definió en un valor de 2, de modo que exista una entrega asegurada del mensaje en los suscriptores y solo se reciba una copia del mensaje.
- Enrutador: la función principal de la puerta de enlace es proporcionar a los dispositivos clientes acceso al intermediario o bróker.

Cabe destacar que para el funcionamiento del sistema el enrutador utiliza diferentes tipos de enrutamiento, por ejemplo, en el caso del ordenador central (Raspberry Pi 3) se utiliza enrutamiento estático, ya que se debe reconocer inequívocamente al servidor local encargado del almacenamiento de la base de datos, el servidor de la herramienta Node-RED, y también el servidor del bróker MQTT para la comunicación entre los distintos clientes suscritos a los temas de interés. En este sentido, resulta necesario establecer los parámetros de funcionamiento de la red local, como también las medidas de seguridad establecidas para su funcionamiento adecuado, esto es:

- Servidor bróker: utiliza el número de puerto correspondiente a 1883 para una comunicación sin encriptación de datos. La seguridad implementada en el bróker MQTT corresponde al uso de credenciales de usuarios y contraseñas para permitir la autenticación de los clientes y su conexión con el intermediario.
- Servidor Node-RED: utiliza la dirección IP del ordenador central seguida del número de puerto correspondiente: (IP:1880). La seguridad implementada en la herramienta Node-RED es el uso de un sistema de identificación de usuario y contraseña que permite el acceso a la interfaz gráfica de flujo de trabajo.

- Interfaz de usuario: utiliza la dirección del servidor Node-RED seguida del acrónimo “ui” (*User Interface*, en inglés): (IP:1880/ui). A este nivel no se establece ningún sistema de seguridad ya que todo usuario dentro de la red local puede acceder a la interfaz de usuario web para el control del sistema de iluminación.

La siguiente Figura 77 presenta el esquema del sistema de comunicación inalámbrico analizado desde la perspectiva del uso del protocolo MQTT, aquí la placa NodeMCU como los dispositivos clientes por medio de la interfaz de usuario web, tienen los roles de publicar y obtener información dentro del sistema, y el ordenador central (bróker) permite que todos los componentes se conecten a él y se comunican entre sí mediante el servicio publicador/suscriptor en un determinado tema de interés.

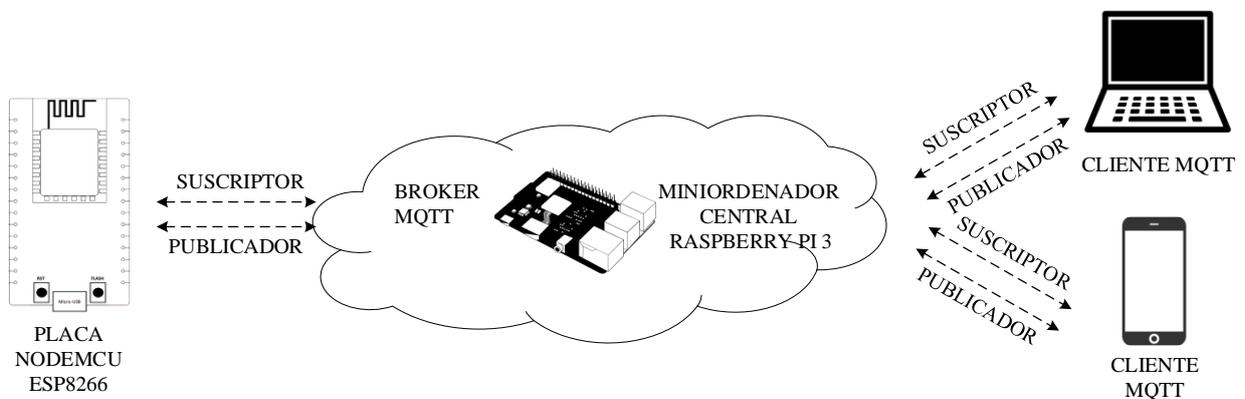


Figura 77 Esquema general del funcionamiento del protocolo MQTT

6.5.2 Instalación y configuración del servidor local MQTT

Para el alojamiento del servidor local MQTT se hace uso de la plataforma embebida Raspberry Pi 3 con el sistema operativo Raspberry Pi OS (Raspbian) instalado en una tarjeta microSD. Así pues, para la instalación del servidor se estableció el uso del bróker Eclipse Mosquitto el cual es un intermediario de mensajes de código abierto (con licencia EPL/EDL) que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT, además proporciona una biblioteca C para configurar clientes MQTT mediante líneas de comandos simples como (`mosquitto_pub` y `mosquitto_sub`) (Light, 2017).

En la instalación del bróker Mosquitto se distinguen los siguientes pasos:

- Verificación de la versión de Raspbian: para ello se ejecuta el comando (`lsb_release -a`) y se verifica la versión instalada. Para el presente trabajo se utiliza la versión *buster*.
- Preparación para instalación: se descarga el paquete Mosquitto del repositorio de Raspbian mediante la importación de la llave de firma del paquete con el comando (`wget`

http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key). Posteriormente se introduce el comando (*sudo apt-key add mosquitto-repo.gpg.key*) y se almacena el paquete en un directorio con el comando (*cd /etc/apt/sources.list.d/*).

- Instalación: para la versión de Raspbian obtenida (*buster*), se instala el paquete en el directorio establecido anteriormente con el comando (*sudo wget http://repo.mosquitto.org/debian/mosquitto-buster.list*).
- Actualización: en el siguiente paso se actualiza la base de datos con el comando (*sudo apt-get update*).
- Instalar clientes MQTT: se procede a instalar Mosquitto clientes en la Raspberry con el comando (*sudo apt install mosquitto mosquitto-clients*).

La configuración del bróker MQTT permite establecer los mecanismos de seguridad dentro del protocolo MQTT, así como parámetros de funcionamiento del servidor, de esta forma para su configuración se estableció el uso de usuarios y contraseñas como mecanismo de autenticación de un cliente (NodeMCU) antes de que se permita su conexión con el bróker MQTT (Raspberry Pi 3). En este contexto, se detallan los siguientes pasos realizados para la configuración deseada:

- Configuración de inicio MQTT: el uso del comando (*sudo systemctl enable mosquitto*) permite que el bróker Mosquitto se ejecute automáticamente al reiniciar la Raspberry.
- Segurizar MQTT: este proceso inicia con la creación del fichero de credenciales que almacena los usuarios y contraseñas útiles para la autenticación de los clientes, para esto se ejecuta el comando (*cd /etc/mosquitto/conf.d/*) seguido del comando (*sudo nano password.txt*) que permite editar el archivo *password.txt* para el registro de las credenciales (usuario:contraseña). Posteriormente se genera un fichero con formato que entienda mqtt con el comando (*sudo mosquitto_passwd -U password.txt*).



```
pi@raspberrypi:~$ ^C
pi@raspberrypi:~$ cat password.txt
cliente1:$7$101$J1x+4Wg1cAboJoQV$znpr2QkgkceAxtl4YKpZ3RCde7/yyDlHY9EAmFI3xYAQaYrgeoCno1dNEJ8T+GDh1
Fjge+0r f0WbqcDSaiQw==
cliente2:$7$101$xvmZMpl3p9f60ddw$59vcyGdx1wk43tSPS4F1e+ZY9IK1kwly6gvI8LQMkLwvWP/13SkUvnmJrcEHJdEYenBP
1KzRdJir0280Z06ljwA==
pi@raspberrypi:~$
```

Figura 78 Terminal Linux, visualización de las contraseñas cifradas MQTT

En último paso se configura MQTT para el uso de credenciales, para ello se accede al fichero de configuración con el comando (*cd /etc/mosquitto/conf.d/*) seguido de (*sudo nano password.conf*) que permite editar el fichero para agregar las siguientes líneas de texto:

- (*allow_anonymous false*): Impide que un usuario anónimo no autenticado con las credenciales establecidas se conecte con el bróker MQTT.
- (*password_file /etc/mosquitto/conf.d/password.txt*): agrega las credenciales a la configuración de autenticación de Mosquitto.

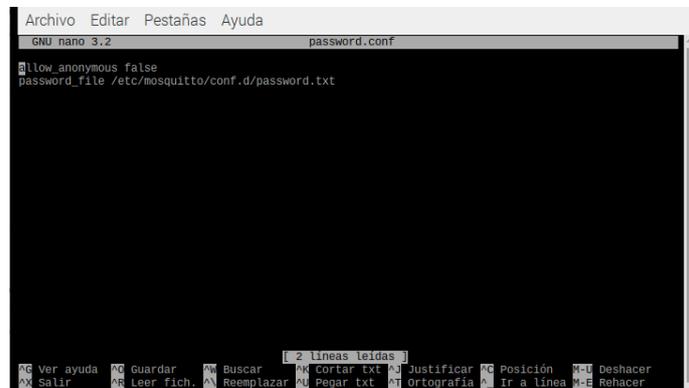


Figura 79 Terminal Linux, edición del fichero de configuración MQTT

- Verificación: se procede a reiniciar el bróker Mosquitto y verificar su nueva configuración, esto con los comandos: (`sudo systemctl restart mosquitto`) y (`systemctl status mosquitto`).

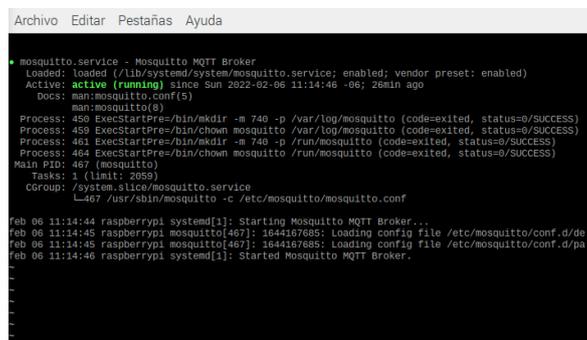
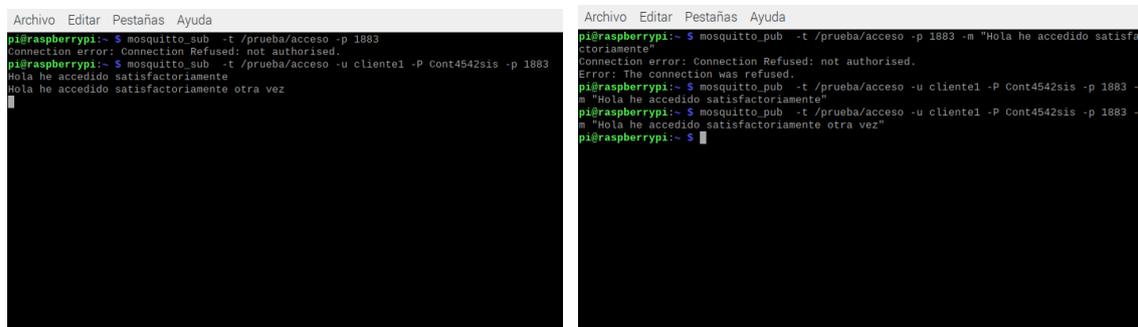


Figura 80 Terminal Linux, visualización del estado del bróker Mosquitto

En última instancia se verifica el funcionamiento del sistema de seguridad implementado en el bróker MQTT, para lo cual dentro de la terminal de Linux se ejecuta el comando (`mosquitto_sub -t /prueba/acceso -p 1883`) para suscribirse a un tema de interés, a continuación se verifica la prohibición de acceso al bróker MQTT. Por otra parte, al establecer las credenciales adecuadas con el comando (`mosquitto_sub -t /prueba/acceso -u cliente1 -P Cont4542sis -p 1883`) se puede suscribir satisfactoriamente a un tema llamado “/prueba/acceso” y permanecer a la espera de mensajes de llegada a dicho tema. En otra terminal se implementa un cliente publicador con el comando (`mosquitto_pub -t /prueba/acceso -u cliente1 -P Cont4542sis -p 1883 -m "Hola he accedido satisfactoriamente"`), de este modo se verifica la llegada de los

mensajes en el cliente suscriptor. Este mecanismo implica que cualquier cliente dentro de la red local (p.ej. un cliente NodeMCU) tenga que usar credenciales para su autenticación con el bróker MQTT previo al envío de datos al intermediario, a su vez imposibilita el acceso de la información a dispositivos no deseados.



```
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ mosquitto_sub -t /prueba/acceso -p 1883
Connection error: Connection Refused: not authorised.
pi@raspberrypi:~$ mosquitto_sub -t /prueba/acceso -u cliente1 -P Cont4542sis -p 1883
Hola he accedido satisfactoriamente
nota he accedido satisfactoriamente otra vez

Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ mosquitto_pub -t /prueba/acceso -p 1883 -m "Hola he accedido satisfactoriamente"
Connection error: Connection Refused: not authorised.
Error: The connection was refused.
pi@raspberrypi:~$ mosquitto_pub -t /prueba/acceso -u cliente1 -P Cont4542sis -p 1883 -m "Hola he accedido satisfactoriamente"
pi@raspberrypi:~$ mosquitto_pub -t /prueba/acceso -u cliente1 -P Cont4542sis -p 1883 -m "Hola he accedido satisfactoriamente otra vez"
pi@raspberrypi:~$
```

Figura 81 Terminales Linux, funcionamiento del proceso de autenticación con el bróker MQTT

Nota. De izquierda a derecha cliente suscriptor y publicador respectivamente

6.6 Herramientas software open source para la gestión de la información

6.6.1 Configuración de la herramienta Node-RED

Previo a la programación en la herramienta Node-RED preinstalada en el sistema operativo de la Raspberry, se realiza la configuración necesaria para su correcto funcionamiento. Por defecto, el editor Node-RED no está protegido y cualquier persona puede acceder a él conociendo su dirección IP, lo cual lo hace vulnerable a cambios no deseados. Por lo tanto, la configuración establecida permite securizar el acceso a la herramienta mediante un mecanismo de autenticación basado en credenciales de nombre de usuario/contraseña, para lo cual se definen los siguientes pasos a seguir.

- Ingreso a la carpeta de ubicación de Node-RED con el comando (`cd ~/.node-red`).
- Instalación de node-red-admin (Una herramienta de línea de comandos para administrar remotamente Node-RED), para esto se ejecuta el comando (`sudo npm install -g node-red-admin`).
- Cifrado de la contraseña: para calcular la contraseña se ejecuta el comando (`node-red-admin hash-pw`) se ingresa la contraseña y una vez cifrada se copia el hash proporcionado.

```
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ cd ~/.node-red
pi@raspberrypi:~/.node-red$ nod-red-admin hash-pw
bash: nod-red-admin: orden no encontrada
pi@raspberrypi:~/.node-red$ nod-red-admin hash-pw
Password:
52a5085DMejsazhdWm1stN1D2Enbu6KcW2PdqVvI5X1DA228/bdrK0i5uVie
pi@raspberrypi:~/.node-red$
```

Figura 82 Terminal Linux, generación de contraseña cifrada Node-RED

- Modificación de la configuración de Node-RED: en este paso se procede a acceder al fichero de configuración con el comando (*nano settings.js*) que permite su edición, para después descomentar la propiedad *adminAuth* y cambiar el *username* por el nombre usuario y la *password* por la contraseña cifrada.

```
GNU nano 3.2 settings.js
//ui: { path: "ui" },
// Securing Node-RED
// -----
// To password protect the Node-RED editor and admin API, the following
// property can be used. See http://nodered.org/docs/security.html for deta5
adminAuth: {
  type: "credentials",
  users: [{
    username: "jordy",
    password: "52a5085q58rYDqCtTzVq7ZBgh/Sg00mMqN0ZdeNod8s5myQILiuXTRHo5
    permissions: "*"
  }
],
// To password protect the node-defined HTTP endpoints (httpNodeRoot), or
// the static content (httpStatic), the following properties can be used.
// The pass field is a berypt hash of the pasword.
```

Figura 83 Terminal Linux, edición del fichero de configuración Node-RED

- Reiniciar el servicio Node-RED: en este último paso se reinicia el servicio para aplicar los cambios con el comando (*sudo systemctl restart nodered.service*), posteriormente se verifica el funcionamiento del uso de credenciales accediendo al servicio de forma local (IP:1880).

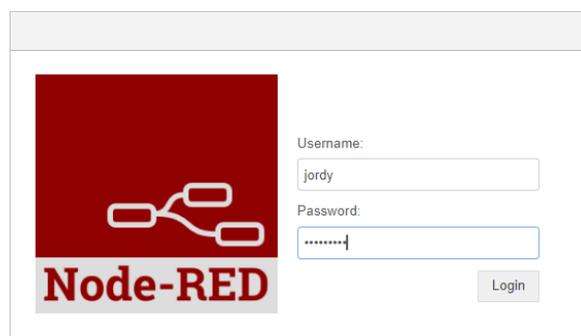


Figura 84 Proceso de login configurado dentro de Node-RED

- En el último paso se configura la herramienta Node-RED para que se inicie automáticamente al iniciar la Raspberry, mediante el comando (*sudo systemctl enable nodered.service*).

6.6.2 *Instalación del servidor local LAMP*

El servidor local instalado en la Raspberry es el encargado de almacenar diariamente los datos del consumo energético (medidos en kWh) del prototipo. Como consecuencia, los datos son accesibles al usuario de manera gráfica y textual por medio de la interfaz web programada en la herramienta Node-RED. Para lograr estas capacidades, el funcionamiento del servidor se concentra en un servidor LAMP por sus características y ventajas analizadas anteriormente (ver apartado 4.20.3). En su causa, el presente apartado detalla brevemente los pasos realizados para la instalación por medio de la consola de la Raspberry, de los distintos paquetes de *software* necesarios para el funcionamiento del sistema de almacenamiento, gestión, y visualización de los datos provenientes del prototipo. Las instrucciones y comandos de instalación se presentan a continuación:

- Instalación de Apache2: Previo a la instalación es necesario actualizar la Raspberry, para lo cual se ejecuta el comando (*sudo apt update && sudo apt upgrade -y*) y después el comando (*sudo apt install apache2 -y*) para la instalación del servidor web.
- Instalación de PHP: el lenguaje de programación PHP permite la conexión entre el servidor y la interfaz de usuario phpMyAdmin, para esto se ejecuta el comando (*sudo apt install php -y*) y a continuación se reinicia Apache2 con el comando (*sudo service apache2 restart*).
- Instalación de MariaDB: para la instalación de los paquetes MySQL Server (MariaDB Server) y PHP-MySQL se ejecuta el comando (*sudo apt install mariadb-server php-mysql -y*), y nuevamente se procede a reiniciar el servicio mediante el comando (*sudo service apache2 restart*). Después de instalar MariaDB Server, se procede a ejecutar el comando (*sudo mysql_secure_installation*) que permite asignar la configuración del servidor MariaDB como la contraseña para el usuario *root* y asegurar su instalación de forma correcta.
- Instalación de phpMyAdmin: para instalar esta herramienta de interfaz web destinada a manejar la administración de MySQL, se ejecuta el comando (*sudo apt install phpmyadmin -y*), aquí se selecciona Apache2 como el servidor web al que se conectará automáticamente y se aceptan y continúan los pasos posteriores. Una vez instalado el paquete se procede a habilitar la extensión PHP MySQLi mediante el comando (*sudo phpenmod mysqli*) y finalmente se reinicia Apache2 para establecer los cambios. Para visualizar la interfaz de phpMyAdmin simplemente se accede a la dirección IP de la Raspberry PI seguida de /phpmyadmin (p.ej. <http://192.168.1.107/phpmyadmin>).



Figura 85 Proceso de login de la interfaz phpMyAdmin

6.6.3 Creación de la base de datos local

La creación y administración de la base de datos MySQL se realizó por medio de la interfaz gráfica de phpMyAdmin. En primer lugar, se creó una nueva base de datos con el nombre “Datos_consumo” en la cual se añadió una tabla con el nombre “Energía Activa” cuya finalidad es la de almacenar diariamente la energía consumida por el sistema y otros valores como la potencia activa del sistema, el factor de potencia y el tiempo de encendido de la luminaria para fines de cálculo del consumo energético analizado en apartados posteriores. Por otra parte, es la herramienta Node-RED la encargada de escribir y leer la información de la tabla, es decir, escribe diariamente los valores de energía consumidos y además extrae dicha información de las filas para permitir su visualización por medio de su interfaz de usuario web. En este sentido para la configuración de la tabla “Energía Activa” se definió los siguientes campos, los cuales poseen las siguientes características.

Tabla 17 Campos configurados en la tabla de la base de datos del sistema dentro de la plataforma phpMyAdmin

Tabla Energía Activa		
Campo	Tipo	Función
ID	int, autoincrementar	Este valor es escrito de forma automática en las filas al escribir un nuevo valor de consumo en kWh.
FECHA	varchar	Es escrita por la herramienta Node-RED de forma periódica. Permite realizar búsquedas rápidas del valor (kWh) para una determinada fecha.
HORA	time	Es escrito de forma automática por phpMyAdmin al momento de ingresarse un nuevo valor (kWh), como huella temporal.

kWh	float	Almacena los kWh consumidos por el prototipo, el valor es escrito en la tabla de forma periódica, y también extraído para su visualización en la interfaz de usuario web.
Watts	float	Es escrito en la tabla de forma periódica, por otra parte, este valor también es extraído de la tabla para cálculos relacionados al consumo energético.
FP	float	Es escrito en la tabla de forma periódica, por otra parte, permite calcular y extraer el FP promedio para un determinado día de funcionamiento del prototipo.
DURACION	varchar	Almacena el tiempo de encendido del sensor PIR para fines relacionados al cálculo del consumo energético.

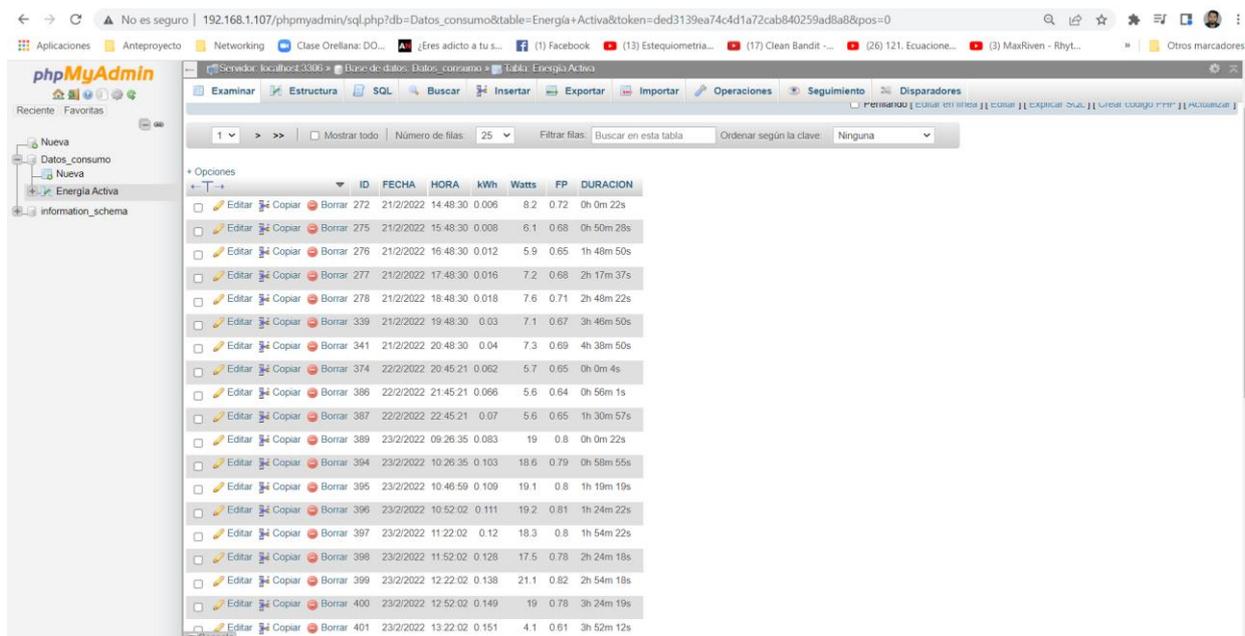


Figura 86 Interfaz de phpMyAdmin, estructura de la tabla de datos

6.7 Sistema de monitoreo, gestión y visualización de la información

El presente apartado tiene la finalidad de describir la elaboración y el funcionamiento del bloque final del prototipo encargado del sistema de monitorio, gestión y visualización de la información, el cual es implementado en el ordenador central del sistema. Una vez configurado el protocolo de comunicación inalámbrico e instalado el servidor de la base de datos local, se describe a continuación, la estructura del flujo de programa elaborado en la herramienta Node-RED, que permite el desarrollo de la interfaz gráfica de usuario web para el cumplimiento de los objetivos propuestos.

6.7.1 Programación basada en flujo dentro de la herramienta Node-RED

El desarrollo de la programación se dividió en diferentes flujos de programas dentro de la interfaz de programación de aplicaciones Node-RED, esto permite dividir la interfaz de usuario web en tres componentes principales, que facilitan la visualización de la información del prototipo, como también la interacción con el usuario, estos son:

- Control de Cargas
- Datos energía
- Datos Ambientales

6.7.1.1 Adquisición de datos.

Dentro de la programación de los diferentes flujos de programa, es necesario describir los nodos utilizados para el proceso de adquisición y envío de la información del sistema. Los datos recibidos en el ordenador central son accesibles por medio de un nodo MQTT de entrada, el cual se conecta al bróker MQTT y se suscribe a los mensajes de un tema específico.

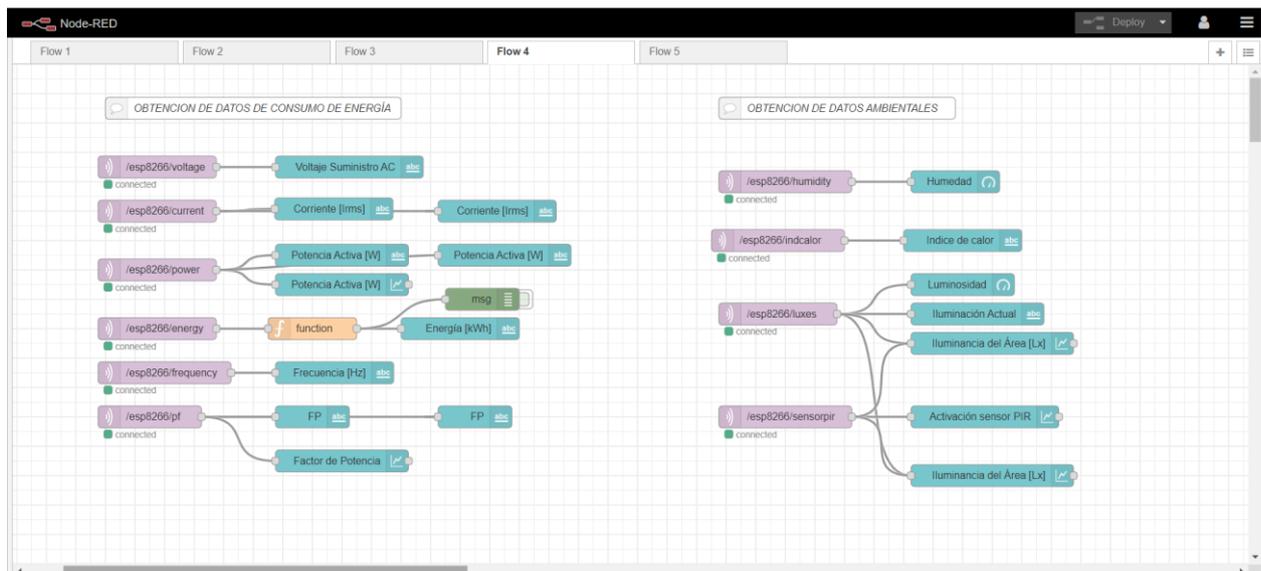


Figura 87 Interfaz Node-RED; flujo para la adquisición de datos

De este modo se analiza la estructura del nodo *mqtt in*, el cual define diferentes propiedades como la dirección IP del servidor, el nombre del tema al cual se suscribe, la QoS elegida (2 para este trabajo), así como el uso de credenciales para el proceso de autenticación con el bróker MQTT, todo esto mediante una sencilla gestión de las propiedades del nodo, como se observa a continuación.

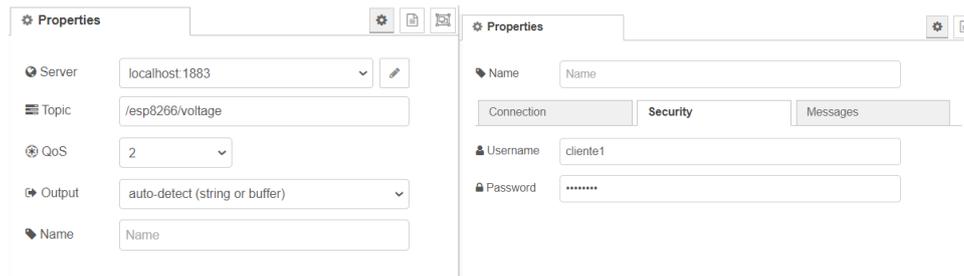


Figura 88 Configuración de las propiedades del nodo mqtt in

En este sentido, la siguiente Tabla 18 presenta la descripción de los nodos de la Figura 87, como se muestra a continuación:

Tabla 18 Descripción de los nodos usados para la adquisición de datos

Nodo	Función
Comment	-Permite agregar comentarios a los flujos.
Mqtt in	-Permite conectarse al bróker MQTT y suscribirse a los mensajes de un tema específico. -La información adquirida la pasa a la propiedad msg.payload.
ui_text	-La propiedad msg.payload de entrada se muestra en un campo de texto no editable en la interfaz de usuario UI.
ui_chart	-La propiedad msg.payload de entrada es convertida en un número para luego trazar estos valores en un gráfico dentro de la UI.
ui_gauge	-Agrega un widget de tipo indicador a la interfaz de usuario. -La propiedad msg.payload de entrada puede ser formateada para mostrarse en el indicador con un tono de color específico para cada valor calculado.

6.7.1.2 Interfaz web para el control de cargas.

Por otro lado, el proceso de envío de información desde la interfaz de usuario web al cliente NodeMCU, hace uso de un nodo MQTT de salida, el cual se configura con las mismas propiedades del nodo *mqtt in* con excepción del *topic* o tema en el cual el cliente publicará la información proporcionada.

De la Figura 89 se puede analizar que el sistema posee dos modos de operación (automático o manual), los cuales son manipulables desde la interfaz de usuario web. La información enviada por los nodos *mqtt out* en los diferentes *topics* es recibida por el cliente NodeMCU el cual está suscrito a dichos temas, para posteriormente efectuar un cambio en el funcionamiento del algoritmo de control desarrollado y detallado en (ver Anexo 13).

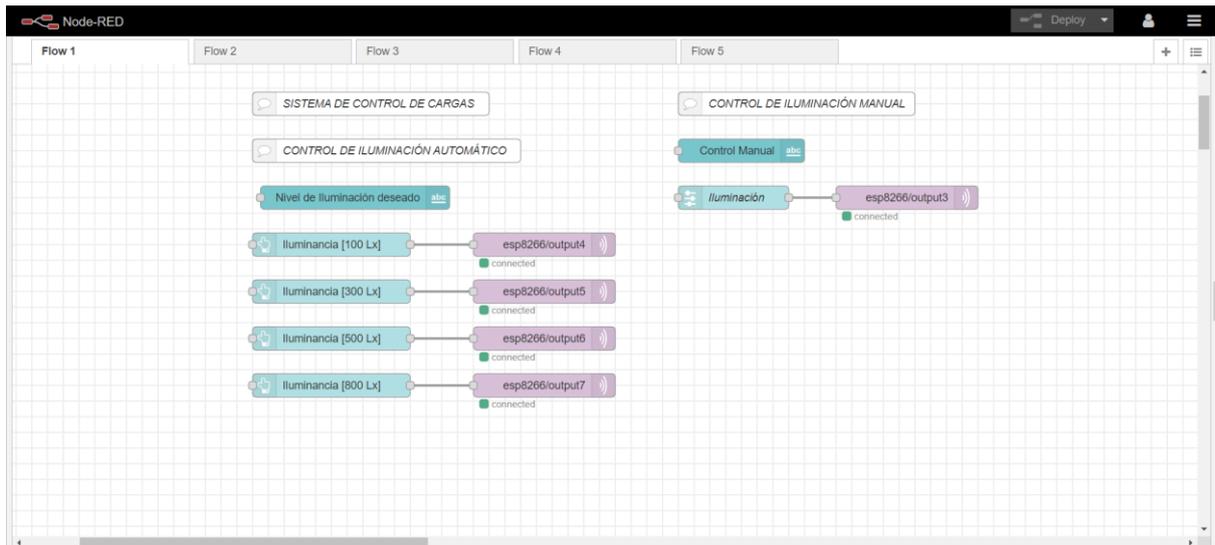


Figura 89 Interfaz Node-RED; flujo para el envío de datos

De igual forma se presenta la descripción de los nodos utilizados, mediante la siguiente Tabla 19.

Tabla 19 Descripción de los nodos usados para el envío de datos

Nodo	Función
ui_slider	-Agrega un widget deslizante a la interfaz de usuario. -El usuario puede cambiar su valor entre los límites (mín. y máx.). Cada cambio de valor genera un mensaje con el valor establecido como msg.payload.
ui_button	-Añade un botón a la interfaz de usuario. -Al hacer clic en el botón, se genera un mensaje con un msg.payload configurado dentro de sus propiedades.
Mqtt out	-Permite conectarse al bróker MQTT y publicar mensajes en un tema específico. -La información que publica corresponde a la propiedad msg.payload de entrada al nodo.

6.7.1.3 Interfaz web para la monitorización de los datos ambientales.

El proceso de monitorización conlleva un análisis constante de los datos adquiridos del sensor DHT22, sensor PIR y el sensor luxómetro BH1750. Como se ha planteado en párrafos anteriores, el desarrollado del sistema de monitorización de datos ambientales permite realizar tareas como:

- Visualización en tiempo real de los parámetros ambientales de temperatura, humedad, índice de calor y luminosidad dentro del área de funcionamiento del prototipo.

- Establecimiento programado de un rango de temperatura adecuado para el ambiente de trabajo para fines de alerta, esto mediante límites de temperatura máxima y mínima configurables dentro de la UI.
- Generación de reportes vía Gmail al detectar variaciones en la temperatura fuera del rango de temperatura óptimo elegido por el usuario.

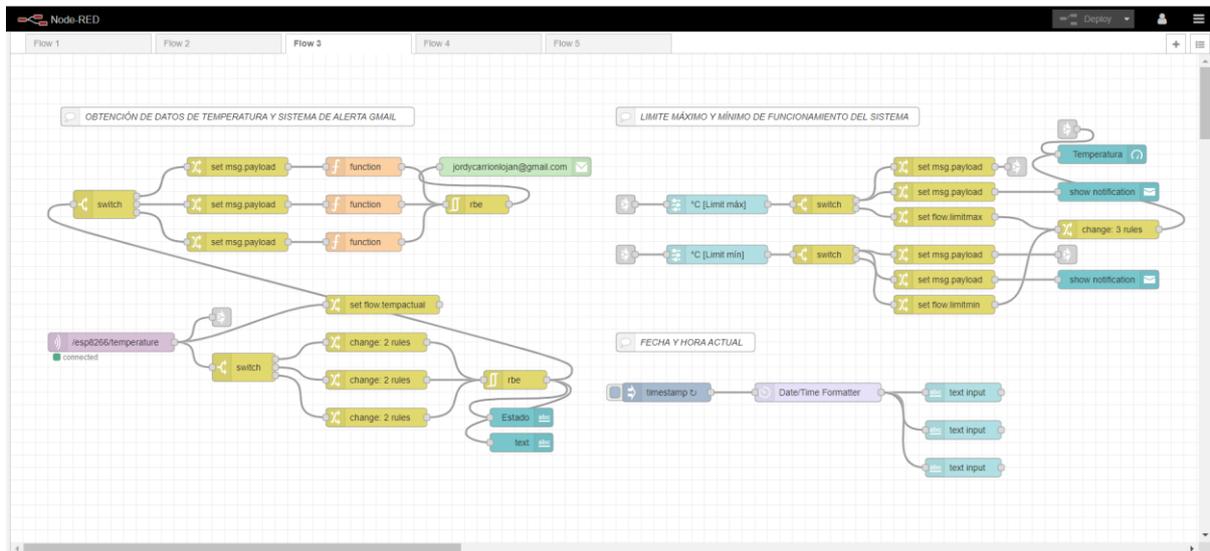


Figura 90 Interfaz Node-RED; flujo para el monitoreo de los datos ambientales

La Figura 90 presenta un flujo de programa en el cual se distinguen tres conjuntos de bloques que permiten realizar tareas específicas como la obtención de datos de temperatura, creación del reporte Gmail, creación de widgets para la interacción con el usuario, así como la introducción de procesos condicionales, reglas y notificaciones para el correcto funcionamiento de la interfaz de usuario web. A fin de cumplir con estas cualidades del sistema se presenta la descripción de algunos nodos utilizados, mediante la siguiente Tabla 20.

Tabla 20 Descripción de los nodos usados para la monitorización de los datos ambientales

Nodo	Función
switch	-Cuando llega un mensaje, el nodo evalúa cada una de las reglas definidas y posteriormente reenvía el mensaje a la salida correspondiente.
rbe	-Analiza la propiedad msg.payload de entrada, y solo transmite a su salida aquellos datos donde el contenido payload o carga útil ha cambiado.
change	-Este nodo permite establecer, cambiar, eliminar o mover propiedades de un mensaje. -Los nodos usados permiten establecer variables de contexto de flujo para su uso dentro de los nodos function y switch, por ejemplo las variables flow.limitmin o flow.limitmax.

function	<p>-Permite crear una función de JavaScript para ejecutarse en los mensajes que recibe el nodo.</p> <p>-La función creada permite acceder a las variables de flujo y la creación de mensajes textuales, esto modificando la propiedad msg.payload que contiene el cuerpo del mensaje.</p>
e-mail	<p>-Permite enviar el msg.payload de entrada como un correo electrónico, con un asunto establecido en la propiedad msg.topic.</p> <p>-El nodo permite configurar el destinatario del mensaje dentro de sus propiedades.</p>
inject	<p>-Inyecta un mensaje en un flujo de forma manual o a intervalos regulares. La carga útil del mensaje puede ser de varios tipos, incluidas cadenas, objetos de JavaScript o la hora actual.</p> <p>-El nodo usado permite iniciar un flujo de proceso a intervalos de tiempo de 2 segundos, para accionar el funcionamiento del nodo moment.</p>
moment	<p>-Este nodo convierte un objeto de fecha/hora o una cadena en un texto o un objeto de fecha/hora con un formato agradable.</p> <p>-La información de fecha/hora a la salida del nodo se usa para su visualización dentro de la UI, por medio de nodos ui_text_input.</p>
ui_toast	<p>-Este nodo muestra el msg.payload de entrada como una notificación emergente o un mensaje de diálogo Aceptar/Cancelar en la interfaz de usuario.</p> <p>-Es usado para mostrar mensajes de alerta para la correcta configuración de los límites de temperatura máxima y mínima.</p>
link out	<p>-Permite crear conexiones virtuales entre flujos de nodos diferentes. El nodo se puede conectar a cualquier nodo link in que exista en cualquier pestaña.</p>
link in	<p>-Permite crear conexiones virtuales entre flujos de nodos diferentes. El nodo se puede conectar a cualquier nodo link out que exista en cualquier pestaña.</p>

6.7.1.4 Interfaz web para la gestión y visualización del consumo energético.

En último lugar se describe el flujo de programa elaborado para la gestión de la información del consumo energético del prototipo, el cual a través de distintos nodos permite cumplir las funciones detalladas a continuación:

- Permite en tiempo real, presentar la información acerca de los parámetros eléctricos y de consumo de energía de la luminaria.

- Lectura diariamente a intervalos de tiempo de media hora el valor de consumo energético del sensor PZEM004T (dado en kWh) y lo almacena en la base de datos local, con la fecha del día.
- El sistema lectura todos los días a una hora determinada el valor de consumo energético y envía un reporte vía Gmail a un destinatario programado.
- Por medio de la interfaz gráfica, el usuario puede acceder a una fecha cualquiera y verificar el consumo de energía. Esto se logra accediendo a la base de datos local y extrayendo los valores para la fecha indicada. Aunado a esto, el usuario puede obtener un reporte vía Gmail del consumo entre la fecha actual y otra fecha escogida dentro de la UI.
- El sistema permite visualizar el tiempo de encendido de la luminaria en el día. De modo idéntico, el usuario puede acceder a información de otro día por medio de la interfaz gráfica, aunado a esto, el sistema obtiene el valor de la potencia activa y factor de potencia promedio del funcionamiento del prototipo, para lo cual analiza todas las mediciones realizadas y almacenadas en la base de datos para una fecha seleccionada.
- Adicional, el usuario puede visualizar por medio de la interfaz gráfica la reducción de consumo energético para los distintos días de funcionamiento, esto se logra analizando el ahorro de consumo en función del sensor PIR, como también, en función del sistema de control de potencia que permite reducir la cantidad de energía suministrada a la luminaria.

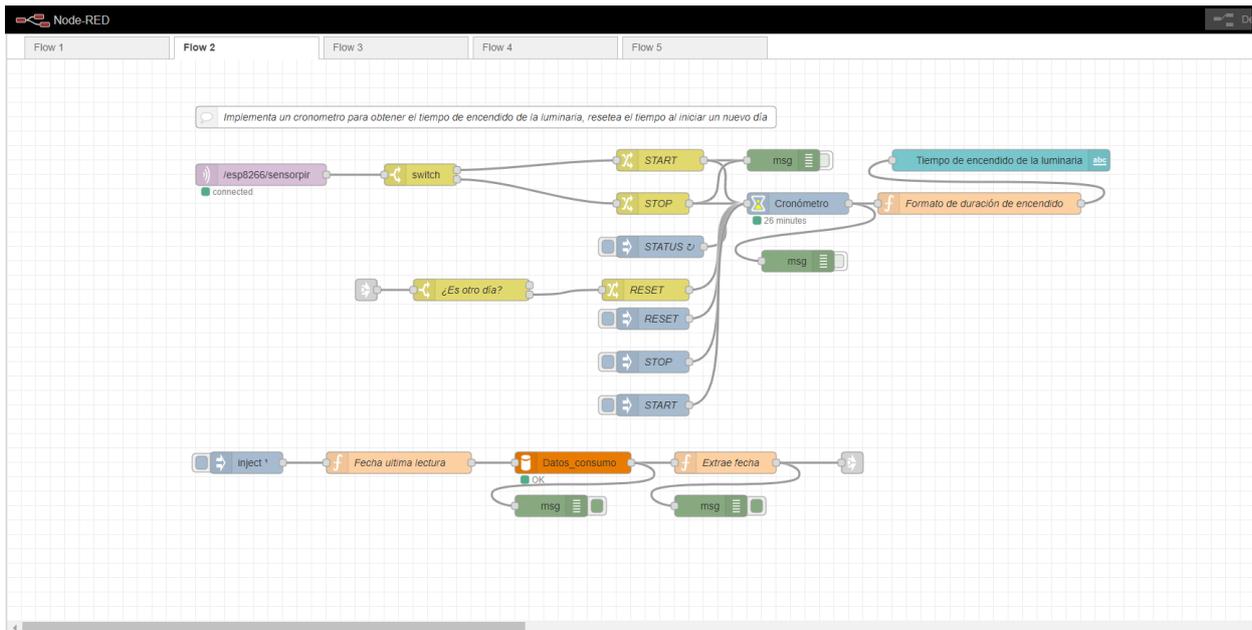


Figura 91 Interfaz Node-RED; cálculo del tiempo de encendido de la luminaria

Nota. Permite implementar un cronómetro que se activa y desactiva en función del sensor PIR, posteriormente almacena el tiempo de encendido del sensor en la base de datos

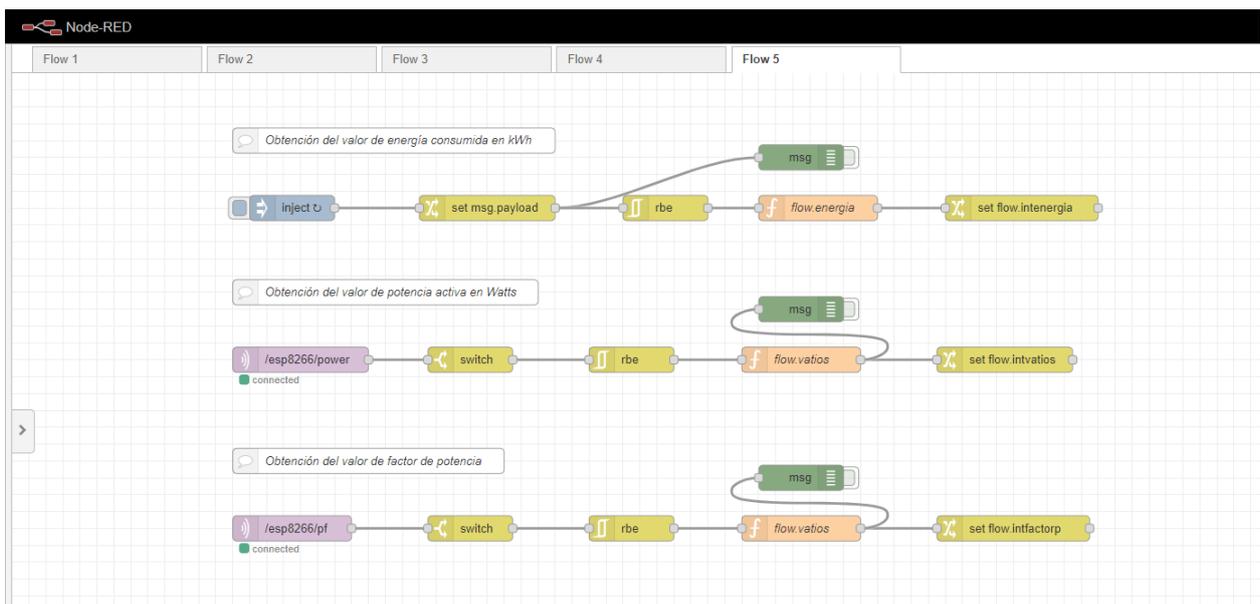


Figura 92 Interfaz Node-RED; obtención de los parámetros eléctricos de la luminaria

El siguiente flujo de programa analizado en la Figura 93, permite al sistema realizar una lectura cada media hora de los parámetros de: potencia activa P, factor de potencia FP, energía consumida en kWh y el tiempo de encendido de la luminaria. Mediante el nodo ‘Escritura’ se realiza el comando SQL necesario para almacenar estos valores en la base de datos según la fecha del día. Adicionalmente, dentro de la base de datos también se escribe el instante de la medición para los cálculos posteriores de ahorro energético.

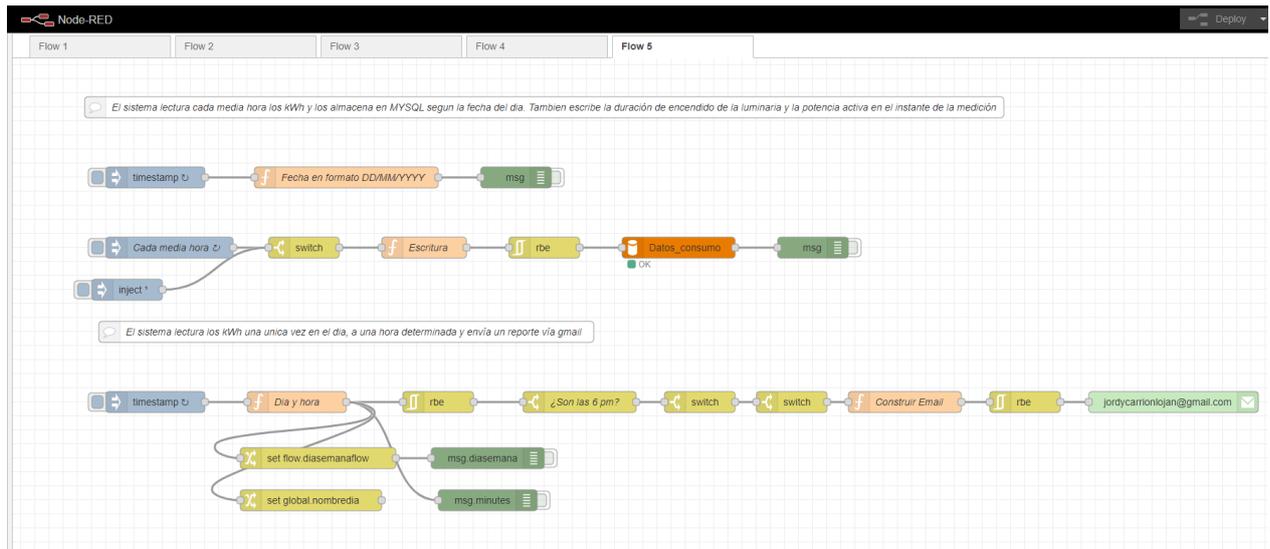


Figura 93 Interfaz Node-RED; escritura en la base de datos del sistema

Nota. El flujo de nodos inferior envía un reporte diario a las 6 pm con el valor de energía consumida hasta la fecha

Posteriormente la siguiente Figura 94 permite implementar un calendario en la interfaz de usuario para la selección de la fecha de consulta. Los nodos *function* permiten extraer los valores de consumo energético de la base de datos para la fecha seleccionada y mostrarlos en forma de gráficos para una visualización más adecuada.

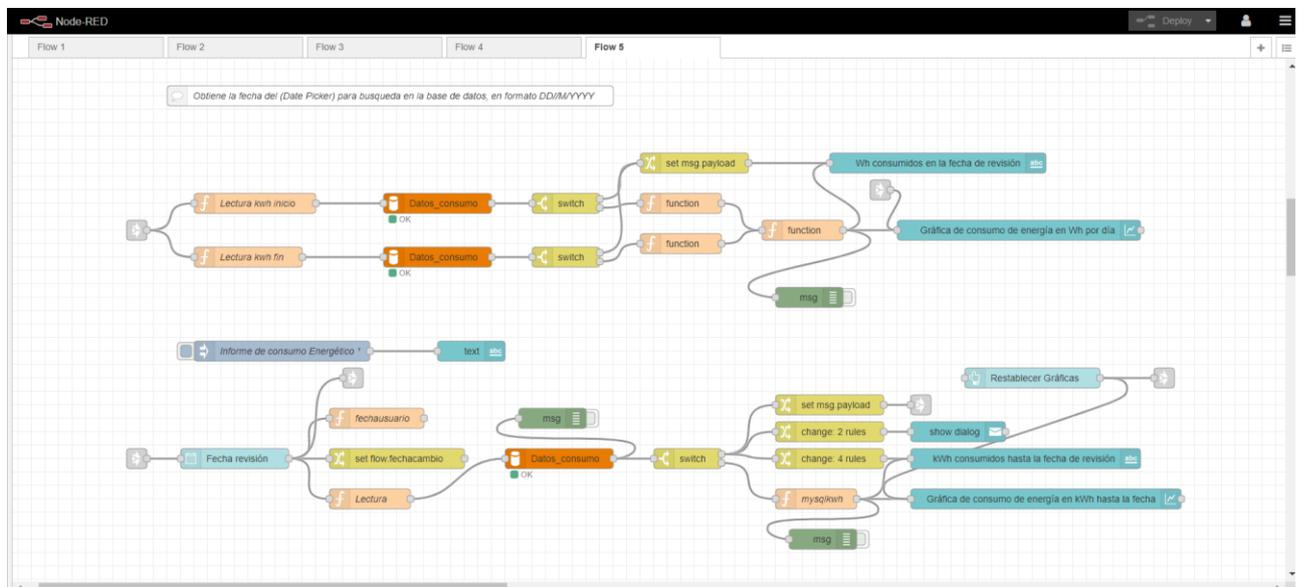


Figura 94 Interfaz Node-RED; obtención de la fecha en el Date Picker para realización de consultas

Una de las funcionalidades del sistema es calcular el ahorro energético de la luminaria, para lo cual, el programa desarrollado extrae de la base de datos el tiempo de encendido de la luminaria. La siguiente Figura 95 ejemplifica este proceso.

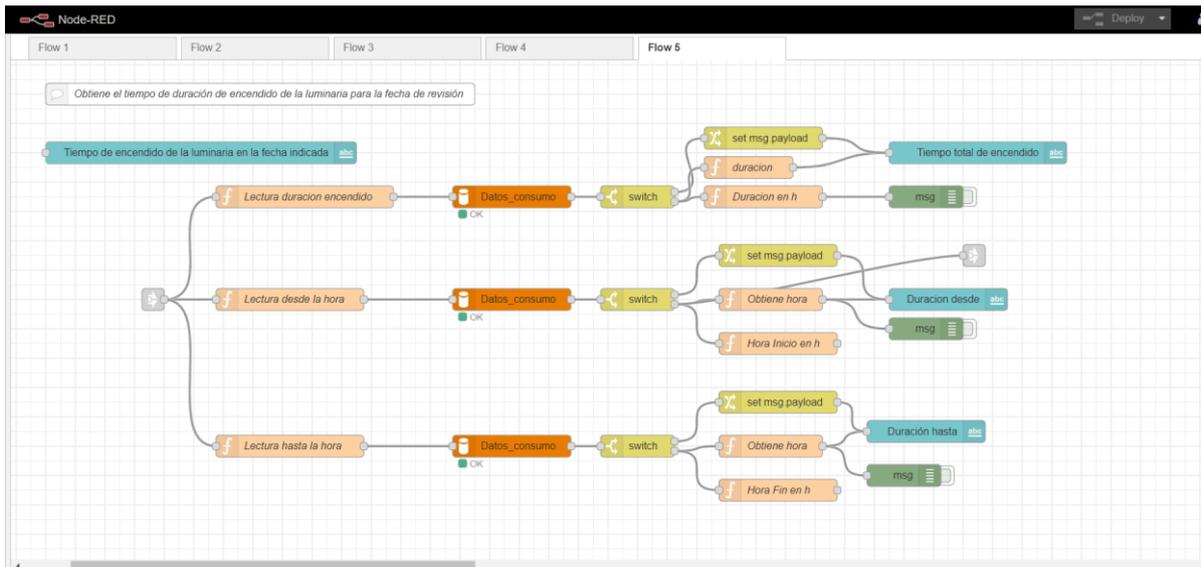


Figura 95 Interfaz Node-RED; obtención de la duración de encendido de la luminaria

Por otra parte, el prototipo permitir obtener los valores promedios de potencia activa P y factor de potencia FP para una fecha de funcionamiento. En este sentido, el uso de funciones SQL permiten simplificar este proceso, por ejemplo, para la extracción de la potencia activa promedio se utilizó el comando SQL detallado en la propiedad msg.topic dentro del nodo 'Lectura potencia activa promedio' de la siguiente forma: (msg.topic=SELECT ROUND(AVG(ALL`Watts`)) FROM `Energía Activa` WHERE `FECHA` LIKE ""+fecha+"" AND `Watts` != 0");. Aunado a esto, los nodos de la parte inferior de la Figura 96, permiten crear un reporte de la energía consumida desde una fecha establecida hasta la última lectura dentro de la fecha actual de la base de datos.

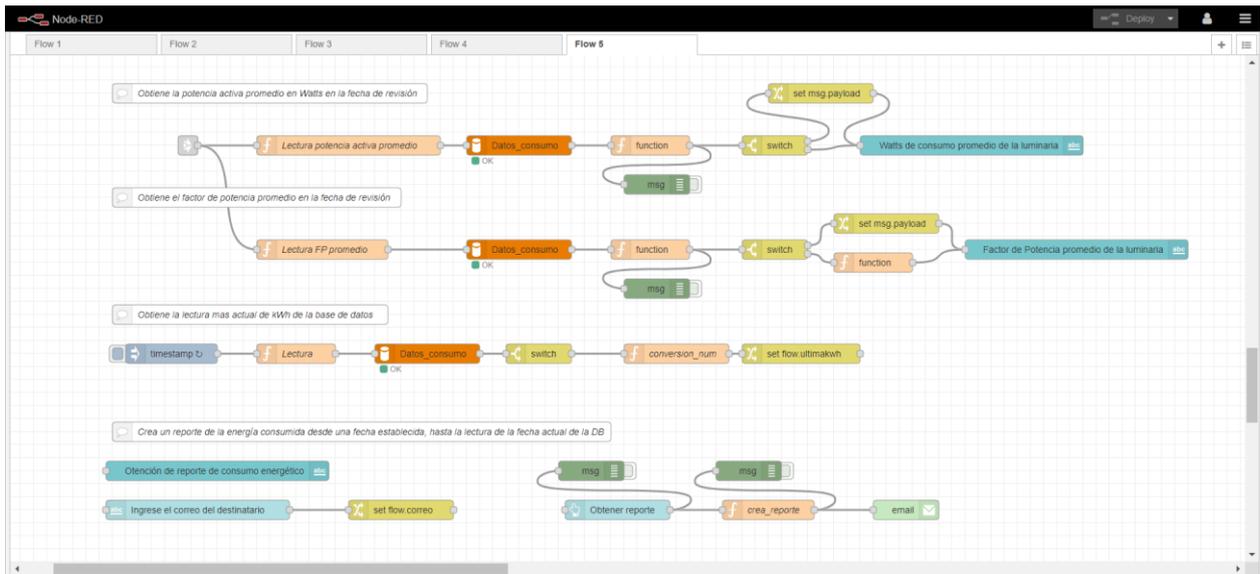


Figura 96 Interfaz Node-RED; obtención de los valores promedio de potencia activa y factor de potencia

Una vez obtenidos los valores necesarios de la base de datos para el cálculo del ahorro energético en una fecha en particular, tales como: la potencia activa promedio, los kWh consumidos, el tiempo de encendido del prototipo, el tiempo de encendido de la luminaria; el desarrollo posterior permitió obtener los valores de ahorro en función del sistema de control de potencia como también en función del sistema de detección de presencia; el análisis de este estudio se plantea detalladamente en el apartado 6.9. Por su parte, la siguiente Figura 97 ejemplifica este proceso.

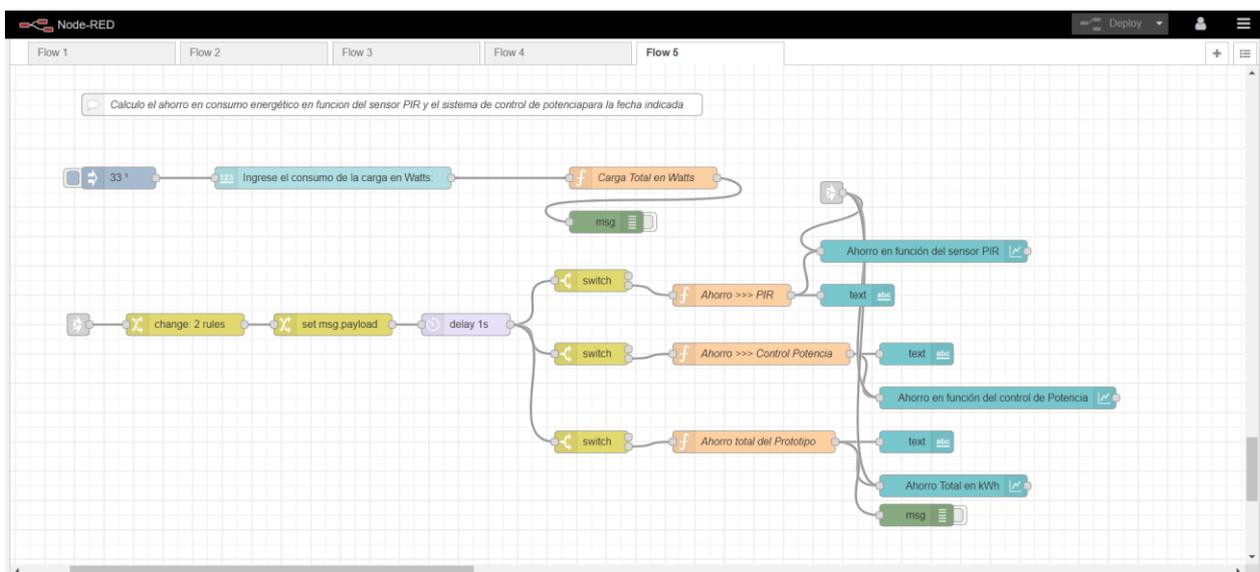


Figura 97 Interfaz Node-RED; cálculo del ahorro energético-económico

Cabe destacar que el funcionamiento del programa se centra en el uso del nodo mysql, el cual permite realizar diversas operaciones en la base de datos (Datos_consumo),

específicamente en la tabla (Energía Activa) gracias al uso de simples comando SQL definidos previamente en los nodos *function*. Evidentemente es necesario la configuración previa de este nodo mysql para permitir el acceso a la base de datos local como se presenta a continuación.

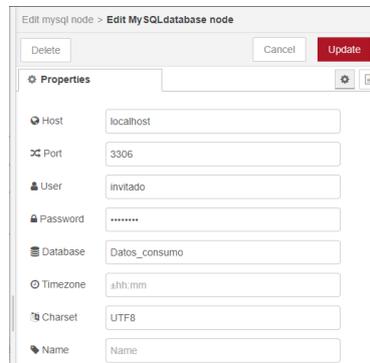


Figura 98 Configuración de las propiedades del nodo mysql

En otro orden de ideas, de los flujos de programa presentados en este contexto, se observan nuevos nodos para la creación de *widgets* de interacción con el usuario, así como nodos para el manejo de reglas y notificaciones de alerta que permiten el correcto funcionamiento de la interfaz de usuario web. Teniendo en cuenta estas características del sistema, se presenta la descripción de algunos nodos utilizados mediante la siguiente Tabla 21.

Tabla 21 Descripción de los nodos usados para la gestión del consumo energético

Nodo	Función
inject	-El nodo usado permite iniciar un flujo de proceso a intervalos de tiempo regulados, por ejemplo, cada media hora para escribir los valores de consumo dentro de la base de datos local.
change	-Los nodos usados permiten establecer variables de contexto de flujo para su uso dentro de los nodos <i>function</i> . Ejemplos de estas variables creadas son: flow.intenergia, flow.correo, o flow.ultimakwh que almacena la última lectura de la base de datos.
function	-Las funciones creadas permiten acceder a las variables de flujo, la creación de reportes Gmail o creación de instrucciones SQL tanto de escritura como de lectura, posteriormente estas son inyectadas dentro del nodo mysql mediante la modificación de la propiedad msg.topic la cual contiene la propia consulta SQL (p.ej. msg.topic="INSERT INTO `Energía Activa` (`ID`, `FECHA`, `HORA`, `kWh`, `Watts`, `FP`, `DURACION`) VALUES (NULL, '"+ fecha+"", CURRENT_TIME(),"+ energia +", "+ vatios +", "+ fp +", "+ duracion +")");).
Mysql	-Permite el acceso básico a una base de datos MySQL. Además, permite realizar operaciones de consulta en la base de datos configurada.

	-La propiedad msg.topic debe contener la consulta de la base de datos y el resultado se devuelve en formato msg.payload.
debug	-Muestra las propiedades del mensaje a la salida de un nodo. De forma predeterminada muestra el msg.payload, pero se puede configurar para mostrar cualquier propiedad, el mensaje completo o el resultado de una expresión JSONata.
switch	-Cuando llega un mensaje, el nodo evalúa cada una de las reglas definidas y posteriormente reenvía el mensaje a la salida correspondiente. -Por ejemplo, este nodo es usado para verificar que sea una hora determinada para el envío de un reporte Gmail.
ui_toast	-Este nodo es usado para mostrar un mensaje de diálogo “No existe lectura para la fecha indicada, por favor ingrese otra fecha” Aceptar/Cancelar dentro de la interfaz de usuario.
Hourglass	-Este nodo permite implementar un cronómetro el cual se activa y desactiva en función del estado del sensor PIR. El tiempo se resetea diariamente mediante la instrucción msg.command=”reset”.

6.7.2 Pruebas y resultados de la interfaz de usuario web

El presente apartado muestra las diferentes pestañas de la interfaz de usuario web realizadas para el manejo del prototipo. Como resultado la Figura 99 muestra la primera pestaña denominada “Datos Ambientales”, la cual permite cumplir con las características y tareas analizadas en (ver apartado 6.7.1.3).

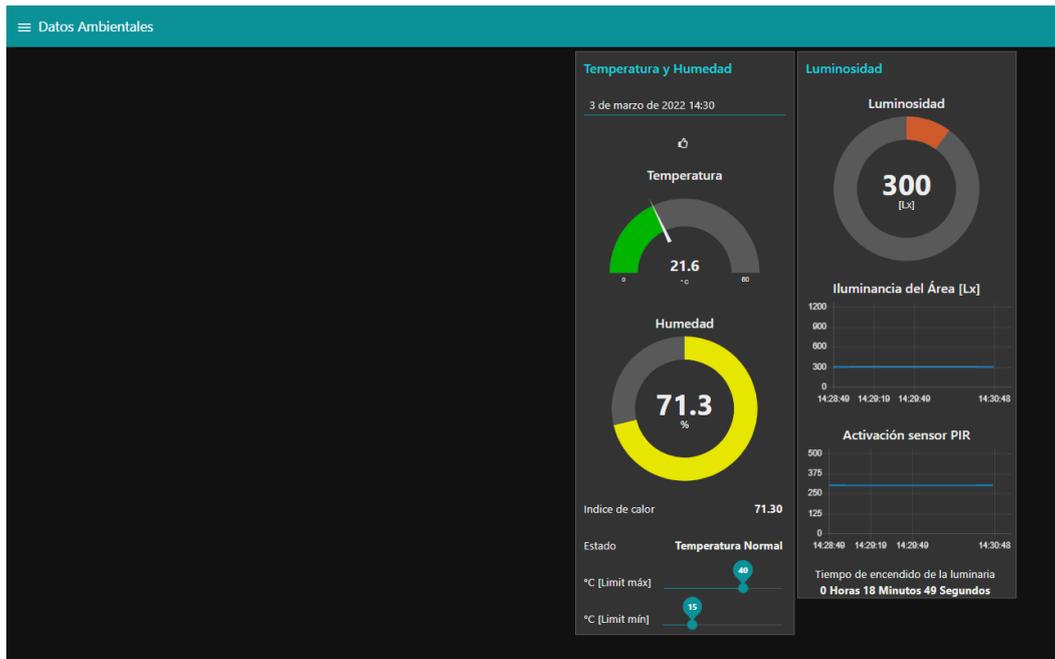


Figura 99 Interfaz de usuario web; Datos ambientales

Para ejemplificar el sistema de alerta Gmail, se configura manualmente el límite máximo y mínimo a los valores de 40 y 15 grados Celsius respectivamente, por ende, al tener una lectura reciente de 21,6 °C del ambiente circundante, el programa arroja el estado de “Temperatura Normal”. Dicho estado es analizado por el programa y posteriormente enviado vía Gmail como se observa en la Figura 100. Ahora bien, de tener otra configuración o valor de temperatura actual, se generarían diferentes reportes, los cuales incluyen:

- Temperatura superior al límite máximo
- Temperatura inferior al límite mínimo
- Temperatura en estado normal

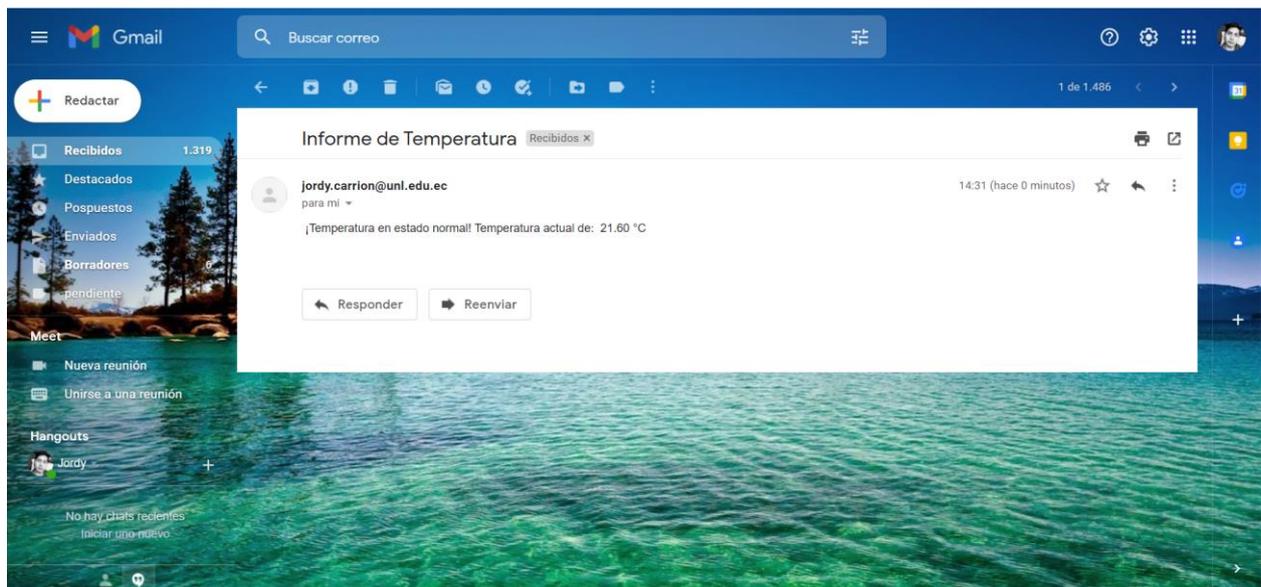


Figura 100 Reporte Gmail del estado de la temperatura actual

En segundo lugar, la interfaz de “Control de Cargas” presenta los distintos elementos para el control del prototipo, además presenta de manera gráfica el nivel de iluminancia dentro del área en combinación con el estado del sensor PIR. La interfaz cuenta con 4 niveles seleccionables y el control manual de la luminaria, ambos modos de operación maximizan el ahorro energético en función de la presencia o ausencia de personas dentro del área.

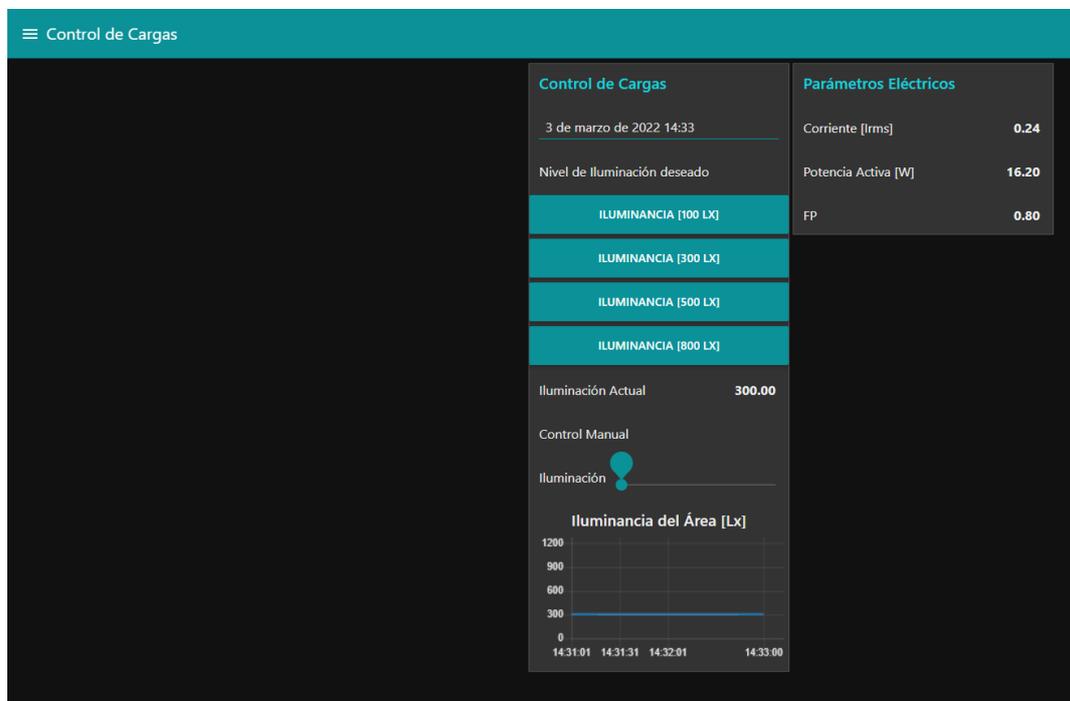


Figura 101 Interfaz de usuario web; Control de Cargas

Por último, la interfaz de “Datos Energía” permite cumplir las tareas programadas y analizadas anteriormente (ver apartado 6.7.1.4). Para ejemplificar la obtención de un reporte

con información acerca del consumo energético se elige una fecha anterior a la actual y se procede a ingresar el correo de un destinatario, de modo que al hacer clic en el botón “obtener reporte” se genera un mensaje con el informe de consumo hasta la fecha actual, y cuya información se observa en la Figura 103.

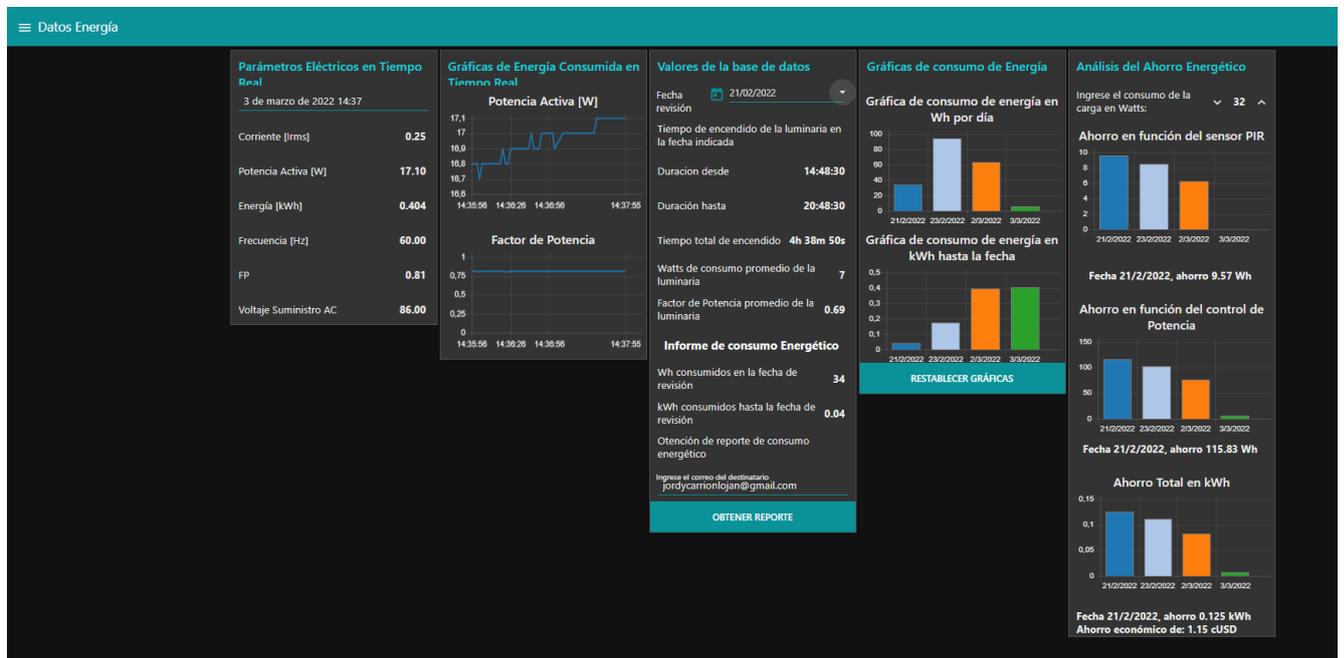


Figura 102 Interfaz de usuario web; Datos Energía

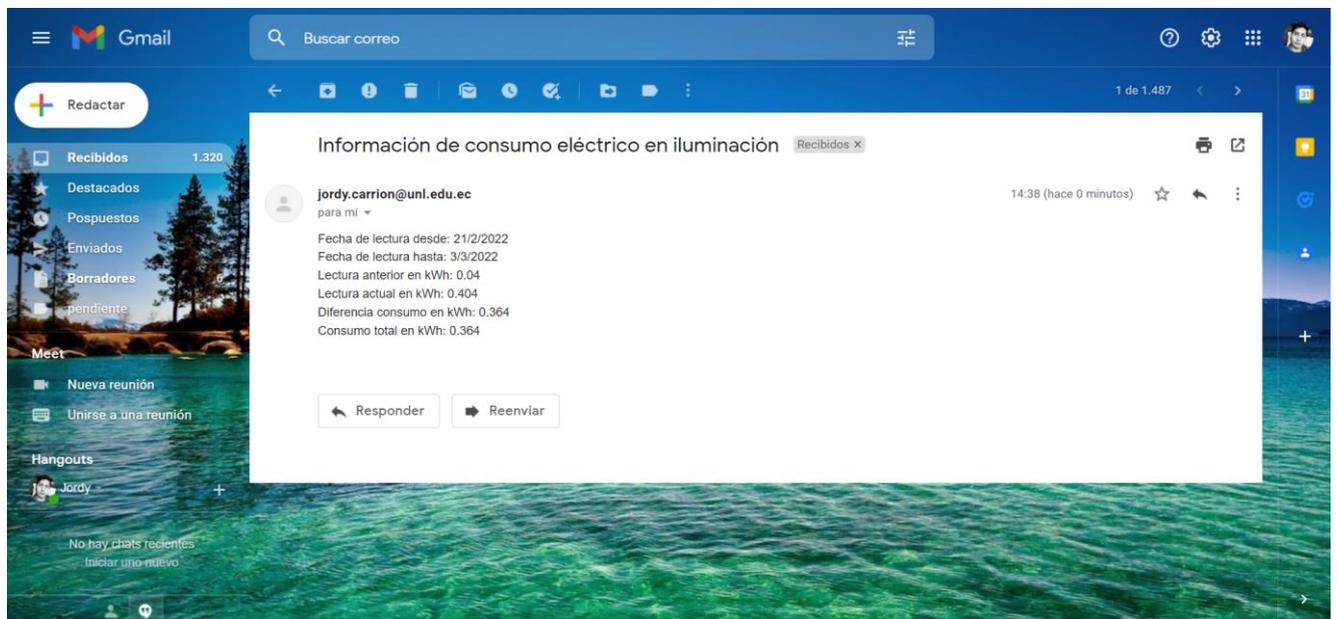


Figura 103 Reporte Gmail del estado de consumo energético

6.7.3 Plataforma de análisis de datos IoT

Una plataforma IoT es el software que provee las herramientas e infraestructura para la realización de diversas tareas, que, dependiendo el tipo de plataforma IoT permitirá realizar

tareas como: conectividad de dispositivos, adquisición de datos, visualización, base de datos, análisis de datos, como también la gestión de acciones en base a los datos procesados. Sobre todo, una plataforma IoT proporciona una funcionalidad extendida para otorgar valor añadido a un producto final, a la vez que simplifica el trabajo de los desarrolladores a la hora de construir una aplicación IoT.

En el desarrollo del presente trabajo se estableció el uso de una plataforma de análisis de datos IoT que permita acceder a la información proveniente del prototipo, esto dentro de un ecosistema de interfaz gráfica amigable y de alto potencial de análisis de datos. La ventaja de la plataforma es que permite acceder a dicha información de forma remota y segura mediante el acceso a Internet desde cualquier lugar con conexión.

6.7.3.1 Plataforma ThingSpeak.

ThingSpeak es un software de código abierto escrito en Ruby, ofrece un servicio de plataforma de análisis de datos IoT que permite agregar, visualizar y analizar flujos de datos en vivo en la nube (Lawlor, 2014). Esta plataforma permite el envío de la información a la nube desde los propios dispositivos como (ESP8266 o Raspberry Pi), además permite el análisis de los datos mediante la programación en MATLAB ® dentro de ThingSpeak, así mismo la plataforma permite crear visualizaciones personalizadas de los datos y compartir la información al público o de forma privada. Una de sus ventajas es que permite a los ingenieros y científicos crear prototipos y sistemas IoT sin configurar servidores ni desarrollar software web. Por esta razón, las principales aplicaciones que permite esta plataforma se orientan al desarrollo de prototipos IoT para el monitoreo ambiental, monitoreo de energía, agricultura inteligente, entre otras. Entre algunas de sus características principales se tiene:

- **API:** ThingSpeak dispone de una API la cual está disponible en GitHub para su descarga en un servidor propio. Es totalmente abierta, por lo que también se puede modificar su código fuente original y así contribuir a la comunidad con nuevas características.
- **Plugins:** estos permiten crear indicadores, gráficos o pantallas personalizadas de Google® utilizando HTML, CSS y JavaScript. Al igual que los canales, los plugins pueden ser públicos o privados.
- **Canales:** Los canales almacenan todos los datos que recopila una aplicación ThingSpeak, cada canal incluye ocho campos que pueden contener cualquier tipo de datos, más tres campos para datos de ubicación y uno para datos de estado.

- **API KEY:** Las claves API permiten escribir o leer datos en un canal, estas se generan automáticamente cuando se crea un nuevo canal. Este mecanismo de seguridad impide que alguien no deseado pueda enviar datos a la web y pueda falsear los datos de un sistema.

En resumen, la plataforma incluye todo lo necesario para poder empezar a trabajar desde una aplicación web con capacidad de gestionar usuarios, claves API, canales y cartografía.

6.7.3.2 Programación de la plataforma ThingSpeak.

Una vez creada la cuenta en ThingSpeak, se procede a ingresar a la plataforma y acceder al apartado “Channels”, aquí se establece la configuración del canal y se añaden las distintas variables a utilizar. La siguiente Figura 104 ejemplifica la configuración inicial del canal creado.

The screenshot shows the 'Configuración de canales' page in the ThingSpeak interface. The channel name is 'Sistema de Iluminación Inteligente' and its ID is 1641064. The page is divided into two main sections: 'Configuración de canales' on the left and 'Ayuda' on the right.

Configuración de canales:

- porcentaje completado: 30%
- Canal ID: 1641064
- Nombre: Sistema de Iluminación Inteligente
- Descripción: (Empty text area)
- Campo 1: Temperatura
- Campo 2: Humedad
- Campo 3: Luminosidad
- Campo 4: Potencia Activa
- Campo 5: Energía Consumida
- Campo 6: Factor de Potencia
- Campo 7:
- Campo 8:

Ayuda:

Los canales almacenan todos los datos que recopila una aplicación ThingSpeak. Cada canal incluye ocho campos que pueden contener cualquier tipo de datos, más tres campos para datos de ubicación y uno para datos de estado. Una vez que recopila datos en un canal, puede usar las aplicaciones de ThingSpeak para analizarlos y visualizarlos.

Configuración de canales:

- **Porcentaje completado:** Calculado en base a los datos ingresados en los distintos campos de un canal. Ingrese el nombre, la descripción, la ubicación, la URL, el video y las etiquetas para completar su canal.
- **Nombre del canal:** ingrese un nombre único para el canal de ThingSpeak.
- **Descripción:** ingrese una descripción del canal de ThingSpeak.
- **Field#:** marque la casilla para habilitar el campo e ingrese un nombre de campo. Cada canal de ThingSpeak puede tener hasta 8 campos.
- **Metadatos:** ingrese información sobre los datos del canal, incluidos los datos JSON, XML o CSV.
- **Etiquetas:** Introduzca palabras clave que identifiquen el canal. Separa las etiquetas con comas.
- **Enlace a sitio externo:** si tiene un sitio web que contiene información sobre su canal de ThingSpeak, especifique la URL.
- **Mostrar ubicación del canal:**
 - **Latitud:** especifique la posición de latitud en grados decimales. Por ejemplo, la latitud de la ciudad de Londres es 51,5072.
 - **Longitud:** especifique la posición de longitud en grados decimales. Por ejemplo, la longitud de la ciudad de Londres es -0,1275.

Figura 104 Configuración de los canales en la plataforma ThingSpeak

El siguiente paso es obtener una API Key para la conexión entre ThingSpeak y el ordenador central del sistema (Raspberry Pi). Existe una API Key para leer y otra para escribir, por lo cual la clave generada para escribir permitirá el envío de datos al canal creado.

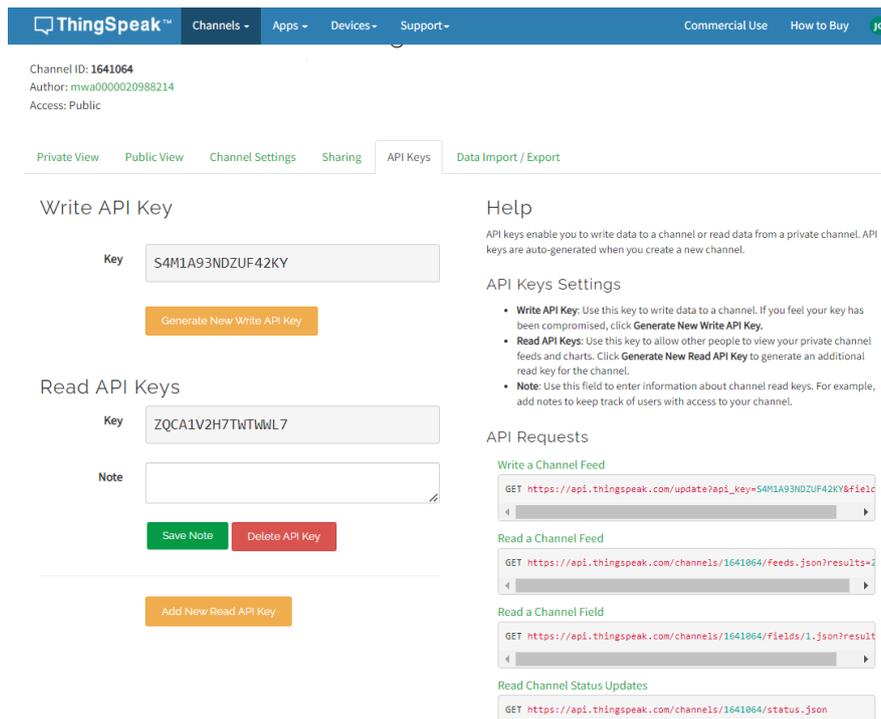


Figura 105 Obtención de API Key dentro de ThingSpeak

En otro orden de ideas, ThingSpeak permite usar el método de publicación de MQTT para actualizar un *feed* (variable) del canal y de igual forma la suscripción de MQTT para recibir mensajes cuando se actualice un canal. En este sentido, los datos que llegan al ordenador central mediante el protocolo MQTT, posteriormente son enviados a la plataforma ThingSpeak mediante la programación y ejecución de un script Python3 dentro del entorno de desarrollo integrado Thonny. El desarrollo del programa se detalla en el Anexo 15.

Finalmente, dentro de las pestañas de vista pública o privada se procede a agregar gráficas y widgets para cada *feed* o variable creada. El desarrollo de la plataforma creada para la visualización de las variables del prototipo se presenta en el Anexo 16. Puesto que la plataforma se encuentra en la nube cualquier usuario puede acceder a la vista pública del sistema accediendo a la página (<https://thingspeak.com/>) y en la sección de canales públicos, ingresar la clave “mwa0000020988214” dentro de la casilla de búsqueda por ID de usuario.

6.8 Implementación y validación del sistema de iluminación inteligente

El presente apartado tiene la finalidad de analizar la implementación práctica del prototipo realizado, como también describir las diferentes partes del módulo de pruebas construido. En este sentido, el módulo permite realizar la instalación de los sensores inteligentes para la lectura de las variables ambientales, la detección de presencia, como también permite la instalación de un circuito electrónico dentro de una carcasa plástica para realizar las funciones

de control de potencia de la iluminación, la adquisición de los datos provenientes de los sensores y la comunicación con el ordenador central del sistema.

6.8.1 Módulo de control de iluminación

En primer lugar, se describe el diseño de la carcasa junto con el hardware instalado en su interior, que consta de las siguientes partes:

-Placa PCB 1: consta de todos los componentes soldados (ver Anexo 6). Realiza las funciones de control automático del sistema, control de potencia de la luminaria, acondicionamiento de los datos provenientes de los sensores BH1750, PIR, e implementa la comunicación inalámbrica bidireccional con el ordenador central. El código programado en el primer dispositivo NodeMCU ESP8266 se puede apreciar detalladamente en (ver Anexo 13).

-Placa PCB 2: esta segunda placa utiliza el mismo diseño elaborado (ver Anexo 6), no obstante, implementa únicamente el montaje de un dispositivo NodeMCU. La razón de incluir esta segunda placa es la realización de las funciones de lectura del sensor DHT22, lectura de los parámetros eléctricos de la luminaria mediante el dispositivo medidor PZEM004T y de modo similar transmitir la información al ordenador central del sistema. El código programado en el segundo dispositivo NodeMCU se puede apreciar detalladamente en (ver Anexo 14).

En este sentido la siguiente Figura 106 muestra la elaboración y montaje de los componentes antes mencionados, para posteriormente ser ubicados sobre la plataforma del módulo de pruebas construido. La estructura plástica de la carcasa permite colocar las placas PCBs atornilladas a la superficie, de forma que exista un aislamiento eléctrico entre ambas para la prevención de cualquier contacto, además, su superficie con orificios permite no bloquear las señales de la antena WiFi del ESP8266. Cabe destacar que el sistema requiere como entrada una conexión a la red eléctrica con suministro 120 V_{AC}, y proporciona como salida un enchufe eléctrico tipo hembra para la conexión directa de la carga (luminaria implementada).



Figura 106 Construcción y montaje del hardware del prototipo

Nota. Placa PCB 1 y PBC 2 de izquierda a derecha respectivamente.

En segundo lugar, se describe el módulo de pruebas construido sobre el cual se monta la carcasa principal del prototipo vista anteriormente. Este módulo de pruebas permite la instalación de los componentes de la luminaria, el cableado eléctrico, y su sujeción en el techo de cualquier superficie previo a su adecuamiento respectivo. Se puede señalar que, para el desarrollo de las pruebas del presente trabajo se configuró un arreglo de 4 focos LED conectados paralelamente, los cuales se enchufan directamente al toma corriente tipo hembra de la figura anterior, esta conexión cierra el circuito y permite suministrar la cantidad de energía necesaria a la carga mediante el circuito de control de potencia, tal como se indicó en la descripción del diseño electrónico del prototipo (ver Figura 40). La siguiente Figura 107 muestra el diseño del módulo de pruebas construido, donde se puede apreciar la colocación del sensor PIR y sensor luxómetro en el centro del módulo para permitir su funcionamiento adecuado, de modo similar el módulo permite la instalación del sensor DHT22 para la lectura en tiempo real de las variables ambientales del lugar de funcionamiento. En este sentido, los sensores son cableados hasta el interior de la carcasa plástica y posteriormente conectados a los pines digitales del MCU por medio de las regletas especificadas en el diseño PCB.



Figura 107 Módulo de pruebas del prototipo de sistema inteligente de iluminación

6.8.2 Pruebas de funcionamiento del módulo de control de iluminación

Una vez descrito el hardware implementado dentro del módulo de control de iluminación, el siguiente apartado presenta la validación práctica del funcionamiento del prototipo, para lo cual se analizan diversos escenarios o configuraciones del nivel de iluminación deseado. Así pues, se analiza la respuesta del sistema en función de parámetros eléctricos como la potencia consumida, el factor de potencia y el nivel de iluminación alcanzado en base al valor directamente establecido desde la interfaz de usuario web, accesible desde cualquier dispositivo inteligente dentro de la red local, con la dirección (<http://192.168.1.107:1880/ui>). Para cumplir con este propósito se consideró la realización de pruebas experimentales y el almacenamiento de los resultados dentro de la base de datos del prototipo, considerando dos escenarios diferentes de funcionamiento como se detallará a continuación.

6.8.2.1 Pruebas del módulo de control de iluminación en horas nocturnas.

La siguiente Figura 108 analiza el primer escenario de pruebas, en este caso se estableció el nivel de ajuste de forma manual (a través del widget deslizante del panel de control de cargas), tal que exista un nivel bajo de luminosidad dentro del área de funcionamiento. Cabe destacar que para el desarrollo de estas pruebas se consideró en primer lugar un escenario en horas nocturnas, de modo que el funcionamiento del sistema pueda ser analizado de mejor manera.

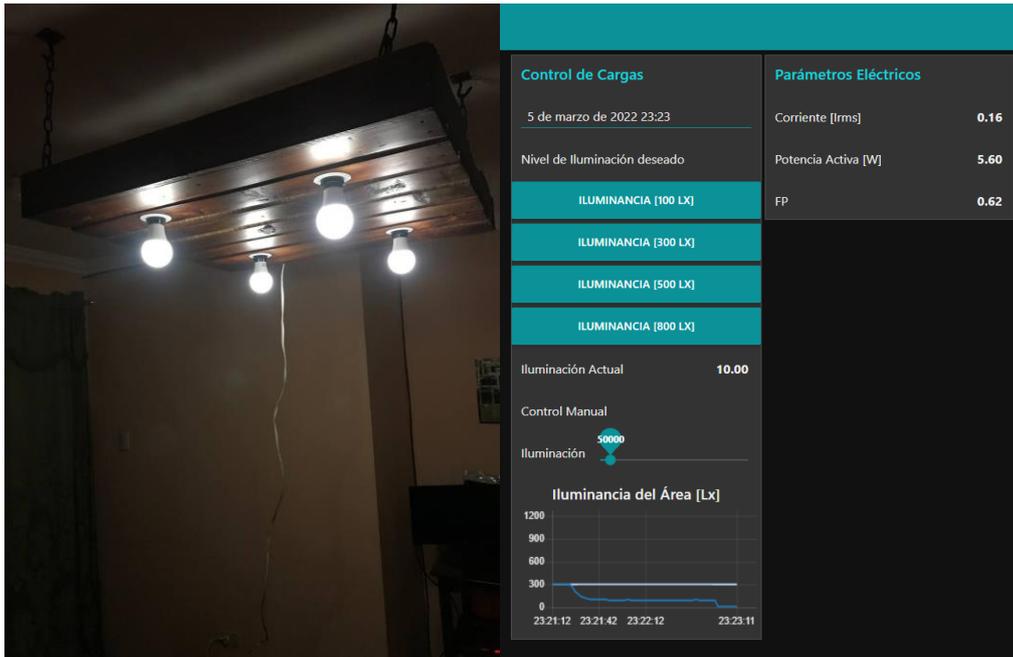


Figura 108 Respuesta del sistema de iluminación en horas nocturnas; Configuración de baja iluminación

Como resultado la imagen ejemplifica un escenario con un nivel de luminosidad de 10 luxes, por otra parte, de la interfaz de usuario se obtiene algunos parámetros eléctricos del funcionamiento del prototipo, tales como: Corriente $I_{RMS}=0,16$ A, $P=5,60$ Vatios y un FP de 0,62.

De modo similar, el segundo escenario ejemplifica una configuración del nivel de luminosidad en 100 luxes, en el cual el sistema mantiene un bajo factor de potencia debido a la poca cantidad de energía suministrada a la luminaria. Cabe destacar que, de aquí en adelante se realiza la configuración del nivel de iluminancia desde los botones del panel de control de cargas dentro de la interfaz de usuario UI.

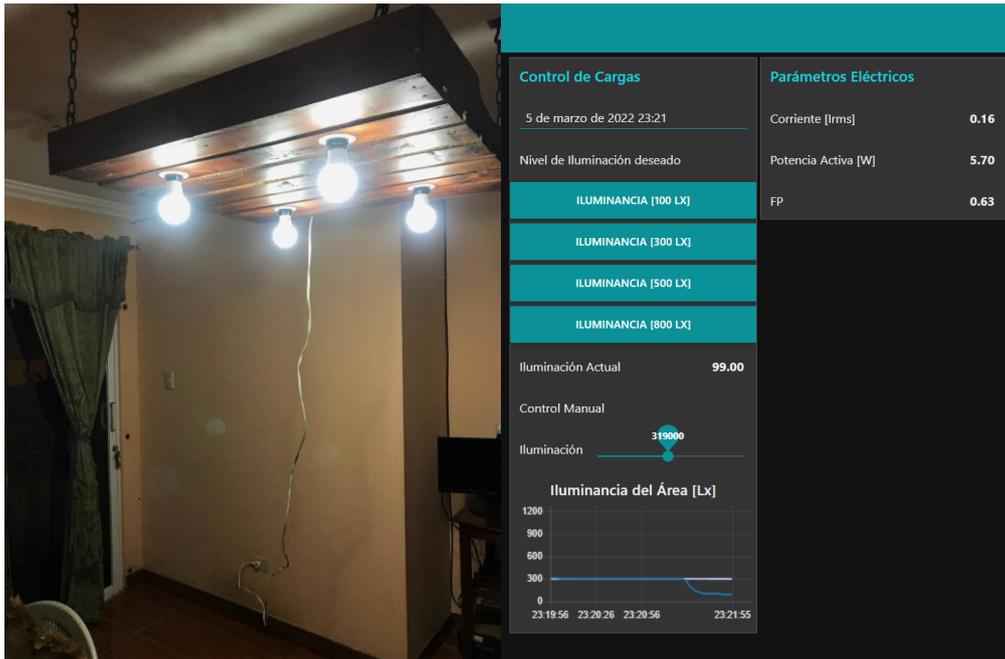


Figura 109 Respuesta del sistema de iluminación en horas nocturnas; Nivel establecido en 100 Lx

En otro escenario se puede analizar de la Figura 110 que el sistema permite llegar satisfactoriamente al nivel de iluminación configurado en 300 luxes, así mismo, de los parámetros eléctricos se verifica el aumento de los valores de potencia activa y corriente a medida que el sistema proporciona más energía a la luminaria. De acuerdo al análisis del factor de potencia en función del circuito de atenuación, se evidencia un aumento significativo del FP llegando casi a su valor nominal de 0,9 para los focos LED implementados.

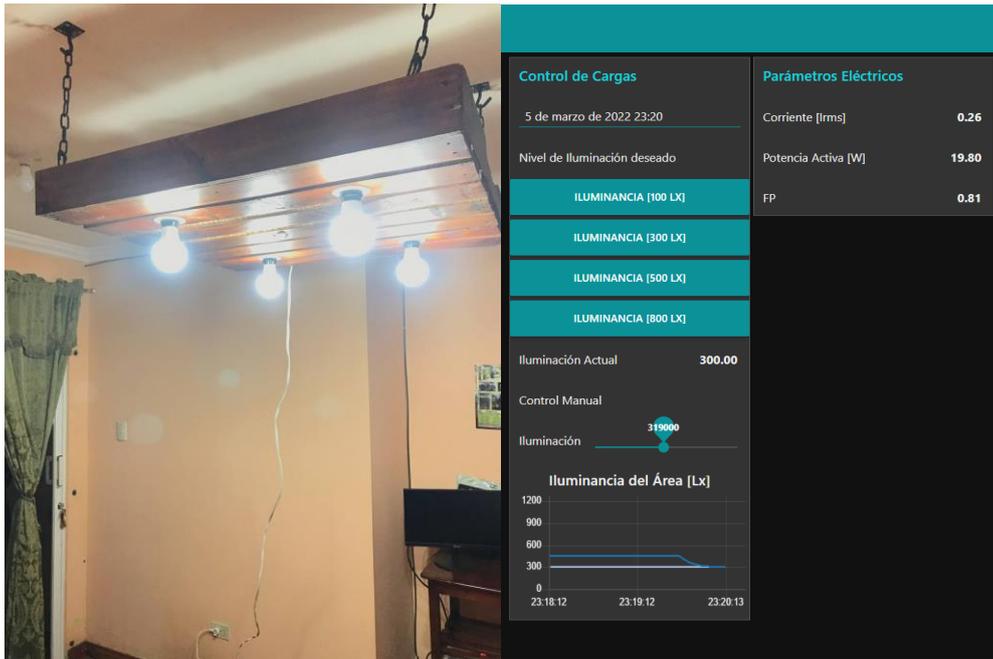


Figura 110 Respuesta del sistema de iluminación en horas nocturnas; Nivel establecido en 300 Lx

En última instancia se configura el sistema a un nivel de iluminancia de 500 luxes. De la Figura 111 se puede analizar que el nivel de iluminación alcanzado permanece constante con un valor de 467 luxes, no obstante, no llega al nivel deseado. Esto es debido a que el sistema está en su capacidad máxima de iluminación, el voltaje de entrada a la luminaria es aproximadamente el mismo de la red eléctrica 120 V_{AC} y el factor de potencia de la carga es de 0,9 aproximadamente. Así mismo se constata que para una carga de 4 focos LED con un consumo individual de 8,3 W obtenido en las pruebas anteriores (ver apartado 6.2.8) se obtiene una luminaria que, en su configuración sin atenuación consume una potencia activa de aproximadamente 33 vatios y una corriente de 300 mA.

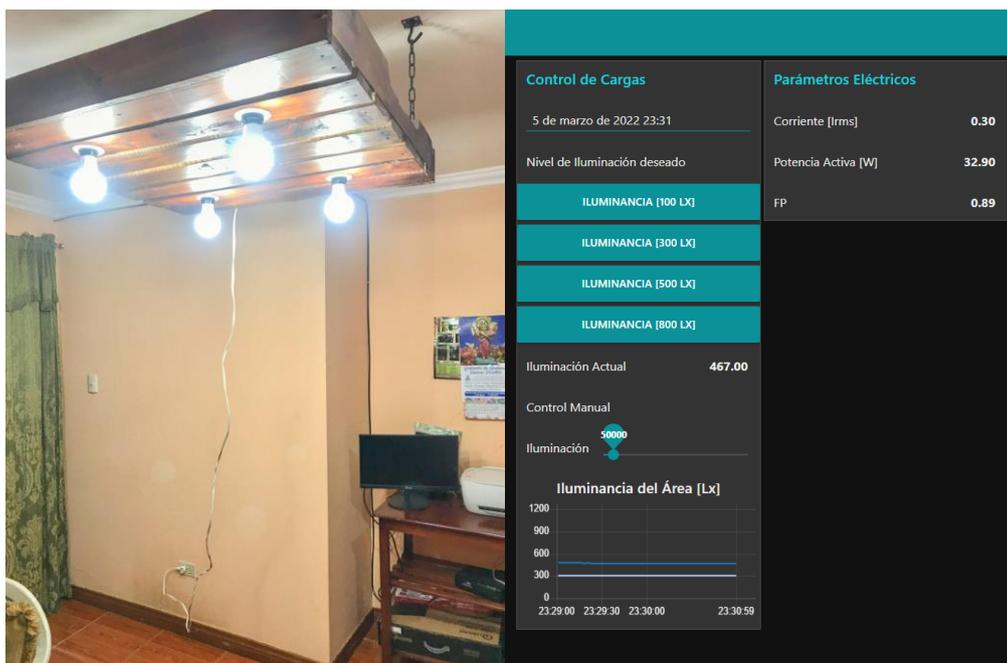


Figura 111 Respuesta del sistema de iluminación en horas nocturnas; Nivel establecido en 500 Lx

En este sentido el sistema puede alcanzar una capacidad mayor de iluminación con la adición de más focos LED conectados paralelamente a la carga del prototipo y colocados en diferentes lugares dentro de la zona de funcionamiento. Si bien en el desarrollo de las pruebas realizadas se configuró una carga con 4 focos LED de la misma marca (ver Anexo 12), el sistema permite el control de una carga de mayor consumo energético, con el límite impuesto por la cantidad de corriente máxima de drenaje I_D permitida en los dispositivos MOSFET utilizados (ver apartado 5.1.9) cuando estos se encuentran en conducción, la cual según el *datasheet* del fabricante especifica un valor máximo de 8 Amperios con un voltaje $V_{GS} = 10\text{ V}$ a 25°C , y una potencia máxima de disipación de 125 W a igual temperatura de 25°C , lo cual indudablemente permite aumentar el número de focos LED instalados en el sistema.

6.8.2.2 Pruebas del módulo de control de iluminación en horas diurnas.

El segundo escenario de pruebas considera un análisis experimental del módulo con la presencia de luz natural dentro de la zona de trabajo. De la misma manera, la siguiente Figura 112 ejemplifica un escenario de trabajo del sistema para un nivel de iluminación establecido en el panel de control de cargas de 100 luxes. De la prueba se puede inferir que el ambiente de trabajo posee una luminancia de 133 luxes sin el encendido de la luminaria del prototipo, ya que, en virtud de que el nivel de iluminación natural excede el nivel de iluminación requerido, el controlador no proporciona energía a la luminaria y esta se encuentra en estado apagado.

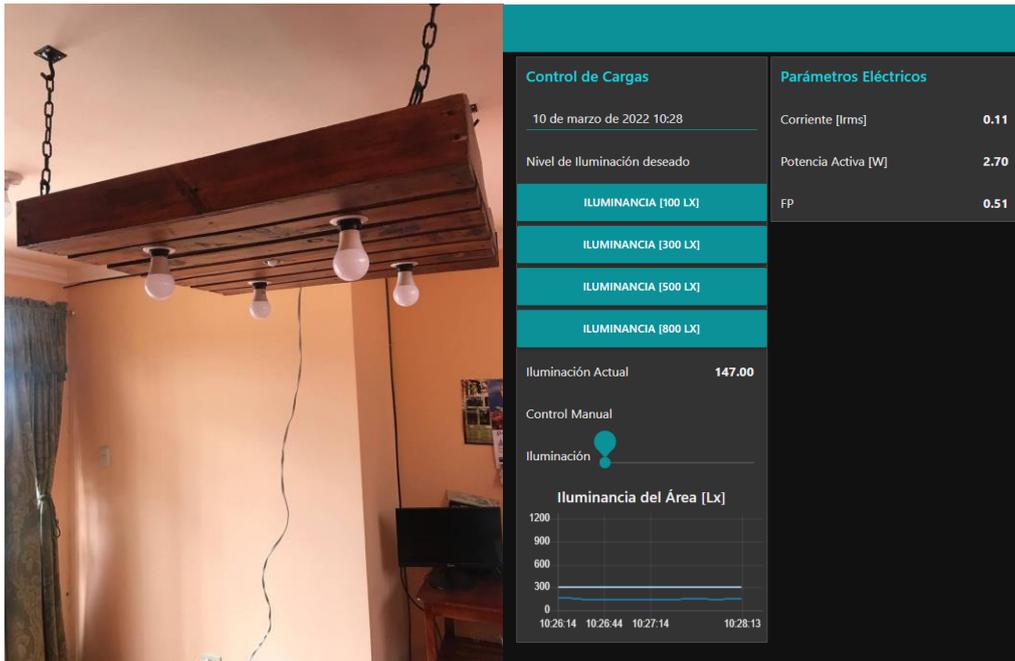


Figura 112 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 100 Lx

Posteriormente se configura el sistema para obtener un nivel de iluminación de 300 luxes. De la Figura 113 se puede observar que el valor de potencia activa P es de 9,2 Vatios, por debajo de los 19,8 Vatios requeridos para obtener el mismo nivel de iluminación en horas de la noche, lo cual indica el ahorro de energía en función de la luz natural presente en la zona.

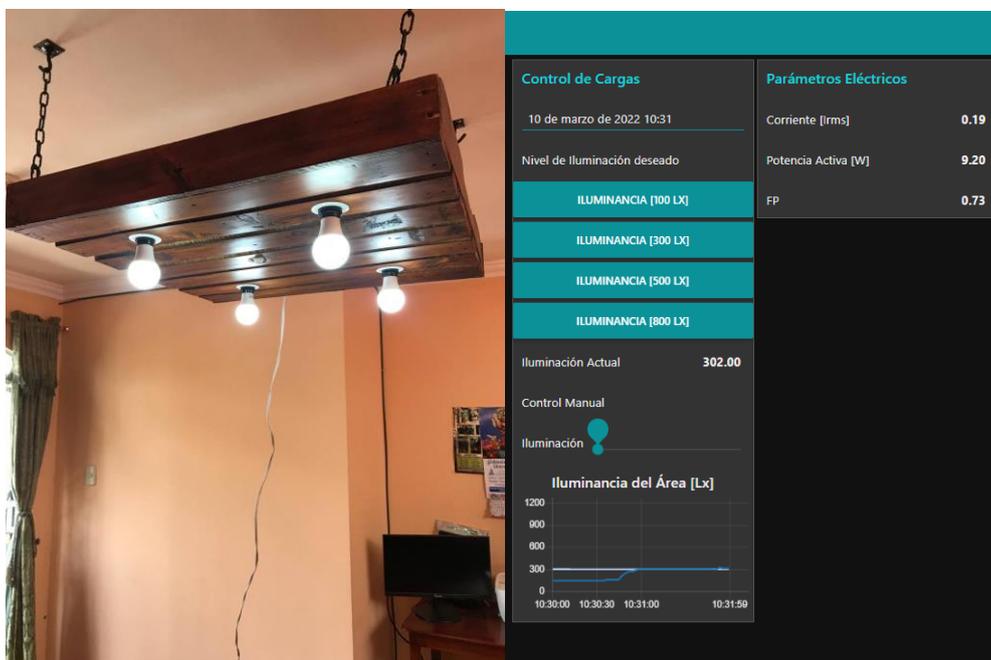


Figura 113 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 300 Lx

De igual modo, el siguiente escenario muestra una configuración del valor de iluminación establecido en 500 luxes. De la Figura 114 se evidencia que el sistema logra

mantener el nivel de iluminación requerido, lo cual resalta la capacidad de iluminación del prototipo en horas del día. Como consecuencia el sistema trabaja en una configuración de elevado consumo de energía activa de aproximadamente 22,8 Vatios lo cual permite obtener un factor de potencia cercano a 0,83.

Finalmente, el último escenario de pruebas ejemplifica un nivel de iluminación requerido de 800 luxes. De los resultados obtenidos en la Figura 115 se evidencia que el sistema de iluminación no alcanza el nivel programado, llegando a un límite de iluminación máximo de 632 luxes. Al igual que el caso analizado con anterioridad, el sistema en su máxima capacidad de iluminación proporciona un alto desempeño energético, logrando aprovechar casi toda la energía suministrada por la red eléctrica, no obstante, el sistema posee un alto consumo energético. Por su parte, el sistema permite lograr ahorros energéticos-económicos debido al sistema de detección de presencia, que apaga las luminarias en ausencia de personas dentro del área de funcionamiento.

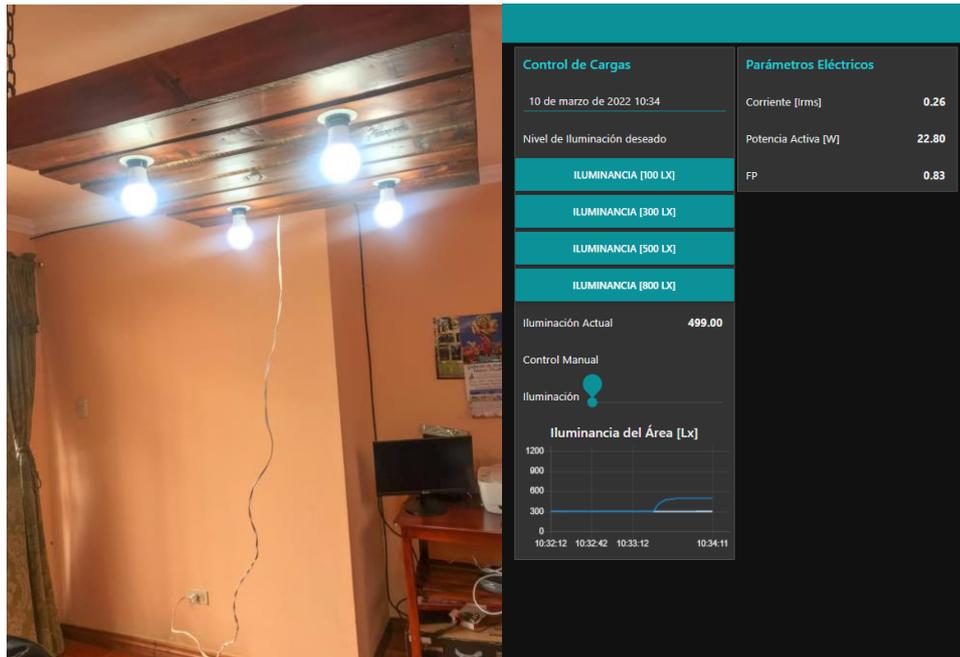


Figura 114 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 500 Lx

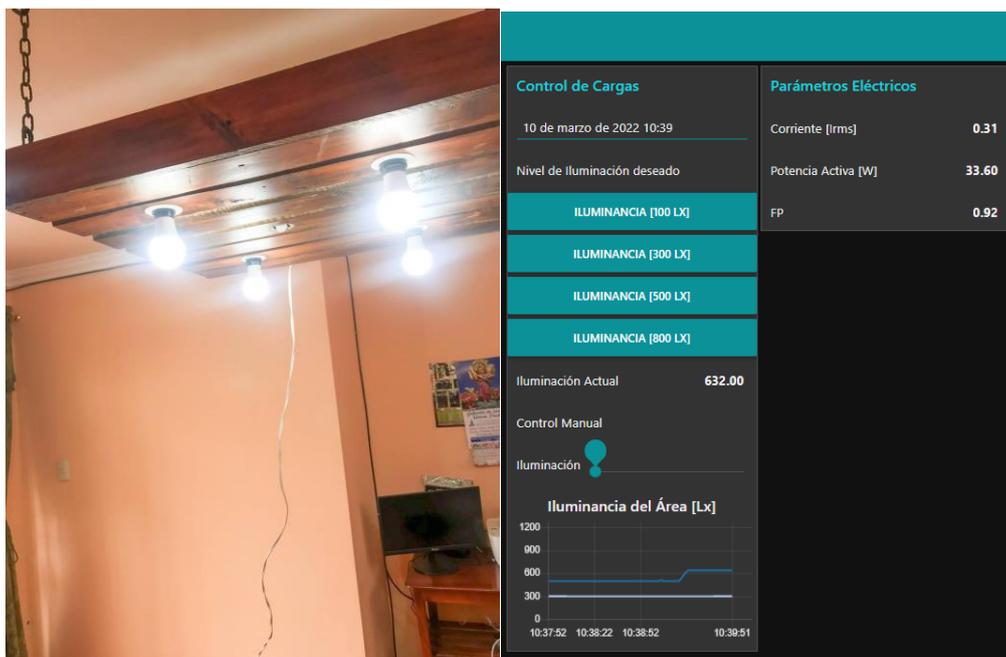


Figura 115 Respuesta del sistema de iluminación en horas diurnas; Nivel establecido en 800 Lx

6.9 Análisis de la reducción del consumo energético-económico

Como se mencionó inicialmente el consumo energético se define como la cantidad de energía que gasta un dispositivo para la realización de una actividad programada. El presente apartado analiza la capacidad de reducción del consumo energético por parte del prototipo elaborado, tomando como partida los datos obtenidos de las diferentes pruebas realizadas por el sistema en varios escenarios de funcionamiento, para finalmente por medio de tablas y diagramas realizar un análisis exhaustivo del comportamiento del prototipo.

Fundamentalmente en el desarrollo del trabajo se ha realizado un análisis de diversas tecnologías de iluminación, destacando principalmente la tecnología de estado sólido LED, la cual además de sus ventajas relacionadas al ahorro de consumo en relación a los sistemas de iluminación tradicionales, permite la posibilidad de control y con ello reducir más su consumo aumentando la eficiencia y el confort del consumidor. Por su parte, el prototipo elaborado facilita el monitoreo y facturación de la energía consumida en iluminación dentro del área de funcionamiento, esto posibilita acceder a información valiosa y precisa sobre el consumo energético, y el tiempo de encendido de la luminaria el cual es un elemento de gran importancia para la reducción de los costos de energía, ya que permite tomar acciones que beneficien la reducción del consumo y por ende las emisiones contaminantes al medio ambiente.

Para cumplir con este análisis se descompone la capacidad de reducción del consumo energético en dos partes, que en conjunto, permiten obtener una medida aproximada del valor de ahorro energético como se detallará a continuación.

6.9.1 Ahorro energético mediante el sistema de detección de presencia

El sistema de detección de presencia implementado con la ayuda del sensor PIR permite reducir las horas de encendido de la luminaria y en consecuencia los kWh consumidos al final del día. Para analizar esta facultad del prototipo se toma en consideración los datos obtenidos y almacenados en la base de datos local, específicamente los datos de la duración de encendido de la luminaria para distintos días de trabajo del sistema. En este sentido, el sistema se analiza independientemente del nivel de ajuste de la intensidad de iluminación, ya que este analiza la duración de encendido del sistema, mas no la capacidad de control de potencia, lo cual es el propósito del siguiente apartado.

De esta forma se presenta la Tabla 22, la cual resume información obtenida de la base de datos local para distintos días de funcionamiento del prototipo. Para ejemplificar el proceso de cálculo del ahorro energético, se toma como partida la fecha del “24/2/2022”. De la base de datos se obtiene que la duración de encendido total del sistema se determinó en “3h. 21min. 32seg.”, por su parte, el tiempo de encendido total del sensor PIR y por ende la luminaria es de “2h. 57min. 19seg.” o un total de 2,96 horas aproximadamente, dando como resultado un tiempo de apagado de la luminaria de 24 minutos o 0,41 horas. En base a las lecturas realizadas y almacenadas en la base de datos, se obtuvo una potencia activa promedio de 25 Vatios para la fecha indicada, lo que quiere decir que el valor de Wh ahorrados ese día corresponde a la siguiente expresión:

$$Wh_{consumo} = Potencia\ activa\ en\ W * horas\ de\ consumo = 25 * 2,9 = 72,5\ Wh$$

$$Wh_{ahorro} = Potencia\ activa\ en\ W * horas\ de\ consumo\ ahorradas = 25 * 0,41 = 10,25\ Wh$$

De esta manera la siguiente Figura 116 ilustra la cantidad de Watt-hora horrados para los días analizados.

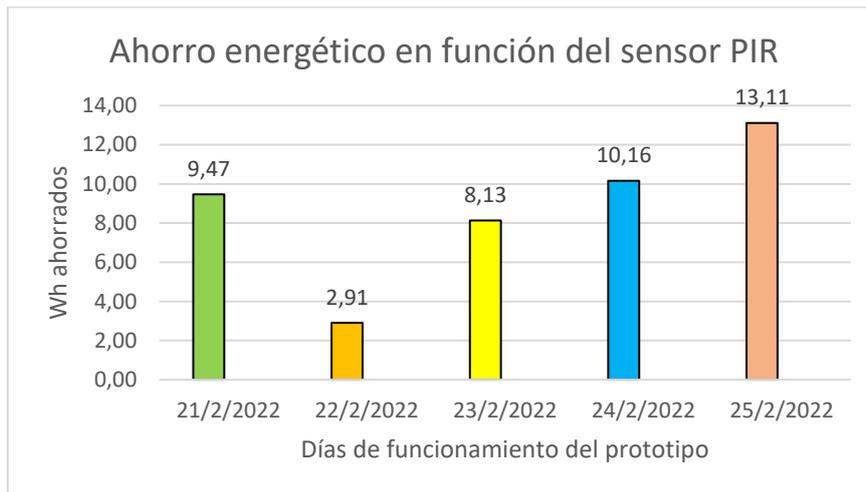


Figura 116 Ahorro energético de Wh, en función del sensor PIR

Nota. Los valores varían en función del tiempo de ahorro, y la potencia activa promedio de la luminaria para cada día analizado

6.9.2 Ahorro energético mediante el sistema automático de control de potencia

El sistema automático de control de potencia permite suministrar a la carga la potencia necesaria para mantener un nivel de iluminación adecuado, de esta forma el sistema maximiza el ahorro de energía en base a la luz incidente en el área de trabajo, como también en función de la presencia de personas. De modo idéntico, para analizar esta facultad del prototipo, se toma en consideración la información almacenada en la base de datos y así permitir su visualización y análisis mediante la Tabla 23.

Para ejemplificar nuevamente el proceso de cálculo del ahorro energético se toma como partida la misma fecha del “24/2/2022”. De la base de datos se obtiene que el tiempo de encendido total de la luminaria es de “2h. 57min. 19seg.” o un total de 2,96 horas aproximadamente, también se extraen los valores de consumo inicial y final en kWh correspondientes a “0,182” y “0,253” respectivamente, lo cual permite calcular los Vatios-hora consumidos en la fecha indicada, dando un valor de 71 Wh.

$$Wh = (kWh_{fin} - kWh_{inicio}) * 1000 = (0,253 - 0,182) * 1000 = 71\ Watts - hora$$

Posteriormente se calcula la potencia activa promedio, y el error porcentual en referencia al valor de potencia activa promedio obtenido de las mediciones almacenadas en la base de datos (25 W), esto es:

$$W_{prom_calc} = \frac{Wh_{consumidos}}{horas_{encendido}} = \frac{71}{2,96} \approx 24$$

$$E_p = \left| \frac{V_V - V_A}{V_V} \right| * 100$$

Donde:

- E_p es el error porcentual.
- V_V es el valor verdadero.
- V_A es el valor aproximado (Valor DB).

$$E_p = \left| \frac{24 - 25}{24} \right| * 100 = 4\%$$

Este valor de E_p del 4% indica la exactitud de los valores obtenidos referentes a la potencia activa promedio, la cual es obtenida diariamente de la base de datos, al igual que el valor de factor de potencia promedio que permite analizar la eficiencia del prototipo. En otro orden de ideas, el sistema de control de potencia en su configuración sin atenuación de la luminaria consume una potencia activa o real de 33 Vatios con un voltaje de suministro de 120 V_{AC}, lo cual permite calcular el valor de ahorro energético del sistema en función del número de horas de encendido para cada día de funcionamiento. Para cumplir con este propósito se calcula el ahorro en Vatios producto de la dimerización de la luminaria, por consiguiente, se restan los valores de potencia activa determinados de la siguiente manera:

$$W_{ahorro} = W_{carga} - W_{prom_calc} = 33 - 24 = 9 \text{ Watts}$$

Finalmente, el ahorro energético en Wh para la fecha indicada se calcula mediante la siguiente expresión:

$$Wh_{ahorro} = W_{ahorro} * horas_{encendido} = 9 * 2,96 = 26,64 \text{ Wh}$$

Cabe destacar que los valores obtenidos de Wh de ahorro diario, varían de acuerdo al nivel de iluminación configurado dentro de la interfaz de usuario web, la cantidad de iluminación natural presente y el tiempo de encendido de la luminaria en función del sensor de presencia. Por otra parte, de la Tabla 23 se puede inferir que a medida que aumenta el ahorro

energético en función del sistema de control de potencia, se disminuye el factor de potencia del sistema, esto como consecuencia directa de la dimerización de la luminaria, no obstante, los valores del FP promedio se mantienen dentro de un rango que no suponen un deterioro significativo en relación a su valor nominal (0,9 para la luminaria implementada).

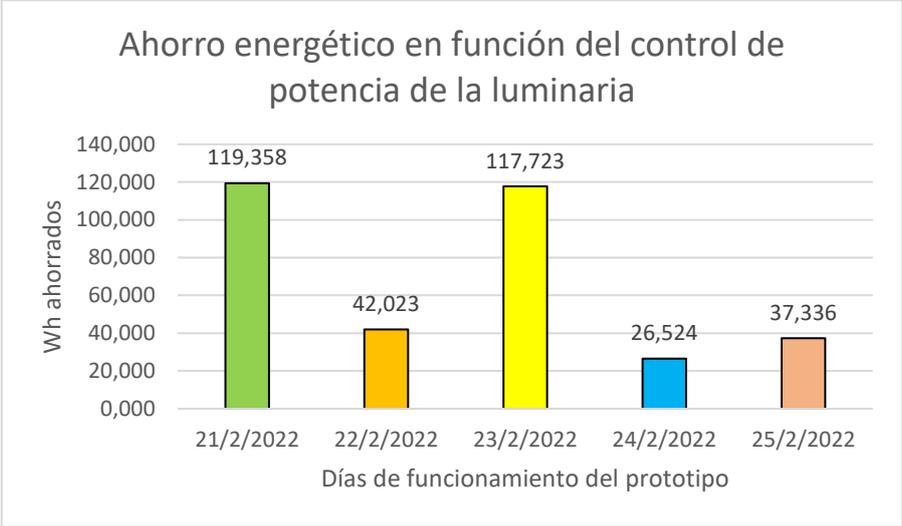


Figura 117 Ahorro energético en función del control de potencia de la luminaria

Tabla 22 Ahorro de Wh en función del sensor PIR

Fecha	Hora inicio	Hora fin	Tiempo Transcurrido	Duración de encendido Luminaria	Potencia Activa Promedio en Vatios (Valor DB)	Tiempo ahorro	Potencia Activa Promedio en Vatios (Valor DB)	Ahorro de Wh, en función del control de presencia
21/2/2022	14:48:30	20:48:30	6:00:00	4:38:50	4,65	1:21:10	1,35	7,00
22/2/2022	20:45:21	22:45:21	2:00:00	1:30:57	1,52	0:29:03	0,48	6,00
23/2/2022	9:26:35	16:22:02	6:55:27	6:24:57	6,42	0:30:30	0,51	16,00
24/2/2022	9:12:42	12:34:14	3:21:32	2:57:19	2,96	0:17:13	0,41	25,00
25/2/2022	9:20:41	13:04:20	3:43:39	3:07:53	3,13	0:35:46	0,60	22,00

Tabla 23 Ahorro de Wh en función de la atenuación de la luminaria

Fecha	Duración de encendido Luminaria	kWh Inicio	kWh Fin	Wh consumidos	Potencia Activa promedio en Vatios (Valor calculado)	Potencia Activa promedio en Vatios (Valor DB)	Error	FP promedio (Valor DB)	Vatios Reales de la carga	Ahorro en Vatios	Ahorro de Wh en función del control de potencia	
21/2/2022	4:38:50	4,65	0,006	0,040	34,00	7,32	7,00	5%	0,69	33	25,68	119,35
22/2/2022	1:30:57	1,52	0,062	0,070	8,00	5,28	6,00	12%	0,65	33	27,72	42,02
23/2/2022	6:24:57	6,42	0,083	0,177	94,00	14,65	16,00	8%	0,76	33	18,35	117,72
24/2/2022	2:57:19	2,96	0,182	0,253	71,00	24,02	25,00	4%	0,83	33	8,98	26,52
25/2/2022	3:07:53	3,13	0,254	0,320	66,00	21,08	22,00	4%	0,83	33	11,92	37,33

6.9.3 Ahorro energético-económico total del prototipo

A fin de analizar el ahorro energético total del prototipo elaborado, se toman los resultados obtenidos de las dos partes detalladas anteriormente. En consecuencia, se obtiene el ahorro energético total para cada día de funcionamiento del prototipo. Finalmente, de acuerdo a (CELEC EP, 2021) se toma como referencia la tarifa promedio a nivel nacional de 9,20 cUSD/kWh, de este modo se obtiene en la última columna de la Tabla 24, el valor de ahorro económico aproximado por la implementación del prototipo para cada una de las fechas analizadas.

Tabla 24 Ahorro energético-económico total del prototipo

Fecha	Wh consumidos	Wh ahorrados en función del sensor PIR	Wh ahorrados en función del control de potencia	Ahorro total del prototipo en kWh	FP promedio	Ahorro Económico en centavos
21/2/2022	34	9,47	119,35	0,129	0,69	1,185
22/2/2022	8	2,91	42,02	0,045	0,65	0,413
23/2/2022	94	8,13	117,72	0,126	0,76	1,158
24/2/2022	71	10,09	26,52	0,037	0,83	0,337
25/2/2022	66	13,11	37,33	0,050	0,83	0,464

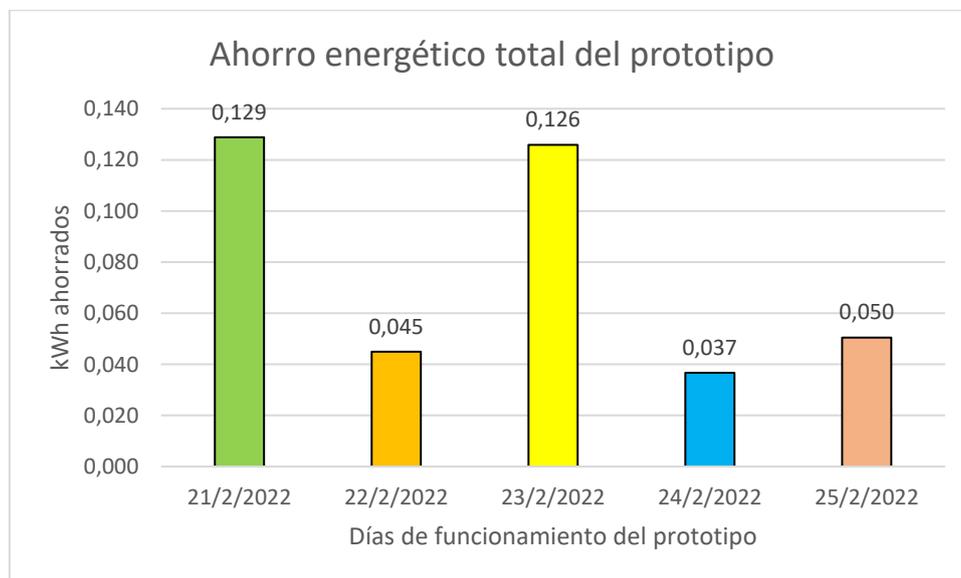


Figura 118 Ahorro energético total por la implementación del prototipo

De acuerdo a los resultados obtenidos, se procede de modo similar a realizar la comparativa con los valores proporcionados mediante la interfaz de usuario web. De la Figura 119 se observa que las gráficas presentadas en la UI se asemejan con exactitud a las obtenidas

en el análisis matemático, en otras palabras, el usuario posee la facultad de realizar un análisis exhaustivo del consumo energético del sistema, con el único trabajo de escoger la fecha para el análisis y visualización de los resultados.



Figura 119 Interfaz Datos energía, visualización del ahorro energético

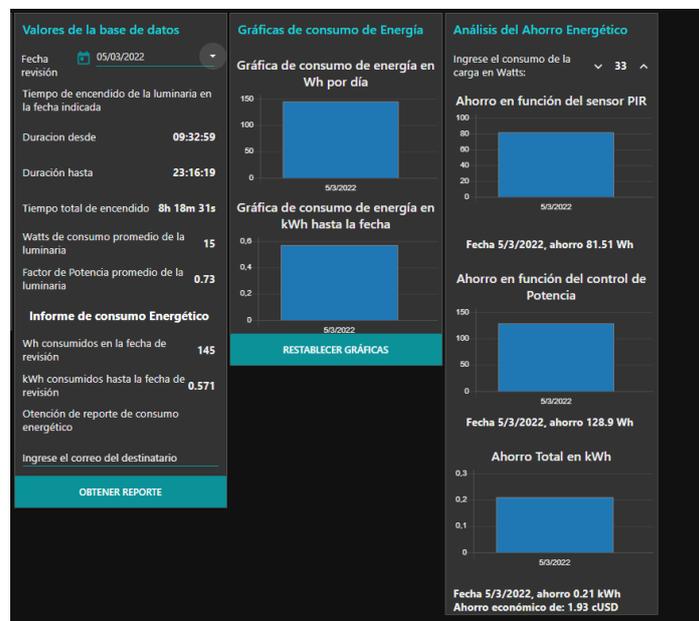


Figura 120 Análisis del consumo energético para un día entero de funcionamiento del prototipo

7 Discusión

Como se ha venido diciendo, la intención del presente trabajo de titulación es la de innovar en el terreno de los sistemas de iluminación inteligentes y eficientes energéticamente, asimismo, aprovechar los conocimientos teóricos como también la tecnología actual para el desarrollo de un prototipo económico que se fundamenta en las técnicas ya desarrolladas en otras situaciones donde su aplicación pueda ser de interés. Se ha seguido un enfoque mixto, basado en la búsqueda de soluciones que ha permitido el desarrollo final del sistema propuesto. De este modo, se pueden mencionar los siguientes hitos como resultados más destacados del presente trabajo.

Del estudio de relés de estado sólido abordado en (Elliott, 2019), se mencionó que un relé MOSFET permite una elevada velocidad de conmutación, capacidad de controlar cargas con un consumo desde unos pocos miliamperios hasta varios amperios, como también la virtud de poseer pocas pérdidas de potencia debido a su baja resistencia interna, lo cual permitió contrastar la teoría con los resultados obtenidos en las pruebas experimentales del circuito de control de potencia (ver apartado 6.2.7) basado en la implementación de un relé AC mediante el uso de dos MOSFET de conmutación de canal N. En este sentido, al comparar los resultados de las señales obtenidas directamente del osciloscopio, con las analizadas en el estudio de los atenuadores de control de fase de borde posterior (ver apartado 4.10.2), se corroboró el correcto funcionamiento del sistema desarrollado para el control de potencia de la luminaria.

Por su parte, el microcontrolador NodeMCU ESP8266 seleccionado para la generación de la señal de activación del relé (señal PWM), permite un óptimo funcionamiento del circuito electrónico, en gran parte debido a su alta capacidad de procesamiento, que sumado a las funciones integradas de Wi-Fi, permitió el cumplimiento del primer objetivo específico abordado: ‘Emplear las tecnologías de microcontroladores y sistemas de desarrollo embebido *open source*, en un sistema de iluminación inteligente’. Además, la placa de desarrollo NodeMCU posibilita la programación en la plataforma de Arduino IDE, permitiendo aprovechar la extensa cantidad de librerías, y con ello lograr una reducción en la complejidad del algoritmo que permita el correcto funcionamiento del sistema.

Ahora bien, en base a la precisión planteada para el funcionamiento del sistema automático de iluminación, tanto en el ámbito doméstico como también empresarial, se realizó un análisis bibliográfico de trabajos relacionados a la automatización y desarrollo de sistemas

de control embebido. En consecuencia, se tomó como partida el trabajo desarrollado por (Huamán, 2017), el cual concluye que un sistema de control automático de iluminación basado en un controlador PID, permite un buen desempeño del proceso automatizado logrando llevar el nivel de iluminación al valor deseado de forma rápida y sin la generación de oscilaciones.

Como producto se desarrolló un sistema de control automático, partiendo del análisis del modelo matemático de la planta, lo cual permite realizar y validar mediante simulación el ajuste de los parámetros del controlador PID (ver apartado 6.4.3.2). En tal sentido, el ajuste del controlador implicó el cálculo adecuado de las ganancias K_p , K_i , K_d , lo cual fue posible gracias al uso de software matemático y la simulación de la respuesta del sistema en estado estacionario. Una vez calculados los valores de ganancias adecuados, se realizó la programación del sistema de control automático para su posterior validación por medio de un enfoque cuantitativo de las pruebas experimentales del funcionamiento del sistema, y cuyos resultados se plantean detalladamente en el apartado 6.4.6. De esta forma se logró dar cumplimiento al segundo objetivo específico planteado: ‘Diseñar e implementar un sistema que permita controlar automáticamente la intensidad de iluminación.’

Una vez planteados los requisitos de funcionamiento del sistema de comunicación inalámbrico, basado en una red de sensores y actuadores WSN, con la Raspberry Pi como ordenador central del sistema (ver apartado 6.5.1), se procedió a realizar la programación para la comunicación y envío de datos entre las dos partes, tomando en consideración ciertas medidas de seguridad y funcionamiento programadas en el bróker MQTT del sistema. De acuerdo con esto, el mecanismo de seguridad implementado requiere de un proceso de autenticación (usuario/contraseña) de los clientes NodeMCU previo al intercambio de información con el bróker, además, el nivel de calidad de servicio QoS para el protocolo se definió en un valor de 2, si bien este valor escogido incrementa la sobrecarga del sistema, las pruebas de funcionamiento arrojaron un resultado favorable, permitiendo así incrementar la confiabilidad del sistema.

Similar al trabajo desarrollado por (Gaona, 2019) el cual implementa un gestor de base de datos para un sistema de monitoreo de variables ambientales dentro de áreas críticas, en el presente trabajo se consideró la implementación de un servidor LAMP (ver apartado 4.20.3) en el ordenador central. Por su parte, para el almacenamiento y posterior realización de consultas y manipulación de la información, se debe tener cierto grado de conocimiento en lo referente al lenguaje SQL, en consecuencia, se recomienda el uso de la interfaz de phpMyAdmin para la administración de la base de datos y realización de operaciones.

Para dar cumplimiento al tercer objetivo específico abordado ‘Diseñar e implementar un sistema de monitoreo en tiempo real *indoor*, de humedad, temperatura, luminosidad, y plasmar la información en una aplicación web’, se analizó el uso de la herramienta Node-RED (ver apartado 4.20.2), la cual permite un entorno de programación de aplicaciones amigable y de alto potencial. Al tratarse de una herramienta de software libre posee una gran comunidad de usuarios que crean y ponen a disposición nuevos nodos para el uso de los desarrolladores, por otra parte, existe una gran cantidad de documentación y foros al alcance a través de su página web oficial (<https://nodered.org/docs/>). En este sentido, el desarrollo de la programación basada en flujo, permitió el uso de los nodos (mqtt in / mqtt out) para el intercambio de información entre el bróker MQTT y los clientes NodeMCU, además, mediante el uso del nodo *Mysql* se logró tener pleno acceso a la base de datos almacenada en el ordenador central, y la realización de consultas mediante comandos SQL previamente definidos a la salida de los nodos *function*. Por su parte, mediante el uso de los nodos pertenecientes a la lista *dashboard*, se realizó el diseño de la interfaz de usuario web, para la visualización de los parámetros requeridos en el sistema de monitoreo en tiempo real.

La ventaja de emplear la herramienta Node-RED es la gran capacidad de gestión de la información, como también, la facilidad de adaptación del sistema a los requisitos de funcionamiento. Otra de las ventajas que se observó, es la alta escalabilidad del sistema, lo cual, mediante la programación y la adición de nodos dentro de los flujos de programa ya analizados anteriormente (ver apartado 6.7.1), se lograría agregar más dispositivos sensores y actuadores dentro de la red WSN, y con esto, permitir el monitoreo y control de la iluminación de más áreas dentro de una edificación.

A diferencia del trabajo elaborado por (Manotoa, 2018), en el cual se calcula la potencia consumida por una luminaria mediante el uso de un sensor de efecto hall ACS712 para la medición de la corriente, en el presente trabajo se optó por el uso del medidor PZEM004T (ver apartado 5.1.7), el cual, permite realizar una lectura no invasiva de la potencia activa consumida por la luminaria LED, como también analizar el factor de potencia, la energía consumida, entre otros parámetros. La ventaja del uso de este medidor es su alta sensibilidad y la certeza de realizar las mediciones con 1 grado de precisión según la información del fabricante. En consecuencia, el sistema desarrollado posee un alto grado de exactitud en el proceso de medición de los valores de consumo de energía durante su funcionamiento, lo que es lo mismo, los valores obtenidos permiten un adecuado cálculo del ahorro energético. Así pues, durante la fase de pruebas del módulo de control de iluminación se consideró un enfoque cuantitativo de

los parámetros eléctricos obtenidos sobre el consumo de la luminaria, en este sentido, el cálculo del ahorro energético-económico realizado en el apartado 6.9.3, permitió corroborar el cumplimiento del último objetivo específico: ' Analizar el ahorro energético y reducción del gasto económico por la implementación del sistema de control automático de iluminación.'

Dado que la mayor parte de los trabajos que implementan sistemas de iluminación inteligentes, reseñados en la literatura, reflejan un ahorro energético, las pruebas y resultados obtenidos permitieron corroborar que, a medida que incrementa la complejidad del sistema de iluminación, se obtienen mayores ventajas para el cumplimiento del último objetivo antes mencionado. En este sentido, los sistemas inteligentes de iluminación actuales están siendo orientados al mejoramiento y la adición de: sistemas para la detección de presencia, circuitos electrónicos para la regulación de la intensidad luminosa, el desarrollo de algoritmos de control automático para el aprovechamiento de luz incidente, o el establecimiento de niveles de iluminación adecuados, además, otros modelos más complejos apuntan al análisis de la densidad de flujo de personas dentro de un área, como también la creación de redes de sensores inalámbricos distribuidos (WSN) para el control de sistemas de gran capacidad aplicados a áreas extendidas como estaciones de metro o inclusive aplicado a los sistemas de alumbrado público.

En definitiva, el desarrollo del trabajo de titulación ha supuesto una mejora personal y profesional, no solo por los conocimientos nuevos abordados y adquiridos durante el desarrollo del proyecto, sino también, por la capacidad de superar cada uno de los problemas y limitantes vinculadas al desarrollo del trabajo, permitiendo así, lograr los objetivos propuestos al inicio del trabajo.

8 Conclusiones

A la vista de los resultados favorables que se han obtenido en el procedimiento experimental del prototipo, se puede afirmar el cumplimiento de los objetivos formulados al comienzo del presente trabajo, desarrollando un prototipo de sistema de iluminación inteligente, que regula la cantidad de energía necesaria consumida por la luminaria, maximizando el ahorro energético en función del nivel de iluminación dentro del área, como también en función de la presencia y ausencia de personas. En este sentido, se estableció conveniente esclarecer las siguientes conclusiones acerca de cada una de las etapas dentro del desarrollo del prototipo.

- El diseño y desarrollo del circuito atenuador de control de fase AC basado en MOSFET permitió un control adecuado de cargas como luminarias incandescentes o luminarias

LED dimerizables. La ventaja de este circuito atenuador es la baja resistencia interna, lo cual permite una menor potencia disipada en pérdidas y por ende una menor temperatura de operación, lo cual facilitó el funcionamiento y control de la luminaria durante largas horas de trabajo. Además, la característica de corte de fase en el apagado de la onda, causa menos interferencias en comparación con la atenuación de borde delantero analizada en los circuitos electrónicos basados en TRIAC, en consecuencia, el circuito elaborado permite prolongar la vida útil de las luminarias implementadas y mejorar el funcionamiento de las mismas.

- El análisis del factor de potencia en el sistema de iluminación implementado permitió concluir que el proceso de atenuación de la iluminación reduce el valor del FP en cargas no lineales como lo es una luminaria LED. Por otro lado, el sistema implementado posee una alta eficiencia energética, ya que este indicador del correcto aprovechamiento de la energía eléctrica se mantuvo por encima de 0,8 en configuraciones de atenuación por encima del 30 % del ciclo de trabajo del circuito de control de potencia. Esta medida puede oscilar dependiendo del tipo y la calidad de la luminaria LED, en este sentido, para el desarrollo del trabajo se consideró el uso de focos LED con un factor de potencia de 0,9 lo cual presenta un alto desempeño energético con la red de suministro eléctrico.
- Las pruebas del algoritmo de control automático de iluminación, elaborado mediante un controlador PID, permitió lograr un buen funcionamiento del prototipo, llevando el nivel de iluminación al valor establecido por medio de la interfaz gráfica de usuario. Así mismo, el controlador desarrollado proporciona funciones de encendido y apagado de las luminarias, tal que, el usuario experimente un buen impacto visual y la atenuación sea agradable con el entorno.
- El desarrollo de la interfaz de usuario web mediante la herramienta Node-RED, permitió la creación de una aplicación para la visualización de las variables ambientales en tiempo real, el control de la luminaria, como también plasmar información de las características eléctricas del funcionamiento y consumo del sistema. Desde una perspectiva más general, el usuario puede realizar un análisis con información del tiempo de encendido de la luminaria, la energía consumida, la cantidad de energía ahorrada y un valor aproximado del ahorro económico logrado para cada día de funcionamiento del prototipo.
- El sistema de monitoreo en tiempo real de temperatura y humedad permitió obtener un prototipo confiable, con la capacidad de enviar correos de alerta vía Gmail cuando el

sistema lectura valores de temperatura fuera del rango de operación configurado desde la interfaz de usuario. Aunado a esto, un usuario puede acceder a información de las variables ambientales como también de los parámetros de funcionamiento del prototipo desde cualquier lugar con acceso a internet, ingresando a la plataforma de análisis de datos ThingSpeak.

- Como resultado de las pruebas y datos obtenidos del módulo de control de iluminación implementado, se concluye que la capacidad de iluminación del módulo permite garantizar un nivel de iluminación mantenido de 300 luxes, lo cual cumple con la Normativa de iluminación de los lugares de trabajo en interiores NTE INEN 2969-1.

De los resultados obtenidos del análisis del consumo energético por medio de la interfaz de usuario web, se puede destacar los resultados para un día de trabajo correspondiente a la fecha del 5 de marzo de 2022 (ver Figura 120). Para este día de funcionamiento, el sistema se mantuvo encendido desde las 9:33 am hasta las 23:16 pm con un nivel iluminación establecido en 300 luxes. Los resultados obtenidos arrojaron un tiempo total de encendido de la luminaria de 8h 18min, a su vez, la potencia activa promedio se estableció en 15 vatios, en consecuencia, el factor de potencia promedio se mantuvo en un valor de 0,73. El análisis posterior determinó un consumo para la fecha dada de 145 Wh y un valor de energía ahorrada total de 0,21 kWh, lo cual se traduce en un ahorro económico de 1,93 cUSD en ese día de funcionamiento.

9 Recomendaciones

El siguiente apartado establece ciertas medidas o sugerencias prácticas que se consideraron en el desarrollo del prototipo.

- Para la replicación práctica del prototipo, puesto que se dispone del diseño PCB elaborado, se recomienda tomar ciertas medidas de protección (guantes, gafas protectoras), para la realización de pruebas y puesta en funcionamiento del circuito. Esto debido a que el prototipo utiliza un suministro eléctrico en corriente alterna de baja tensión (120 V_{AC}), y una mala conexión de los cables o dispositivo dañado podría provocar un accidente en la zona de trabajo.
- En la implementación elaborada se estableció en primera instancia el uso de un único módulo de control de la iluminación dentro de un área de trabajo. Sin embargo, puesto que el módulo se encuentra dentro de la red WSN como se detalló en la topología del sistema de comunicación inalámbrico (ver Figura 76), con la Raspberry Pi como ordenador central, se puede incrementar el número de módulos de control para la

instalación de más sensores y actuadores e incrementar las zonas de trabajo dentro de una edificación. Esto debido a la capacidad de crecimiento y escalabilidad del sistema desarrollado, y la facilidad del servidor MQTT de conectar múltiples clientes para el envío y recepción de la información basado en el modelo publicador/suscriptor.

- Por otra parte, para una posible aplicación comercial a futuro, se recomienda mantener el ESP8266 en modo sueño ligero o *light sleep*, en el cual el módulo WiFi se desactiva y se suspende la CPU, permitiendo un consumo de hasta 0,9 mA muy inferior al consumo del ESP8266 en estado activo, entre 56 mA y 170 mA. De este modo se lograría un ahorro de energía adicional cuando el sistema detecte la ausencia de personas, así pues, mantendría la luminaria apagada, y pausaría el envío de la información al ordenador central (iluminancia en luxes), hasta un nuevo evento de activación por parte del sensor PIR, aumentando más la eficiencia del sistema.
- A manera de recomendación, se aconseja la realización de copias de seguridad periódicas (*backup* de la SD del Raspberry Pi). Esta medida permite salvaguardar la configuración, los programas, actualizaciones, base de datos y la programación desarrollada en la herramienta Node-RED. De esta forma, en caso de fallo del hardware o software del ordenador central, el sistema podrá ser restaurado a su última versión en la cual el funcionamiento era pleno.
- Entre las recomendaciones más importantes del trabajo, se incluye la promoción de la iluminación LED como medida para la reducción del consumo energético y la mitigación de las emisiones de gases de efecto invernadero que afectan la salud humana y el medio ambiente.

Por otra parte, para futuros estudios académicos asociados a los sistemas de iluminación inteligentes dentro de áreas de trabajo en interiores, se establece las siguientes sugerencias:

- Se propone la adición de un sistema de visión artificial que permita un sistema de detección de presencia de mayor robustez y fiabilidad. Esto debido a que un sensor PIR no proporciona un adecuado funcionamiento, y genera eventos de activación en función de cualquier cuerpo que emita energía electromagnética infrarroja (p.ej. personas, animales). Este sistema sugerido permitiría la detección exclusiva de personas dentro del área, utilizando técnicas de reconocimiento y software de procesado.
- Se propone la mejora del algoritmo de control automático PID, mediante el estudio y desarrollo de un algoritmo de aprendizaje automático que permita adaptar y regular el nivel de iluminación, en función de decisiones tomadas anteriormente por los usuarios

y aprendidas por el sistema, para ciertas condiciones del ambiente y hora de funcionamiento.

De cara a desarrollos futuros, queda la puerta abierta a reutilizar lo expuesto en este trabajo para expandir el desarrollo a otros sistemas que permitan alcanzar una mayor eficiencia energética y gestión del consumo, mediante la creación de prototipos a partir del desarrollo de sistemas embebidos, y la aplicación de la teoría de los sistemas de control, que son la base principal del sistema propuesto.

10 Bibliografía

- InnovatorsGuru. (2019). *PZEM-004T V3*. Obtenido de <https://innovatorsguru.com/pzem-004t-v3/>
- ABC PROYECTOS ELECTRÓNICOS. (2014). *Sensor de humedad y temperatura DHT 11 y DHT22*.
- Abu-Khalaf, M., Chen, R., & Turevskiy, A. (2009). *PID Control Design Made Easy*.
- ACADEMIA. (2020). Obtenido de https://www.academia.edu/28289361/Internet_of_Threats_IoT_Una_visi%C3%B3n_
- Aguayo, P. (10 de Noviembre de 2004). *INTRODUCCIÓN AL MICROCONTROLADOR*. Obtenido de <https://d1wqtxts1xzle7.cloudfront.net/39407044/micro-with-cover-page-v2.pdf?Expires=1637811076&Signature=VIhrzgcHzpiCOd8shze4yfS3g~UCTWGgD70rwpTO9M5EeGSwMI-IVgRo9rMzZaHymJfkcjbnNrd3M6AvBiT~Lg3J3Y1fdYg1cjMtX~cZxEubfIY1WnZFDtj2GRH5YRVXWfdLHAZlnHZYMKpx8QcLGP>
- Alejandro, S. (9 de Junio de 2021). *GSL Industrias*. Obtenido de <https://www.industriasgsl.com/blog/post/que-es-un-sistema-de-control>
- Almeida, J. J. (2016). *Eficiencia energética e implementación de focos LED en el sistema residencial ecuatoriano*. Quito.
- Aosong Electronics. (s.f.). *Digital-output relative humidity & temperature sensor/module DHT22*. Obtenido de <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>

- ARROW Electronics. (2021). *Controladores LED*. Obtenido de <https://www.arrow.com/es-mx/categories/drivers-and-interfaces/drivers/led-drivers>
- ATMEL. (2013). *Atmel 8-bit AVR Microcontroller with 2/4/8K Bytes In-System Programmable Flash*.
- Avago Technologies. (19 de Junio de 2007). *Phototransistor Optocoupler General Purpose Type*. Obtenido de <https://manualzz.com/doc/8501428/data-sheet-4n35-phototransistor-optocoupler-general-purpo...>
- AX, Desarrollos. (29 de Septiembre de 2013). *Explicación Control de fase directo*. Obtenido de <http://axdesarrollos.blogspot.com/2013/09/explicacion-control-de-fase-directo.html>
- BBC News. (2020). *El tipo de iluminación que puede salvar al mundo*.
- Beauregard, B. (2011). *Arduino PID*.
- Bigelow, S. J. (9 de Abril de 2021). *Sistema operativo Linux*. Obtenido de <https://www.computerweekly.com/es/definicion/Sistema-operativo-Linux>
- Borgues, E. (17 de Marzo de 2019). *infranetworking*. Obtenido de <https://blog.infranetworking.com/servidor-base-de-datos/>
- CELEC EP. (2021). *VIGENTE EL MECANISMO PARA SUBSIDIO ELÉCTRICO*. Obtenido de <https://www.celec.gob.ec/termopichincha/index.php/86-noticias/398-vigente-el-mecanismo-para-subsidio-#:~:text=La%20tarifa%20promedio%20a%20nivel,9%2C20%20cUSD%2FkWh.>
- CONELEC. (2001). *Calidad del servicio electrico de distribución*. Quito.
- CONELEC. (2013). *Aspectos de sustentabilidad y sostenibilidad social y ambiental*. Quito.
- Corona, L., Abarca, G., & Carreño, J. (2014). *Sensores y Actuadores*. México: GRUPO EDITORIAL PATRICIA, S.A. DE C.V.
- DOMÓTICA SISTEMAS. (2019). *¿Qué es Domótica e Inmótica?* Obtenido de https://domoticasistemas.com/tienda/tutoriales/2_diferencias-domotica-e-inmotica.html
- Durán, C. (Diciembre de 2004). *ELECTRO INDUSTRIA*. Obtenido de <http://www.emb.cl/electroindustria/articulo.mvc?xid=193>

- EfectoLED Blog. (Julio de 2017). *Tipos de diodos LED que existen en el mercado*. Obtenido de <https://www.efectoled.com/blog/es/tipos-de-chips-led-que-existen-en-el-mercado/>
- Elliot, R. (2015). *Lighting Dimmers - Part 2*. Obtenido de <https://sound-au.com/lamps/dimmers2.html>
- Elliott, R. (Noviembre de 2017). *Lighting Dimmers*. Obtenido de <https://sound-au.com/lamps/dimmers.html>
- Elliott, R. (Diciembre de 2019). *MOSFET Solid State Relays*. Obtenido de <https://sound-au.com/articles/mosfet-relay.htm>
- Elliott, R. (Julio de 2020). *Factor de potencia: la realidad*. Obtenido de <https://sound-au.com/lamps/power-factor.html>
- ELUNIVERSO. (14 de Octubre de 2018). *Empresa impulsa el uso de focos led en Ecuador*.
- FAIRCHILD. (1 de Mayo de 2016). *FL5150/60 IGBT and MOSFET AC Phase Cut Dimmer Controller*. Obtenido de <https://www.mouser.com/datasheet/2/149/FL5150-964157.pdf>
- FAIRCHILD SEMICONDUCTOR. (2002). *IRF840 Data Sheet*.
- Fernandez, A. (2013). *Python 3 al descubierto*. Madrid: Alfaomega Grupo Editor.
- Gago, A., & Fraile, J. (2012). *ILUMINACIÓN CON TECNOLOGÍA LED*. España: Paraninfo.
- Gaona, C. E. (2019). *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO EN TIEMPO REAL DE VARIABLES AMBIENTALES DE ÁREAS CRÍTICAS DEL EDIFICIO SIS ECU-911 LOJA*. Loja.
- García, F. (25 de Noviembre de 2021). *¿Qué es phpMyAdmin y cómo usarlo?* Obtenido de <https://www.arsys.es/blog/phpmyadmin>
- Gomez, A. (19 de Julio de 2016). *Diseño de una estación metereológica autónoma sin hilos y de bajo consumo*. Obtenido de <https://sites.google.com/site/tfgalbertogomez/home/proyectos/Sensores/DHT22>
- González, B. R. (2019). *DISEÑO LUMÍNICO PARA EL ESTADIO DE LA UNIVERSIDAD NACIONAL DE LOJA*. Loja.
- Greengard, S. (2021). *Energy Management for Internet of Things via Distributed*. MIT press.

- Guanín, C. R. (2015). *DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE ILUMINACIÓN AUTOSUSTENTABLE FOTOVOLTAICO PARA UNA PARADA DE BUSES Y SU VAYA INFORMATIVA DEL SISTEMA INTEGRADO DE TRANSPORTE DE CUENCA*. Cuenca.
- Hardwarelibre. (2019). Obtenido de <https://www.hwlibre.com/esp8266/#:~:text=ESP8266%3A%20el%20m%C3%B3dulo%20WIFI%20para%20Arduino%20%7C%20Hardware%20libre>
- Heath, N. (13 de Marzo de 2014). *How IBM's Node-RED is hacking together the internet of things*. Obtenido de <https://www.techrepublic.com/article/node-red/>
- Huamán, J. (2017). *CONTROL INTELIGENTE DE SISTEMAS DE ILUMINACIÓN EN EDIFICIOS*. Piura.
- Hunkeler, U. a.-C. (2008). MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08).
- Ipiña, D. (24 de Febrero de 2005). *Programación Web Open Source con LAMP*. Obtenido de <https://paginaspersonales.deusto.es/dipina/cursos/LAMP.pdf>
- Kim, J.-H., Jung, J.-H., Ryu, M.-H., & Baek, J.-W. (2011). A SIMPLE DIMMER USING A MOSFET FOR AC DRIVEN LAMP. *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, 2872-2876.
- Kuo, B. C. (1996). *SISTEMAS DE CONTROL AUTOMÁTICO*. Pearson Educación.
- Lawlor, L. (28 de Octubre de 2014). *GitHub*. Obtenido de <https://github.com/lawlor/thingspeak>
- LEDBOX. (15 de Mayo de 2019). *Reguladores por corte de fase*. Obtenido de <https://blog.ledbox.es/informacion-led/casos-practicos-led/reguladores-por-corte-de-fase>
- LEDBOX Blog. (15 de Noviembre de 2017). *La importancia del los drivers en la iluminación LED*. Obtenido de <https://blog.ledbox.es/informacion-led/instalacion-de-luminarias/la-importancia-del-los-drivers-en-la-iluminacion-led>

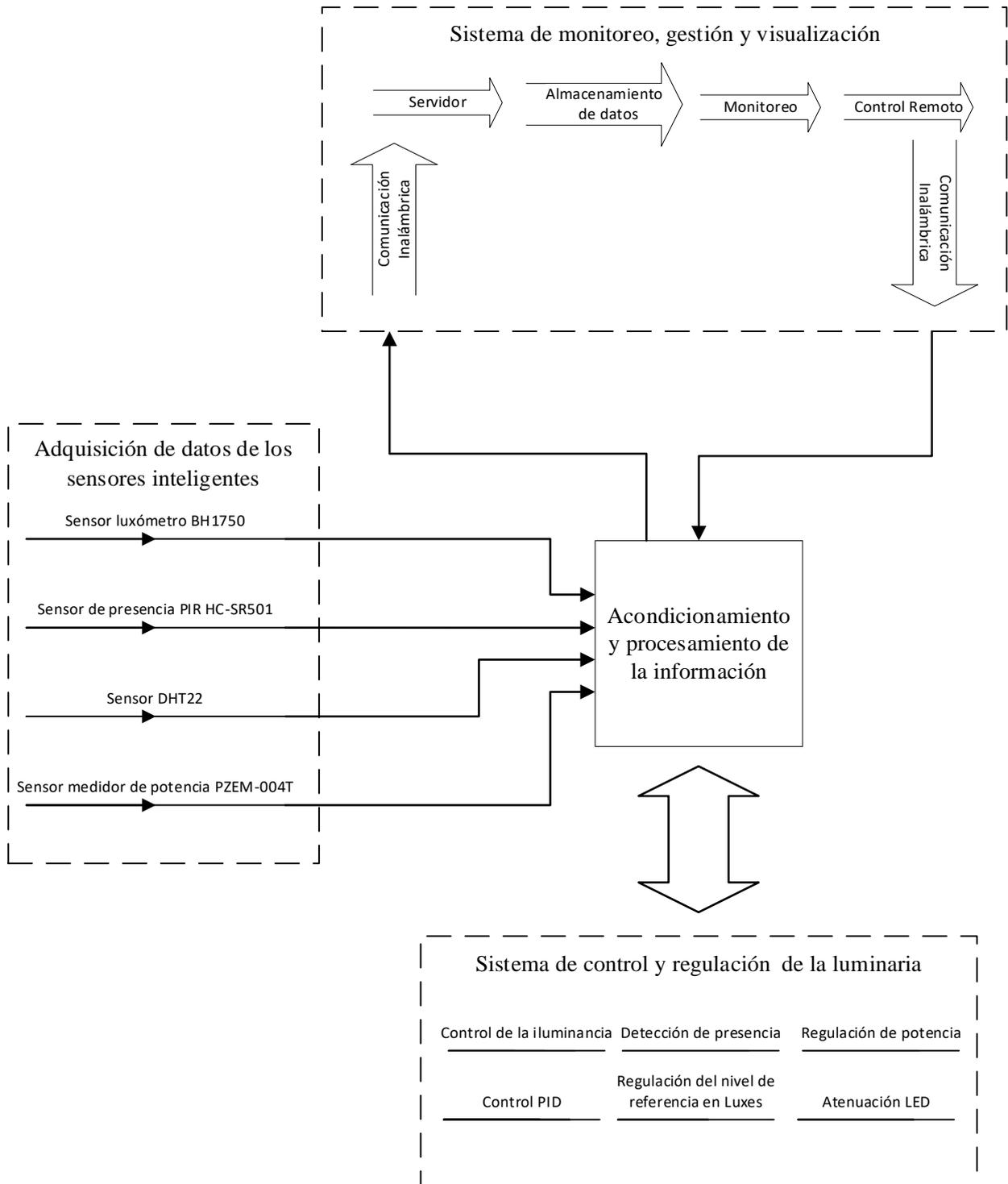
- Light, R. A. (13 de Mayo de 2017). *Mosquitto: server and client implementation of the MQTT protooco*. Obtenido de <https://joss.theoj.org/papers/10.21105/joss.00265>
- López, E. (2010). *Identificación de Sistemas*. Obtenido de <http://www.ie.tec.ac.cr/einteriano/control/Laboratorio/3.7Identificacion%20de%20sistemas.PDF>
- Luis, C., & Danny, C. (2015). *EFICIENCIA ENERGÉTICA EN EL ALUMBRADO PÚBLICO DEL CENTRO HISTÓRICO DE CUENCA: TELEGESTIÓN Y SUSTITUCIÓN DE LUMINARIAS*. Cuenca.
- Malvino, A. P. (2000). *Principios de Electrónica*. McGraw-Hill/Interamericana de España, S. A. U.
- Manotoa, D. A. (2018). “*PLATAFORMA IoT DE CONTROL INTELIGENTE DE UN SISTEMA DE ILUMINACIÓN LED CON SUMINISTRO ELÉCTRICO EN CORRIENTE CONTINUA LVDC*”. Ambato.
- Mazzone, V. (Marzo de 2002). *Controladores PID*. Obtenido de <https://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>
- Microchip Technology Inc. (2009). *12-Bit Digital-to-Analog Converter with EEPROM Memory in SOT-23-6*.
- Miguel, G. (2018). *Diseño de un driver de led de 200 W desde la red electrica con control DALI*. Gijón.
- NEC. (2013). *Instalaciones Electromecánicas*. Quito.
- NTE INEN 2969-1. (2015). *ILUMINACIÓN DE LOS LUGARES DE TRABAJO EN INTERIORES*. Quito.
- Ogata, K. (2010). *Ingeniería de control moderna*. Madrid: PEARSON EDUCACIÓN, S.A.
- Outeiriño, E. B. (2020). *TECNOLOGÍAS FOTÓNICAS Y MODELOS MATEMÁTICOS PARA ALUMBRADO INTELIGENTE Y EFICIENCIA ENERGÉTICA*.
- Parihar, Y. S. (2019). Internet of Things and Nodemcu. 1-4.
- Picuíno. (Noviembre de 2021). *Controlador PID*. Obtenido de <https://www.picuino.com/es/control-pid.html>

- Proyecto987. (s.f.). *Potencia en un circuito de corriente alterna*. Obtenido de http://www.proyecto987.es/corriente_alterna_11.html
- Punto Flotante S.A. (2017). *Sensor infrarrojo de movimiento PIR HC-SR501* . Obtenido de <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>
- (s.f.). *PZEM-004T V3.0 User Manual*.
- Red Hat. (19 de Marzo de 2018). *El concepto de Linux*. Obtenido de <https://www.redhat.com/es/topics/linux>
- Reyes, D. G. (2006). *Bases de datos distribuidas con una solución LAMP (Linux, Apache, MySQL y PHP)*. México.
- Reynada, L. P. (2012). *Microprocesadores*. México: RED TERCER MILENIO S.C. .
- Rodger, L. (1 de Enero de 2016). *Node RED Programming Guide*. Obtenido de <http://noderedguide.com/>
- ROHM SEMICONDUCTOR. (2009). *Digital 16bit Serial Output Type Ambient Light Sensor IC*.
- Sałuch, M., Tokarski, D., & Adamczewski, G. (2018). Raspberry PI 3B + microcomputer as a central control unit in intelligent building automation management systems. *MATEC Web of Conferences 196 , 04032 (2018)*, 1-6.
- Sánchez, V. M. (2018). *Internet de las Cosas. Horizonte 2050*.
- SangCheol, M., Koo, G.-B., & Gun-Woo, M. (2015). DIMMING-FEEDBACK CONTROL METHOD FOR TRIAC DIMMABLE LED DRIVERS . *IEEE Transactions on Industrial Electronics*, 960-965.
- Stallman, R. M. (Marzo de 2015). *Hardware libre y diseños libres para hardware*. Obtenido de <https://www.gnu.org/philosophy/free-hardware-designs.html>
- Texas Instruments Incorporated. (Agosto de 2016). *Self-Powered AC Solid State Relay With MOSFETs*.
- UNIT Electronics. (2016). *Transistor IRF840*. Obtenido de <https://uelectronics.com/producto/transistor-irf840/>

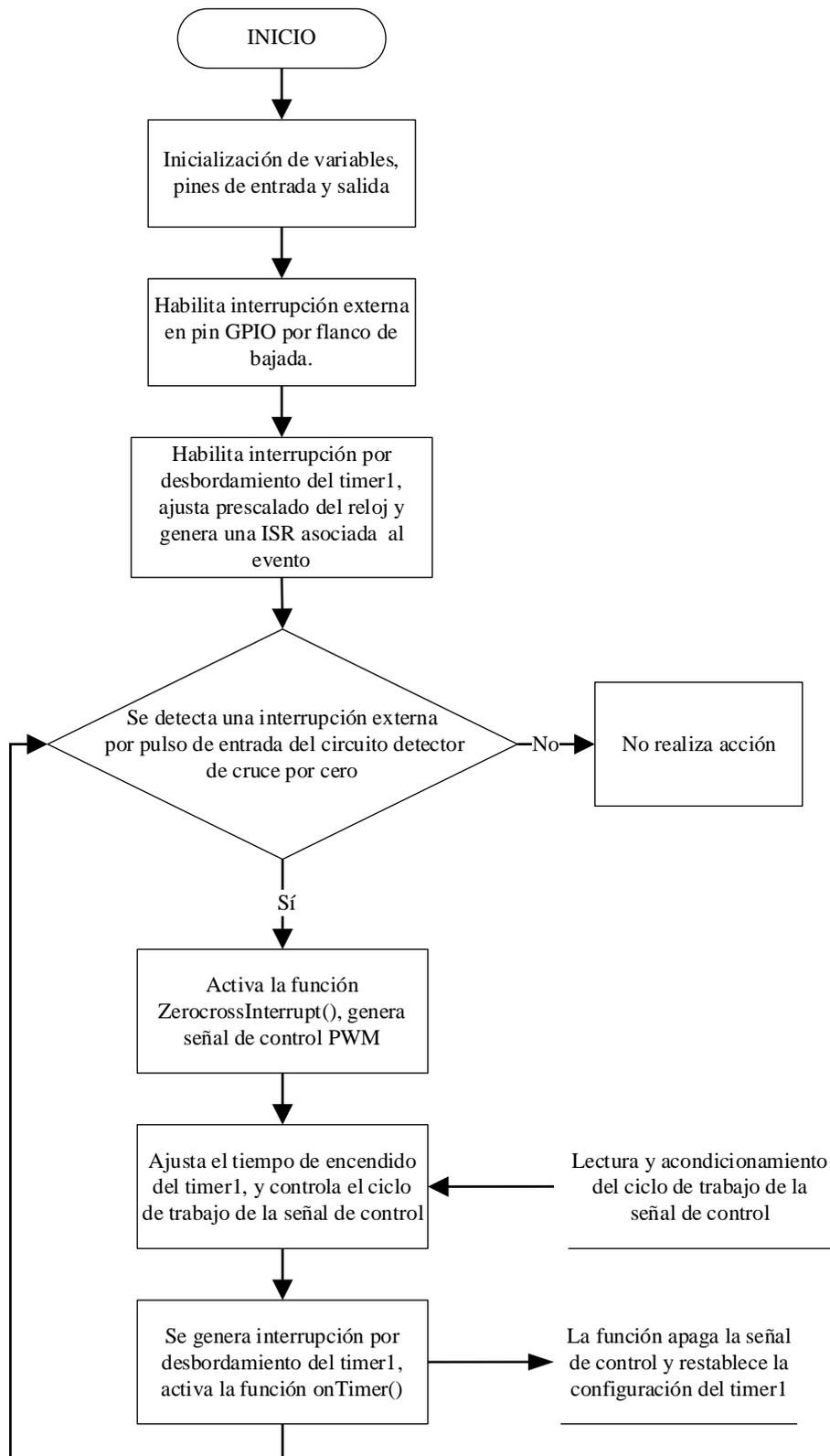
- UNIT Electronics. (2021). *Transistor IRF840*. Obtenido de <https://uelectronics.com/producto/transistor-irf840/>
- Uriarte, J. M. (10 de Marzo de 2020). *"Microprocesador"*. Obtenido de <https://www.caracteristicas.co/microprocesador/>
- Vicmar, R. (2020). *Evaluación y diseño de una Plataforma IoT con soporte de protocolo MQTT para un entorno de red empresarial*. Lima.
- Victoria. (20 de Noviembre de 2017). *Diferencia entre lenguajes compilados e interpretados*. Obtenido de <https://yosoy.dev/diferencia-entre-lenguajes-compilados-e-interpretados/>
- Won-Seok, K. (18 de Septiembre de 2013). *Electronic Products*. Obtenido de <https://www.electronicproducts.com/competitive-analysis-of-mosfet-and-igbt-in-a-400-w-phase-cut-dimmer/>
- XIAMEN HUZDA . (31 de Julio de 2019). *Atenuadores De Corte De Fase*. Obtenido de https://es.huzzdaled.com/blog/phase-cut-dimmers-foward-phase-reverse-phase-dimmer_b1
- YMANT. (10 de Octubre de 2021). Obtenido de <https://www.ymant.com/blog/lamp/>
- Yuan, M. (4 de Octubre de 2017). *Conozca MQTT*. Obtenido de <https://developer.ibm.com/es/articles/iot-mqtt-why-good-for-iot/>
- Yusi, C., Chen, F., & Jingfeng, Y. (2020). *DESIGN AND APPLICATION OF A SMART LIGHTING SYSTEM BASED ON DISTRIBUTED WIRELESS SENSOR NETWORKS*. Nanjing: Multidisciplinary Digital Publishing Institute MPDI.

11 Anexos

Anexo 1 Diagrama de bloques general del prototipo de sistema inteligente de iluminación



Anexo 2 Diagrama de flujo del algoritmo de control de potencia del circuito atenuador



SYLVANIA

LED Toledo
TOLEDO A60 10W WW 15H DIM CJ
P27960



Bombilla LED en formato bulbo para iluminación domestica, su tecnología y diseño proporciona una mejor iluminación interior. Ahorra hasta el 90% de energía comparado con bombillas incandescentes.

CARACTERÍSTICAS

Diseño de bulbo tradicional
 Cuerpo con acabado opalizado
 Tecnología de chip LED SMD

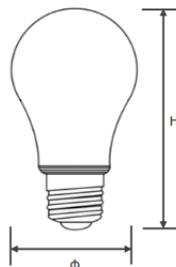
APLICACIONES

Iluminación decorativa
 Espacios domésticos
 Iluminación residencial



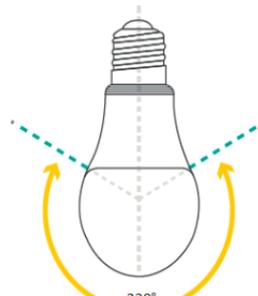
DATOS ÓPTICOS		DATOS FÍSICOS		DATOS ELÉCTRICOS	
Temperatura de color	3000K	Base	E27	Potencia de entrada	10W
Flujo luminoso	800 lm	Acabado	Opalizado	Tensión de operación	120V 50/60Hz
Ángulo de apertura	230°	Dimensiones (Φ x H)	60x110 mm	Corriente de entrada	0.12A @120V
Tipo de distribución	Directa simétrica	Temperatura de operación Ta	-10°C ~ +45°C	Frecuencia de operación	50/60Hz
Reproducción de color (IRC)	>80	Ambiente de Uso	Iluminación interior	Factor de potencia	0.7
Vida útil	15000h L70			Atenuable	SI
Eficacia	80 lm/W				

DIMENSIONES



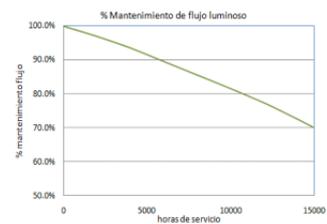
Las características de los productos pueden ser modificadas sin previo aviso según la evolución de la tecnología LED. 07/18

APERTURA DE HAZ



Producto Ecológico:
 Permite ahorrar energía comparado con productos tradicionales. Libre de mercurio.

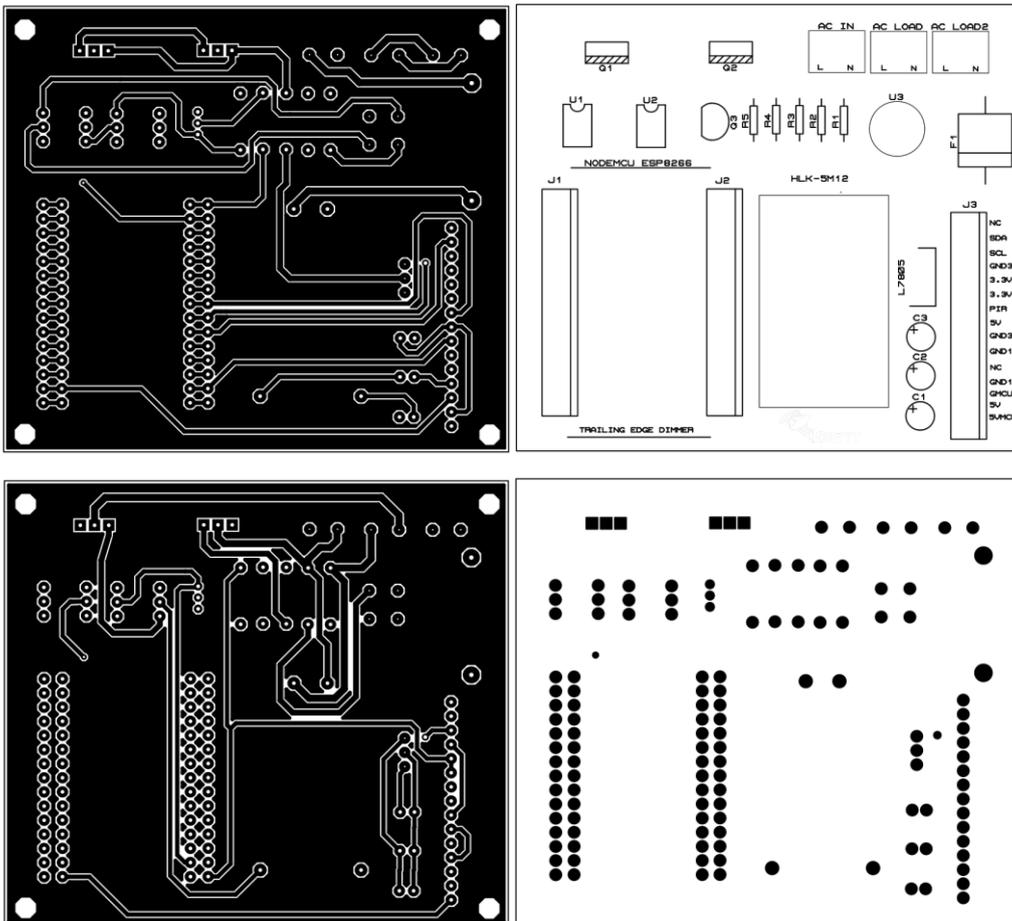
CURVA DE DEPRECIACION



by FEILO SYLVANIA

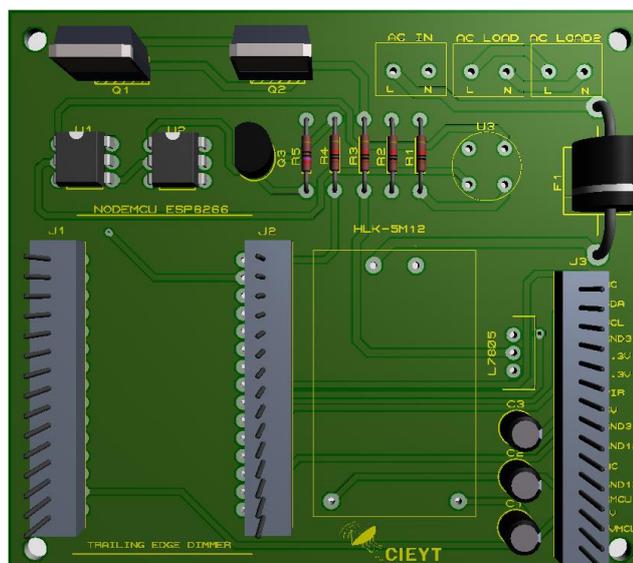
Nota. Adaptado de <https://sylvania.com.ec/wp-content/uploads/2020/01/1.-FT-P29362.pdf>

Anexo 4 Diseño PCB del prototipo de sistema de iluminación inteligente

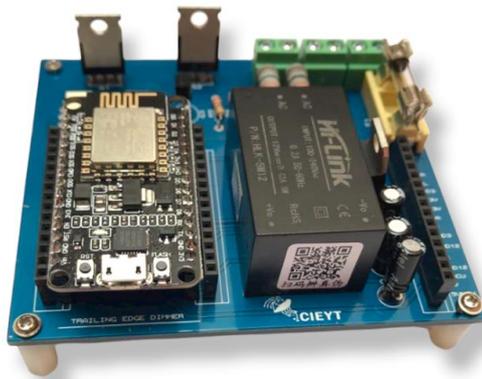
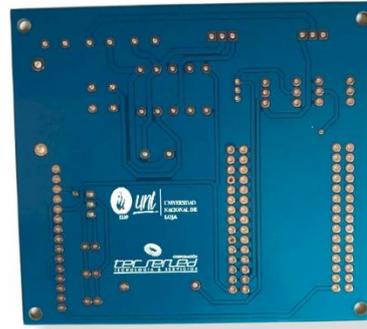
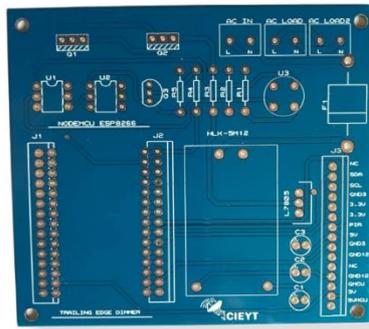


Nota. De arriba hacia abajo, capas de cobre superior (Top Copper), cobre inferior (Bottom Copper), capa de serigrafía (Top Silk) y capa de perforaciones (Drill Hole)

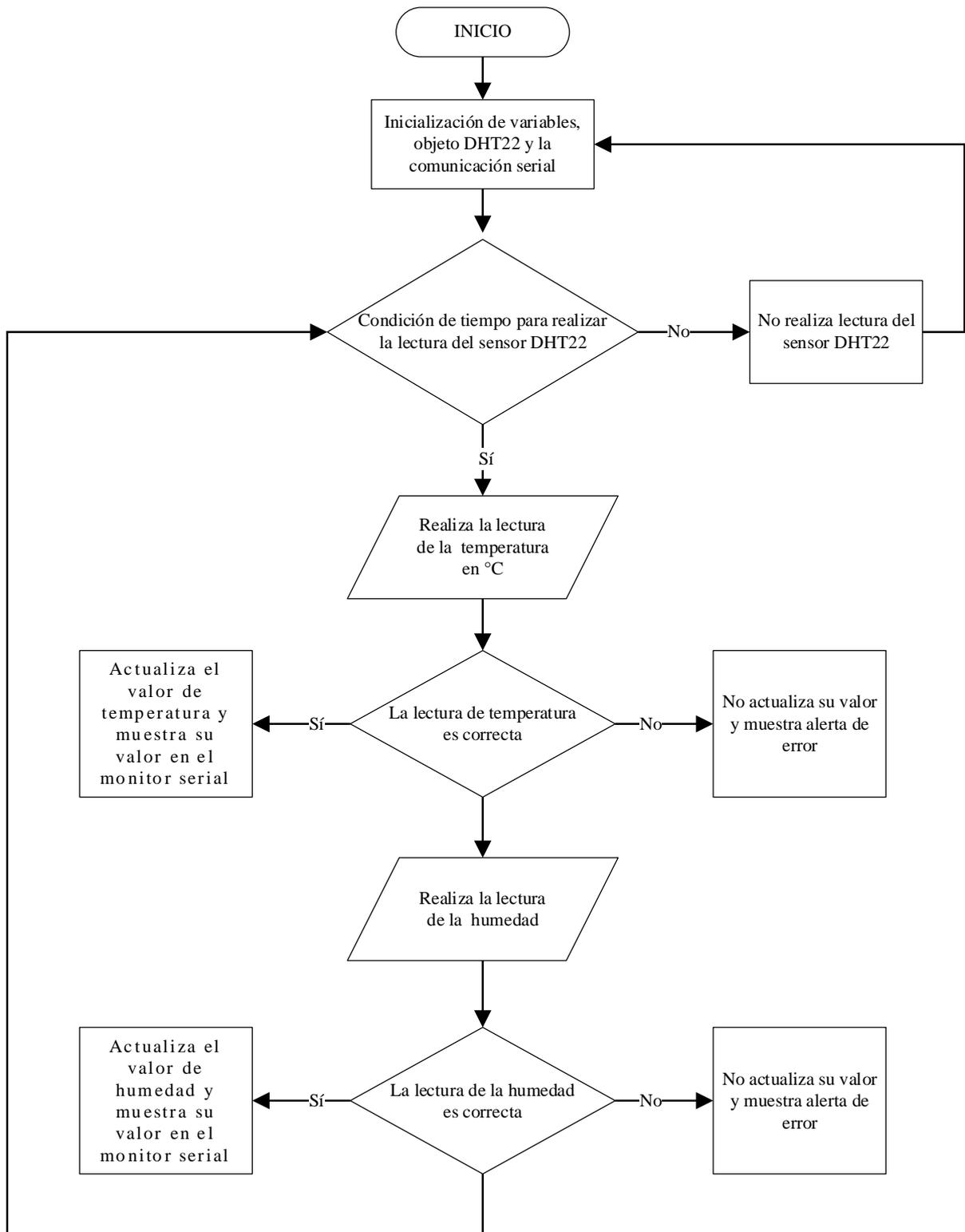
Anexo 5 Simulación 3D del diseño de placa PCB



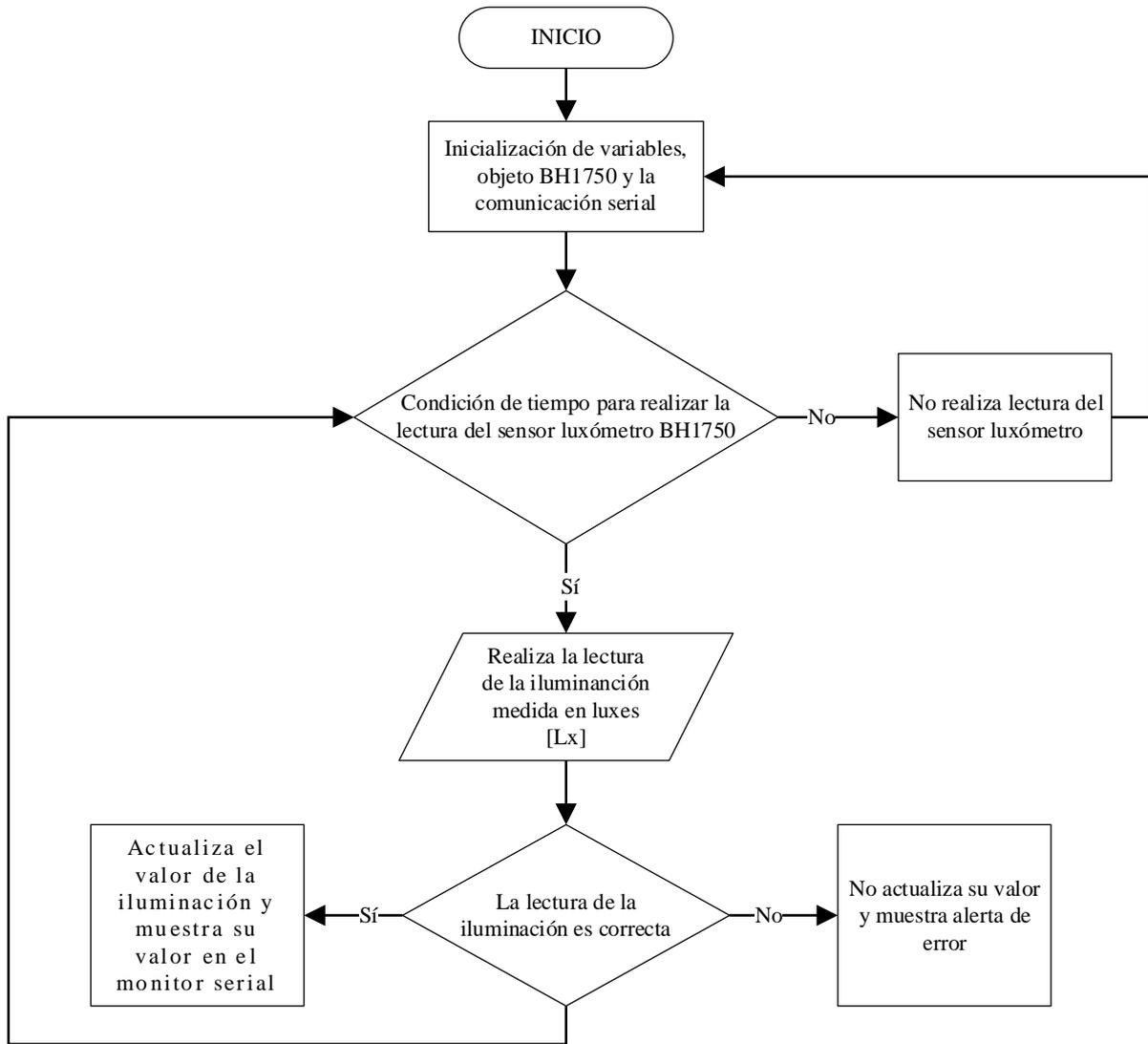
Anexo 6 Resultado final de la construcción de las placas PCB y circuito electrónico



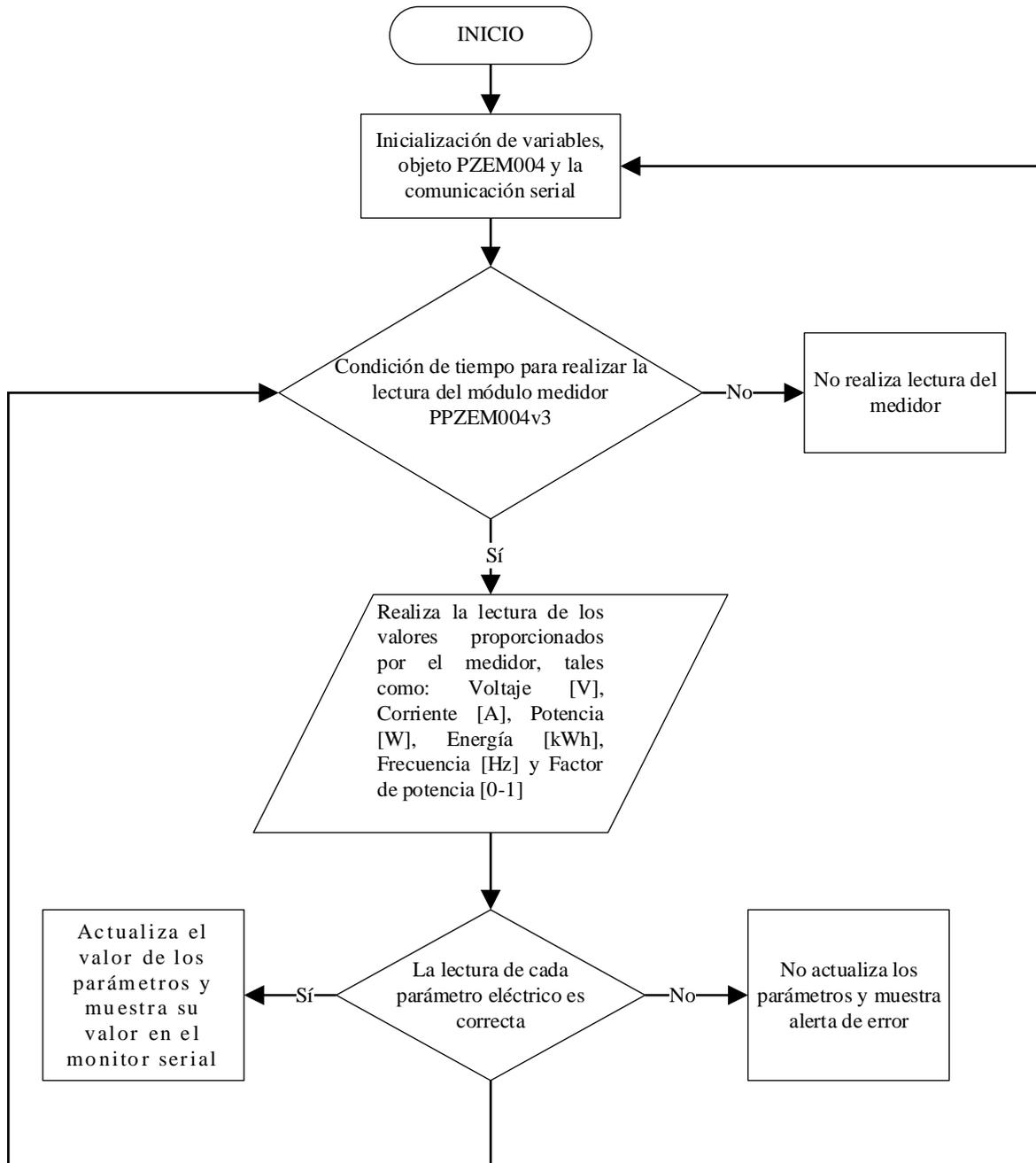
Anexo 7 Diagrama de flujo para la adquisición de datos del sensor DHT22



Anexo 8 Diagrama de flujo para la adquisición de datos del sensor BH1750



Anexo 9 Diagrama de flujo para la adquisición de datos del módulo PZEM-004T



Código implementado
<pre> /***** DECLARACIÓN DE LAS LIBRERIAS, DEFINICIÓN DE PINES GPIO, VARIABLES Y CONSTANTES *****/ #include <Adafruit_Sensor.h> #include <DHT.h> #include <Wire.h> #include <PZEM004Tv30.h> #include <BH1750.h> #define DHTPIN 4 // Define el pin digital D2 o GPIO4 conectado al sensor DHT22 #define DHTTYPE DHT22 // Define el tipo de sensor DHT usado, DHT22 para este trabajo DHT dht(DHTPIN, DHTTYPE); // Declara el objeto DHT con los parámetros pin y tipo de DHT. PZEM004Tv30 pzem(2, 0); // Declara los pines TX y RX para la comunicación pzem(TX,RX)---> pzem(D4,D3) o pzem(GPIO2,GPIO0) BH1750 lightMeter; float lux = 0.0; //Variable que almacena el nivel de Iluminancia [Lux] extraído de la lectura del sensor BHT1750 float t = 0.0; // Variable para mantener los valores actuales de temperatura del sensor DHT22 float h = 0.0; // Variable para mantener los valores actuales de humedad del sensor DHT22 unsigned long previousMillis = 0; // Variable de tiempo que almacenará la última vez que se actualizó los sensores const int interval = 2000; // Variable que actualiza las lecturas de los sensores cada (interval) milisegundos /***** LECTURA DEL SENSOR BH1750 *****/ void leer_luxometro() { Wire.begin(13, 12); // Habilita la comunicación serial(Bus Serial) para el sensor luxómetro BH1750 con la conexión de SDA en D7 o GPIO13 y SCL en D6 O GPIO12 lightMeter.begin(); // Establece el modo de lectura por defecto BH1750_CONTINUOUS_HIGH_RES_MODE--->High Resolution Mode - (1 lx precisión, 120ms measurement time) uint16_t newlux = lightMeter.readLightLevel(); // Realiza la lectura del sensor y lo almacena en la variable newlux if (isnan(newlux)) { // Si la lectura de la iluminancia falla, no cambie el valor de luxes Serial.println("Error obteniendo los datos del sensor BH1750!"); } else { lux = newlux; Serial.print("Iluminancia:"); Serial.print(lux); </pre>

```

Serial.print("[x]");
Serial.print("\t");
Serial.print("\n");
}
}
/*****
LECTURA SENSOR PZEM004TV3
* *****/
void leer_pzem004t() {
  Serial.print("Custom Address:");
  Serial.println(pzem.readAddress(), HEX);
  // Lee los datos eléctricos del módulo PZEM004Tv3
  float voltage = pzem.voltage();
  float current = pzem.current();
  float power = pzem.power();
  float energy = pzem.energy();
  float frequency = pzem.frequency();
  float pf = pzem.pf();
  // Verifica que la lectura de los datos sea válida
  if (isnan(voltage)) {
    Serial.println("Error en la lectura de voltage!");
  } else if (isnan(current)) {
    Serial.println("Error en la lectura de corriente!");
  } else if (isnan(power)) {
    Serial.println("Error en la lectura del potencia!");
  } else if (isnan(energy)) {
    Serial.println("Error en la lectura de energía!");
  } else if (isnan(frequency)) {
    Serial.println("Error en la lectura de frecuencia!");
  } else if (isnan(pf)) {
    Serial.println("Error en la lectura del factor de potencia!");
  } else {

    // Presenta los datos eléctricos a través del monitor serie
    Serial.print("Voltaje: "); Serial.print(voltage); Serial.println(" [V]");
    Serial.print("Corriente: "); Serial.print(current); Serial.println(" [A]");
    Serial.print("Potencia: "); Serial.print(power); Serial.println(" [W]");
    Serial.print("Energía: "); Serial.print(energy, 3); Serial.println(" [kWh]");
    Serial.print("Frecuencia: "); Serial.print(frequency, 1); Serial.println(" [Hz]");
    Serial.print("PF: "); Serial.println(pf);
  }

  Serial.println("");
}
/*****
LECTURA DEL SENSOR DHT22
* *****/
void leer_dht22() {
  float newT = dht.readTemperature(); // Leer la temperatura como Celsius (valor por
defecto)

```

```

//float newT = dht.readTemperature(true); // Leer la temperatura como Fahrenheit (is
Fahrenheit = true)
if (isnan(newT)) { //Si la lectura de la temperatura falla, no cambie el valor de t
  Serial.println("Error obteniendo los datos del sensor DHT22!");
}
else { //Si la lectura de la temperatura es correcta, actualiza el valor de t
  t = newT;
  Serial.print("Temperatura: ");
  Serial.print(t);
  Serial.print(" °C \t");
}
float newH = dht.readHumidity(); // Lee la humedad
if (isnan(newH)) { // Si la lectura de la humedad falla, no cambie el valor de h
  Serial.println("Error obteniendo los datos del sensor DHT22!");
}
else { // Si la lectura de la humedad es correcta, actualiza el valor de h
  h = newH;
  Serial.print("Humedad: ");
  Serial.print(h);
  Serial.print(" %\t");
}
float hic = dht.computeHeatIndex(t, h, false); //Calcula el índice de calor en grados
centígrados
Serial.print("Índice de calor: ");
Serial.print(hic);
Serial.print(" °C \t");
Serial.println("");
}

/*****
COMANDOS DE ARRANQUE DEL MCU
* *****/
void setup() {
  Serial.begin(115200); // Inicializa el puerto serie para fines de depuración
  Serial.print("Reset Energy");
  pzem.resetEnergy(); // Resetea el contador de energía activa en kWh
  pinMode(DHTPIN, INPUT_PULLUP); // Activa resistencia Pull-Up del pin D2
  dht.begin();
}

/*****
COMANDOS DE LA FUNCION DE BUCLE DEL MCU
* *****/
void loop() {
  unsigned long currentMillis = millis(); // Mediante la función millis() realiza la lectura
del tiempo actual
  if (currentMillis - previousMillis >= interval) // Condicional que realiza la lectura de los
sensores cada (interval) milisegundos
  {
    previousMillis = currentMillis; // Guarda la última vez que se actualizaron los

```

```
valores
  leer_luxometro();
  leer_dht22();
  leer_pzem004t();
  Serial.println("");
}
}
```

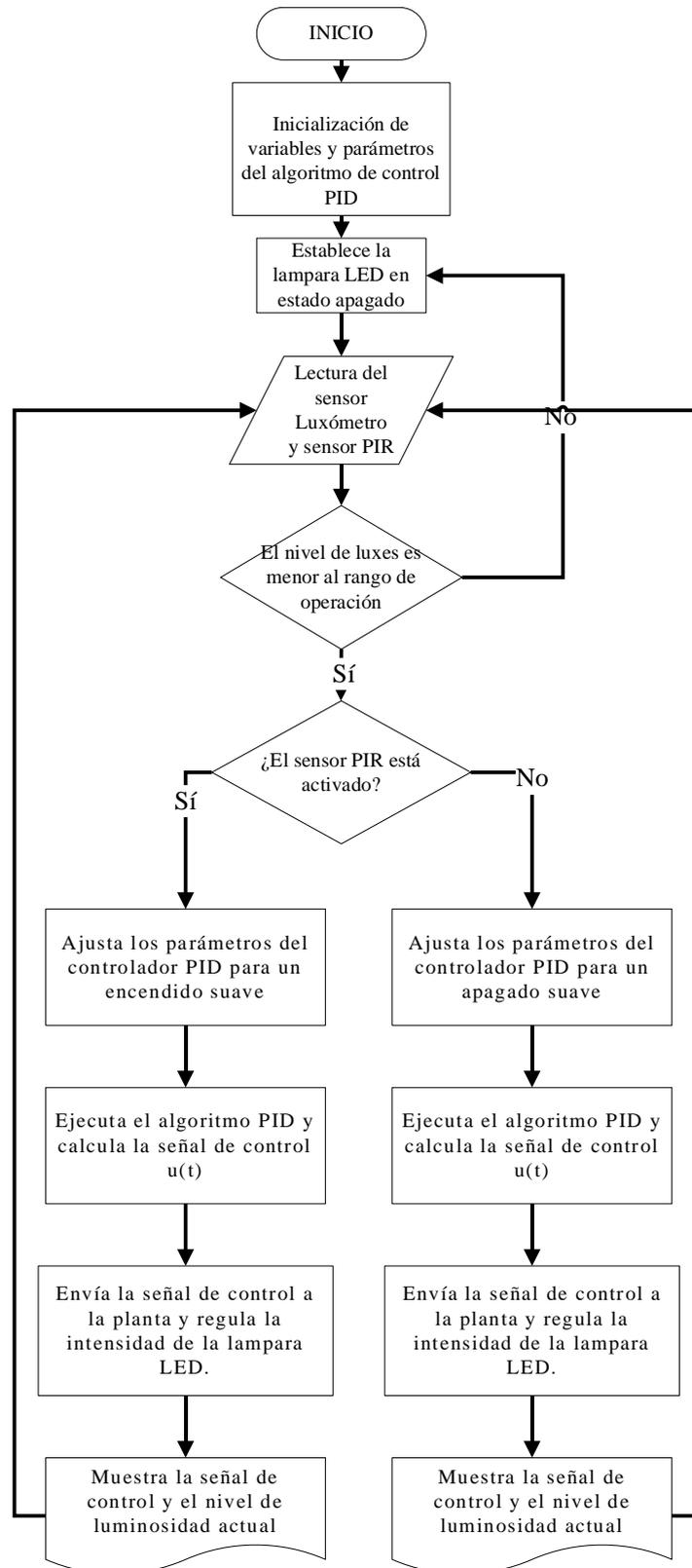
Resultados obtenidos en el monitor serie de ARDUINO

```
16:57:23.199 -> Iluminancia:96.00[lx]
16:57:23.239 -> Temperatura: 22.20 °C    Humedad: 59.50 %    Índice de calor: 22.03 °C

16:57:23.239 -> Custom Address:1
16:57:23.319 -> Voltaje: 120.90 [V]
16:57:23.319 -> Corriente: 0.11 [A]
16:57:23.319 -> Potencia: 9.50 [W]
16:57:23.319 -> Energía: 0.001 [kWh]
16:57:23.319 -> Frecuencia: 60.0 [Hz]
16:57:23.319 -> PF: 0.70
16:57:23.319 ->
16:57:23.319 ->
16:57:25.199 -> Iluminancia:96.00[lx]
16:57:25.239 -> Temperatura: 22.20 °C    Humedad: 59.50 %    Índice de calor: 22.03 °C

16:57:25.239 -> Custom Address:1
16:57:25.319 -> Voltaje: 120.90 [V]
16:57:25.319 -> Corriente: 0.11 [A]
16:57:25.319 -> Potencia: 9.50 [W]
16:57:25.319 -> Energía: 0.001 [kWh]
16:57:25.319 -> Frecuencia: 59.9 [Hz]
16:57:25.319 -> PF: 0.70
```

Anexo 11 Diagrama del algoritmo de control inteligente de iluminación



SERIE A REGULABLE

Descripción del producto

Focos de tamaño estándar, de fácil instalación, diseñados para sustituir focos incandescentes o ahorradores. Cuentan con encendido instantáneo pero pueden usarse con atenuadores de luz.

Aplicaciones

Residencial y comercial.
Luz cálida y luz fría.

Características del producto



Especificaciones técnicas

#	CÓDIGO CAJA	BUSTER	CCT	POTENCIA	FLUJO LUMINOSO	ÁNGULO APERTURA	CRI	BASE	VOLTAJE	VIDA ÚTIL	PCS
1	IL070133	-	3000K	9W	806lm	240°	80	E27	110-240V	25000H	50
2	-	IL070055	6500K	9W	806lm	240°	80	E27	110-240V	25000H	50

Nota. Adaptado de <https://improselec.com/wp-content/uploads/2019/06/catalogo-maviju-improselec-1-1.pdf>

Anexo 13 Código implementado para el sistema de control PID y la comunicación con el bróker MQTT

Código implementado

```

/*****
DECLARACIÓN DE LAS LIBRERIAS, DEFINICIÓN DE PINES
GPIO, VARIABLES Y CONSTANTES
* *****/
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>
#include <PID_v1.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <BH1750.h>
BH1750 lightMeter;
#define GATE 16 // Pin D0 para activación de los MOSFET
#define CRUCE0 5 // Pin D1 O GPIO5 para la detección de cruce por cero de la señal de
entrada AC(120V 60Hz) y activación de la ISR
#define PIR 14 // Pin D5 o GPIO14 para la lectura del sensor PIR HC-SR501

float escalado; // Variable para calcular del tiempo de activación del Timer1 en modo de
    
```

```

comparación
float x = 0; // Variable para realizar conversiones
float aux; // Variable auxiliar para establecer estados de proceso
volatile float lux = 0; // Variable que almacena el porcentaje de iluminación extraído de la
lectura del sensor BHT1750
volatile float newlux; // Variable que almacena el valor de iluminación de la lectura del
sensor
volatile float luxref = 300; // Variable que almacena el nivel de iluminación establecido en la
interfaz de usuario (300 luxes inicialmente)
volatile bool zero = 0; // Variable de indicador de estado que muestra si se ha producido
un punto de cruce por cero
volatile int ticks = 20000; // Variable volatile que almacena el número de ticks que va a
escribir el timer1; esto generera los tiempos de encendido del GATE (MODO
AUTOMÁTICO)
volatile int ticks2 = 20000; // Variable volatile que almacena el número de ticks que va a
escribir el timer1; esto generera los tiempos de encendido del GATE (MODO MANUAL)
volatile int pwms; // Variable volatile que almacena el ciclo de trabajo de la señal pwm
de 0-100%
volatile bool LPIR = 0; // Variable volatile que almacena el estado del sensor PIR HC-
SR501
volatile int VALUPIR; // Variable volatile que almacena el estado del sensor PIR HC-
SR501 de 0-100 para visualizacion de la señal de estado lógico
volatile bool ELAMP = 0; // Variable volatile que almacena el estado de encendido o
apagado de la lampara LED
volatile bool CONTROL_LUM = 1; // Variable volatile que determina el tipo de control de
la luminaria AUTOMÁTICA=1, MANUAL=0
double Setpoint, Input, Output; // Variables del controlador PID
double Kp = 1, Ki = 20, Kd = 0.0001; // Parámetros del controlador PID
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT); //Inicia las instancias del
controlador PID
unsigned long previousMillis = 0; // Variable de tiempo que almacenará la última vez que
se actualizó el sensor
const int Ts = 160; // Variables que actualiza las lecturas de los sensores cada Ts
mseg y ejecuta el cálculo del controlador PID
bool aux1; // Variable booleana de estado auxiliar
volatile int ESTPIR; // Variable de estado auxiliar para el sensor PIR
volatile int limitlux = 3050; // Variable que establece el límite de luxes sobre el cual
trabaja el sistema de control PID
volatile int rangolux; // Variable que establece el límite de luxes en un valor de
porcentaje de 0-100% o de 0-1000 luxes
/*****
VARIABLES PARA LA CONEXIÓN WIFI
* *****/
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
char msg[50];
int value = 0;
// Declara constantes para la conexión Wi-Fi local
const char* ssid = "Nombre_Red";

```

```

const char* password = "Contraseña";
// Declara los parámetros de seguridad para la conexión al bróker MQTT
const char* mqtt_server = "192.168.1.107";
const char *mqtt_usuario = "cliente2";
const char *mqtt_clave = "Cont1565sis";
/*****

  FUNCIÓN PARA CONEXIÓN A LA RED WIFI
  *****/
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Conectando a: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
/*****

  FUNCIÓN PARA RECONECTARSE AL BRÓKER Y SUSCRIBIRSE A LOS TEMAS
  *****/
void reconnect() {
  // Bucle hasta que se reconecte
  while (!client.connected()) {
    Serial.print("Intentando conexión MQTT...");
    // Crea un ID de cliente aleatorio
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Intento de conexión usando las credenciales configuradas en el servidor bróker MQTT
    if (client.connect(clientId.c_str(), mqtt_usuario, mqtt_clave )) {
      Serial.println("Cliente conectado");
      client.subscribe("esp8266/output3");
      client.subscribe("esp8266/output4");
      client.subscribe("esp8266/output5");
      client.subscribe("esp8266/output6");
      client.subscribe("esp8266/output7");
    }
    else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" Reintento de conexión en 5 segundos");
      delay(5000);
    }
  }
}
}

```

```

/*****
  FUNCIÓN PARA RECIBIR DATOS DEL BRÓKER
  *****/
void callback(char* topic, byte* message, unsigned int length) {
  Serial.print("Mensaje recibido en topic: ");
  Serial.print(topic);
  Serial.print(", Mensaje: ");
  String messageTemp;
  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp += (char)message[i];
  }
  Serial.println();
  // Mensaje recibido usando el topic o tema: esp8266/output3
  if (String(topic) == "esp8266/output3")
  {
    CONTROL_LUM = 0; // Establece el control de la luminaria en modo MANUAL
    Serial.print("Cambio del controlador a modo MANUAL ");
    Serial.println("");
    if (LPIR == 1) {
      ELAMP = 1;
    }
    else {
      ELAMP = 0;
    }
    volatile int value;
    value = messageTemp.toInt(); // Realiza conversión a tipo enteros
    ticks2 = value; // Variable volatile que almacena el número de ticks que va a
    escribir el timer1; esto generará los tiempos de encendido del GATE
    pwms = (value * 100) / 662400; // Variable que calcula el porcentaje del ciclo de trabajo
    PWM de 0-100%
    Serial.print("Cambio de salida PWM: ");
    Serial.print("\t");
    Serial.print(pwms);
    Serial.println("");
  }
  // Mensaje recibido usando el topic o tema: esp8266/output4
  if (String(topic) == "esp8266/output4")
  {
    CONTROL_LUM = 1; // Establece el control de la luminaria en modo AUTOMÁTICO
    Serial.print("Cambio del controlador a modo AUTOMÁTICO ");
    Serial.println("");
    Serial.print("Cambio de Setpoint: ");
    Serial.print("\t");
    Serial.print(messageTemp);
    Serial.println("");
    luxref = 100; // Cambia el Setpoint del controlador PID a 100 luxes
  }
  // Mensaje recibido usando el topic o tema: esp8266/output5
  if (String(topic) == "esp8266/output5")

```

```

{
CONTROL_LUM = 1; // Establece el control de la luminaria en modo AUTOMÁTICO
Serial.print("Cambio del controlador a modo AUTOMÁTICO ");
Serial.println("");
Serial.print("Cambio de Setpoint: ");
Serial.print("\t");
Serial.print(messageTemp);
Serial.println("");
luxref = 300; // Cambia el Setpoint del controlador PID a 300 luxes
}
// Mensaje recibido usando el topic o tema: esp8266/output6
if (String(topic) == "esp8266/output6")
{
CONTROL_LUM = 1; // Establece el control de la luminaria en modo AUTOMÁTICO
Serial.print("Cambio del controlador a modo AUTOMÁTICO ");
Serial.println("");
Serial.print("Cambio de Setpoint: ");
Serial.print("\t");
Serial.print(messageTemp);
Serial.println("");
luxref = 500; // Cambia el Setpoint del controlador PID a 500 luxes
}
// Mensaje recibido usando el topic o tema: esp8266/output7
if (String(topic) == "esp8266/output7")
{
CONTROL_LUM = 1; // Establece el control de la luminaria en modo AUTOMÁTICO
Serial.print("Cambio del controlador a modo AUTOMÁTICO ");
Serial.println("");
Serial.print("Cambio de Setpoint: ");
Serial.print("\t");
Serial.print(messageTemp);
Serial.println("");
luxref = 800; // Cambia el Setpoint del controlador PID a 800 luxes
}
}
}
/*****
FUNCION PARA ENVIAR DATOS AL BRÓKER CORRESPONDIETES A UN TOPIC
*****/
void enviar_datos() {
// Convierte las variables de tipo double en su representación ASCII y los almacena como
una cadena.
static char luxString[7];
int valorlux = newlux;
dtostrf(valorlux, 6, 2, luxString);
client.publish("/esp8266/luxes", luxString);
Serial.println("");
int valorpir = VALUPIR;
static char pirString[7];
dtostrf(valorpir, 6, 2, pirString);
client.publish("/esp8266/sensorpir", pirString);
}

```

```

}
/*****
LECTURA DEL SENSOR BHT1750
*****/

float leer_lux() {
  //Habilita la comunicación serial para el sensor luxómetro BH1750
  Wire.begin(13, 12); // SDA en D7 o GPIO13 y SCL en D6 O GPIO12 para el luxómetro
  BH1750
  lightMeter.begin(); // Establece el modo de lectura por defecto
  BH1750_CONTINUOUS_HIGH_RES_MODE--->High Resolution Mode - (1 lx precision,
  120ms measurement time)
  newlux = lightMeter.readLightLevel(); // Realiza la lectura del sensor luxómetro en luxes.
  if (isnan(newlux)) { // Si la lectura de la iluminancia falla, no cambia el valor de
  luxes
  Serial.println("Error obteniendo los datos del sensor BH1750!");
  }
  else {
  Serial.print("LUXES: ");
  Serial.print(newlux);
  Serial.print("\t");
  Serial.println("");
  lux = (newlux * 100) / 1000; // Función de escalamiento para control PID, que establece
  un porcentaje de iluminación de 0-100%, o de 0-1000 luxes.
  return lux;
  }
}
/*****
RUTINA DE INTERRUPCIÓN POR DESBORDAMIENTO DEL TIMER1
*****/
void ICACHE_RAM_ATTR onTimer() {
  if (zero == 1) {
    timer1_disable();
    digitalWrite(GATE, HIGH);
    zero = 0;
  }
}
/*****
RUTINA DE INTERRUPCIÓN EXTERNA
*****/
void ICACHE_RAM_ATTR ZerocrossInterrupt() {
  if (ELAMP == 1) {
    digitalWrite(GATE, LOW); // Activa la salida del GATE en 0
    timer1_enable(TIM_DIV1, TIM_EDGE, TIM_SINGLE);
    if (CONTROL_LUM == 1) {
      timer1_write(ticks); // Escribe la salida del GATE con un valor de(20000-662400
      ticks)para el ajuste de brillo de la lampara LED
    }
  }
  else {
    timer1_write(ticks2); // Escribe la salida del GATE con un valor de(20000-662400

```

```

ticks)para el ajuste de brillo de la lampara LED
}
zero = 1;          // Actualiza el estado de detección de cruce por cero a True
}
else {
    digitalWrite(GATE, HIGH); // Activa la salida del GATE en 1
    zero = 0;          //Actualiza el estado de detección de cruce por cero a False
}
}
}
/*****
COMANDOS DE ARRANQUE DEL MCU
* *****/
void setup()
{
    Serial.begin(9600);          // Habilita la comunicación serial a 9600 baudios
    pinMode(PIR, INPUT);        // Declara el pin PIR como entrada
    pinMode(GATE, OUTPUT);      // Declara el pin GATE como salida
    digitalWrite(GATE, HIGH);   // Inicia el estado del GATE en alto, para tener la lampara
    apagada.
    pinMode(CRUCE0, INPUT_PULLUP); // Activa la resistencia Pull-Up del pin D1
    attachInterrupt(digitalPinToInterrupt(CRUCE0), ZerocrossInterrupt, FALLING); // Activa
    la interrupción externa en el pin CRUCE0 por flanco de bajada.
    timer1_attachInterrupt(onTimer); // Activa la interrupción por
    desbordamiento del timer1, y genera una ISR(onTimer)
    timer1_enable(TIM_DIV1, TIM_EDGE, TIM_SINGLE); // Configura
    el timer1 con prescalado TIM_DIV1, temporizador TIM-EDGE y recarga del temporizador
    en SINGLE
    myPID.SetMode(AUTOMATIC); // Establece el controlador PID en modo automático
    myPID.SetTunings(Kp, Ki, Kd); // Declara los parámetros de ajuste del controlador PID
    myPID.SetOutputLimits(0, 4095); // Establece el límite de los valores de salida del
    controlador PID
    Setpoint = luxref * 0.1; // Ajusta el punto de funcionamiento del PID de 0-500 luxes al
    60% que corresponde a 300 luxes
    rangolux = limitlux * 0.1; // Establece el rango máximo en el cual trabajará el controlador
    PID 3050*0.1= 305%
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}
/*****
COMANDOS DE LA FUNCION DE BUCLE DEL MCU
* *****/
void loop() {
    // Conecta al cliente MCU con el bróker MQTT
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    unsigned long now = millis();

```

```

// Función que envía los datos al bróker MQTT cada 2 segundos
if (now - lastMsg > 2000) {
  lastMsg = now;
  enviar_datos();
}
LPIR = digitalRead(PIR); //Realiza lectura del estado del sensor PIR (Alto o Bajo)
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= Ts) { // Función que permite realizar un muestreo del
programa cada Ts mseg
  previousMillis = currentMillis;
  Input = leer_lux(); // Actualiza el porcentaje de iluminación que existe actualmente de
0-100% o de 0-1000 luxes
  if (CONTROL_LUM == 1) { // Verifica si el control de la luminaria está en modo
AUTOMÁTICO
    if (Input < rangolux) { // Si el nivel de luminosidad está por debajo del rango máximo de
operación activa el controlador PID
      if (LPIR == 1) { // Condicional que verifica que la salida del sensor PIR este encendida
        VALUPIR = 300;
        Serial.print("PIR: ");
        Serial.print(VALUPIR);
        Serial.print("\t");
        Serial.print(" Setpoint: ");
        Serial.print(luxref);
        Serial.print("\t");
        double Kp = 1, Ki = 20, Kd = 0.0001;
        myPID.SetTunings(Kp, Ki, Kd); // Establece los nuevos parámetros del
controlador PID
        myPID.SetControllerDirection(DIRECT); // Establece la dirección de la salida del
controlador de forma ascendente
        Setpoint = luxref * 0.1; // Ajusta el setpoint del PID
        myPID.Compute(); // Realiza el cálculo del controlador PID y genera la
señal de control Output
      }
      else { // Condicional que verifica que la salida del sensor PIR este apagada, y permite
realizar un apagado suave de la lampara
        VALUPIR = 10;
        Serial.print("PIR: ");
        Serial.print(VALUPIR);
        Serial.print("\t");
        Serial.print(" Setpoint: ");
        Serial.print(luxref);
        Serial.print("\t");
        Setpoint = 300.0; // Ajusta el setpoint del PID A 3000 luxes
        double Kp = 1, Ki = 3, Kd = 0; // Ajusta los parámetros para realizar un apagado
suave de la lampara en aproximadamente 3 segundos
        myPID.SetTunings(Kp, Ki, Kd); // Establece los nuevos parámetros del
controlador PID
        myPID.SetControllerDirection(REVERSE); // Establece la dirección de la salida del
controlador de forma descendente
        myPID.Compute(); // Realiza el cálculo del controlador PID y genera la

```



```

#include <PubSubClient.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <PZEM004Tv30.h>
//#include <BH1750.h>
#include <DHT.h>
#define DHTPIN 4 // Define el pin digital D2 o GPIO4 conectado al sensor DHT22
#define DHTTYPE DHT22 // Define el tipo de sensor DHT usado, DHT22 para este
trabajo
DHT dht(DHTPIN, DHTTYPE); // Declara el objeto DHT con los parámetros pin y tipo de
DHT.
PZEM004Tv30 pzem(12,13); // Declara los pines TX y RX para la
comunicación pzem(TX,RX)---> pzem(D6,D7) o pzem(GPIO12,GPIO13)
// Variables para la lectura del módulo PZEM004Tv3
volatile float voltage;
volatile float current;
volatile float power;
volatile float energy;
volatile float frequency;
volatile float pf;
//Variables para la lectura del sensor DHT22
volatile float t;
volatile float h;
/*****
VARIABLES PARA LA CONEXIÓN WIFI
*****/
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
char msg[50];
int value = 0;
// Declara las constantes para la conexión Wi-Fi local
const char* ssid = "Nombre_Red";
const char* password = "Contraseña";
// Declara los parámetros de seguridad para la conexión al bróker MQTT
const char* mqtt_server = "192.168.1.107";
const char *mqtt_usuario = "cliente1";
const char *mqtt_clave = "Cont4542sis";

/*****
* FUNCIÓN PARA CONEXIÓN A LA RED WIFI
*****/
void setup_wifi() {

    delay(10);
    Serial.println();
    Serial.print("Conectando a: ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

```

```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi conectado");
Serial.println("Dirección IP : ");
Serial.println(WiFi.localIP());
}
/*****
* FUNCIÓN PARA RECIBIR DATOS DEL BRÓKER
* *****/
void callback(char* topic, byte* message, unsigned int length) {
  Serial.print("Mensaje recibido en topic: ");
  Serial.print(topic);
  Serial.print(", Mensaje: ");
  String messageTemp;
  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp+=(char)message[i];
  }
  Serial.println();
}
/*****
* FUNCIÓN PARA RECONECTARSE AL BRÓKER Y SUSCRIBIRSE A LOS TEMAS
* *****/
void reconnect() {
  // Bucle hasta que se reconecte
  while (!client.connected()) {
    Serial.print("Intentando conexión MQTT...");
    // Crea un ID de cliente aleatorio
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Intento de conexión usando las credenciales configuradas en el servidor bróker MQTT
    if (client.connect(clientId.c_str(), mqtt_usuario, mqtt_clave )) {
      Serial.println("Cliente conectado");
    } else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" Intente de nuevo en 5 segundos");
      delay(5000);
    }
  }
}

/*****
* LECTURA SENSOR PZEM004TV3
* *****/
void leer_pzem004t(){

```

```

//Serial.print("Custom Address:");
//Serial.println(pzem.readAddress(), HEX);
// Lee los datos eléctricos del módulo PZEM004Tv3
voltage = pzem.voltage();
current = pzem.current();
power = pzem.power();
energy = pzem.energy();
frequency = pzem.frequency();
pf = pzem.pf();
// Verifica que lectura de los datos sea válida
if(isnan(voltage)){
  Serial.println("Error en la lectura de voltage!");
} else if (isnan(current)) {
  Serial.println("Error en la lectura de corriente!");
} else if (isnan(power)) {
  Serial.println("Error en la lectura del potencia!");
} else if (isnan(energy)) {
  Serial.println("Error en la lectura de energía!");
} else if (isnan(frequency)) {
  Serial.println("Error en la lectura de frecuencia!");
} else if (isnan(pf)) {
  Serial.println("Error en la lectura del factor de potencia!");
} else {

  // Presenta los datos eléctricos a través del monitor serie
  Serial.print("Voltaje: "); Serial.print(voltage); Serial.println(" [V]");
  Serial.print("Corriente: "); Serial.print(current); Serial.println(" [A]");
  Serial.print("Potencia: "); Serial.print(power); Serial.println(" [W]");
  Serial.print("Energía: "); Serial.print(energy,3); Serial.println(" [kWh]");
  Serial.print("Frecuencia: "); Serial.print(frequency, 1); Serial.println(" [Hz]");
  Serial.print("PF: "); Serial.println(pf);
}

Serial.println("");
}
/*****
* LECTURA DEL SENSOR DHT22
* *****/
void leer_dht22(){
  float newT = dht.readTemperature(); // Leer la temperatura como Celsius (valor por defecto)
  //float newT = dht.readTemperature(true); // Leer la temperatura como Fahrenheit (is Fahrenheit = true)
  if (isnan(newT)) { //Si la lectura de la temperatura falla, no cambie el valor de t
    Serial.println("Error obteniendo los datos del sensor DHT22!");
  }
  else { //Si la lectura de la temperatura es correcta, actualiza el valor de t
    t = newT;
    Serial.print("Temperatura: ");
    Serial.print(t);
  }
}

```

```

Serial.print(" °C \t");
static char tempString[7];
dtostrf(t, 6, 2, tempString);
client.publish("/esp8266/temperature", tempString); // Envía el valor de temperatura al
bróker MQTT
}
float newH = dht.readHumidity(); // Lee la humedad
if (isnan(newH)) { // Si la lectura de la humedad falla, no cambie el valor de h
Serial.println("Error obteniendo los datos del sensor DHT22!");
}
else { // Si la lectura de la humedad es correcta, actualiza el valor de h
h = newH;
Serial.print("Humedad: ");
Serial.print(h);
Serial.print(" %\t");
static char humString[7];
dtostrf(h, 6, 2, humString);
client.publish("/esp8266/humidity", humString); // Envía el valor de humedad al bróker
MQTT
}
float hic = dht.computeHeatIndex(t, h, false); // Calcula el índice de calor en grados
centígrados
Serial.print("Índice de calor: ");
Serial.print(hic);
Serial.print(" °C \t");
Serial.println("");
static char hicString[7];
dtostrf(hic, 6, 2, hicString);
client.publish("/esp8266/indcalor", hicString); // Envía el valor de índice de calor al bróker
MQTT
}
/*****
* FUNCION PARA ENVIAR LOS DATOS A SU TÓPICO DENTRO DEL CORREDOR
* *****/
void enviar_datos(){
// Convierte las variables de tipo double en su representación ASCII y los almacena como
una cadena.
static char voltageString[7];
dtostrf(voltage, 6, 2, voltageString);
static char currentString[7];
dtostrf(current, 6, 2, currentString);
static char powerString[7];
dtostrf(power, 6, 2, powerString);
static char energyString[7];
dtostrf(energy, 6, 3, energyString);
static char freqString[7];
dtostrf(frequency, 6, 2, freqString);
static char pfString[7];
dtostrf(pf, 6, 2, pfString);
client.publish("/esp8266/voltage", voltageString);

```

```

client.publish("/esp8266/current", currentString);
client.publish("/esp8266/power", powerString);
client.publish("/esp8266/energy", energyString);
client.publish("/esp8266/frequency", freqString);
client.publish("/esp8266/pf", pfString);
}
/*****
* COMANDOS DE ARRANQUE DEL MCU
* *****/
void setup() {
  dht.begin();
  Serial.begin(9600);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  //pzem.resetEnergy(); // Resetea el contador de energía activa en kWh
}
/*****
* COMANDOS DE LA FUNCION DE BUCLE DEL MCU
* *****/
void loop() {
  // Conecta al cliente MCU con el bróker MQTT
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  unsigned long now = millis();
  // Función que envía los datos al bróker MQTT cada 2 segundos
  if (now - lastMsg > 2000) {
    lastMsg = now;
    leer_dht22();
    leer_pzem004t();
    enviar_datos();
  }
}
}

```

Anexo 15 Código implementado para el envío de datos a la plataforma ThingSpeak

Código implementado

```

# Importa la librería que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT.
import paho.mqtt.client as mqtt
#Importa Python urllib3 que permite capturar datos, publicar datos, transmitir datos, trabajar
con JSON
import urllib3

```

```

# Define la clave API generada para escribir en ThingSpeak
myAPI = 'S4M1A93NDZUF42KY'
# Define la URL generada en ThingSpeak
baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI
http = urllib3.PoolManager() #Crea una instancia PoolManager para realizar solicitudes
# Define las variables globales de tipo str para el envío de la información
val=""
val2=""
val3=""
val4=""
val5=""
val6=""
# Se llama cuando el cliente recibe un mensaje CONNACK (Acuse de recibo de conexión)
del bróker en respuesta a la conexión aceptada
def on_connect(client,userdata, flags, rc):
    print("La conexión devolvió el resultado: "+str(rc))
    #Suscribe al cliente a los diferentes temas
    client.subscribe("/esp8266/temperature")
    client.subscribe("/esp8266/humidity")
    client.subscribe("/esp8266/luxes")
    client.subscribe("/esp8266/power")
    client.subscribe("/esp8266/energy")
    client.subscribe("/esp8266/pf")

# Se llama cuando se recibe un mensaje sobre un tema al que se suscribe el cliente
def on_message(client, userdata, message):
    print("Mensaje recibido " + str(message.payload) + " sobre el tema " + message.topic)
    # Inicializa las variables
    humi = 0
    temp = 0
    lux=0
    pot=0
    energ=0
    pf=0
    # Verifica el tema o topic del mensaje de llegada y actualiza el valor de las variables
    if message.topic == "/esp8266/temperature":
        print("Temperatura actualizada")
        temp = str(message.payload, 'UTF-8')
        temp = temp.strip()
        print(temp)
        global val
        val = temp

    if message.topic == "/esp8266/humidity":
        print("Humedad actualizada")
        humi = str(message.payload, 'UTF-8')
        humi = humi.strip()
        print(humi)
        global val2

```

```

val2 = humi

if message.topic == "/esp8266/luxes":
    print("Luminosidad actualizada ")
    lux = str(message.payload, 'UTF-8')
    lux = lux.strip()
    print(lux)
    global val3
    val3 = lux

if message.topic == "/esp8266/power":
    print("Potencia actualizada ")
    pot = str(message.payload, 'UTF-8')
    pot = pot.strip()
    print(pot)
    global val4
    val4 = pot

if message.topic == "/esp8266/energy":
    print("Energía actualizada ")
    energ = str(message.payload, 'UTF-8')
    energ = energ.strip()
    print(energ)
    global val5
    val5 = energ

if message.topic == "/esp8266/pf":
    print("PF actualizado ")
    pf = str(message.payload, 'UTF-8')
    pf = pf.strip()
    print(pf)
    global val6
    val6 = pf

# Verifica que los valores no sean nulos
if val != "" and val2 != "" and val3 != "" and val4 != "" and val5 != "" and val6 != "":
    # El método http.request envía una solicitud GET a la base URL para escribir los valores
    # calculados dentro de los campos field en ThingSpeak, y obtiene una respuesta.
    conn = http.request('GET', baseURL +
    '&field1=%s&field2=%s&field3=%s&field4=%s&field5=%s&field6=%s'%(val,val2,val3,v
    al4,val5,val6))
    # Imprime el estado de la respuesta conn
    print(conn.status)
    conn.read()
    conn.close()
    val = ""
    val2 = ""
    val3 = ""
    val4 = ""
    val5 = ""

```

```
val6 = "  
  
def main():  
    mqtt_client = mqtt.Client()  
    # Se establece el usuario y contraseña para la autenticación con el bróker MQTT  
    mqtt_client.username_pw_set(username="cliente1",password="Cont4542sis")  
    mqtt_client.on_connect = on_connect  
    mqtt_client.on_message = on_message  
    # Se conecta al cliente con el intermediario --> connect(host, port, keepalive)  
    mqtt_client.connect('localhost', 1883, 60)  
    # Conecta con el servidor MQTT y procesa los mensajes en un hilo de fondo.  
    mqtt_client.loop_start() #Esta llamada también maneja la reconexión con el bróker.  
  
    # Función Principal  
if __name__ == '__main__':  
    main()
```

Anexo 16 Visualización de los canales del sistema de iluminación inteligente en ThingSpeak





FINE-TUNED ENGLISH LANGUAGE INSTITUTE

Líderes en la Enseñanza del Inglés

Ing. María Belén Novillo Sánchez.

ENGLISH TEACHER - FINE TUNED ENGLISH CIA LTDA.

CERTIFICA:

Que el documento aquí compuesto es fiel traducción del idioma español al idioma inglés del trabajo de titulación **“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA INTELIGENTE DE ILUMINACIÓN Y MONITOREO EN TIEMPO REAL DE VARIABLES AMBIENTALES PARA LA EMPRESA DE TECNOLOGÍA Y SERVICIOS TECSERLED”** autoría de **Jordy Carlos Carrión Lojan** con número de cédula **1104312747**, estudiante de la Carrera de Ingeniería en Electrónica y Telecomunicaciones de la Universidad Nacional de Loja.

Lo certifico en honor a la verdad y autorizo al interesado hacer uso del presente en lo que a sus intereses convenga.

Loja, 16 de agosto del 2022



Ing. María Belén Novillo Sánchez.

ENGLISH TEACHER- FINE TUNED ENGLISH CIA LTDA.

Líderes en la Enseñanza del Inglés

Matriz - Loja: Macará 205-51 entre Rocafuerte y Miguel Riofrío - Teléfono: 072578899
Zamora: García Moreno y Pasaje 12 de Febrero - Teléfono: 072608169
Yantzaza: Jorge Mosquera y Luis Bastidas - Edificio Sindicato de Choferes - Teléfono: 072301329

www.fte.edu.ec