

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad De Ingeniería En Ciencias Aplicadas  
Carrera De Ingeniería En Mantenimiento Eléctrico

## **IOT PARA LA SUPERVISIÓN Y CONTROL DE CIRCUITOS DE ILUMINACIÓN**

Trabajo de grado presentado ante la ilustre universidad técnica del norte previo a la obtención del título de Ingeniero en mantenimiento eléctrico

Autor:

Diego Fernando Puetate Paspuezan

Director:

MSc. Francisco Roberto Naranjo Cobo

IBARRA, 2022



**UNIVERSIDAD TÉCNICA DEL NORTE  
BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN  
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

**1. IDENTIFICACIÓN DE LA OBRA**

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CÉDULA DE IDENTIDAD:</b>	0401635875		
<b>APELLIDOS Y NOMBRES:</b>	Puetate Paspuezan Diego Fernando		
<b>DIRECCIÓN:</b>	Sucre y Quito (Tulcán-Ecuador)		
<b>EMAIL:</b>	dfpuetatep@utn.edu.ec		
<b>TELÉFONO FIJO:</b>	256052	<b>TELÉFONO MÓVIL:</b>	0986768110

DATOS DE LA OBRA	
<b>TÍTULO:</b>	IOT PARA LA SUPERVISIÓN Y CONTROL DE CIRCUITOS DE ILUMINACIÓN
<b>AUTOR (ES):</b>	Puetate Paspuezan Diego Fernando
<b>FECHA: DD/MM/AAAA</b>	01 de febrero del 2022
SOLO PARA TRABAJOS DE GRADO	
<b>PROGRAMA:</b>	PREGRADO
<b>TITULO POR EL QUE OPTA:</b>	Ingeniería en Mantenimiento Eléctrico
<b>ASESOR /DIRECTOR:</b>	MSc. Francisco Roberto Naranjo Cobo

**2. CONSTANCIAS**

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 01 días del mes de febrero de 2022

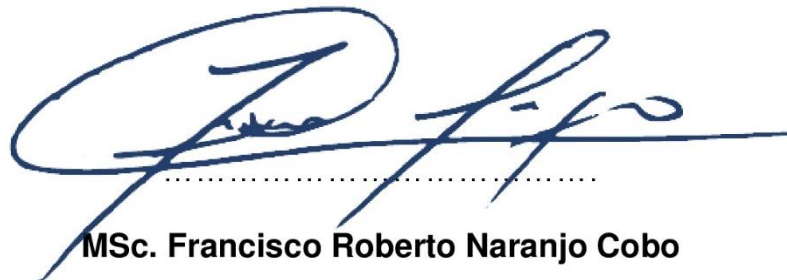
**EL AUTOR:**

(Firma):

Nombre: Puetate Paspuezan Diego Fernando

## CERTIFICACION DEL DIRECTOR DE TRABAJO DE GRADO

Yo, Francisco Roberto Naranjo Cobo en calidad de tutor del señor, Puetate Paspuezan Diego Fernando certifico que ha cumplido con las normas establecidas en la elaboración del trabajo de investigación titulado: **“IoT para la supervisión y control de circuitos de iluminación”**. Para la obtención de título de Ingeniero en Mantenimiento Eléctrico; aprobando la defensa, impresión y empastado.

A handwritten signature in blue ink, consisting of a large, stylized initial 'F' followed by 'R. Naranjo Cobo' in a cursive script. The signature is written over a horizontal dotted line.

**MSc. Francisco Roberto Naranjo Cobo**

**DIRECTOR DE TRABAJO DE GRADO**

## **DEDICATORIA**

Dedico este trabajo a mi familia, Tomas mi hijo y de manera muy especial a la persona que lo hizo todo posible y me enseñó que al final de la tormenta existe la calma a mi esposa y compañera de vida Tania Cuacés por regalarme el don de la paciencia y además por estar conmigo en todo momento alentándome, impulsándome a ser mejor y enseñándome a creer en mí.

## **AGRADECIMIENTO**

A mis padres María Paspuezan y Marcial Puetate por estar siempre brindándome sus consejos e impulsándome a trabajar duro cada día, a mi hermana por el afecto la amistad y todos los momentos compartidos en el transcurso de toda la vida universitaria.

A todas aquellas personas que estuvieron conmigo apoyándome de una u otra manera y me impulsaron a seguir mejorando cada día para demostrar que se puede lograr lo que se propone en la vida.

De manera muy especial al MSc. Francisco Naranjo por ser participe en la elaboración de este proyecto contribuyendo con sus conocimientos, tiempo y paciencia, a mis amigos de los cuales aprendí a fortalecer muchas debilidades y estuvieron presentes en muchas horas de estudio y compartiendo vivencias día a día tanto dentro como fuera de las aulas.

**PUETATE PASPUEZAN DIEGO FERNANDO**

## RESUMEN

El presente proyecto basa su desarrollo en el manejo del control de iluminación en un entorno controlado como es el laboratorio de electricidad de la facultad Fica en la universidad técnica del norte, utilizando la tecnología orientada al IoT se procede a controlar por medio de sensores y actuadores la regulación de la potencia de las luminarias para generar un ambiente óptimo de luminancia para los docentes y estudiantes que hacen uso de las instalaciones.

El proyecto se realizó tomando en cuenta una normativa de lúmenes por ambiente, para obtener los valores requeridos se utilizó sensores de luxes en el entorno los cuales se conectan de forma inalámbrica por medio del protocolo MQTT a un servidor que se encarga de almacenar estos datos en una base de datos y presentarlos al usuario final de manera que pueda mantener monitorización en tiempo real cada uno de los sensores, el sistema se encarga de controlar la potencia por medio de TRIACs controlados por un microcontrolador para establecer las medidas en ambiente normadas, también se procede a controlar una persiana de forma remota.

De los resultados obtenidos se puede concluir que los módulos funcionan de forma íntegra entre ellos, el tiempo de censado y actuado solo vienen limitadas por las tecnologías que se utilizan dentro del proyecto que para este caso son RS-232 y MQTT por medios inalámbricos. El sistema funciona de manera automática y manual según se requiera por el usuario.

## **ABSTRACT**

This project bases its development on the management of lighting control in a controlled environment such as the electricity laboratory of the Fica faculty at the technical university of the north, using IoT oriented technology we proceed to control by means of sensors and actuators the regulation of the power of the luminaires to generate an optimal environment of luminance for teachers and students who make use of the facilities.

The project was carried out taking into account a normative of lumens per environment, to obtain the required values were used luxes sensors in the environment which are connected wirelessly via the MQTT protocol to a server that is responsible for storing these data in a database and present them to the end user so you can maintain motorization in real time each of the sensors, the system is responsible for controlling the power through TRIACs controlled by a microcontroller to establish measures in standard environment, it is also proceeds to control a blind remotely.

From the results obtained it can be concluded that the modules work integrally with each other, the time of sensing and acting are only limited by the technologies used within the project, which in this case are RS-232 and MQTT by wireless means. The system works automatically and manually as required by the user.

## TABLA DE CONTENIDOS

PORTADA .....	I
IDENTIFICACIÓN DE LA OBRA .....	II
CONSTANCIAS.....	II
CERTIFICACION DEL DIRECTOR.....	III
DEDICATORIA .....	IV
AGRADECIMIENTO .....	IV
RESUMEN.....	V
ABSTRACT .....	VI
TABLA DE CONTENIDOS.....	VII
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS .....	XII
Introducción.....	1
Antecedentes .....	1
1.1. Planteamiento del Problema.....	1
1.2. Objetivos .....	1
1.2.1. Objetivo General.....	1
1.2.2. Objetivos Específicos.....	1
1.3. Alcance .....	1
1.1. Justificación.....	2
Capítulo I: Fundamentos Teóricos .....	3
2.1. Internet de las Cosas (IoT) .....	3
2.1.1. Introducción.....	3
2.1.2. Concepto.....	3
2.1.3. Arquitectura .....	4
2.1.4. Aplicaciones .....	6
2.1.5. Protocolos de comunicación.....	7
2.2. Norma europea sobre la iluminación para interiores-UNE 12464.1.....	8
2.2.1. Requisitos de iluminación según actividad.....	9
2.2.2. Establecimientos Educativos .....	9
2.3. Luminarias .....	11

2.3.1.	Flujo luminoso .....	11
2.3.2.	Iluminación o iluminancia .....	11
2.3.3.	Intensidad luminosa .....	12
2.3.4.	Luminancia .....	12
2.3.5.	Rendimiento y eficiencia luminosa .....	13
2.3.6.	Curva de distribución luminosa .....	14
2.3.7.	Deslumbramiento .....	14
2.4.	Raspberry Pi.....	16
2.4.1.	Modelos.....	16
2.5.	Microcontrolador .....	20
2.5.1.	Arduino.....	20
2.5.2.	Expressif-ESP .....	23
2.6.	Sensores de luz .....	27
2.6.1.	Fotorresistencia o LDR .....	27
2.6.2.	BH1750 .....	28
2.7.	Motor DC.....	28
Capítulo II: Desarrollo .....		30
3.1.	Desarrollo Experimental .....	30
3.2.	Metodología.....	30
3.3.	Modelo en V .....	30
3.3.1.	Análisis.....	31
3.3.2.	Propósito y ámbito del sistema .....	33
3.3.3.	Requerimientos.....	33
3.3.4.	Elección de Hardware y Software para el diseño.....	38
3.3.5.	Recursos.....	41
3.4.	Diseño del prototipo .....	42
3.4.1.	Módulo de Control .....	44
3.4.3.	Módulo de Distribución de Potencia .....	50
3.4.5.	Módulo de Censado .....	59
3.5.	Implementación.....	64
3.5.1.	Módulo de Control y distribución de potencia .....	64
3.5.2.	Módulo de Censado .....	66
3.6.	Pruebas de funcionamiento .....	68
3.6.1.	Prueba #1: .....	68



3.6.2.	Prueba #2:	69
3.6.3.	Prueba #3:	70
3.6.4.	Prueba #4:	71
3.6.5.	Prueba #5:	72
Capitulo III: Resultados		74
4.1.	Resultados	74
4.1.1.	Prueba #1:	74
4.1.2.	Prueba #2:	75
4.1.3.	Prueba #3:	75
4.1.4.	Prueba #4:	76
4.1.5.	Prueba #5:	76
Conclusiones		78
Recomendaciones		79
Bibliografía		80
Anexos		81
Anexo 1. Código control de cruce por cero Arduino IDE.		81
Anexo 2. Código control de fase dimmer Arduino UNO		82
Anexo 3. Código lectura sensor BH1750 con Esp32		83
Anexo 4. Diseño caja sensor en 3d		85
Anexo 5. Hoja de especificaciones MOC3021		86
Anexo 6. Hoja de especificaciones TRIAC BT136-600D		88
Anexo 7. Hoja de especificaciones PC817c		89
Anexo 8. Manual de operación y mantenimiento de prototipo		90
Anexo 9. Código Python pendiente.py		92
Anexo 10. Instalación de Node-RED en Raspberry Pi		94
Anexo 11. Instalación de Grafana		95
Anexo 12. Código Python arduinoS.py		96
Anexo 13. Código Python lectura.py		96
Anexo 14. Código Python promedio.py		96
Anexo 15. Código Python publicar.py		97

## ÍNDICE DE FIGURAS

Fig. 1. Modelo de referencia IoT. Fuente: (UIT-T, 2012) .....	5
Fig. 2. Modelo MQTT. Fuente: (Shaw, 2018) .....	7
Fig. 3. Flujo Luminoso. Fuente: (Castro Guaman & Posligua Murillo, 2015). .....	11
Fig. 4. Iluminancia. Fuente: (Castro Guaman & Posligua Murillo, 2015). .....	12
Fig. 5. Intensidad Luminosa. Fuente: (Castro Guaman & Posligua Murillo, 2015).....	12
Fig. 6. Luminancia. Fuente: (Castro Guaman & Posligua Murillo, 2015). .....	13
Fig. 7. Rendimiento. Fuente: (Castro Guaman & Posligua Murillo, 2015). .....	13
Fig. 8. Curva Fotométrica. Fuente: (Castro Guaman & Posligua Murillo, 2015).....	14
Fig. 9. Deslumbramiento. Fuente: (Castro Guaman & Posligua Murillo, 2015). .....	14
Fig. 10. Pinout Raspberry Pi 1. Fuente: (Shaw, 2018) .....	16
Fig. 11. Pinout Raspberry Pi 2. Fuente: (Shaw, 2018).....	17
Fig. 12. Pinout Raspberry Pi 3. Fuente: (Crespo, 2017).....	18
Fig. 13. Pinout Raspberry Pi 4. Fuente: (WONKSKNOW LLC, 2019).....	19
Fig. 14. Pinout Arduino Uno. Fuente: (Teja, 2021). .....	21
Fig. 15. Pinout Arduino Mega. Fuente: (Teja, 2021). .....	22
Fig. 16. Pinout Arduino Nano. Fuente: (Damián, 2020).....	23
Fig. 17. Pinout ESP8266. Fuente: (Naylamp Mechatronics, 2021).....	23
Fig. 18. Pinout Esp32. Fuente: (Naylamp Mechatronics, 2021) .....	25
Fig. 19. Pinout Esp01. Fuente: (Naylamp Mechatronics, 2021) .....	26
Fig. 20. Fotorresistencia. Fuente: (prometec, 2021).....	27
Fig. 21. Pinout BH1750. Fuente: (Naylamp Mechatronics, 2021) .....	28
Fig. 22. Motor DC. Fuente. Bibliográfica (Pernía, 2011) .....	29
Fig. 23. Etapas modelo en V. Fuente: Elaborado por el autor .....	30
Fig. 24. Medidas Laboratorio de máquinas y toma de luxes. Fuente: Elaborado por el autor.....	31
Fig. 25. Conexión eléctrica laboratorio de máquinas. Fuente: Elaborado por el autor .....	32
Fig. 26. Medidas ventana laboratorio de máquinas. Fuente: Elaborado por el autor.....	33
Fig. 27. Medidas soporte luminarias laboratorio de máquinas. Fuente: Elaborado por el autor .	33
Fig. 28. Sistemas del proyecto. Fuente: Elaborado por el autor.....	43
Fig. 29. Sistemas del proyecto. Fuente: Elaborado por el autor.....	43
Fig. 30. Módulo de Control. Fuente: Elaborado por el autor .....	44
Fig. 31. Diagrama de Flujo Módulo de Control. Fuente: Elaborado por el autor.....	45
Fig. 32. Interfaz de Node-RED. Fuente: Elaborado por el autor.....	46
Fig. 32. Nodo MQTT. Fuente: Elaborado por el autor .....	46
Fig. 34. Nodo MQTT. Fuente: Elaborado por el autor .....	46
Fig. 35. Visualización lúmenes. Fuente: Elaborado por el autor.....	47
Fig. 36. Visualización lúmenes. Fuente: Elaborado por el autor.....	47
Fig. 37. Interfaz control motor persiana. Fuente: Elaborado por el autor .....	47

Fig. 38. Interfaz Grafana lúmenes por tiempo. Fuente: Elaborado por el autor .....	48
Fig. 39. Funcionamientos tópicos. Fuente: Elaborado por el autor .....	48
Fig. 40. Estructura influxdb. Fuente: Elaborado por el autor .....	49
Fig. 41. Diagrama de Flujo código en Python. Fuente: Elaborado por el autor .....	50
Fig. 42. Detector de onda completo. Fuente: Elaborado por el autor .....	51
Fig. 43. Señal detectora de onda completo. Fuente: Elaborado por el autor .....	52
Fig. 44. Circuito cruce por cero con Arduino. Fuente: Elaborado por el autor .....	52
Fig. 45. Circuito control de fase tipo dimmer. Fuente: Elaborado por el autor .....	53
Fig. 46. Esquemático Kicad. Fuente: Elaborado por el autor.....	54
Fig. 47. Circuito para imprimir Kicad. Fuente: Elaborado por el autor.....	54
Fig. 48. Circuito impreso CNC. Fuente: Elaborado por el autor .....	55
Fig. 49. Circuito con soldadura de componentes. Fuente: Elaborado por el autor.....	56
Fig. 50. Luminarias con valor 7000. Fuente: Elaborado por el autor .....	56
Fig. 51. Luminarias con valor 6000. Fuente: Elaborado por el autor .....	57
Fig. 52. Luminarias con valor 2000. Fuente: Elaborado por el autor .....	57
Fig. 53. Tabla de valores lúmenes por tiempo (ms). Fuente: Elaborado por el autor .....	58
Fig. 54. Variación lúmenes en tiempo(ms). Fuente: Elaborado por el autor.....	58
Fig. 55. Circuito alimentación Esp8266. Fuente: Elaborado por el autor .....	60
Fig. 56. Conexión Esp8266 BH1750. Fuente: Elaborado por el autor.....	60
Fig. 57. Circuito de censado. Fuente: Elaborado por el autor.....	61
Fig. 58. Diagrama de Flujo módulo de censado. Fuente: Elaborado por el autor .....	62
Fig. 59. Diagrama de bloques persiana motorizada. Fuente: Elaborado por el autor.....	63
Fig. 60. Diagrama de flujo motor persiana. Fuente: Elaborado por el autor .....	64
Fig. 61. Laboratorio Electricidad. Fuente: Elaborado por el autor .....	64
Fig. 62. Cambio de luminarias. Fuente: Elaborado por el autor .....	65
Fig. 63. Conexión luminarias. Fuente: Elaborado por el autor.....	65
Fig. 64. Conexión eléctrica en los laboratorios. Fuente: Elaborado por el autor .....	66
Fig. 65. Conexión eléctrica en los laboratorios. Fuente: Elaborado por el autor .....	66
Fig. 66. Instalación de servo motor en persiana. Fuente: Elaborado por el autor .....	67
Fig. 67. Instalación de servo motor en persiana. Fuente: Elaborado por el autor .....	67
Fig. 68. Sensores de lúmenes. Fuente: Elaborado por el autor .....	68
Fig. 69. Conexión Modulo control y Modulo de distribución de potencia. Fuente: Elaborado por el autor .....	69
Fig. 70. Conexión Modulo de censado .....	69
Fig. 71. Conexión Modulo control y Modulo de distribución de censado .....	70
Fig. 72. Conexión Modulo control y control de persiana.....	71
Fig. 73. Conexión Módulos .....	72
Fig. 74. Intensidad luminaria. Fuente: Elaborado por el autor .....	74
Fig. 75. Lectura de sensores. Fuente: Elaborado por el autor.....	75
Fig. 76. Lectura de sensores. Fuente: Elaborado por el autor.....	76
Fig. 77. Botones control de motor. Fuente: Elaborado por el autor .....	76
Fig. 78. Botones control de motor. Fuente: Elaborado por el autor .....	77

## ÍNDICE DE TABLAS

TABLA 1 ÍNDICE DE DESLUMBRAMIENTO.....	9
TABLA 2 NIVELES DE ILUMINACIÓN EN EDIFICIOS EDUCATIVOS.....	10
TABLA 3 ÍNDICE DE DESLUMBRAMIENTO.....	15
TABLA 4 COMPARATIVA HARDWARE MODELOS RASPBERRY PI.....	19
TABLA 5 LECTURA LUMENES EN LUMINARIAS DENTRO DEL LABORATORIO DE MAQUINAS.....	32
TABLA 6 ABREVIATURA DE REQUERIMIENTOS.....	34
TABLA 7 REQUERIMIENTOS OPERACIONALES.....	34
TABLA 8 REQUERIMIENTOS DEL SISTEMA.....	35
TABLA 9 REQUERIMIENTOS DE LA ARQUITECTURA.....	36
TABLA 10 VALORACIÓN REQUERIMIENTO DE LENGUAJE DE PROGRAMACIÓN.....	38
TABLA 11 VALORACIÓN REQUERIMIENTO DE PLATAFORMA IOT.....	38
TABLA 12 VALORACIÓN REQUERIMIENTO DE HARDWARE SISTEMA DE POTENCIA.....	39
TABLA 13 VALORACIÓN REQUERIMIENTO DE HARDWARE SISTEMA DE CENSADO.....	40
TABLA 14 VALORACIÓN REQUERIMIENTO DE HARDWARE SISTEMA DE CONTROL.....	40
TABLA 15 RECURSOS HUMANOS.....	41
TABLA 16 RECURSOS ECONÓMICOS.....	41
TABLA 17 RECURSOS TECNOLÓGICOS.....	42
TABLA 18 MATERIALES SISTEMA REGULADOR DE POTENCIA.....	54
TABLA 19 POTENCIA LÚMENES POR TIEMPO (MS).....	57
TABLA 20 RESULTADOS MÓDULOS DE CENSADO.....	75
TABLA 21 RESULTADOS PRACTICAS.....	77

## **Introducción**

### **Antecedentes**

#### **1.1. Planteamiento del Problema**

Uno de los aspectos más importantes para tener buena salud laboral es la iluminación. La inadecuada iluminación es contraproducente dentro del entorno laboral o educativo, el exceso o la falta de luz en el ambiente puede provocar: fatiga visual, dolor de cabeza, irritabilidad etc. (Quirónprevención, 2019).

En algunos lugares la iluminación puede llegar a ser excesiva o carente para determinadas tareas de aquí nace la necesidad de decrementar o incrementar el nivel iluminación de alguna forma. Por otra parte, debido al aumento o disminución de la cantidad de luz natural según la hora del día, es necesario aprovechar al máximo la luz natural ahorrando energía y prolongando la vida útil de las luminarias.

¿Cómo controlar la luminosidad en espacios cerrados de forma automática?

#### **1.2. Objetivos**

##### **1.2.1. Objetivo General**

- Desarrollar un prototipo de sistema IoT para el control de luminosidad mediante un microcontrolador.

##### **1.2.2. Objetivos Específicos**

- Elaborar el marco teórico que fundamente el control de luminosidad orientada a IoT.
- Diseñar un prototipo de sistema IoT para el control de luminarias.
- Construir el prototipo de sistema IoT.

#### **1.3. Alcance**

El presente proyecto de investigación busca realizar la implementación de un prototipo de sistema IoT en el laboratorio de Máquinas Eléctricas de la carrera de Electricidad en la Universidad Técnica del Norte, el cual permita realizar el control automático, así como control remoto de circuitos de iluminación vía wifi, el sistema debe controlar luminarias tipo led conectadas a la red de corriente alterna y una persiana controlada mediante microcontroladores a los cuales estarán conectados sensores de luminosidad y actuadores para controlar la cantidad de luz natural y artificial según los requerimientos del usuario. La visualización y gestión del

sistema se desarrollará a través de una página web alojada en un Raspberry Pi permitiendo el acceso desde cualquier dispositivo que esté conectado en la red local.

Se usará como referencia la normativa española UNE 12464.1.

### **1.1. Justificación**

Una correcta iluminación es esencial para poder desempeñar tareas sin dificultades ya sea en el trabajo, en el lugar de estudios o en nuestra residencia. Si la iluminación es deficiente, aumenta la posibilidad de que los trabajadores, estudiantes o cualquier persona, cometan fallos y como consecuencia puedan producirse accidentes, también como otra consecuencia está la fatiga visual que deriva en un sin número de problemas, como dolor de cabeza, cansancio, irritabilidad, mal humor etc. (quirónprevención, 2019).

Al hablar de sistemas de monitoreo y control de iluminación se hace alusión a una solución usada con todo tipo de tecnologías lumínicas que permiten monitorear y controlar de manera práctica, sencilla y personalizable cualquier sistema de iluminación artificial. Por lo general estos sistemas se han concebido únicamente para llevar a cabo tareas propias de control como: el encendido y apagado de luces, control horario, etc.

Varios son los beneficios que se obtiene a través de los sistemas de iluminación como: el aprovechamiento de la iluminación natural, el uso de la energía cuando es absolutamente necesario dando lugar al ahorro energético y la protección del medio ambiente, prolongamos la vida útil de las luminarias, todo esto nos permite tener espacios adaptados y personalizados a las necesidades del usuario. (Electric, 2020)

Según (Electronics, 2017) un control de iluminación adecuado puede tener varios beneficios en diversos aspectos como:

- Mayor comodidad y aumento de la productividad debido a la utilización de atenuadores, sensores y persianas, generando un ambiente visual cómodo en el que se evita la fatiga visual.
- Ahorro económico al implementar atenuadores ya que las luminarias no se encienden al 100% de su capacidad lumínica generando un ambiente optimo en combinación con la luz natural.
- Ahorro económico al prolongar la vida útil de las luminarias

Además, los sistemas de control de iluminación son sistemas que nos permiten adaptar la iluminación de los espacios a diferentes escenarios, con la finalidad de realizar un ahorro en el consumo de energía eléctrica y mejorar el confort de las personas (Guasch, 2019).

# Capítulo I: Fundamentos Teóricos

El objetivo de este capítulo es la presentación teórica de cada concepto que se ve relacionado de forma directa e indirecta con la implementación de un sistema de control de luminosidad por medio de IoT.

Se presenta los conceptos de lo que compete al internet de las cosas, su arquitectura, aplicaciones y protocolos de comunicación. Luego se presenta la normativa utilizada para normar la iluminación en base al tipo de actividad o sitio donde se desempeña una labor humana.

Por último, se procede a detallar de forma teórica en base a características y funcionalidad los componentes que se utilizaran para la elaboración del proyecto a implementar.

## 2.1. Internet de las Cosas (IoT)

### 2.1.1. Introducción

La evolución de las herramientas tecnológicas como sensores o actuadores se produce por la industrialización de las tecnologías y su producción de forma masiva, lo que permite que se logre avances para mejorar procesamiento y reducir de costos en los componentes electrónicos, lo que ha significado que muchos usuarios desarrollen sus propios sistemas de censado y control de procesos en sitios como hogares y lugares de trabajo. Esta tendencia genera que exista una gran demanda de procesos y datos que gracias a nuevas tecnologías y conceptos como son la domótica y el almacenamiento de datos en la nube, permita al usuario final elaborar un control remoto en tiempo real, en base a los datos recolectados con los sensores y utilizar los actuadores para la activación de procesos por medio de plataformas o privadas amigables con el usuario final.

Por esta razón desde hace algunos años el internet de las cosas se ha convertido en un realidad en las sociedades, tomando en consideración que el concepto de comunicar computadores o dispositivos inteligentes con sensores ya se ha visto implementado y actualmente casi todo dispositivo del uso diario en un hogar o sitio de trabajo puede ser enlazado a un servidor para la recolección y análisis de datos, la tecnología se presenta como una gran ventaja para la toma de decisiones y desarrollo de nuevas tecnologías que mejoren procesos.

### 2.1.2. Concepto

Se trata de una tecnología que como concepto general se trata de la conexión de objetos de uso diario a Internet, los mismos que se encargan de intercambiar, agregar y procesar información sobre el entorno que los rodea y agregarles un valor añadido a los usuarios finales. En última instancia se trata de eliminar las barreras entre los objetos físicos y su representación de datos en las redes de comunicación y las aplicaciones.

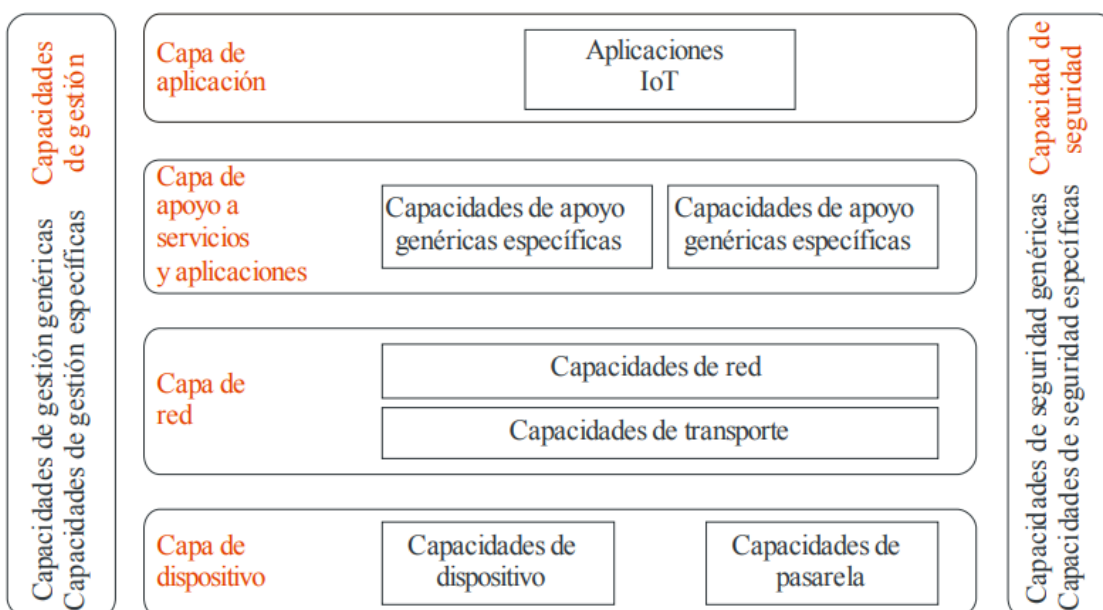
El término Internet de las Cosas (IoT) fue acuñado por primera vez por el pionero de la tecnología británica Kevin Ashton<sup>9</sup> en una presentación que realizó en 1999 para la multinacional Procter & Gamble, donde describía un sistema en el cual los objetos en el mundo físico podrían conectarse a Internet a través de sensores para automatizar la recogida de datos, propugnando su aplicación en la cadena de suministro añadiendo etiquetas RFID (o identificación por radiofrecuencia, *Radio Frequency Identification*), que más adelante examinaremos con detalles. (Andres, Internet de las cosas, 2018)

Concretamente, el padre del término destacaba cómo, hasta esa fecha, la información en la cadena de suministro era introducida de forma manual por las personas, con los consiguientes retrasos y posibles errores. Sin embargo, si la información provenía directamente de los objetos, entonces se podría hacer un *seguimiento en tiempo real* de su utilización, sus características, su vida útil, la necesidad de nuevos aprovisionamientos, su estado de funcionamiento, etc., lo cual se traduciría en aumentos de productividad y consiguiente reducción de costes. (Andres, Internet de las cosas, 2018)

Una realidad en donde los objetos físicos de uso diario se integran perfectamente en la red de información, en la cual los objetos físicos pueden convertirse actores participativos en los procesos empresariales usando los servicios disponibles para convertirse en un objeto en inteligente a través del internet, además utilizando todos los recursos necesarios para dar seguridad y privacidad a estos procesos.

### 2.1.3. Arquitectura

Es una arquitectura multicapa en donde los objetos inteligentes se enlazan y se utilizan para brindar varios servicios por medio de las capas que son: capa aplicación, capa soporte de servicio y aplicación, capa de red y la capa de dispositivo.





Como se muestra en la Figura 1 se tiene las siguientes etapas y se procede a describirlas:

- **Capa de aplicación.**

Se refiere a las aplicaciones HMI que interactúan con el usuario final.

- **Capa de soporte de servicios y aplicaciones.**

Consiste en dos grupos de capacidades:

- Capacidades de soporte genéricas: Se refiere a cómo utilizar diferentes aplicaciones IoT que son procesamiento y almacenamiento de datos.
- Capacidades de soporte específicas: Son diversos grupos de capacidades precisas para distintas funciones para aplicaciones IoT.

- **Capa de red.**

Consiste en dos grupos que son:

- Capacidades de red: Ofrecen funciones para el control de acceso y recursos de transporte, gestión de movilidad y autenticación, autorización y contabilización.
- Capacidad de transporte: es la forma en que se transporta la información y datos de servicios.

- **Capa de dispositivo.**

Se utiliza una clasificación lógica de las capacidades de la capa de dispositivos que son:

- Interacción directa con la red de comunicaciones: representa que los dispositivos inteligentes pueden recabar y cargar información directamente con la red de comunicación.
- Interacción indirecta con la red de comunicación: Representa los datos que se actualizan hacia los dispositivos desde las redes de comunicación por medio de pasarelas o VPNs.
- Redes ad-hoc: Se puede construir redes de manera ad-hoc para mejorar la capacidad evolutiva y la velocidad de despliegue.
- Modos reposo y activo: Las capacidades de dispositivo para pasar de modo reposo a activo para ahorrar energía.
- Soporte de interfaces múltiples: Se puede utilizar diferentes buses de red de control mediante distintas tecnologías como son ZigBee, Bluetooth, Wifi, redes 2G-3G-4G, redes Ethernet.
- Conversión de protocolo: Es la interoperabilidad entre tecnologías distintas de comunicación como podría ser ZigBee y Bluetooth.

- **Capacidades de gestión.**

Se refiere a la gestión de datos, configuración, gestión del rendimiento y de la seguridad las cuales se clasifican en:

- Gestión de dispositivos: se refiere a la activación y desactivación, actualización, diagnóstico del dispositivo.
- Gestión de la topología de red local.
- Gestión de tráfico y la congestión: la detección de las condiciones de red y la reserva de recursos para los flujos de datos esenciales para la seguridad de la información.
- **Capacidades de seguridad.**

Las capacidades de seguridad se pueden definir por cada capa como se observa a continuación:

- Capa aplicación: autorización, autenticación, confidencialidad, integridad, privacidad, seguridad y protección de virus.
- Capa de red: autorización, autenticación, confidencialidad y protección de la integridad de datos en señalización.
- Capa de dispositivo: autenticación, autorización, validación de la integridad del dispositivo, control de acceso, confidencialidad e integridad de datos.

#### 2.1.4. Aplicaciones

Las aplicaciones de tecnología en la actualidad son casi infinitas porque cualquier dispositivo que tenga conectividad a internet permite la recopilación de datos y acciones en tiempo real en base a Big Data. Entre las principales aplicaciones se puede destacar las siguientes:

- **Wearables:** Se puede destacar como cualquier producto que un usuario pueda llevar como prenda de vestir o accesorio corporal los cuales pueden ser: gafas, bandas de entrenamiento, pulseras, cinturones GPS, etc. En su defecto son dispositivos pequeños y con eficiencia energética que incluyen sensores para lectura y recopilación de datos e información.
- **Salud:** eHealth e utiliza para enlazarse a los pacientes y permite a los doctores realizar un seguimiento de las condiciones de salud fuera del hospital y en tiempo real, mediante el uso de métricas y alertas automáticas sobre los signos vitales por medio de plataformas en línea.
- **Tráfico:** se puede utilizar para la gestión de tránsito por medio de semáforos inteligentes y estadísticas de flujo de tráfico con el uso de cámaras de seguridad, además se puede lograr una optimización extra cuando por medio de este análisis de datos se presenta al usuario final rutas alternas que se encuentren libres para una mejor circulación.
- **Agricultura:** Mediante la implementación de sensores IoT es fácil obtener gran cantidad de data sobre el estado y las etapas de los suelos, generando eficiencia

de uso de agua y tiempos de cosecha para de esta manera optimizar recursos y obteniendo mejores prestaciones.

- **Hotelería:** presenta mejora en la calidad de servicio por medio de llaves electrónicas, que son enviadas directamente a los dispositivos móviles de cada uno de los huéspedes, así como ofertas y presentación de actividades diarias, realizando también una base de datos que permita por análisis de datos presentar al usuario opciones acorde a sus gustos.
- **Educación:** Por medio de herramientas y análisis de datos se puede crear aulas y laboratorios inteligentes que presenten mejoras con respecto a todas las prestaciones actuales para generar mayor interés en el estudiante o docente al momento de impartir conocimiento.

## 2.1.5. Protocolos de comunicación

### 2.1.5.1. El protocolo de red MQ Telemetry Transport (MQTT)

Se basa en la pila TCP/IP como base para la comunicación, pero en MQTT cada conexión de datos se mantiene enlazada o abierta y se utiliza para establecer comunicación entre cada uno de los publicadores y los suscriptores. Este protocolo fue creado por el Dr. Andy Stanford-Clark de IBM y Arlen Nipper de Arcom en 1999 para conectar dispositivos empleados en la industria petrolera.

Es un servicio de mensajería push con patrón publicador/suscriptor, todos los clientes se conectan con un servidor central denominado bróker. Para seleccionar los mensajes que son enviados se disponen tópicos organizados jerárquicamente.

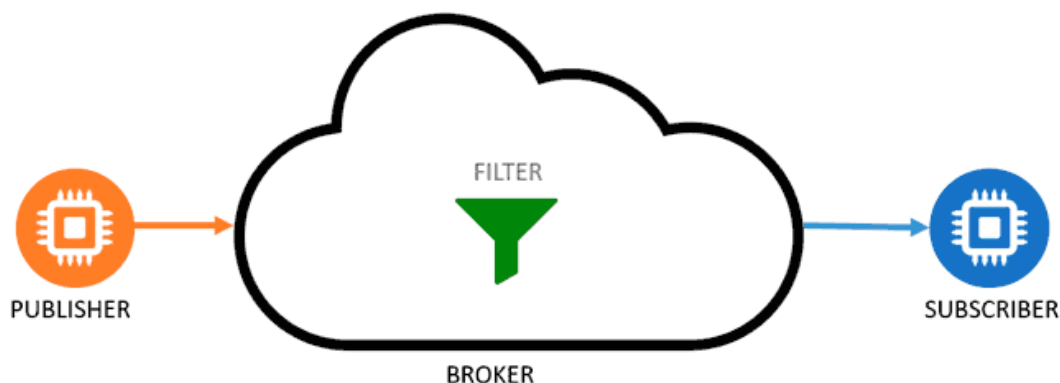


Fig. 2. Modelo MQTT. Fuente: (Shaw, 2018)

Como se observa en la figura 2 el Bróker es donde se realizan los registros de los clientes que se suscriben a un tópico, estos pueden publicar o recibir la información dentro del tópico, toda esta comunicación se mantiene activa mientras el cliente lo decida, esta comunicación es del tipo TCP/IP y usualmente utiliza puerto 1883.

### **2.1.5.2. Protocolo de aplicación restringida (CoAP)**

Se trata de un protocolo de nivel aplicación que se usa en dispositivos electrónicos simples en IoT, está centrado en la transferencia web para el uso de nodos y redes restringidos. Estos nodos en su defecto suelen tener microcontroladores de 8-bit con una pequeña memoria ROM y RAM, y se utilizan en redes restringidas a través de Low-Power Wireless Personal Area Networks.

El grupo que diseñó el protocolo CoAP lo crearon con las siguientes características en mente:

- Sobrecarga y complejidad de análisis
- URI y soporte de tipos de contenido
- Soporte de descubrimiento de recursos provisto por servicios conocidos de CoAP
- Una simple suscripción de recursos y resultados de push notifications
- Almacenamiento simple en caché basado en la capacidad máxima del servidor

### **2.1.5.3. Arquitectura unificada (OPC UA)**

Es el estándar de nueva generación que le sigue a OPC Foundation. OPC clásico es bien conocido en la industria y provee una interfaz estándar para comunicarse con los PLC (Programmable Logic Controller, 'controlador lógico programable'). OPC UA pretende expandir la compatibilidad de OPC al nivel de los dispositivos y de las empresas.

OPC UA es un protocolo cliente/servidor. Los clientes se conectan, navegan, leen y escriben al equipamiento industrial. UA define la comunicación desde la aplicación hacia la capa de transporte, lo que lo hace compatible entre vendedores de iniciativa privada. También es muy seguro, y usa mensajes bidireccionales firmados y encriptación de transporte.

## **2.2. Norma europea sobre la iluminación para interiores-UNE 12464.1**

Dentro de lo que cabe en las normativas existentes actualmente la que más detalla los niveles y distribución de iluminación en espacios cerrados se puede notar que la norma UNE 12464.1 está completamente detalla y delimita todos los procesos necesarios para optimizar luz en lugares cerrados.

Se exige un Ra(unidad de medida del índice cromático) > 80 en la escala de 0 a 100 para iluminar cualquier tarea visual en salas o recintos en los que la ocupación sea de gran duración o permanente, y no ocasional como podría suceder en corredores o pasillos. Estas prescripciones recogidas convenientemente en esta nueva Norma contribuirán a diseñar y ejecutar instalaciones de iluminación en interiores mucho más "humanas" y protectoras de la calidad de vida y condiciones de trabajo en el quehacer cotidiano. Cabe pensar que hay que felicitarse porque la Comisión Europea de Normalización y los países de la Unión Europea hayan refrendado los deseos de los usuarios de las instalaciones satisfaciendo sus ya antiguas

reivindicaciones en cuanto al tratamiento de los colores y del confort visual además de la seguridad. (UIT-T, 2012)

### 2.2.1. Requisitos de iluminación según actividad.

Se debe llegar a complacer tres necesidades humanas básicas las cuales son:

- Confort visual: lo que genera en el usuario final una sensación de bienestar y a su vez un nivel elevado de productividad.
- Prestaciones visuales: donde los trabajadores pueden realizar las actividades que se les asigne incluso en circunstancias difíciles y en periodos prolongados de trabajo.
- Seguridad: Permitiendo una correcta visualización de todas las áreas.

### 2.2.2. Establecimientos Educativos

Dentro de las instituciones educativas se tiene varias dependencias que requieren una variabilidad de configuración lumínica para proporcionar un entorno visual confortable y suficiente, en base a las tareas y el tiempo de uso de las actividades que se van a realizar para la enseñanza. Si se aplican criterios de calidad al momento de diseñar, instalar y mantener los elementos que intervienen al momento de buscar una buena iluminación se puede obtener los resultados que se requieren para garantizar una mayor eficiencia energética.

Una buena iluminación proporciona a los estudiantes y profesores un ambiente estimulante, y el confort visual que permite que mantengan unas actividades sin sobre esfuerzo visual reduciendo el cansancio que provoca una mala iluminación. Dentro de las problemáticas que existen en un centro educativo se encuentran:

- Luminarias las que provocan deslumbramiento directo o indirecto.
- Lámparas de temperatura de color y potencia las que afectan directamente en la lectura de un cuaderno o libro de texto, por ejemplo, las lámparas de luz fría proporcionan un ambiente similar a aire frío y las de colores cálidos dan una sensación de lugares sociales y relajados.

TABLA 1  
ÍNDICE DE DESLUMBRAMIENTO.

<b>Jardines de infancia y Guarderías</b>						
Tipo de actividad	interior, tarea	$E_mLux$	$UGR_L$	$U_o$	$R_a$	Observaciones
SALA DE JUEGOS		300	22	0,4	80	Deberían evitarse altas luminancias en las direcciones de visión desde abajo mediante la utilización de coberturas difusas.
GUARDERÍA		300	22	0,4	80	Deberían evitarse altas luminancias en las direcciones de visión desde abajo mediante la utilización de coberturas difusas.
SALA DE MANUALIDADES		300	19	0,6	80	

Fuente: (UIT-T, 2012).

TABLA 2  
NIVELES DE ILUMINACIÓN EN EDIFICIOS EDUCATIVOS.

### Niveles de iluminación en Edificios Educativos

Tipo de actividad	interior, tarea	$E_mLux$	$UGR_L$	$U_o$	$R_a$	Observaciones
AULAS DE TUTORIA		300	19	0,6	80	La iluminación debería ser controlable.
AULAS PARA CLASES NOCTURNAS Y EDUCACIÓN DE ADULTOS		500	19	0,6	80	La iluminación debería ser controlable.
SALA DE LECTURA		500	19	0,6	80	La iluminación debería ser controlable para colocar varias A/V necesarias.
PIZARRA		500	19	0,7	80	Deben evitarse las reflexiones especulares el presentador/profesor debe iluminarse con la iluminancia vertical adecuada.
MESA DEMOSTRACIONES	DE	500	19	0,7	80	En salas de lectura 750 LUX
AULAS DE ARTE		500	19	0,6	80	
AULAS DE ARTE EN ESCUELAS DE ARTE		750	19	0,7	90	5000 K – 6500 K
AULAS DE DIBUJO TÉCNICO		750	16	0,7	80	
AULAS DE PRÁCTICAS Y LABORATORIOS		500	19	0,6	80	
AULAS DE MANUALIDADES		500	19	0,6	80	
TALLERES DE ENSEÑANZA		500	19	0,6	80	
AULAS DE PRÁCTICAS DE MUSICA		300	19	0,6	80	
AULAS DE PRÁCTICA DE INFORMÁTICA		300	19	0,6	80	
LABORATORIOS DE LENGUAS	DE	300	19	0,6	80	
AULAS DE PREPARACIÓN Y TALLERES		500	22	0,6	80	
HALLS DE ENTRADA		200	22	0,4	80	
AREAS DE CIRCULACIÓN, PASILLOS		100	25	0,4	80	
ESCALERAS		150	25	0,4	80	
AULAS COMUNES DE ESTUDIO Y AULAS DE REUNION	DE	200	22	0,4	80	
SALAS DE PROFESORES		300	19	0,6	80	
BIBLIOTECA ESTANTERÍAS		200	19	0,6	80	
BIBLIOTECAS SALA DE LECTURA	SALA DE	500	19	0,6	80	
ALMACENES DE MATERIAL DE PROFESORES		100	25	0,4	80	
SALAS DE DEPORTE, GIMNASIOS, PISCINAS (USO GENERAL)		300	22	0,6	80	
CANTINAS ESCOLARES		200	22	0,6	80	

COCINA	500	22	0,6	80
--------	-----	----	-----	----

Fuente: (UIT-T, 2012).

### 2.3. Luminarias

Son aparatos que sirven de soporte para las conexiones eléctricas de las lámparas, para que cumplan sus funciones de eficiencia deben cumplir con varias características ópticas, mecánicas y eléctricas requeridas.

Las luminarias en nivel adecuado son responsables del control y distribución de la luz que emite la lámpara. Es necesario que a través de su diseño de sistema óptico se cuida la forma y distribución de la luz.

En lo que vendría a ser una óptima iluminación de espacios se debe tomar en cuenta varias características que generen un mejor rendimiento para el usuario final las cuales se detallan en los subíndices siguientes.

#### 2.3.1. Flujo luminoso

Es la energía que de luz que se emite en todas las direcciones por una fuente luminosa por unidad de tiempo tomando en cuenta que su medida se da en Lumen [lm] que se representa con el símbolo  $\Phi$ .



Fig. 3. Flujo Luminoso. Fuente: (Castro Guaman & Posligua Murillo, 2015).

Cuando se toman dos focos del mismo tipo, pero a diferente potencia se puede notar de manera básica como el que tiene mayor potencia abarca un espacio mayor de luz.

#### 2.3.2. Iluminación o iluminancia.

Es el índice representativo para la densidad del flujo luminoso sobre una superficie. Se puede entender como la relación entre el flujo luminoso que incide sobre la superficie y el tamaño de la misma. Su unidad de medida es el Lux [lumen/m<sup>2</sup>] y se representa por la letra E. Entonces la formula seria:

$$E = \frac{\Phi}{S}$$

En donde:

E: Iluminación o intensidad de iluminación. [lx]

S: Área de la superficie a iluminar [m<sup>2</sup>]

$\Phi$ : Flujo Luminoso [lm]

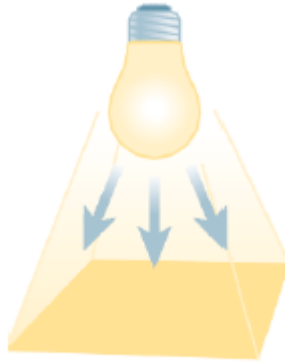


Fig. 4. Iluminancia. Fuente: (Castro Guaman & Posligua Murillo, 2015).

### 2.3.3. Intensidad luminosa

Una fuente de luz emite una cantidad de flujo luminoso en varias direcciones con mayor o menor igualdad dependiendo de la fuente que la origina. La intensidad luminosa es el flujo en una dirección determinada por el ángulo sólido que lo contiene. La unidad de medida es la candela[cd], es la unidad básica de luminotecnia de la que se derivan el resto de magnitudes y se representa con la letra I. Entonces se tiene que:

$$I = \frac{\Phi}{\omega}$$

Siendo:

I: Intensidad lumínica [cd]

$\Phi$ : Flujo Luminoso [lm]

$\Omega$ : Ángulo sólido [sr]



Fig. 5. Intensidad Luminosa. Fuente: (Castro Guaman & Posligua Murillo, 2015).

### 2.3.4. Luminancia

La luminancia representa la cantidad de luz emitida en una dirección dada por una superficie iluminada. La unidad de medida que la representa es candela por superficie [ $cd/m^2$ ] y se representa por la letra L.



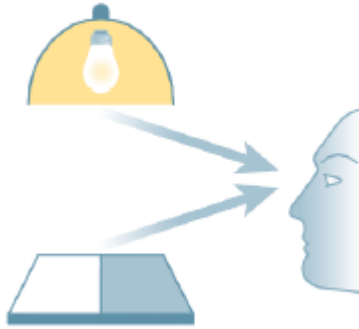


Fig. 6. Luminancia. Fuente: (Castro Guaman & Poslígua Murillo, 2015).

La superficie proyectada será proporcional al coseno del ángulo que se forma de modo que se tiene:

$$L = \frac{I_{\alpha}}{A(m^2) * \cos(\alpha)}$$

Simplificando la teoría de iluminación es el flujo luminoso que llega a una superficie y la luminancia es la luz que procede de esa misma superficie.

### 2.3.5. Rendimiento y eficiencia luminosa

Cuando se requiere diseñar un sistema con eficiencia luminosa se toma en cuenta el rendimiento de las luminarias. El rendimiento es la relación entre el flujo luminoso emitido por una luminaria y el flujo luminoso de la lámpara, la eficiencia luminosa es la relación entre el flujo luminoso entregado, en lumen, y la potencia consumida, en vatios ( $\Phi[\text{lm}]/P[\text{W}]$ ). Cuando entra a funcionar la lámpara, una parte de la energía eléctrica se transforma en luz emitida y otra parte se convierte en calor. Exceptuando unos pocos casos, no deseamos que una bombilla caliente el ambiente, sino que lo ilumine. Una lámpara eficiente convierte la mayor parte de la energía eléctrica en emisión de luz. Si dos luminarias tienen la misma emisión de luz, pero usamos la que tiene mejor eficiencia, se ahorra energía y dinero.



Fig. 7. Rendimiento. Fuente: (Castro Guaman & Poslígua Murillo, 2015).

### 2.3.6. Curva de distribución luminosa

Es también conocida como curva fotométrica, y define la forma y dirección de la distribución de la luz emitida por la luminaria en el espacio. Esta resulta de tomar mediciones de la intensidad luminosa en varios ángulos alrededor de una luminaria y representarlos en forma gráfica, como coordenadas polares. Este parametro se utiliza para conocer la distribución de la luz y hacer una correcta selección al momento de utilizarla.

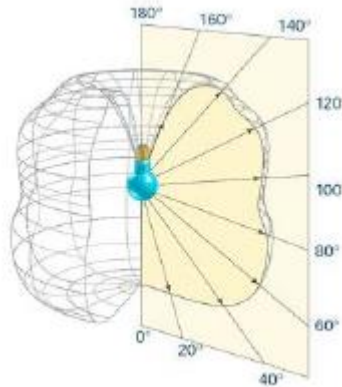


Fig. 8. Curva Fotométrica. Fuente: (Castro Guaman & Posligua Murillo, 2015).

### 2.3.7. Deslumbramiento

Ocurre cuando en el campo de visión se puede observar una fuente de luz directa, por lo cual solo se percibe el objeto de mayor intensidad sin poder distinguir bien nuestro entorno debido a la saturación que se produce en el campo visual. Es necesario controlarlo para eliminar o disminuir la medida de luz y así evitar visión borrosa o fatiga por recibir luz de manera constante.



Fig. 9. Deslumbramiento. Fuente: (Castro Guaman & Posligua Murillo, 2015).

La medida de deslumbramiento se mide usando el URG Unified Gare Rating (Índice de deslumbramiento Unificado), creado por la CIE el año 1995, que describe el brillo molesto según la posición del observador con respecto a la dirección de la vista.

Para sacar el cálculo de deslumbramiento se usa los diagramas de Sollner, primero se debe determinar el ángulo con el cual un usuario observa las partes desnudas de una luminaria o fuente de luz. Se recomienda que el ángulo sea máximo de 30°. El diagrama establece unas curvas de deslumbramiento según la iluminación del local y calidad deseada. En base a la cantidad de flujo luminoso se tienen distintos rangos de deslumbramiento, como se observa en la Tabla 3.

TABLA 3  
ÍNDICE DE DESLUMBRAMIENTO.

Clase de Calidad	Índice de deslumbramiento	Tipo de actividad a desarrollar
A – Muy Alta	1.15	Tareas visuales muy exactas
B – Alta	1.5	Tareas con gran demanda visual o con alta concentración
C – Media	1.85	Tareas con demandas visuales o concentración moderada. Trabajador en movimiento ocasional
D – Baja	2.2	Tareas con demanda visual o concentracióón baja. Trabajador en movimiento
E – Muy Baja	2.55	Tareas de baja demanda, con movimiento de una zona a otra. Zonas no utilizadas generalmente por las mismas personas.

Fuente: (Castro Guaman & Posligua Murillo, 2015).

La fórmula que permite el cálculo de deslumbramiento es:

$$URG = 8 \log_{10} \left( \frac{0.25}{L_b} \sum \frac{L^2 \omega}{p^2} \right)$$

Donde:

- $L_b$ : Es la iluminación de fondo en  $cd/m^2$
- $L$ : Es la luminancia en  $cd/m^2$  de las partes luminosas de cada luminaria en la dirección del observador
- $\omega$ : Es el ángulo solido en relacion de las partes luminosas de cada luminaria en el ojo del observador
- $p$ : Es el índice de posición para cada luminaria individual que se refiere a su desplazamiento de la línea de visión.

## 2.4. Raspberry Pi

Es un ordenador de placa reducida con bajo coste económico, que se elaboró en el Reino Unido con el objetivo de estimular la enseñanza de ciencias de computación en escuelas y mejorar las oportunidades de acceso a equipos de informática a todos los lugares.

El diseño incluye System-on-a-chip Broadcom BMC2835, que contiene un procesador central ARM1176JZF-S a 700 MHz con un modo turbo para que el usuario puede hacer overclock de hasta 1 GHz, un procesador gráfico Video Core IV y memoria RAM.

Para lo que es el sistema operativo el raspberry pi utiliza sistemas basado en núcleo de Linux. El más utilizado es rasbian, que es una distribución basada en debian que está optimizada para el hardware de Raspberry, se lanzó durante julio 2012.

Se origino como una solución para estimular en las escuelas el acercamiento de las ciencias de la computación a más personas y lugares a bajo costo, el primer desarrollo surgió en 2006 aplicando un microcontrolador Atmel Atmega644 similar a Arduino, por ser código abierto todos los diagramas, esquemas y datos para el diseño se pueden descargar en la web.

### 2.4.1. Modelos.

#### 2.4.1.1. Raspberry Pi 1

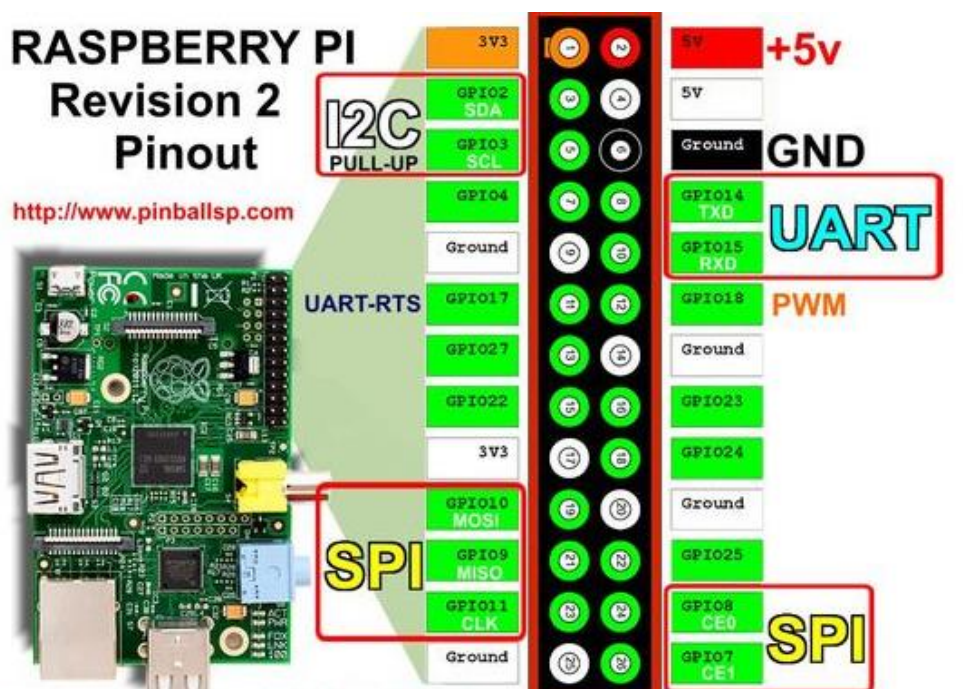



Fig. 10. Pinout Raspberry Pi 1. Fuente: (Shaw, 2018)

Consta de un SOC Broadcom BCM2835, chip gráfico VideoCore IV y procesador ARM 1176JZF-S de un núcleo a 700MHz, aunque con overclock puede llegar hasta los 1000MHz.

Además, cuenta en su modelo A con 256Mb de SDRAM, 1 puerto USB, y no cuenta con conexión Ethernet. Mientras que en el modelo B son 512Mb SDRAM, 2 puertos USB, y añade un puerto Ethernet, las dos cuentan con una tarjeta SD para lo que sería el disco duro que se debe asignar.

Todos los modelos 1 cuentan con salidas de video RCA, HDMI, y DSI para un panel LCD y salida de audio Jack 3.5mm y además de audio por medio de HDMI, además de tener un puerto de conexiones electrónicas que permiten 8 puertos de GPIO, SPI, I2C y UART.

### 2.4.1.2. Raspberry Pi 2



GPIO#	2nd func	Physical Pins		2nd func	GPIO#
		pin#	pin#		
N/A	+3V3	1	2	+5V	N/A
GPIO2	SDA1 (I2C)	3	4	+5V	N/A
GPIO3	SCL1 (I2C)	5	6	GND	N/A
GPIO4	GCLK	7	8	TXD0 (UART)	GPIO14
N/A	GND	9	10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11	12	GEN1	GPIO18
GPIO27	GEN2	13	14	GND	N/A
GPIO22	GEN3	15	16	GEN4	GPIO23
N/A	+3V3	17	18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19	20	GND	N/A
GPIO9	MISO (SPI)	21	22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23	24	CE0_N (SPI)	GPIO8
N/A	GND	25	26	CE1_N (SPI)	GPIO7
EEPROM	ID_SD	27	28	ID_SC	EEPROM
GPIO5	N/A	29	30	GND	N/A
GPIO6	N/A	31	32	-	GPIO12
GPIO13	N/A	33	34	GND	N/A
GPIO19	N/A	35	36	N/A	GPIO16
GPIO26	N/A	37	38	N/A	GPIO20
N/A	GND	39	40	N/A	GPIO21

Fig. 11. Pinout Raspberry Pi 2. Fuente: (Shaw, 2018).

Es una versión actualizada del anterior pero que en cambios a destacar se tiene la potencia de cálculo computacional muy superior. Además, utiliza un SOC Broadcom BCM2836, un procesador ARM Cortex A7 de cuatro núcleos a 900Mhz y un overclock de 1050Mhz, 1Gb SDRAM. Mantiene el chip gráfico VideoCore IV.

Cuenta con 4 puertos USB, puerto Ethernet 10/100Mb. Tarjeta de memoria micro SD como disco duro, El número de puertos GPIO aumenta a 17 y mantiene funciones SPI, I2C y UART.



### 2.4.1.3. Raspberry Pi 3

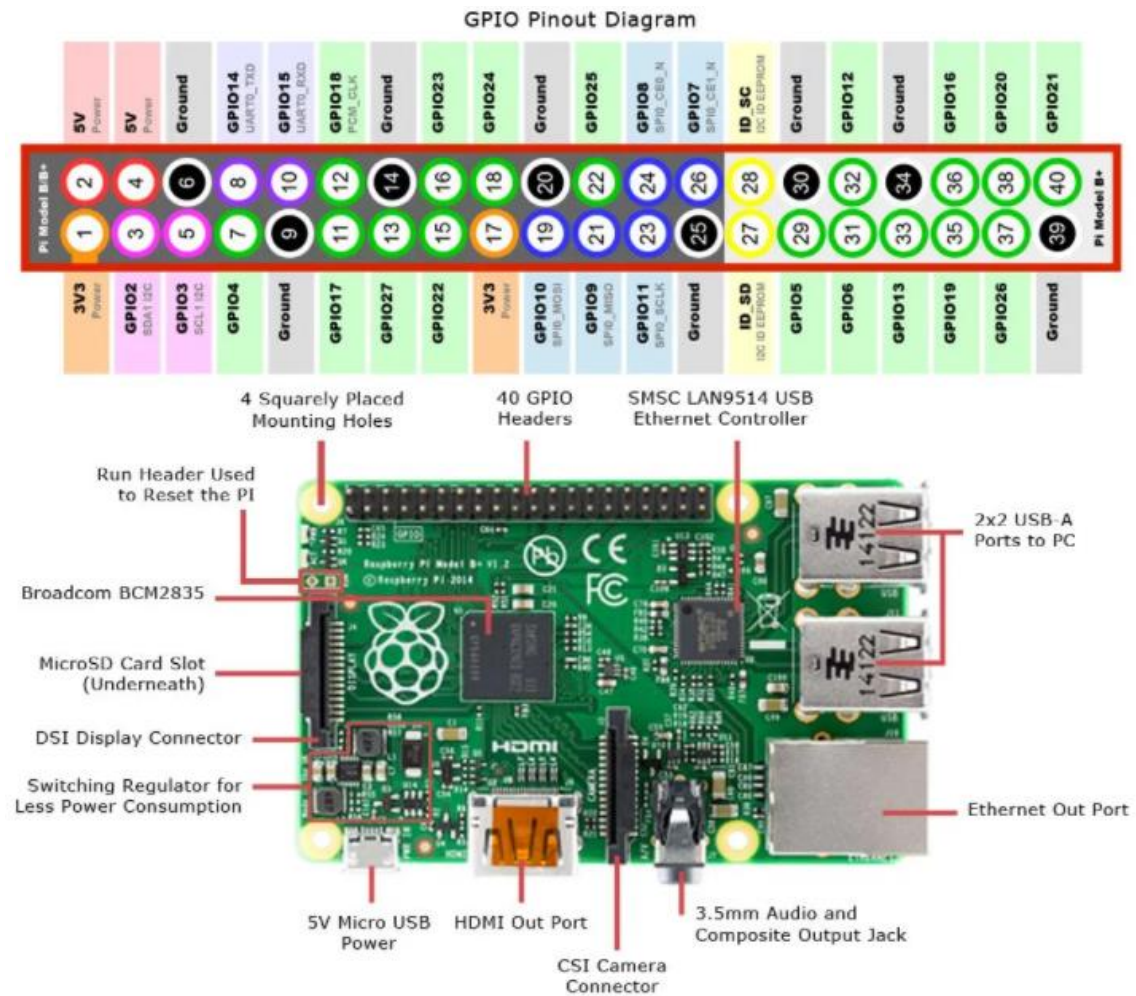


Fig. 12. Pinout Raspberry Pi 3. Fuente: (Crespo, 2017).

Comenzó la evolución a la comunicación con el entorno y el IoT actuales por medio de la implementación de un Bluetooth 4.1 y Wifi 802.11n.

La potencia se amplía con un SOC Broadcom BMC287, y un procesador ARMv8 de cuatro núcleos a 1.2GHz de 64 bits. Se mantiene el chip gráfico VideoCore IV, 1 hasta 2GB de memoria SDRAM, puerto Ethernet, 4 puertos USB y 17 pines GPIO con funciones SPI, I2C y UART.

### 2.4.1.4. Raspberry Pi 4

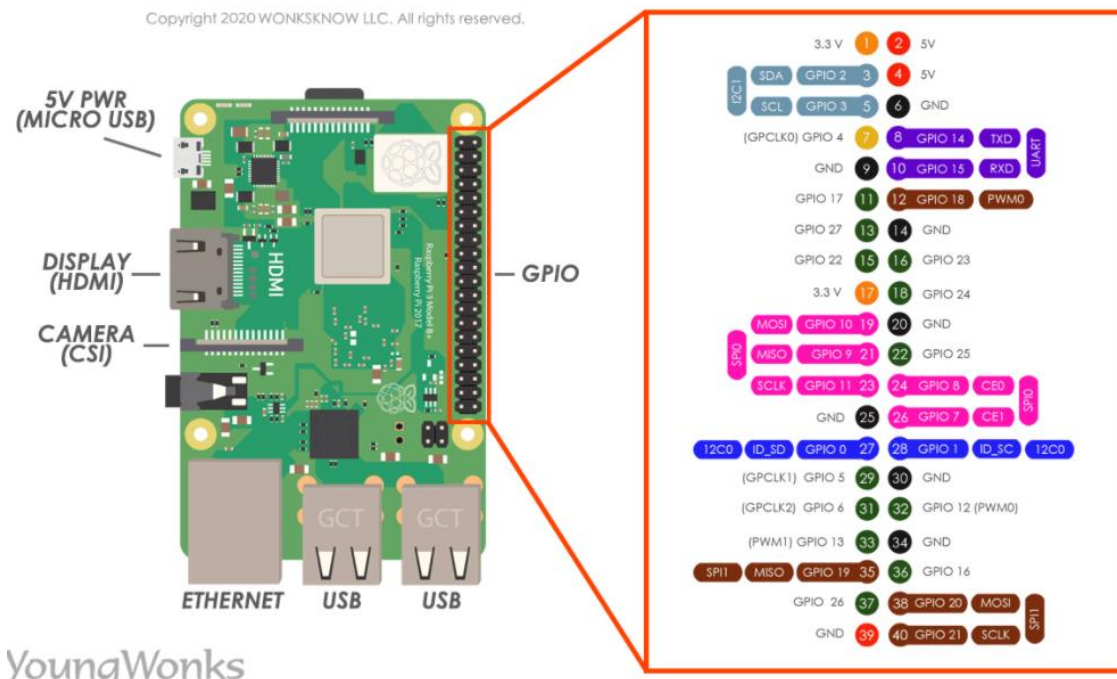


Fig. 13. Pinout Raspberry Pi 4. Fuente: (WONKSKNOW LLC, 2019).

Es una actualización enorme con un procesador ARM Cortex-172 con cuatro núcleos de 1,5GHz por lo cual todos los componentes y la potencia del dispositivo ha sido modificada.

Se anunció en junio de 2019 y se modificó los puertos HDMI de tamaño completo por dos puertos microHDMI, cuenta con la capacidad de manejar dos pantallas de 4K a 60Hz. Se incluye un puerto USB 3.0 y el puerto Ethernet ya no limitado a 300Mbps.

Tiene Bluetooth 5.0 y Wi-Fi 802.11ac para conexión inalámbrica, además el conector microUSB para la alimentación es tipo C que suma 500 mA extra de energía para alcanzar un total de 1.2A, y cuenta con varias capacidades de memoria RAM de 1GB, 2GB, 4GB y 8GB.

### 2.4.1.5. Características

Es necesario en base a la Tabla 4, establecer las características de cada uno de los modelos y sus características de hardware para tener una clara diferenciación.

TABLA 4  
COMPARATIVA HARDWARE MODELOS RASPBERRY PI.

Modelo	Raspberry Pi 1 B+	Raspberry Pi 2 B	Raspberry Pi 3 B	Raspberry Pi 4
Fecha	2012 Feb 15	2015 Feb 1	2016 Feb 29	2019 Jun 24
Precio	30	35	35	100
SOC	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2711B0
Core	ARM1176JZF-S	Cortex-A7	Cortex-A53 64-bit	Cortex-A72
Nº Cores	1	4	4	4
CPU Clock	700 MHz	900 MHz	1.2 GHz	1.5 GHz

RAM	512 MB	1 GB	1 GB	1/2/4/8 GB
Memoria	Micro SD	Micro SD	Micro SD	Micro SD
USB	2	4	4	1 microUSB
Ethernet	Si	Si	Si	Si
Wi-Fi	No	No	Si	Si
Bluetooth	No	No	Si	5.0
HDMI	Si	Si	Si	2 Mini
GPIO	8	17	17	17
UART	Si	Si	Si	Si
SPI	Si	Si	Si	Si
I2C	Si	Si	Si	Si
DSI (LCD)	Si	Si	Si	Si
Cámara	Si	Si	Si	Si
Altura	85.6 mm	85.6 mm	85.6 mm	85.6 mm
Ancho	53.98 mm	56.5 mm	56.5 mm	56.5 mm
Profundidad	17 mm	17 mm	17 mm	17 mm
Peso	45 g	45 g	45 g	45 g
Consumo	700 mA	820 mA	1400 mA	3000 mA

Fuente: Elaborado por el autor

## 2.5. Microcontrolador

### 2.5.1. Arduino

Es una plataforma de creación electrónica código abierto, que se basa en hardware y software libre, lo que implica que su código fuente y los esquemáticos están disponibles para el que quiera utilizarlos. Esta tecnología permite crear diferentes tipos de microordenadores que se integran en una sola placa y se utiliza en varias aplicaciones.

El proyecto nació en 2003, cuando algunos estudiantes del Instituto de Diseño interactivo de Ivrea, Italia con el objetivo de facilitar el acceso y uso de la electrónica a los estudiantes de electrónica a bajo costo.

Se consigue como resultado Arduino que es una placa que integra todos los elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador, y que puedan ser programados en varios sistemas operativos.

Es necesario destacar que existen algunos modelos o placas que se integran dentro de la familia Arduino y las más populares son:



### 2.5.1.1. Arduino Uno

La placa está basada en un microcontrolador atmega328P con una arquitectura AVR de 8 bits basada en el MCU de ATMEL. Tiene dos presentaciones en la actualidad tanto con un microcontrolador tipo DIP y otro tipo SMD para soldadura superficial.

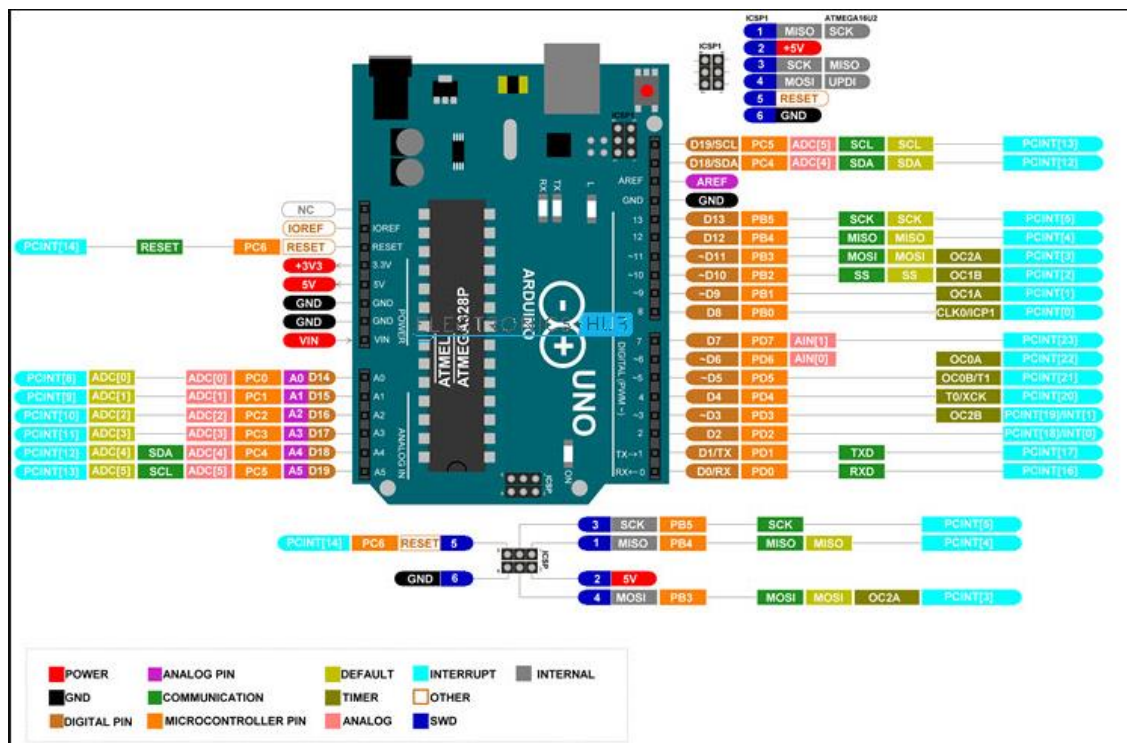


Fig. 14. Pinout Arduino Uno. Fuente: (Teja, 2021).

En la Figura 14 se puede observar definido cada uno de sus puertos y sus respectivas como sus funciones entre las cuales se puede destacar: 24 pines digitales; 6 pines analógicos; comunicación RS232; comunicación SPI; comunicación I2C.

### 2.5.1.2. Arduino Mega

Es una placa completamente diferente a Arduino Uno llegando a considerarse el hermano mayor de la familia Arduino, tanto en tamaño como en características técnicas. Está basado en el microcontrolador Atmega2560, con arquitectura AVR de 8 bits de ATMEL, está disponible en un paquete de 100 pines. Tiene más pines digitales y analógicos además de mayor memoria flash y RAM.

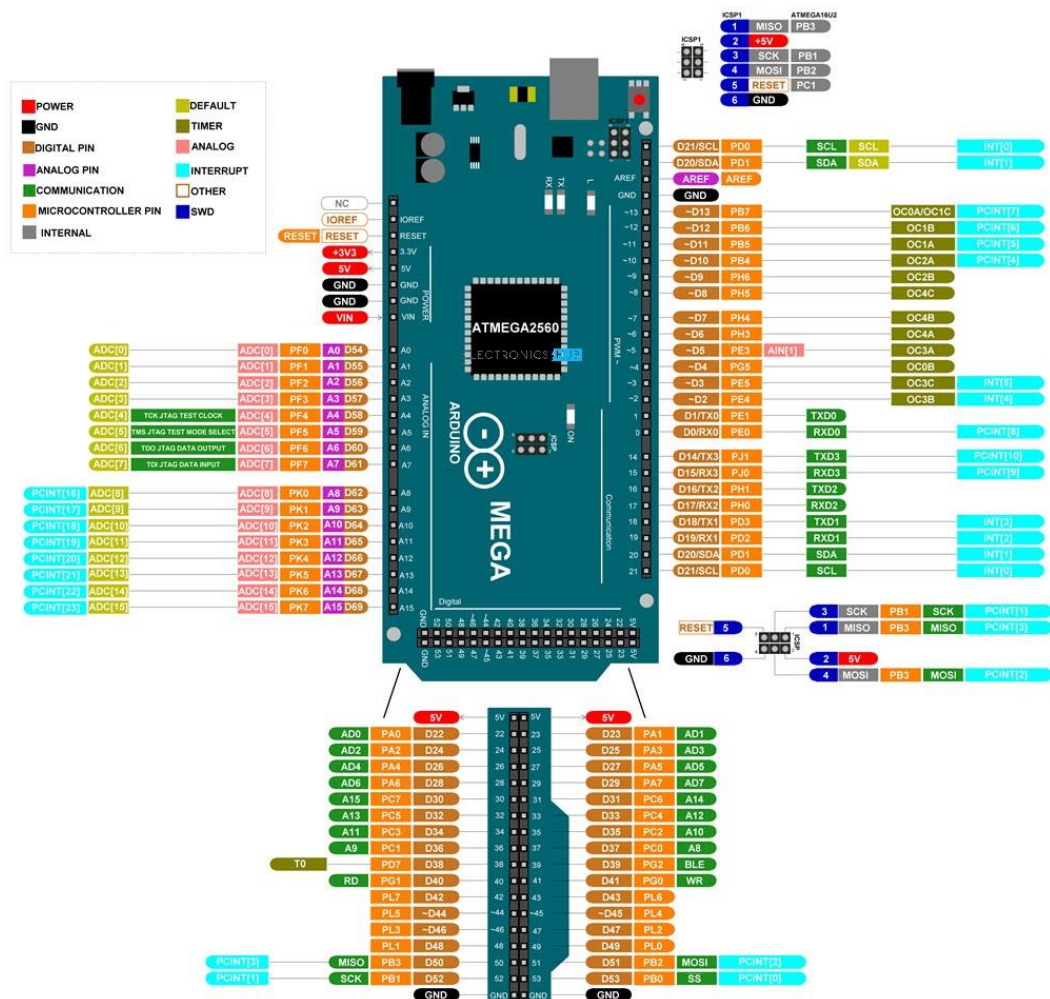


Fig. 15. Pinout Arduino Mega. Fuente: (Teja, 2021).

Como se observa en la Figura 14, este microcontrolador cuenta con: 72 pines de entrada y salida; 54 pines digitales para entrada y salida; 16 pines analógicos con resolución de 10 bits en ADC; 3 puertos para comunicación RS232; comunicación SPI; comunicación I2C.

### 2.5.1.3. Arduino Nano

Tiene funcionalidades similares a las de Arduino Duemilanove pero con un paquete diferente. La Nano basada en la ATmega328 (Arduino Nano 3.x) incorpora la ATmega328P, la misma que el Arduino Uno. La principal diferencia entre ellos es que el Arduino Uno se presenta en forma de PDIP (Plastic Dual-In-line Package) con 30 pines y la Arduino Nano está disponible en TQFP (plastic quad flat pack) con 32 pines. Los 2 pines adicionales de la Arduino Nano sirven para las funcionalidades del ADC, mientras que UNO tiene 6 puertos ADC, pero la Nano tiene 8 puertos ADC.

La placa Arduino Nano no tiene una toma de alimentación DC como otras placas Arduino, sino que tiene un puerto mini-USB. Este puerto se utiliza tanto para la programación como para la monitorización en serie. La característica fascinante de Nano es que elegirá la fuente de

energía más fuerte con su diferencia de potencial, y la fuente de energía que selecciona el puente no es válida.

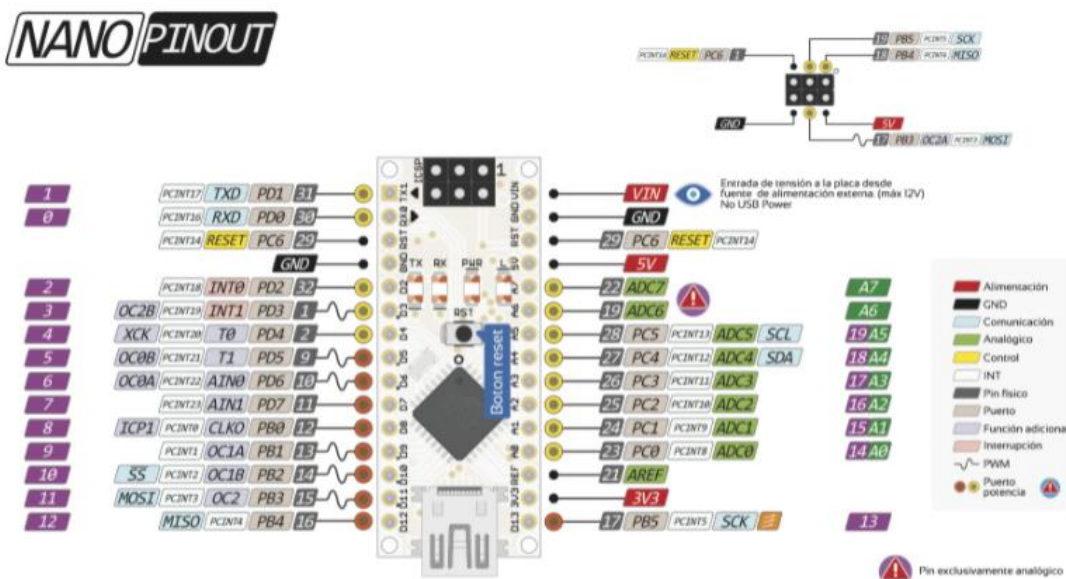


Fig. 16. Pinout Arduino Nano. Fuente: (Damián, 2020).

## 2.5.2. Expressif-ESP

Es una empresa que se dedica al diseño de potentes chips para IoT, y también su sistema operativo y lugares de aplicación. Además, genera certificaciones y capacitaciones que permiten al usuario del producto implementar sistemas integrados a miles de aplicaciones comerciales.

### 2.5.2.1. NODEMCU V2 ESP8266 WIFI

Es una plataforma similar a Arduino, pero orientada al IoT. Esta placa tiene como núcleo al SoM ESP-12E que a su vez está basado en el Soc WiFi ESP8266, integrando un USB-Serial TTL CP2102 y un conector micro-USB con el que se puede programar.

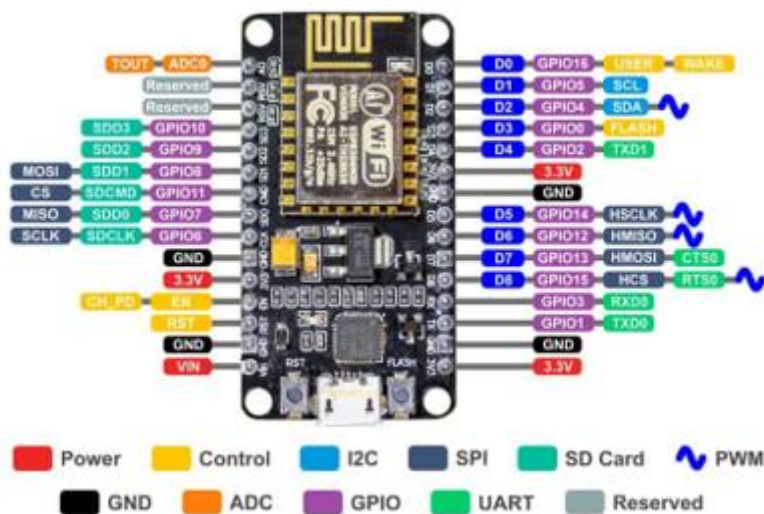


Fig. 17. Pinout ESP8266. Fuente: (Naylamp Mechatronics, 2021).

Se tiene que las características técnicas y físicas del componente son:

- Voltaje de Alimentación: 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC (No usar 5V)
- Placa: NodeMCU v2 (Amica)
- Chip conversor USB-serial: CP2102
- SoM: ESP-12E (Ai-Thinker)
- SoC: ESP8266 (Espressif)
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Instruction RAM: 32KB
- Data RAM: 96KB
- Memoria Flash Externa: 4MB
- Pines Digitales GPIO: 17 (4 pueden configurarse como PWM a 3.3V)
- Pin Analógico ADC: 1 (0-1V)
- Puerto Serial UART: 2
- Certificación FCC
- Antena en PCB
- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado
- PLLs, reguladores, DCXO y manejo de poder integrados
- Potencia de salida de +19.5dBm en modo 802.11b
- Corriente de fuga menor a 10uA
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Consumo de potencia Standby < 1.0mW (DTIM3)
- Pulsador RESET y FLASH
- Leds indicadores: 2
- Dimensiones: 49\*26\*12 mm
- Peso: 9 gramos

En cuestión de tecnologías de comunicación se obtiene que posee lo siguiente:

- SDIO 2.0, SPI, UART
- Integra RF switch, balun, 24dBm PA, DCXO y PMU
- Posee un procesador RISC, memoria en chip e interface para memoria externa
- Procesador MAC/Baseband integrado



- Interface I2S para aplicaciones de audio de alta calidad
- Reguladores de voltaje lineales "low-dropout" en chip
- Arquitectura propietaria de generación de clock "spurious free"
- Módulos WEP, TKIP, AES y WAPI integrados

### 2.5.2.2. NODEMCU-32 - ESP32 WIFI

Es una herramienta muy potente para proyectos IoT, que viene integrada con una placa SoMESP-WROOM-32 y como base al SoC Esp32, se usa un conversor USB-Serial CP2102 para programar por medio del USB el ESP32, esta viene a ser una evolución del ESP8266 mejorando sus capacidades de comunicación y procesamiento computacional. En lo que refiere a conectividad cuenta con protocolos de comunicaciones como WiFi, Bluetooth y BLE. En lo que refiere a procesamiento su CPU 32-bit de dos núcleos de hasta 240Mhz que pueden funcionar de manera independiente.

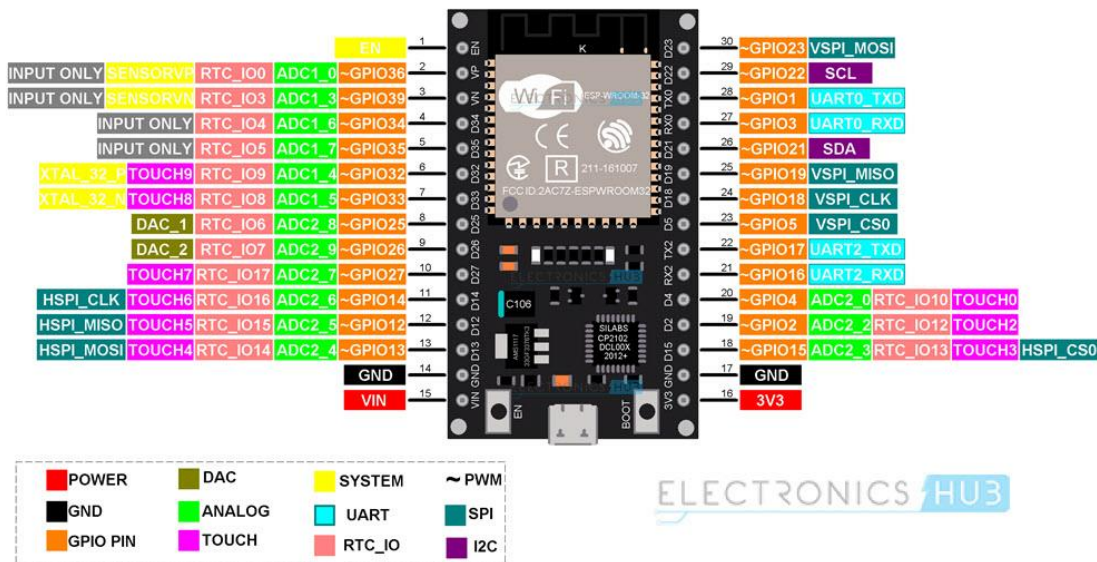


Fig. 18. Pinout Esp32. Fuente: (Naylamp Mechatronics, 2021)

Como se observa en la Figura 18 se puede notar las siguientes características:

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- Placa: ESP32 DEVKIT V1 (Espressif)
- SoM: ESP-WROOM-32 (Espressif)
- SoC: ESP32 (ESP32-D0WDQ6)
- CPU: Dual core Tensilica Xtensa LX6 (32 bit)
- Frecuencia de Reloj: hasta 240Mhz
- Desempeño: Hasta 600 DMIPS
- Procesador secundario: Permite hacer operaciones básicas en modo de ultra bajo consumo

- Wifi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz hasta 150 Mbit/s)
- Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)
- Memoria: 448 KByte ROM; 520 KByte SRAM; 16 KByte SRAM in RTC; QSPI Flash/SRAM, 4 MBytes
- Pines Digitales GPIO: 24 (Algunos pines solo como entrada)
- Pines PWM: 16
- Pines Analógicos ADC: 18 (3.3V, 12bit: 4095, tipo SAR, ganancia programable)
- Conversor Digital a Analógico DAC: 2 (8bit)
- UART: 2
- Chip USB-Serial: CP2102
- Antena en PCB
- Seguridad:
  - Estándares IEEE 802.11 incluyendo WFA, WPA/WPA2 and WAPI
  - 1024-bit OTP, up to 768-bit for customers
  - Aceleración criptográfica por hardware: AES, HASH (SHA-2), RSA, ECC, RNG
- Dimensiones: 55\*28 mm

### 2.5.2.3. ESP-01 ESP8266 WIFI-SERIAL

Este módulo está basado en el SOC ESP8266 y diseñado para las necesidades IoT, tiene un procesador de 32 bits y conectividad Wi-Fi, contando con un modo de WiFi Station o punto de acceso. Tiene cargado con comandos AT, además se puede flashear el firmware para usarlo con el IDE de Arduino mediante un módulo CP2102, tiene un voltaje de trabajo de 3.3V

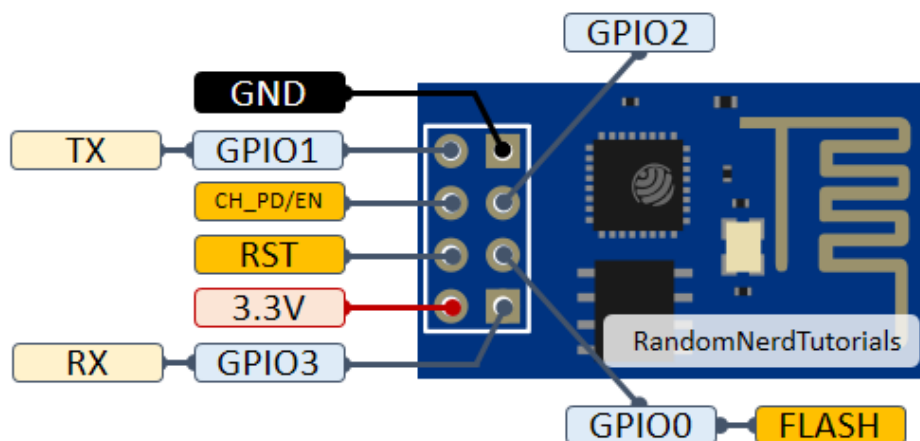


Fig. 19. Pinout Esp01. Fuente: (Naylamp Mechatronics, 2021)

Como se observa en la Figura 19 se encuentra la distribución de cada uno de sus pines y como características físicas están:

- Memoria Flash Externa: 4MB

- Pines Digitales GPIO: 4
- UART: 1
- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado
- PLLs, reguladores, DCXO y manejo de poder integrados
- Potencia de salida de +19.5dBm en modo 802.11b
- Corriente de fuga menor a 10uA
- El procesador integrado de 32-bit puede ser utilizado como procesador de aplicación
- SDIO 1.1/2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Consumo de potencia Standby < 1.0mW (DTIM3)
- Dimensiones: 25\*15 mm
- Peso: 5 gramos

## 2.6. Sensores de luz

### 2.6.1. Fotorresistencia o LDR

Es un componente foto electrónico que su resistencia interna varía en función de la luz que incide sobre él, funciona cuando la luz incide en él y va aumentando su resistencia cuando la intensidad luminosa decrece.

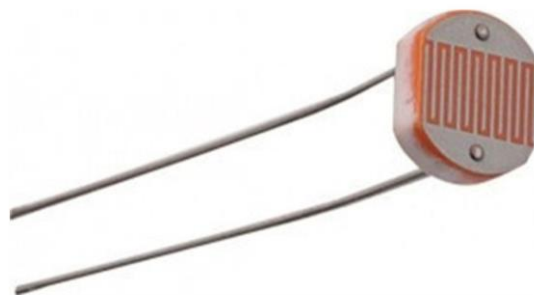


Fig. 20. Fotorresistencia. Fuente: (prometec, 2021)

Se puede denotar que posee las siguientes especificaciones técnicas:

- Sensor: LDR GL5528
- Resistencia en luz (10 lux): 8K-20K Ohm
- Resistencia en oscuridad: 1M Ohm
- Voltaje máx: 150V

- Potencia máx: 100mW
- Material fotosensible: CdS (Sulfato de Sodio)
- Frecuencia de luz pico: 540 nm
- Tamaño de 5mm.

### 2.6.2. BH1750

Es un sensor que sirve para medir el flujo luminoso y fue desarrollado por la empresa Rohm Semiconductor. Tiene un conversor de 16 bit, y entrega una salida I2C y funciona de forma más precisa que un resistor LDR, ya que no requiere convertir el voltaje para interpretar la señal.

La entrega de datos por parte del sensor se realiza en forma de unidades Lux (Lx), lo que equivale a un  $\text{lumen/m}^2$ . Se utiliza en fotometría como unidad de medida de luminancia en base a un modelo estándar de la sensibilidad del ojo humano.

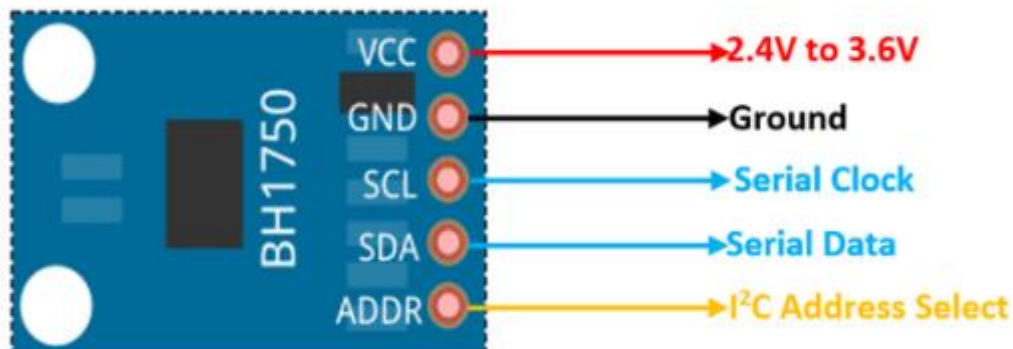


Fig. 21. Pinout BH1750. Fuente: (Naylamp Mechatronics, 2021)

Las especificaciones técnicas del sensor son las siguientes:

- Voltaje de Operación: 3V – 5V
- Interfaz digital a través de bus I2C con capacidad de seleccionar entre 2 direcciones
- Respuesta espectral similar a la del ojo humano
- Realiza mediciones de iluminancia y convierte el resultado a una palabra digital
- Amplio rango de medición 1-65535 lux
- Modo de bajo consumo de energía
- Rechazo de ruido a 50/60 Hz
- Baja dependencia de la medición contra la fuente de luz: halógeno, led, incandescente, luz de día, etc.

### 2.7. Motor DC

Son máquinas capaces de convertir la energía eléctrica en mecánica y de forma inversamente proporcional, todo esto debido a las leyes del electromagnetismo.



El funcionamiento de corriente continua se define en base al rechazo que se ejerce en los polos magnéticos de un imán en base a las leyes de Lorentz, el electroimán se llama rotor y su eje le permite girar entre cada uno de los polos.

Este tipo de motores tienen que ser revisados periódicamente ya que por la utilización se generan desgastes en los carbones y a su vez en el bobinado lo que hace que la teoría electromagnética pierda su estabilidad.

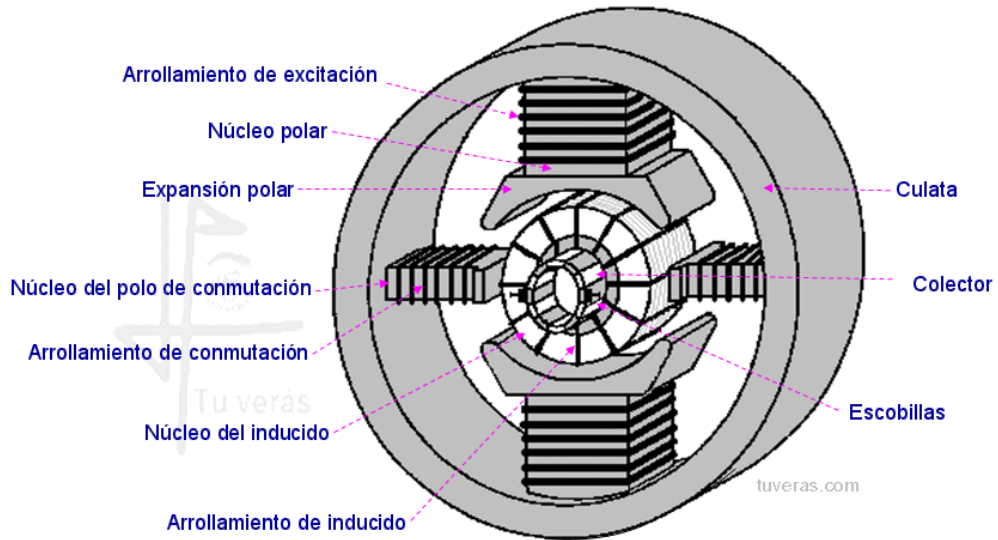


Fig. 22. Motor DC. Fuente. Bibliográfica (Pernía, 2011)

## Capítulo II: Desarrollo

En este capítulo se presenta las fases que permiten elaborar un proyecto, las cuales se pueden definir como: la metodología, la selección de los componentes de software y hardware, las delimitaciones de módulos que integran el prototipo de control de luminancia, la implementación de los componentes y las pruebas de funcionamiento hasta dejar en marcha el sistema en su totalidad.

### 3.1. Desarrollo Experimental

En lo que se refiere a este capítulo su principal objetivo consiste en seleccionar Hardware y Software necesarios para la investigación basado en el Estándar IEEE 29148, y para cumplirlo se utilizará la metodología de selección de sistema y el modelo en V.

### 3.2. Metodología

Este tipo de metodología parte de un enfoque experimental donde el investigador manipula una o varias variables de estudio para controlar el aumento o disminución de las variables y sus conductas observadas, todo esto de manera controlada para obtener los resultados deseados en base a pruebas y cambios de variables.

### 3.3. Modelo en V

Es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño. Como se observa en la Figura 23, se tiene una condición en forma de V, con el análisis y diseño en la parte izquierda y las pruebas de funcionamiento a la derecha.

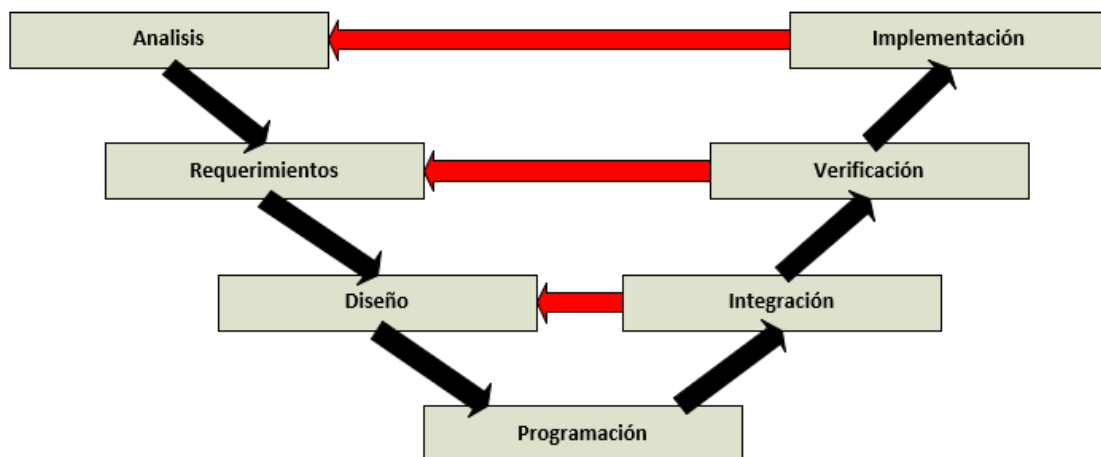


Fig. 23. Etapas modelo en V. Fuente: Elaborado por el autor

Como se observa en la Figura 23 dentro de todas las etapas existe una fase paralela correspondiente de verificación y validación. Esta estructura a cada fase del desarrollo debe existir un resultado verificable.

- Nivel 1: Se orienta al inicio del proyecto y al fin del mismo ya que constituyen los dos extremos del ciclo. Integra el análisis de requisitos y especificaciones.
- Nivel 2: se refiere a las características funcionales del sistema, establece funciones directas o indirectas visibles por el usuario final.
- Nivel 3: se define los componentes hardware y software del sistema final, establece el diseño del dispositivo.
- Nivel 4: se refiere a la parte de la implementación.

Para cumplir con este modelo se establece fases del sistema, entre las cuales se puede destacar: Análisis, Requerimientos, Diseño, Programación, Integración, Verificación, Implementación.

### 3.3.1. Análisis

#### 3.3.1.1. Delimitación campo de aplicación

Es necesario que se tenga en cuenta el lugar de aplicación del sistema que se planea implementar, para determinar requisitos y materiales necesarios para la elaboración.

El presente proyecto está pensado para implementarlo en el Laboratorio de máquinas de la carrera CIELE, como se pudo definir en los objetivos se busca realizar un control lumínico de los laboratorios que permita regular la luminancia en los laboratorios.

#### Laboratorio de maquinas

Para delimitar el campo de aplicación se puede tener como situación inicial, las medidas y distribución de las luminarias en el laboratorio se muestran en la Figura 24.

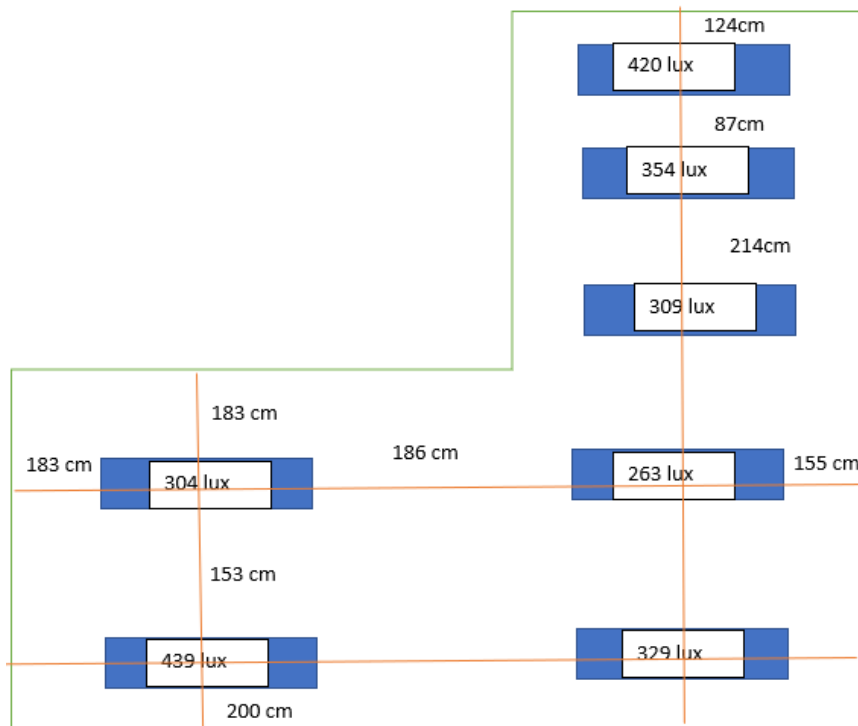


Fig. 24. Medidas Laboratorio de máquinas y toma de luxes. Fuente: Elaborado por el autor

Como se puede observar en la Figura 24 se tiene 7 luminarias que se distribuyen de manera uniforme en el laboratorio, gracias a las medidas realizadas por medio del luxómetro se tiene los valores como se detalla en la Tabla 5

TABLA 5  
LECTURA LUMENES EN LUMINARIAS DENTRO DEL LABORATORIO DE MAQUINAS

Luminarias	Lúmenes (lux)
Luminaria 1	420
Luminaria 2	354
Luminaria 3	309
Luminaria 4	263
Luminaria 5	304
Luminaria 6	329
Luminaria 7	439

Fuente: Elaborado por el autor

Además, se puede observar en a Figura 25 la conexión eléctrica de cada una de las luminarias dentro del laboratorio de máquinas de la carrera de mantenimiento eléctrico.

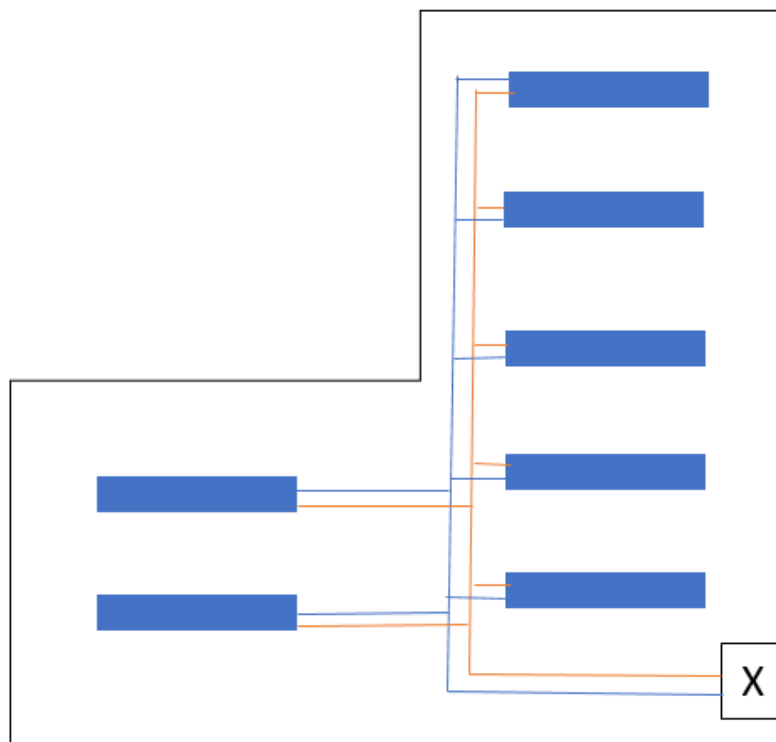


Fig. 25. Conexión eléctrica laboratorio de máquinas. Fuente: Elaborado por el autor

Las medidas de las luminarias y de las ventanas se pueden observar en la Figura 26 y Figura 27 respectivamente.

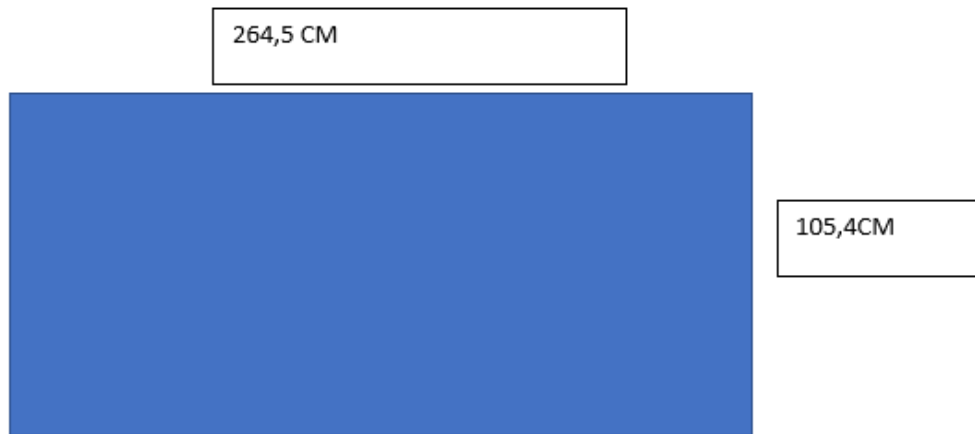


Fig. 26. Medidas ventana laboratorio de máquinas. Fuente: Elaborado por el autor

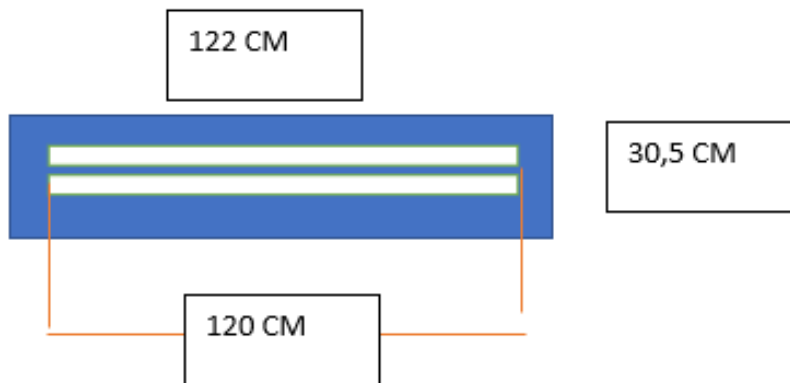


Fig. 27. Medidas soporte luminarias laboratorio de máquinas. Fuente: Elaborado por el autor

### 3.3.2. Propósito y ámbito del sistema

Se propone elaborar un sistema IoT para los laboratorios de la carrera CIELE que serán implementados en el laboratorio de máquinas y que se encargara de controlar las luminarias por medio de un grupo de señales obtenidas de parte de los sensores de luz.

Este sistema utilizará sensores de luz ubicados uno por cada luminaria que se encuentra en la zona de trabajo para la elaboración de medición de la cantidad de luz que tiene el aula de esta manera controlar la intensidad lumínica de las lámparas que se encuentran instaladas y así optimizar los recursos disponibles.

### 3.3.3. Requerimientos

Realizando un análisis previo con la situación actual, el propósito y descripción del problema se puede continuar con el requerimiento del sistema. Según el estándar internacional ISO/IEC/IEEE 29148 se debe plantear los requerimientos del sistema a realizarse y de su arquitectura; cada uno de estos requerimientos tienen las siguientes características:

- Ser verificable
- Que se pueda cumplir
- Condiciones medibles y restricciones

- Defina el rendimiento del sistema

Además, cada requerimiento consta de varios atributos como es una identificación, una prioridad, dependencia, riesgo y dificultad. Son requerimientos que reflejan las necesidades de las personas relacionadas de manera directa o indirectamente con la implantación del sistema, con una valoración de acuerdo a su prioridad.

### Nomenclatura

En la siguiente tabla se detalla una nomenclatura para identificar los requerimientos, lo cual ayudará a tener un mejor manejo de los datos de los requerimientos.

*TABLA 6*  
ABREVIATURA DE REQUERIMIENTOS.

Requerimiento	Abreviatura
Stakeholders	STSR
Sistema	SYSR
Arquitectura	SRSR

Fuente: Elaborado por el autor

#### 3.3.3.1. Requerimientos de Operación y de Usuario

Se dan a conocer los parámetros dados por los stakeholders y usuarios del sistema, los requerimientos operacionales se detallan a continuación en la tabla 7.

*TABLA 7*  
REQUERIMIENTOS OPERACIONALES

Requerimiento: StRS – Operación					
Número	Requerimiento	StRS			Relación
		Prioridad			
		Alta	Media	Baja	
<b>StRS1</b>	Interoperabilidad del sistema	X			
<b>StRS2</b>	Alimentación portable		X		
<b>StRS3</b>	Comunicación Inalámbrica entre nodos	X			
<b>StRS4</b>	Soporte comunicación I2C		X		
<b>StRS5</b>	Portable		X		

Fuente: Elaborado por el autor

Estos requerimientos se destacan de las observaciones que realizó el tutor encargado del proyecto para la implementación final en el entorno de trabajo definido como laboratorio de máquinas eléctricas.

### 3.3.3.2. Requerimientos del sistema

Son los requerimientos que se basan en las funciones que necesita realizar el sistema para su funcionamiento, los cuales se detallan en la Tabla 8.

TABLA 8  
REQUERIMIENTOS DEL SISTEMA

SYSR				
REQUERIMIENTO DE USO				
Número	REQUERIMIENTO	PRIORIDAD		
		ALTA	MEDIA	BAJA
SYSR1	Adaptable y fácil de colocar	X		
SYSR2	Lectura rápida de datos		X	
SYSR3	Disponibilidad de funcionamiento		X	
REQUERIMIENTO DE RENDIMIENTO				
SYSR4	Censado de datos por tiempo prolongado		X	
SYSR5	Almacenamiento de datos en base de datos	X		
REQUERIMIENTO DE INTERFACES				
SYSR6	Conectividad Wi-Fi	X		
SYSR7	Interfaz RS232		X	
SYSR8	Interfaz I2C		X	
SYSR9	Puertos digitales y analógicos	X		
REQUERIMIENTO DE MODOS O ESTADOS				
SYSR10	Modo reposo			X
SYSR11	Comunicación MQTT constante	X		
REQUERIMIENTO DE FÍSICOS				
SYSR12	Tamaño reducido de los componentes		X	
SYSR13	Instalación práctica		X	
SYSR14	Sensores Portátiles	X		

Fuente: Elaborado por el autor

En base a los requerimientos del sistema establecidos es necesario definir que se necesitarán varios componentes que puedan proveer de lo establecido.

### 3.3.3.3. Requerimientos de Arquitectura

Los requerimientos de arquitectura hacen referencia a las necesidades y las características para el software y hardware que se presentan en el sistema, se detalla en la tabla 9.

TABLA 9  
REQUERIMIENTOS DE LA ARQUITECTURA

REQUERIMIENTOS DE ARQUITECTURA					
Número	Requerimiento	Prioridad			Relación
		Alta	Media	Baja	
<b>Requerimiento: SRSH – Lógicos</b>					
<b>SRSH1</b>	Compatibilidad de software y hardware que faciliten la implementación del sistema	X			
<b>SRSH2</b>	El microprocesador debe tener puerto serial	X			
<b>SRSH3</b>	Interfaz de visualización de datos.	X			
<b>Requerimiento: SRSH – Software</b>					
<b>SRSH4</b>	Compatibilidad de librerías y fiabilidad en la comunicación.	X			
<b>SRSH5</b>	Placa electrónica de desarrollo que utilice software libre.	X			



<b>SRSH6</b>	Utilización de comunicación I2C o conversión ADC	X
<b>SRSH7</b>	Soporte protocolos MQTT	X
<b>SRSH8</b>	Protocolos actuales para IoT, Bluetooth 5.0 y Wifi 802.11ac	X
<b>SRSH9</b>	Capacidad de almacenamiento hasta 32GB	X
<b>SRSH10</b>	Procesamiento de datos de 1.5Ghz	X
<b>Requerimiento: SRSH –Hardware</b>		
<b>SRSH11</b>	Compatibilidad de librerías y fiabilidad en la comunicación con Wi-Fi	X
<b>SRSH12</b>	Compatibilidad de hardware con los diferentes componentes para reproducir las diferentes instrucciones.	X
<b>SRSH13</b>	Tener mínimo 8 pines PWM	X
<b>SRSH14</b>	Integración de módulos Bluetooth 5.0 y Wifi 802.11ac	X
<b>SRSH15</b>	Soporte para memoria desde 32GB	X

<b>SRSH16</b>	Procesador de 1.5Ghz	X
<b>SRSH17</b>	Voltaje 110v – 5v	X

Fuente: Elaborado por el autor

### 3.3.4. Elección de Hardware y Software para el diseño

La elección del Hardware y Software del sistema se establecerán de acuerdo con los requerimientos obtenidos anteriormente en función a las necesidades, de manera que se garantice la funcionalidad del sistema a desarrollarse.

#### 3.3.4.1. Elección del Software

Es importante seleccionar el software que se aplicará en el proyecto en base a las tablas de requerimientos en base a la tabla 7, 8 y 9. De esta manera se obtendrá la selección idónea para cada una de las funcionalidades del proyecto.

- Lenguaje de programación

Corresponde al software que se utilizará para la programación del proyecto final, esto se observa en la Tabla 10.

TABLA 10  
VALORACIÓN REQUERIMIENTO DE LENGUAJE DE PROGRAMACIÓN

Lenguaje de Programación							
Tipo	Requerimiento						Valoración
	SYSR1	SRSH3	SRSH4	SRSH7	SRSH8	SRSH4	
Arduino IDE(Esp32, Arduino) (Lenguaje C)	1	1	1	1	1	1	6
Raspberry (Python)	1	1	1	1	1	1	6

Fuente: Elaborado por el autor

Elección: El lenguaje de programación C para Arduino IDE y Python para Raspberry cumple con los requisitos de manejar librerías, comunicación mqtt, manejo de un interfaz de usuario final y ser fácil de adaptar.

- Plataforma IoT

De varias plataformas existentes en la actualidad se puede destacar las siguientes que son las de mayor utilización, en la Tabla 11, se tiene las observaciones de la plataforma.

TABLA 11  
VALORACIÓN REQUERIMIENTO DE PLATAFORMA IOT

Plataforma IoT							
Tipo	Requerimiento						Valoración
	SiRS1	SYSR1	SYSR2	SYSR4	SYSR5	SRSH5	

Node Red	1	1	1	1	1	1	6
Grafana	1	1	1	1	1	0	5
Ubidots	1	1	1	1	0	0	4

Fuente: Elaborado por el autor

Elección: Se seleccionó Node Red debido a la valoración máxima total que se observa, además se calificó tomando en cuenta la interoperabilidad, que sea adaptable a cualquier sistema, realice lecturas rápidas, haga censado continuo, permita el almacenamiento y utilice software libre.

### 3.3.4.2. Elección del Hardware

Para la selección del Hardware se considera el propósito del desarrollo e implementación del sistema para que solviente todas las necesidades que se tenga para todos los involucrados en la investigación.

Se analizan placas de Arduino, ESP32, donde se dará una puntuación de “1” a las funciones que cumpla este dispositivo y con el valor de “0” cuando no cumpla, al final se sumará los valores y se determinara cual es el mejor.

#### Módulo de distribución de potencia

Para el sistema de distribución de potencia es necesario realizar una selección del microcontrolador que se usará para el control de PWM y el censado del paso por cero de la corriente alterna, estos requerimientos se pueden observar en la Tabla 12.

TABLA 12  
VALORACIÓN REQUERIMIENTO DE HARDWARE SISTEMA DE POTENCIA

Hardware Sistema de Potencia							
Tipo	Requerimiento						Valoración
	SYSR7	SYSR9	SYSR12	SRSH12	SRSH13	SRSH17	
Arduino Uno	1	1	1	1	1	1	6
Arduino Mega	1	1	0	1	1	0	5
ESP32	1	0	1	1	0	0	3

Fuente: Elaborado por el autor

Elección: Como se observa el Arduino Uno cumple con los requisitos necesarios para la elaboración del sistema de distribución de potencia en donde se tiene que es compatible en hardware, contar con mínimo 8 puertos PWM, manejar voltaje de entrada 5v, contar con puerto de comunicación serían RS232, contar con puertos analógicos y digitales, además tener un tamaño reducido.

### Módulo de censado lumínico

Para el sistema de censado lumínico es necesario realizar una selección del microcontrolador que se usará para la lectura de sensores y la portabilidad inalámbrica, estos requerimientos se pueden observar en la Tabla 13.

TABLA 13  
VALORACIÓN REQUERIMIENTO DE HARDWARE SISTEMA DE CENSADO

Hardware Sistema de Censado											
Tipo	Requerimiento										Valoración
	SiRS2	SiRS3	SiRS4	SiRS5	SYSR3	SYSR4	SYSR6	SYSR310	SRSH11	SRSH14	
Arduino UNO	1	0	1	1	1	1	0	0	0	0	5
ESP8266	1	1	1	1	1	1	1	1	1	1	10
ESP01	1	1	0	1	1	0	1	0	1	1	7

Fuente: Elaborado por el autor

Elección: Como se observa en se escoge el ESP82 al cumplir los requisitos necesarios para la elaboración del sistema de censado por tener: Alimentación portable; Comunicación inalámbrica entre nodos; Soporte I2C; portable por su pequeño tamaño; Lectura rápida de sensores; conectividad Wi-Fi; Modo reposo; Compatibilidad Wi-Fi.

### Módulo de control

Para el sistema de distribución de potencia es necesario realizar una selección del microcontrolador que se usará para el control de PWM y el censado del paso por cero de la corriente alterna, estos requerimientos se pueden observar en la Tabla 14.

TABLA 14  
VALORACIÓN REQUERIMIENTO DE HARDWARE SISTEMA DE CONTROL

Hardware Sistema de Control											
Tipo	Requerimiento										Valoración
	SYSR1	SYSR2	SYSR3	SRSH3	SRSH5	SRSH7	SRSH11	SRSH14	SRSH15	SRSH16	
Raspberry2	1	1	0	1	1	1	0	0	0	0	5
Raspberry3	1	1	1	1	1	1	1	0	1	0	8
Raspberry4	1	1	1	1	1	1	1	1	1	1	10

Fuente: Elaborado por el autor

Elección: Como se observa en la Tabla 14 Raspberry Pi 4 cumple con los requisitos para elaborar el módulo de control los cuales son: sistema adaptable y fácil de instalar; lectura rápida de los datos; disponibilidad de funcionamiento; posibilidad de visualización de los datos; utilización de software libre; compatibilidad de protocolo MQTT; compatibilidad con la última tecnología Wi-Fi y Bluetooth 5.0; capacidad de almacenamiento superior 32 Gb y un procesador igual o superior a 1.5Ghz con memoria RAM de 4MB.

### 3.3.5. Recursos

#### 3.3.5.1. Recursos Humanos

Comprende a los seres humanos que intervienen en el trabajo. Se especifica en la siguiente tabla 15.

TABLA 15  
RECURSOS HUMANOS

Recursos Humanos	
Estudiantes	Estudiantes de la carrera de Electricidad
Tutor	Ing. Francisco Naranja
Empresa	UNIVERSIDAD TÉCNICA DEL NORTE
Desarrollador	Diego Fernando Puetate Paspuezan

Fuente: Elaborado por el autor

#### 3.3.5.2. Recursos Económicos

Permite evaluar un costo aproximado del prototipo final. La tabla de recursos económicos se presenta a continuación.

TABLA 16  
RECURSOS ECONÓMICOS

ECONÓMICO			
#	MATERIAL	COSTO UNITARIO	COSTO FINAL
11	Resistencia 10kΩ	0,1	1,1
10	Resistencia 330Ω	0,1	1
10	MOC3021	0,4	4
10	TRIAC BT136	0,54	5,4
2	Resistencia 47kΩ	0,1	0,2
1	PC817c	0,54	0,54
1	Puente de diodos	0,31	0,31
1	Resistencia 1k	0,1	0,1
1	Arduino Uno	9	9
1	Raspberry Pi4	115	115
6	Esp32	13	78
5	Sensor luz BH1750	7,4	37

5	Modulo carga baterías litio	2	10
5	Batería de litio	9	45
1	Motor DC	15	15
1	Driver motor	15	15
<b>Total</b>			<b>336,65</b>

Fuente: Elaborado por el autor

### 3.3.5.3. Recursos Tecnológicos

El uso de software que permitirá culminar el sistema. La información se resume en la siguiente tabla.

TABLA 17  
RECURSOS TECNOLÓGICOS

<b>TECNOLÓGICOS</b>	
<b>SOFTWARE</b>	<b>COSTO</b>
ARDUINO IDE	0
RASPBIAN	0
NODE RED	0
GRAFANA	0
EAGLE	0
FRITZING	0
<b>TOTAL</b>	<b>0</b>

Fuente: Elaborado por el autor

### 3.4. Diseño del prototipo

El sistema consiste en un grupo de sensores que se encargará de realizar mediciones de los lúmenes en el área para enviarlos hacia un controlador el que en base a los datos que recibe se encargue de tomar decisiones para variar la intensidad de las luminarias o realizar la apertura de la persiana designada para su apertura en caso de tener luz natural suficiente para cubrir la demanda de lúmenes necesaria.

Esta información se encontrará almacenada en una base de datos que permita tener un histórico de las variaciones en el transcurso de los días de los lúmenes presentes en el aula para de esta manera tener registro de las variaciones y cambios en las zonas de trabajo.

Para poder establecer los requisitos del sistema se debe tomar en cuenta el sistema general que se busca implementar por lo cual se tiene en la Figura 28 un diagrama de bloques de los módulos integrados.

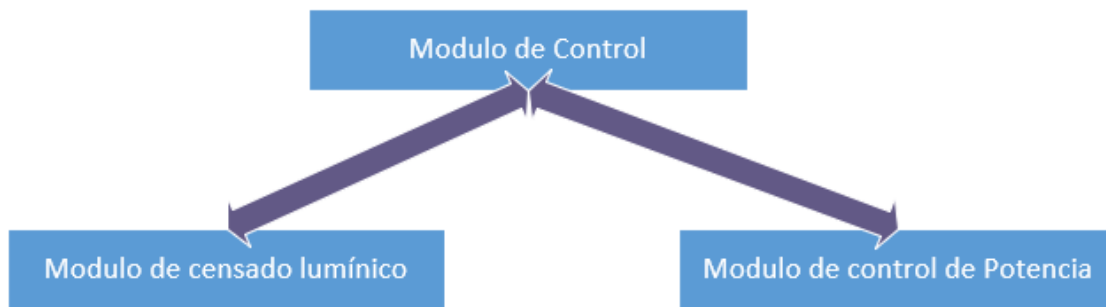


Fig. 28. Sistemas del proyecto. Fuente: Elaborado por el autor

En la Figura 29 se muestra el sistema completo y cada uno de los módulos que lo integran de una forma más detallada.

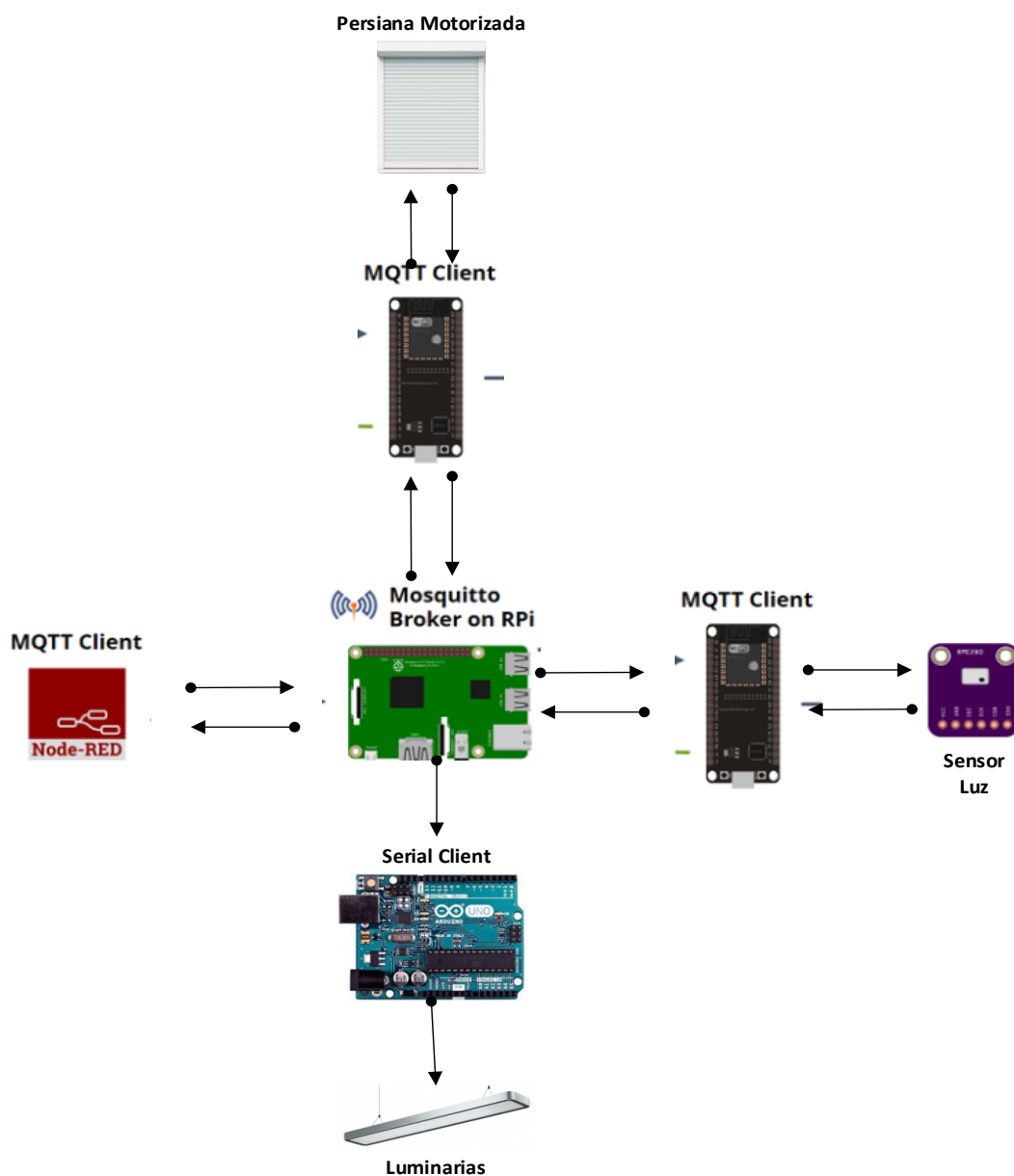


Fig. 29. Sistemas del proyecto. Fuente: Elaborado por el autor

### 3.4.1. Módulo de Control

Este módulo consiste en la parte central tanto de recepción y envío de datos desde el módulo de distribución de potencia hacia el módulo de censado, todo esto como se observa en la Figura 30.

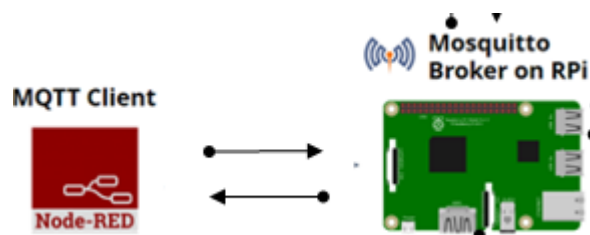


Fig. 30. Módulo de Control. Fuente: Elaborado por el autor

Para esto se necesita utilizar un Raspberry Pi 4, e instalar todos los módulos necesarios los cuales se enumeran a continuación:

- Mosquito MQTT
- Node Red
- InfluxDb

El mosquito mqtt se encargará de recibir las interacciones por medio de tópicos desde y hacia los sensores, mientras que node red provee el interfaz de comunicación y usuario para visualización y toma de decisiones, e Influx Db realizará la integración de todos los datos a una base de datos para crear un histórico de información.

### 3.4.2. Lógica de funcionamiento

Como se observa en la Figura 31 el módulo envía una señal por MQTT a los módulos de censado y espera una respuesta, caso contrario se verifica que los módulos se encuentren conectados, si recibe una respuesta se envía los datos hacia una base de datos y se grafica en el interfaz como es Grafana y Node red para visualizarlo por medio de una interfaz web, al finalizar envía los datos de lectura al módulo de control de potencia para variar la intensidad de las luminarias.



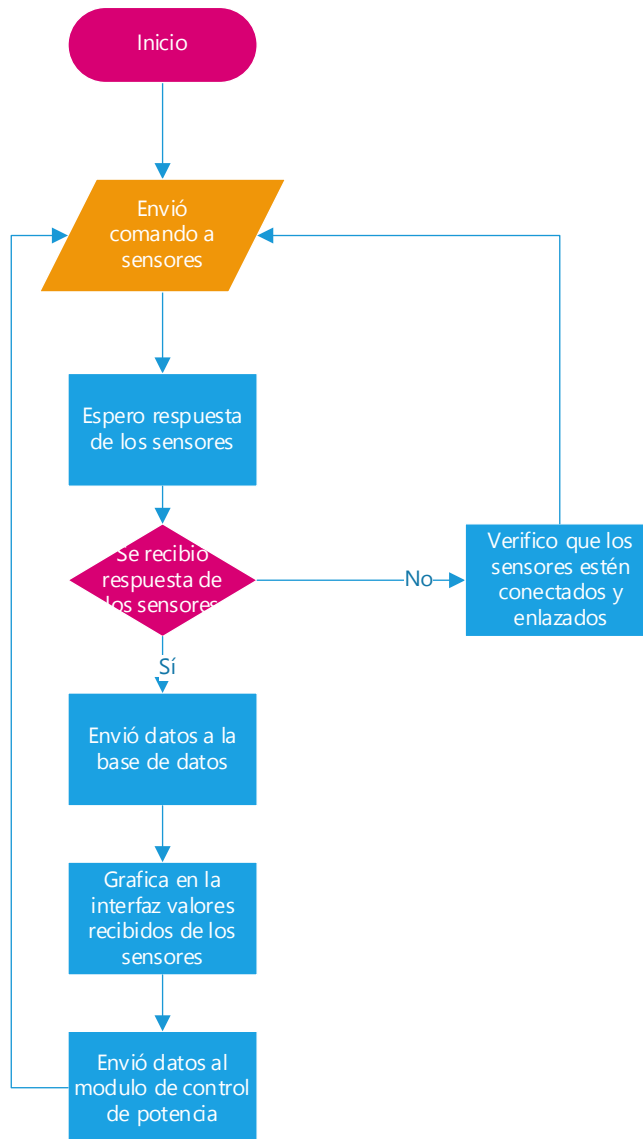


Fig. 31. Diagrama de Flujo Módulo de Control. Fuente: Elaborado por el autor

### 3.4.2.1. Elaboración

Para la elaboración del módulo de control se toma en cuenta los siguientes requisitos que se establecerán que son:

#### Interfaz de monitorización de lúmenes por luminaria

Para este proceso de monitorización se procedió a usar Node red que permite procesar los datos recibidos por medio de MQTT desde cada uno de los sensores que se utilizan en el área de trabajo. El proceso de instalación de node red se puede observar en el Anexo 10, una vez instalado se usa un navegador para ingresar al modo de configuración por medio de la ip 192.168.0.4:1880 y se observara la pantalla que se muestra en la Figura 32.

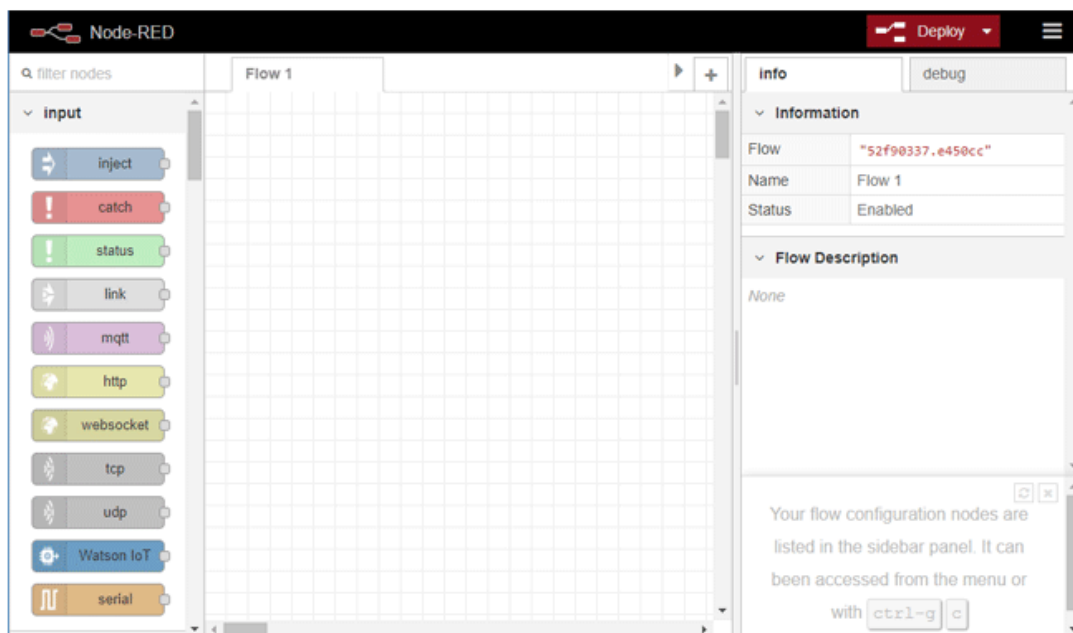


Fig. 32. Interfaz de Node-RED. Fuente: Elaborado por el autor

Dentro de este interfaz se procede a realizar la configuración de la lectura de tópicos en MQTT para esto se realiza el siguiente procedimiento, Primero se selecciona el cuadro de input MQTT como se muestra en la Figura 33.



Fig. 33. Nodo MQTT. Fuente: Elaborado por el autor

En este nodo es necesario conectar el servidor MQTT que para el proyecto se definió como luminarias y password UTN2021 y el topic al que se enlazaran las lámparas es /zona1/lumenes/1 donde 1 es el número de luminaria y se lo cambiara dependiente de la luminaria a configurar.

Una vez enlazado cada vez que llegue una lectura de los sensores se almacenara en una base de datos con el tiempo de recepción, ese valor se visualizara en los indicadores y se enviara a un archivo de texto en donde el programa principal utilizara esos datos para realizar los cálculos y el establecimiento de los lúmenes requeridos. El programa final para una luminaria se observa en la Figura 33.

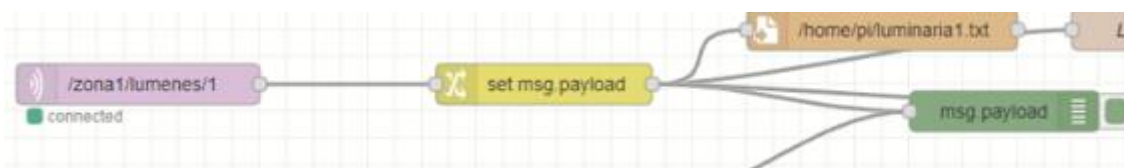


Fig. 34. Nodo MQTT. Fuente: Elaborado por el autor

Para observar los parámetros en el interfaz de node red se usa la dirección 192.168.4.0/ui/dashboard como se observa en la Figura 35.

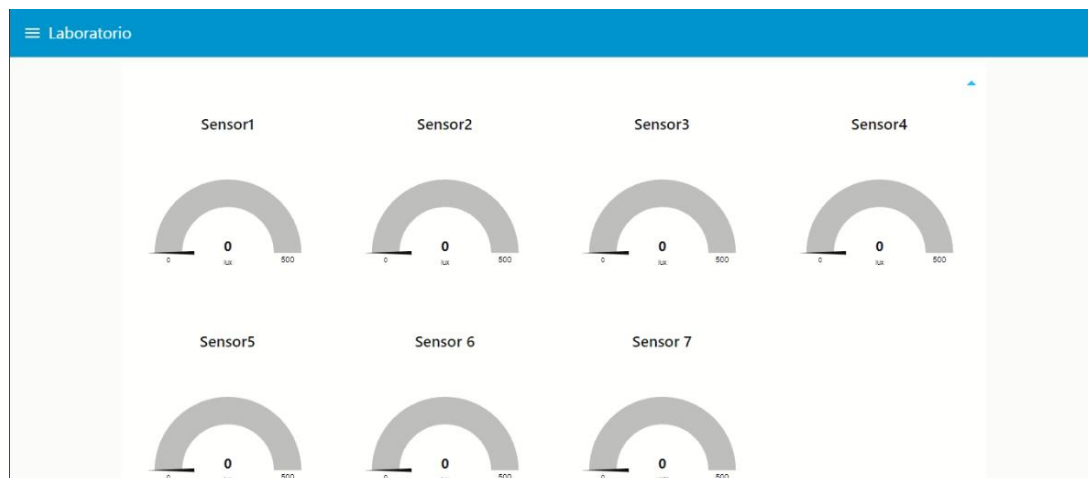


Fig. 35. Visualización lúmenes. Fuente: Elaborado por el autor

Además, se procede a la elaboración de un sistema que permita enviar los datos al motor para controlar la persiana de forma remota por medio del interfaz web, esto se realiza agregando un botón dentro del interfaz, como se muestra en la Figura 36 por medio de un bloque de botón y otro para envié de datos MQTT al topic /laboratorio/luminaria/1.

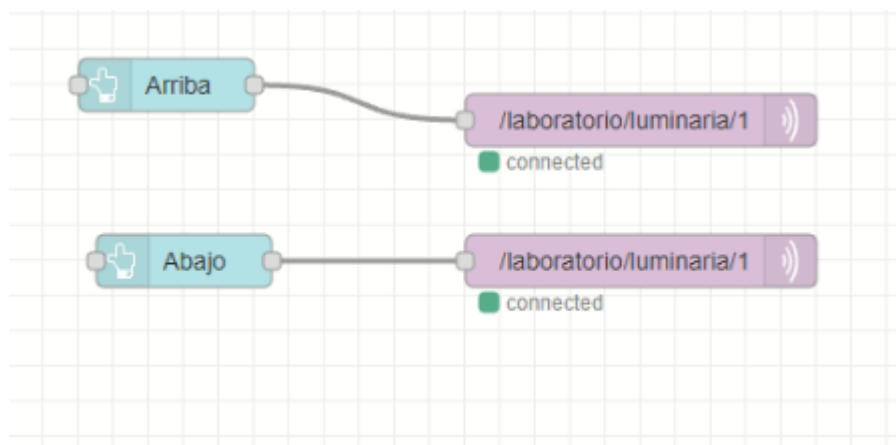


Fig. 36. Visualización lúmenes. Fuente: Elaborado por el autor

Y el interfaz se mostrará en la página del servidor como se muestra en la Figura 37.

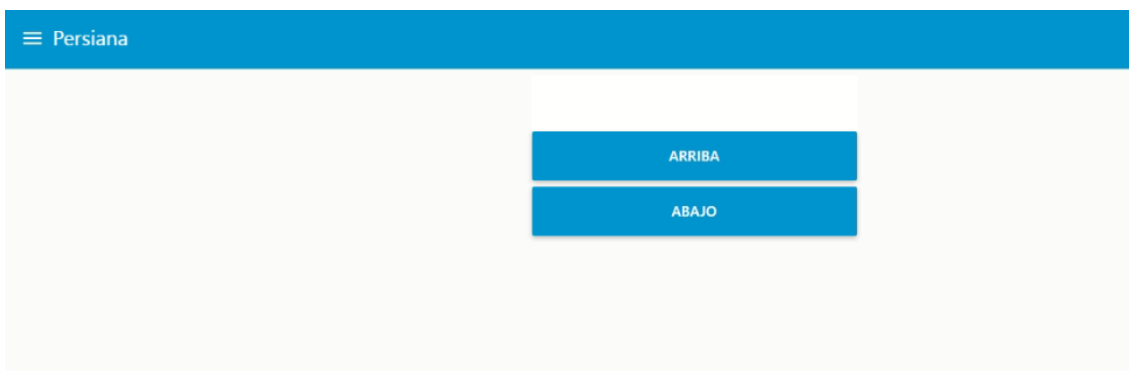


Fig. 37. Interfaz control motor persiana. Fuente: Elaborado por el autor

## Interfaz de monitorización de valores de lúmenes por el tiempo de trabajo

Además de los servidores descritos con anterioridad también se procede a utilizar Grafana para realizar la monitorización de cada uno de las luminarias en el tiempo de forma gráfica utilizando la base de datos creada para cada una de las luminarias, lo que se muestra en la Figura 38. La instalación del servidor esta descrita en el Anexo 11.

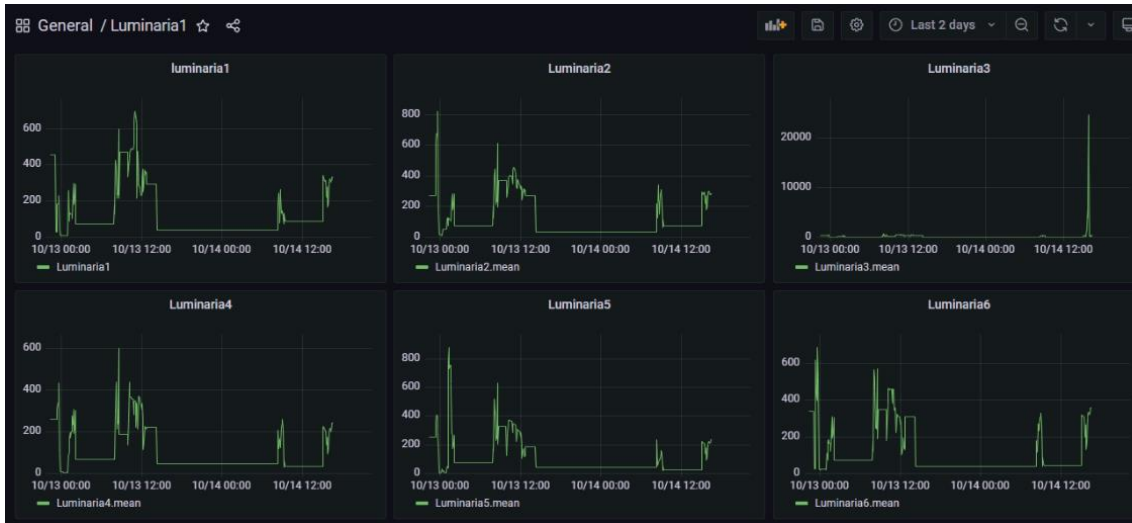


Fig. 38. Interfaz Grafana lúmenes por tiempo. Fuente: Elaborado por el autor

## Servidor MQTT

El servidor se encarga de generar los tópicos y las comunicaciones necesarias para la realización de envío y recepción de mensajes desde el módulo central hacia los módulos de censado con su respectiva luminaria, el funcionamiento se puede observar en la Figura 39.

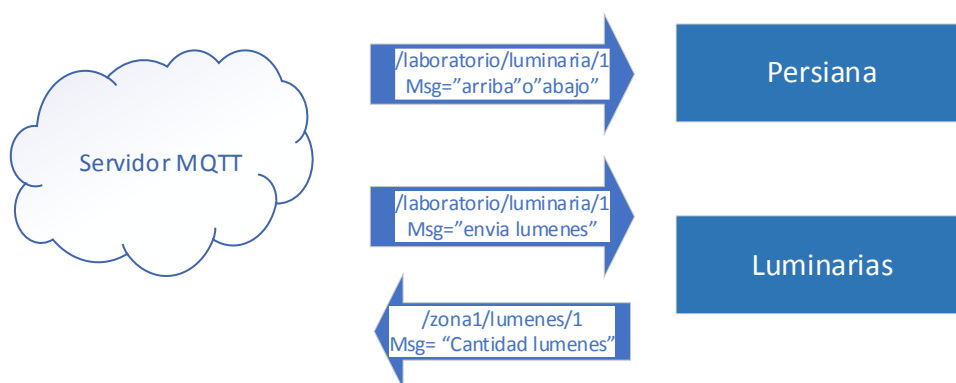


Fig. 39. Funcionamientos tópicos. Fuente: Elaborado por el autor

El servidor solicita por medio de mensajes publicados en un tópico para envío que reciben todos los clientes MQTT y de la misma manera los clientes publican su respuesta en su respectivo tópico.

## Base de datos

Se encarga de almacenar los datos que provienen de cada una de los sensores y así poder visualizar un estadístico en relación a la fecha de captura permitiendo de esta forma obtener un histórico de los procesos y funcionamiento del sistema. Se utilizo influx db que es una base de datos basada en mysql que permite crear una base de datos reducida para sistemas o procesos menos complejos, para instalar lo único necesario es ingresar el siguiente comando:

```
sudo apt-get install influxdb
```

Lo único que se realizo es crear una base de datos por sensor así se almacena los datos de manera independiente.

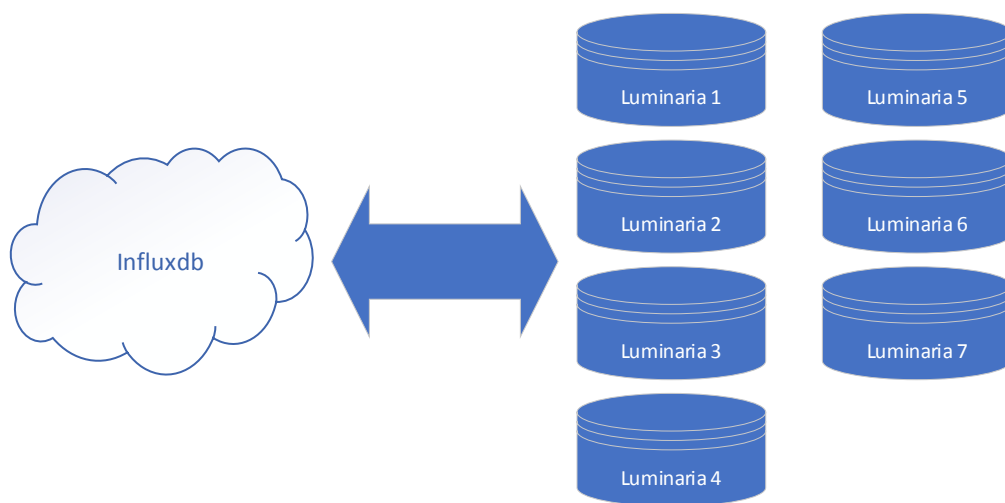


Fig. 40. Estructura influxdb. Fuente: Elaborado por el autor

En la Figura 40 se puede observar las bases de datos que integran el influxdb, de esta forma cada uno de los sensores entrega la lectura que se almacenara por tiempo.

## Programa para censado y control de luminancia

El programa se desarrolló en Python usando la lógica que se presenta en el diagrama de flujo que se muestra en la Figura 41.

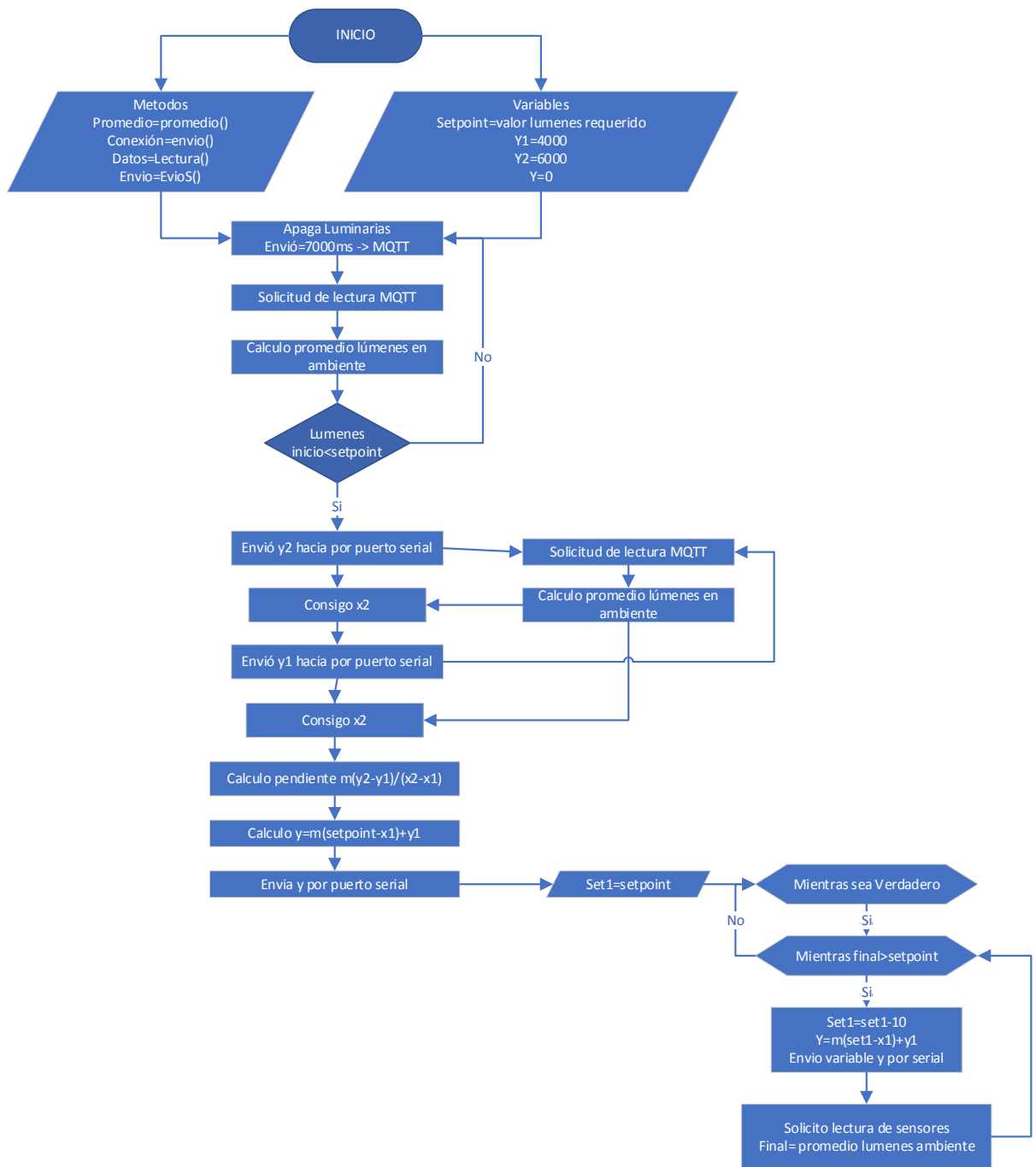


Fig. 41. Diagrama de Flujo código en Python. Fuente: Elaborado por el autor

### 3.4.3. Módulo de Distribución de Potencia

Para realizar el control de las luminarias se utilizó el concepto de dimmer digital controlado que será controlado por un microcontrolador como en este caso será un Arduino Uno. Dentro de las partes que conforman este circuito se tendrá una etapa de control de cruce por cero para poder recortar la onda senoidal que recibe la luminaria y de esta forma controlar la cantidad de potencia que se envía a la luminaria. Para este propósito existen dos etapas elaboradas dentro de un mismo circuito las cuales son:

- Circuito de cruce por cero
- Circuito de control de potencia o dimmer digital

### 3.4.3.1. Circuito de cruce por cero

Este circuito tiene la finalidad detectar cuando la onda alterna hace su pase por cero para indicar en qué momento se tiene un cruce o cambio de frecuencia, esto se emplea para poder sincronizar los componentes eléctricos y electrónicos en base a la frecuencia de trabajo de la corriente alterna.

El diagrama del detector de onda se observa en la Figura 42, donde se consta con todos los componentes.

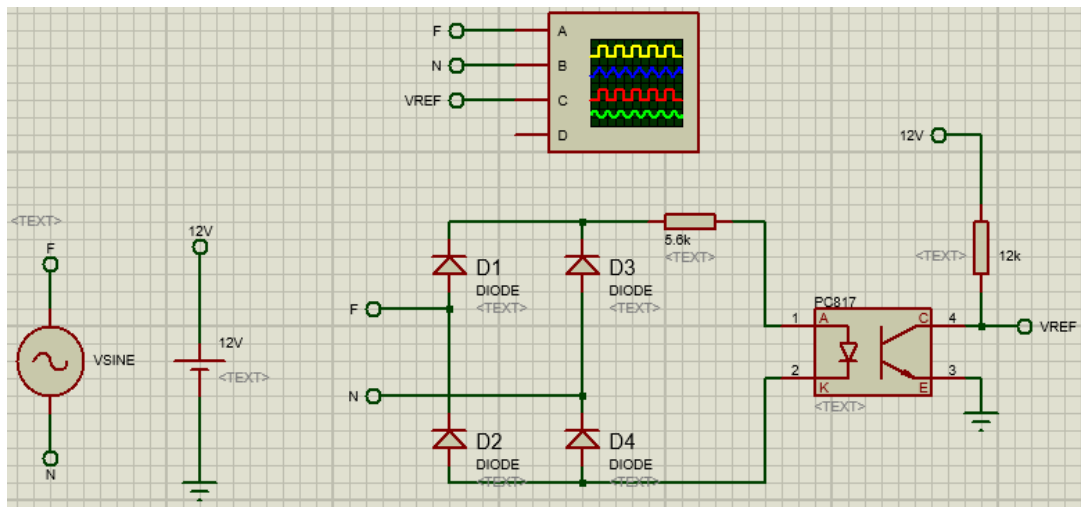


Fig. 42. Detector de onda completo. Fuente: Elaborado por el autor

Y en la Figura 43 se tiene tanto la onda senoidal como la onda filtrada que permite elaborar un conteo del tiempo necesario entre cada cambio para poder cortar la potencia que recibe la luminaria.

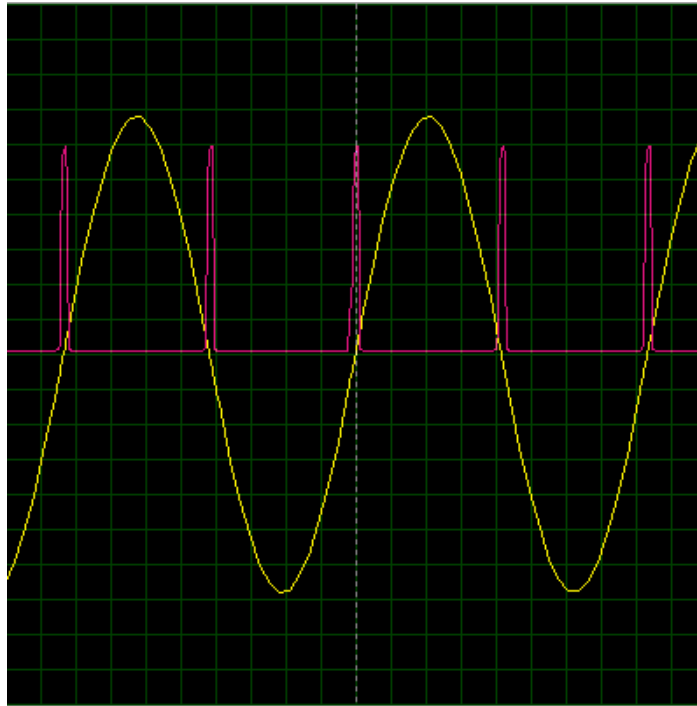


Fig. 43. Señal detectora de onda completo. Fuente: Elaborado por el autor

Como se puede observar en la Figura 43 tomada por el osciloscopio en la simulación se tiene que la onda de color amarillo es la onda de la corriente AC, mientras que la señal en color rosa es la salida que se tiene después del PC817c y se enviar a un pin digital de un microcontrolador para la detección en modo de interrupción.

#### 3.4.4. Circuito de control de potencia o dimmer digital

Para el control de potencia en las luminarias se utilizará un control de fase y para esto es necesario un pin digital que permita disparar un TRIAC que será el encargado de permitir la activación de las luminarias en el momento indicado.

El objetivo de este circuito es realizar un control de activación de fase en base al periodo que permita obtener una onda de mayor o menor potencia dependiendo del momento en que se active el PIN desde Arduino para obtener las ondas que se muestran en la Figura 44 y permitan regular la potencia de la luminaria.

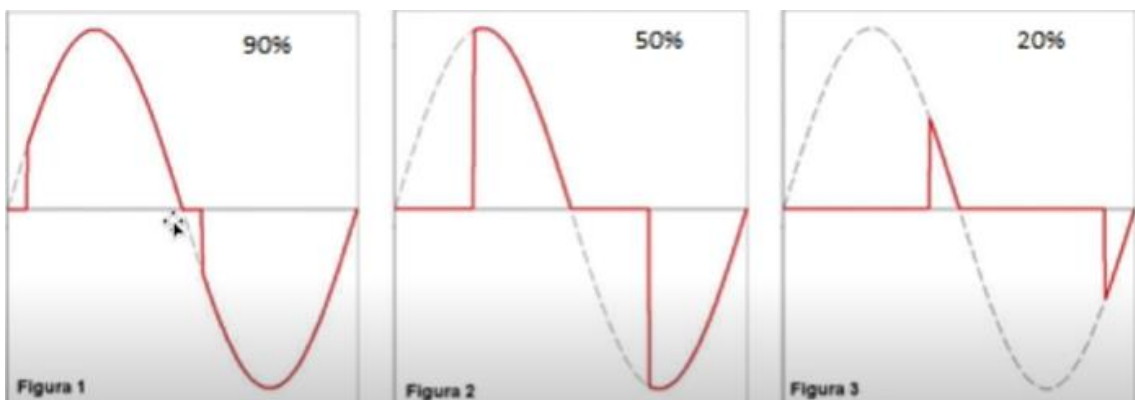


Fig. 44. Circuito cruce por cero con Arduino. Fuente: Elaborado por el autor



Una vez descrito los dos circuitos a utilizar es necesario tomar en cuenta que los dos se complementan ya que el circuito de cruce por cero se encargará de detectar el cruce por cero mientras que el circuito dimmer activará el TRIAC para que la potencia se pueda controlar en base al tiempo y la cantidad de la onda que permita que se transmita hacia la luminaria.

En la Figura 45 se puede observar cómo queda el circuito final que permite elaborar un control de potencia de luminarias por medio de Arduino, además en el Anexo 2 se obtiene el código que permitirá controlar por medio del puerto serial la potencia de una luminaria de cualquier tipo variando solo el tiempo en base a la frecuencia y potencia de la misma.

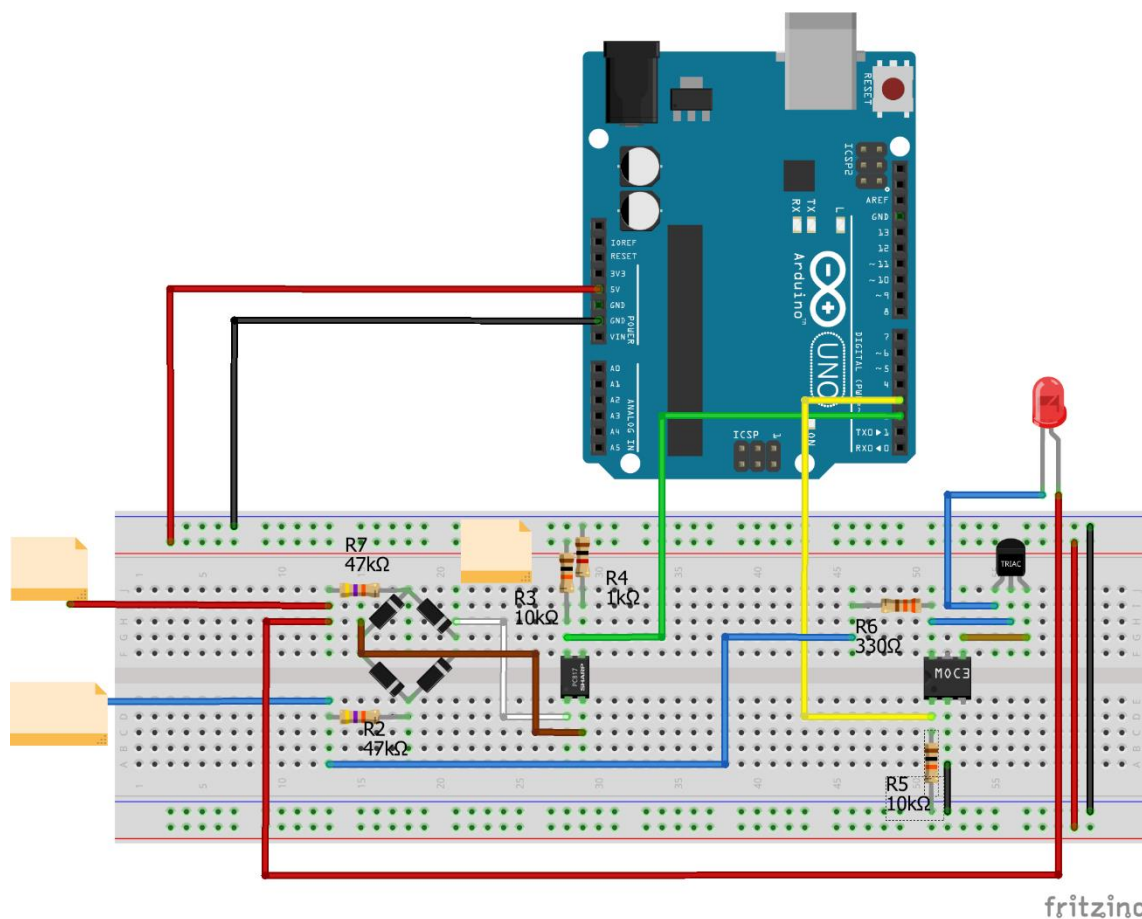


Fig. 45. Circuito control de fase tipo dimmer. Fuente: Elaborado por el autor

Una vez probado el circuito se necesita en base a las 8 luminarias elaborar un circuito en baquelita para la presentación final, para esto se utilizará el software KICAD en donde como se observa en la Figura 46 se tiene el diagrama de los componentes.

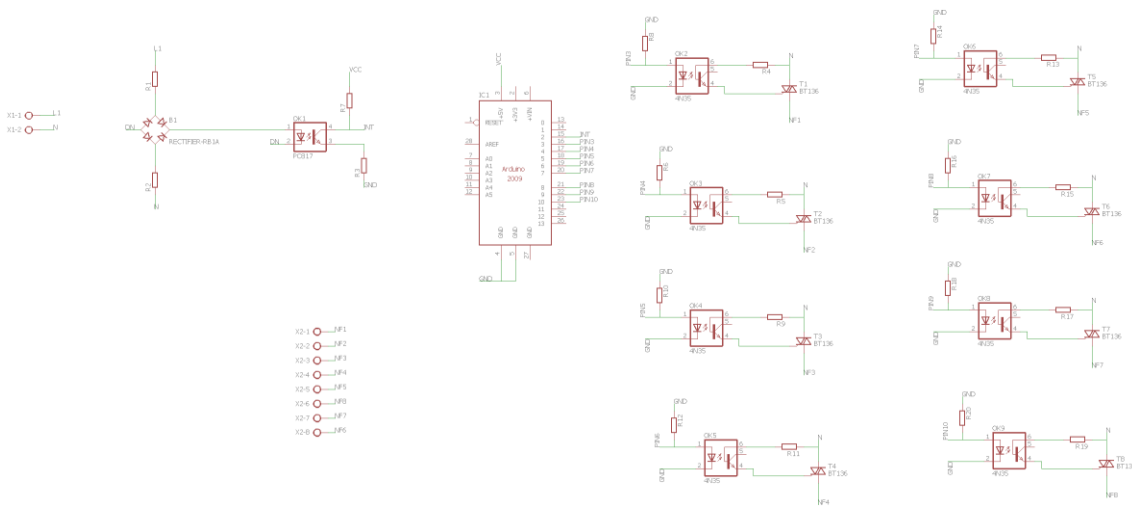


Fig. 46. Esquemático Kicad. Fuente: Elaborado por el autor

Es necesario una vez elaborado el esquemático generar el archivo para realizar el circuito impreso por lo cual en la Figura 47 se observa el archivo para imprimir las pistas y componentes.

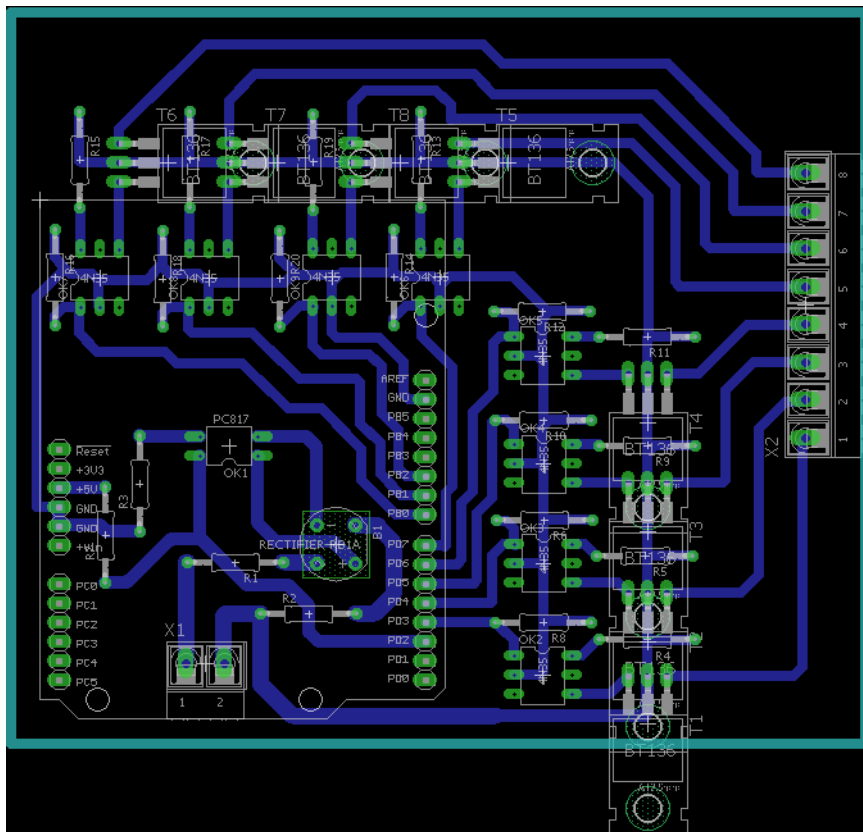


Fig. 47. Circuito para imprimir Kicad. Fuente: Elaborado por el autor

Es necesario tomar en cuenta que existen 7 luminarias en total en las zonas de aplicación del sistema por lo cual se requiere los siguientes materiales en total para cumplir con ese propósito y además elaborar el circuito impreso, estos materiales se observan en la Tabla 18.

TABLA 18  
MATERIALES SISTEMA REGULADOR DE POTENCIA

Cantidad	Material
11	Resistencia 10kΩ

8	Resistencia 330Ω
8	MOC3021
8	TRIAC BT136
2	Resistencia 47kΩ
1	PC817c
1	Puente de diodos
1	Resistencia 1k

Fuente: Elaborado por el autor

### 3.4.4.1. Elaboración

El proceso de elaboración empieza por medio de la impresión del circuito diseñado que se muestra en la Figura 47 y se obtiene por medio de un proceso utilizando la maquina CNC, la cual se encarga de taladrar las pistas de cobre necesarias para la circuitería interna. Este circuito terminado se muestra en la Figura 48.

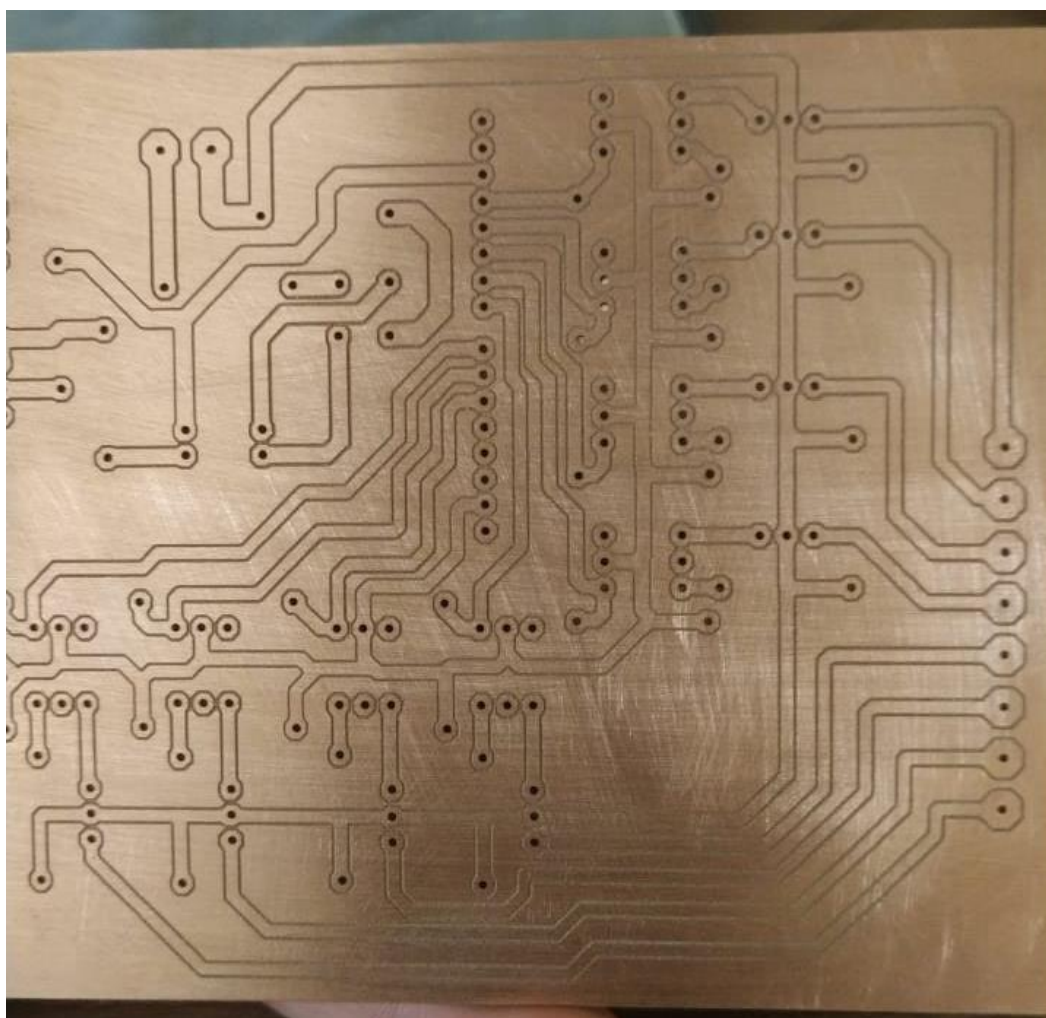


Fig. 48. Circuito impreso CNC. Fuente: Elaborado por el autor

Una vez realizado el circuito se procede a soldar los componentes que se detallaron con anterioridad para tener el resultado final que se muestra en la Figura 49.

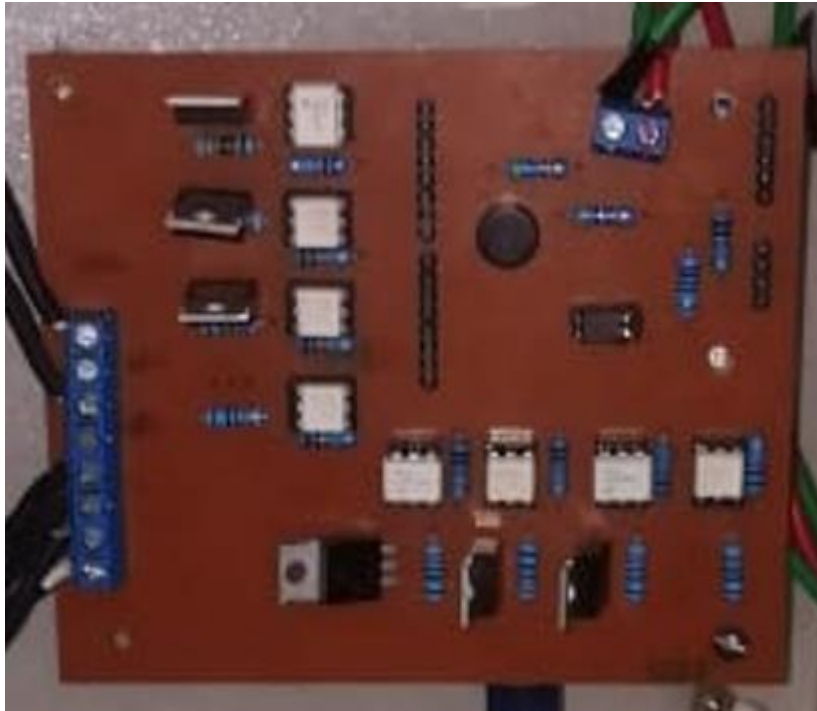


Fig. 49. Circuito con soldadura de componentes. Fuente: Elaborado por el autor

Una vez realizado el proceso de soldadura se procede a probar que los componentes funcionen de la forma deseada para esto se conecta por medio de Arduino y se procede a cargar dentro de Arduino el código necesario para realizar las pruebas de funcionamiento. Para cargar el código se utiliza el código que se encuentra en el Anexo 9, una vez cargado se envía por el puerto serial un valor entre 2000 y 7000 y como se observa en la figura 50, Figura 51 y Figura 52 la luminaria tiene una variación en su potencia.



Fig. 50. Luminarias con valor 7000. Fuente: Elaborado por el autor

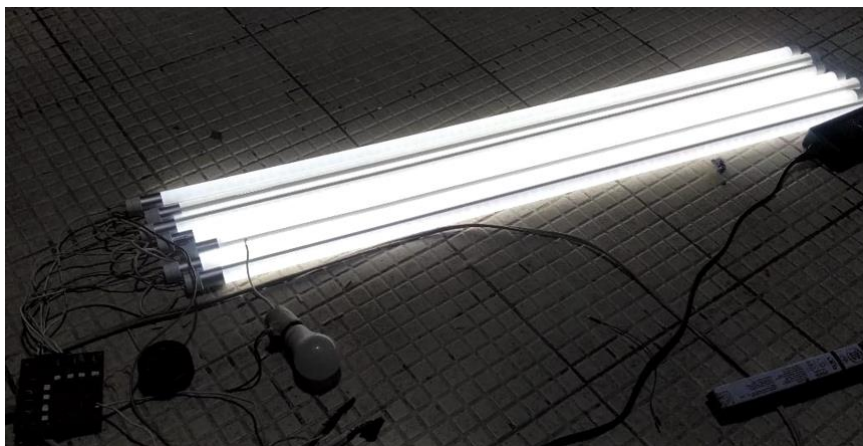


Fig. 51. Luminarias con valor 6000. Fuente: Elaborado por el autor

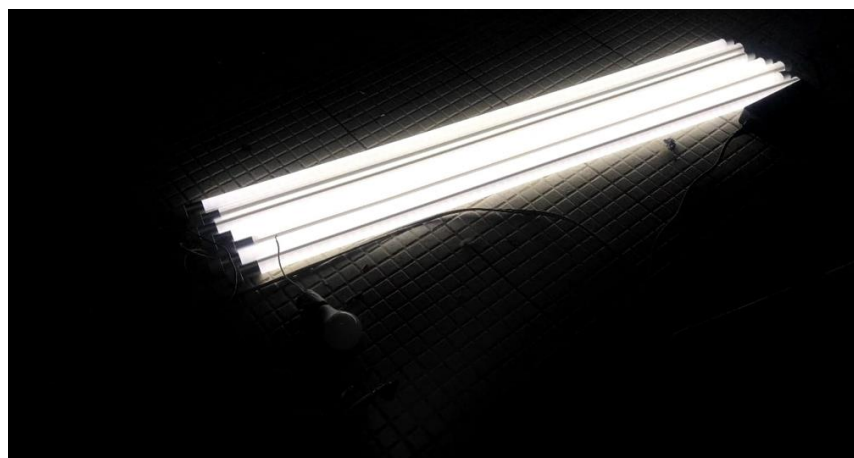


Fig. 52. Luminarias con valor 2000. Fuente: Elaborado por el autor

De estos valores encontrados con pruebas se procedió a elaborar una tabla de lúmenes irradiados en referencia a los valores de tiempo entre 2000 y 6000 ms que se envían al Arduino.

TABLA 19  
POTENCIA LÚMENES POR TIEMPO (MS)

LÚMENES	TIEMPO
50	6000
54	5900
70	5800
80	5700
90	5600
101	5500
112	5400
122	5300
133	5200
142	5100
153	5000
165	4900
173	4800
182	4700
194	4600



199	4500
203	4400
208	4300
213	4200
214	4100
225	1500

Fuente: Elaborado por el autor

Como se observa en la Tabla 19 se tiene que a menos tiempo en milisegundos mayor potencia en lúmenes y que desde los 1500 hasta los 4100 no existe una variación significativa en la potencia de las luminarias y a partir de los 6000 en adelante las luminarias se apagan.

Por lo cual se puede definir una gráfica en base a los valores que se observan en la Figura 53 en donde se puede notar que por igualación al punto más cercano se observa una línea relativamente recta para su interpretación matemática.

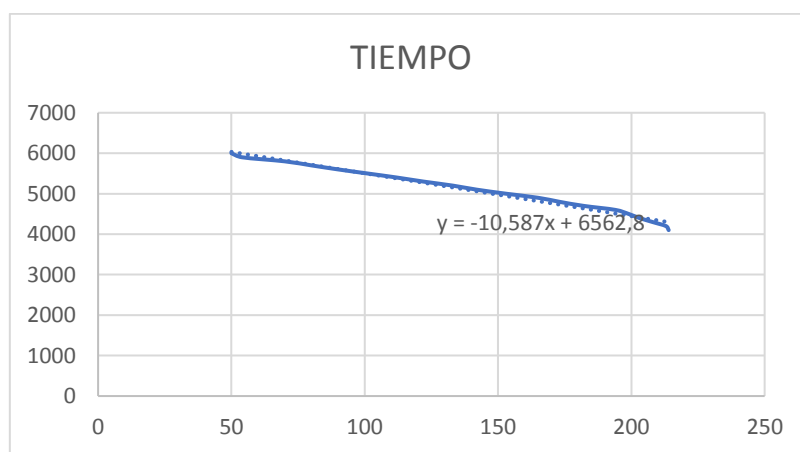


Fig. 53. Tabla de valores lúmenes por tiempo (ms). Fuente: Elaborado por el autor

Además, se utilizó el Grafana para visualizar la lectura de los sensores en todas las variaciones de tiempo para obtener una gráfica similar en tiempo real que se observa en la Figura 54.

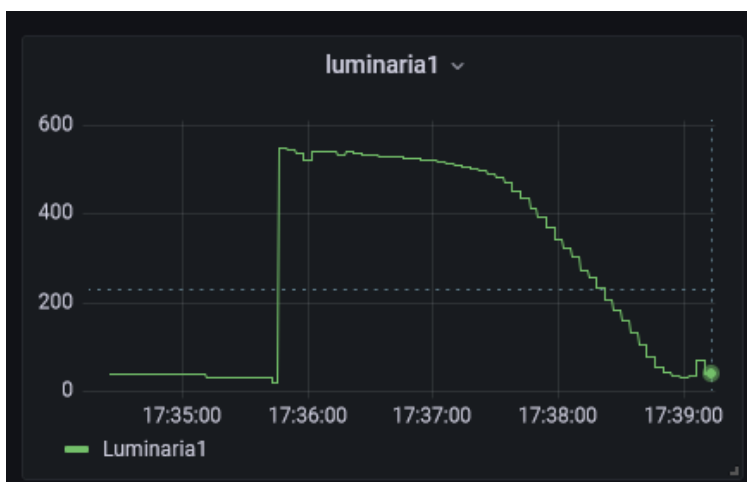


Fig. 54. Variación lúmenes en tiempo(ms). Fuente: Elaborado por el autor

En este punto es necesario notar que se usara la ecuación de la recta para interpretar los resultados y realizar el cálculo de aproximación al nivel de lúmenes que se requiere en el ambiente, la fórmula de la línea recta está definida por:

$$y = m(x - x_1) + y_1$$

En donde:

y = es igual al valor en milisegundos que se quiere establecer

m= es la pendiente de la recta

x= valor en lúmenes deseado

x1= valor en lúmenes de un punto previamente obtenido

y1= valor en milisegundos del punto previamente obtenido

Para la aplicación de esta fórmula, se procede en el código que controla la Raspberry a establecer un punto mínimo y un punto máximo de lúmenes en base a la tabla que seria 6000 (y1) milisegundos como punto mínimo de potencia y 2000 (y2) como punto máximo de potencia. Al establecer estos valores de tiempo el Arduino procederá a encender las luminarias y con los sensores se obtiene la lectura de lúmenes (x1) con mínima potencia y el valor en lúmenes (x2) con máxima potencia para de esta forma obtener la pendiente de la ecuación por medio de la siguiente formula:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

En la elaboración del código se aplica la lógica de tomar el punto máximo y mínimo, para de esta manera encontrar nuestro valor en lúmenes deseado, este programa que se elaboró en Python se encuentra en la carpeta /home/pi/Proyecto del Raspberry Pi, en donde el archivo creado se llama pendienteT.py que se encuentra detallado en el Anexo 9.

#### **3.4.5. Módulo de Censado**

El módulo consiste en elaborar un sistema de censado que permita enviar la información de los lúmenes mediante MQTT.

#### **Elementos que conforman el sistema**

En base a los estudios y elementos disponibles en el mercado se procede a seleccionar los componentes necesarios para la elaboración de un sistema completo de censado, para este propósito se prefirió elaborar un módulo inalámbrico que permita comunicación remota con el sistema de distribución de potencia y el sistema de control y puedan interactuar entre ellos.

Para la parte de autonomía energética se procede a realizar el circuito que se muestra en la Figura 55.

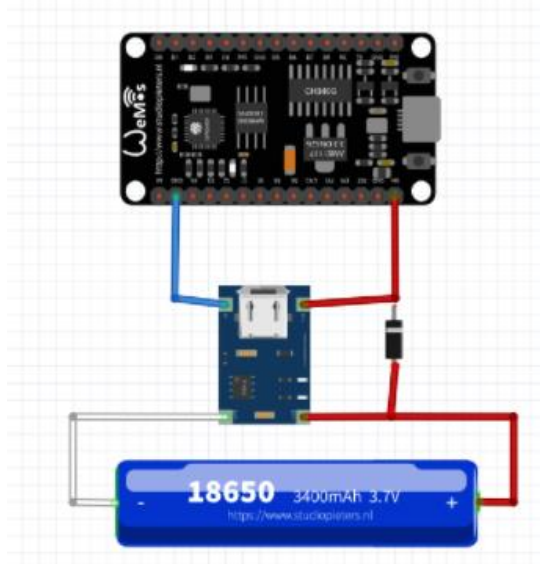


Fig. 55. Circuito alimentación Esp8266. Fuente: Elaborado por el autor

Los materiales que se utilizarán para el sistema de alimentación se utilizan:

- 1 módulo de carga tp4056
- 1 batería de Litio

El circuito para la elaboración de censado se procede a realizar la conexión entre el módulo Esp8266 y el sensor de luz BH1750 como se observa en la Figura 56, para realizar las pruebas de funcionamiento se procede a la carga del código que se encuentra en el Anexo 3.

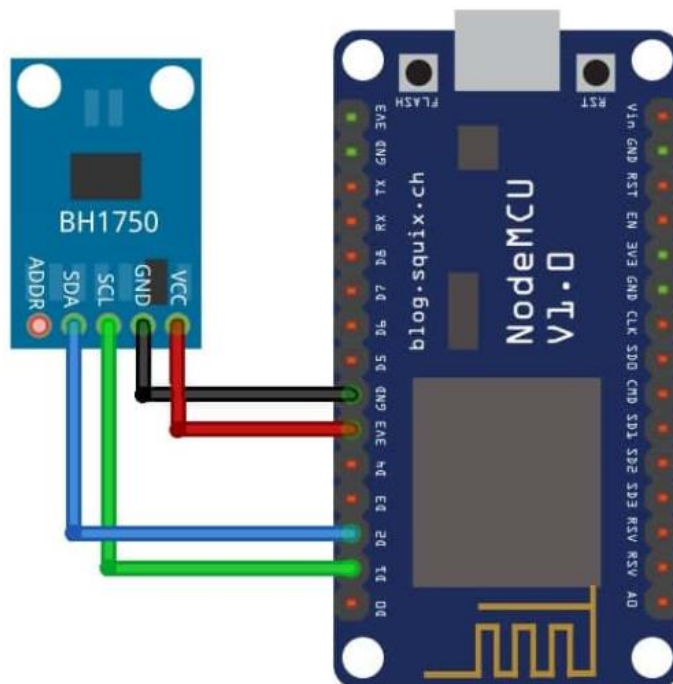


Fig. 56. Conexión Esp8266 BH1750. Fuente: Elaborado por el autor

El sistema completo a implementar se observa en la Figura 57. Esto se debe reproducir en un pequeño componente que permita manejar autonomía y funcionalidad inalámbrica, debido a esto se ocupará un cableado sencillo y un código de envió de datos al módulo de control.



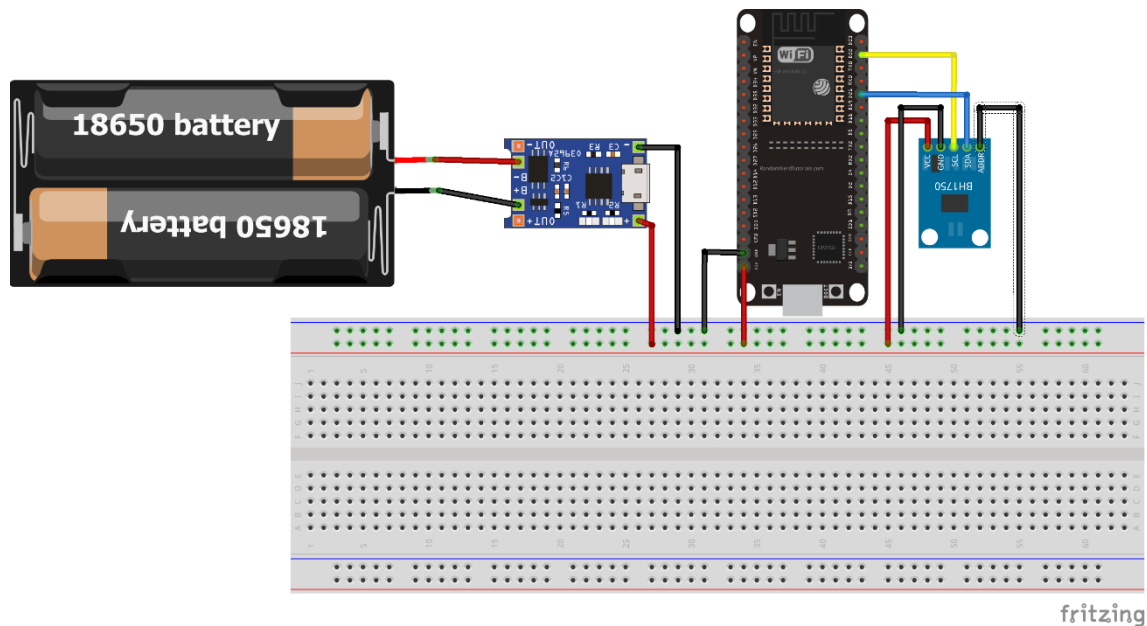


Fig. 57. Circuito de censado. Fuente: Elaborado por el autor

### Calculo autonomía de batería

Es necesario calcular la autonomía del módulo de censado con la batería para lo cual se utilizará la siguiente formula:

$$H = \frac{Wb}{Wc}$$

En donde H es la autonomía y Wb es la potencia de la batería y Wc es la potencia consumida por el sistema.

Además, se obtiene Wb por medio de la siguiente formula:

$$Wb = Vb * Ib$$

En donde Vb es el voltaje de la batería y Ib es la intensidad o amperaje por hora que tiene la batería.

Y para encontrar Wc se utiliza la siguiente formula:

$$Wc = Vb * Ic$$

En donde Ic es el consumo del sistema en amperios hora.

Del sistema que se elabora se utilizara una batería 18650 que tiene un Ib de 2200mAh y el módulo de censado sus componentes que son: el esp8266 consume 150mAh y el sensor BH1750 consume 190uAH por lo cual aplicando la formula se obtiene:

$$Wb = 4,2 * 2200mAh$$

$$Wb = 92400mW$$

$$Wc = 4,2 * 151mAh$$

$$Wc = 634,2mW$$

$$H = \frac{92400mW}{634,2mW}$$

$$H = 145,7$$

Lo que nos indica que el sistema tendrá una autonomía por medio de baterías antes de necesitar carga de 145,7 horas lo que equivale aproximadamente a 6 días.

### Funcionamiento

Como se puede observar en la Figura 58, se tiene el diagrama de flujo que indica que el módulo espera por medio de MQTT recibir una señal para de esta forma censar el entorno y enviar por medio de MQTT la señal censada al módulo de control.

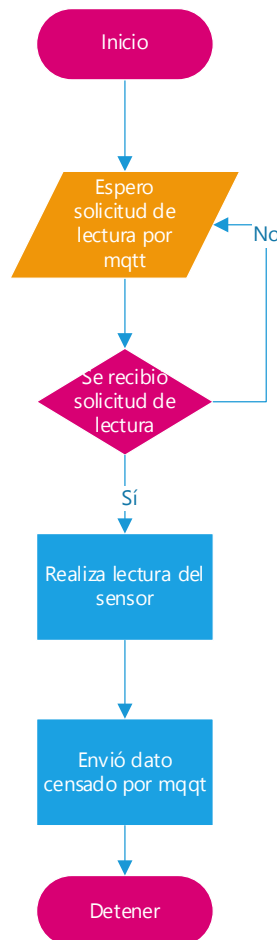


Fig. 58. Diagrama de Flujo módulo de censado. Fuente: Elaborado por el autor

## Control persiana motorizada

Dentro del sistema de censado se procedió a realizar el control de la persiana para de esta manera utilizar un interruptor y a su vez el control remoto por medio de interfaz web.

### Elementos que conforman el sistema

Para cuestiones prácticas se procedió a comprar una persiana que cubra la ventana del laboratorio, un servo motor de engrane metálico, un esp32 y un puente H que permita el cambio de sentido del giro del motor, el diagrama de funcionamiento se puede observar en la figura 59.

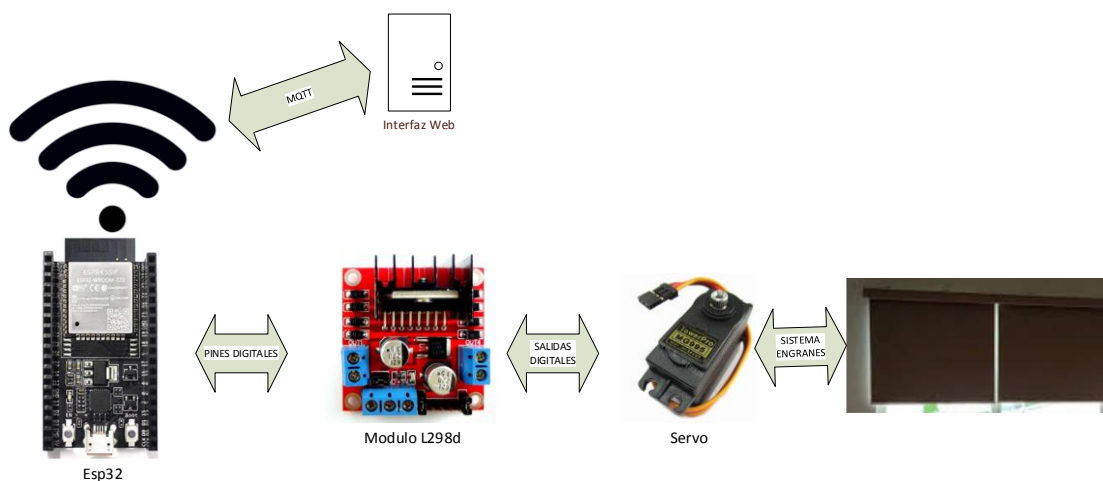


Fig. 59. Diagrama de bloques persiana motorizada. Fuente: Elaborado por el autor

Como se observa en la Figura 42 se tiene que el módulo Esp32 se conecta por medio de WiFi y MQTT con el interfaz web para proceder a realizar las tareas de movimiento de la persiana, a su vez necesita un controlador para el motor que permita controlar el motor protegiendo al microcontrolador de la carga que genera el motor al entrar en funcionamiento, se realizara un diseño de un acople que encaje en la persiana para realizar el movimiento de subida y bajada.

### Funcionamiento

En la Figura 60 se tiene el diagrama de flujo que indica los procesos que realiza el control de persiana, que empieza por conectarse a la red local donde se encuentra el servidor, después se conecta al servidor MQTT donde por medio de caracteres se controla el movimiento hacia arriba o abajo, a su vez también permite que se realice el control por medio de un interruptor para subir o bajar la persiana.

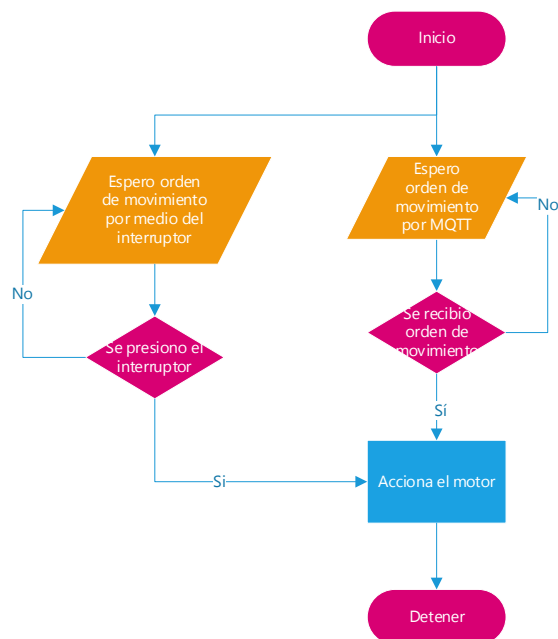


Fig. 60. Diagrama de flujo motor persiana. Fuente: Elaborado por el autor

### 3.5. Implementación

#### 3.5.1. Módulo de Control y distribución de potencia

El área de trabajo se adaptó para la elaboración del proyecto procediendo al cambio de las luminarias originales ya que por características técnicas descritas con anterioridad no se podían dimmerizar, se usaron luminarias de 18w tipo led que permiten el control de potencia por regulación de onda. El tipo de luminarias y como se encuentran instaladas se puede observar en la Figura 61.



Fig. 61. Laboratorio Electricidad. Fuente: Elaborado por el autor

El cambio de las luminarias se observa en la Figura 62.



Fig. 62. Cambio de luminarias. Fuente: Elaborado por el autor

Se adaptó el cableado para que todas las fases estén conectadas de forma común y el neutro se conecte a cada una de las borneras del circuito de control de potencia, como se muestra en la Figura 63.

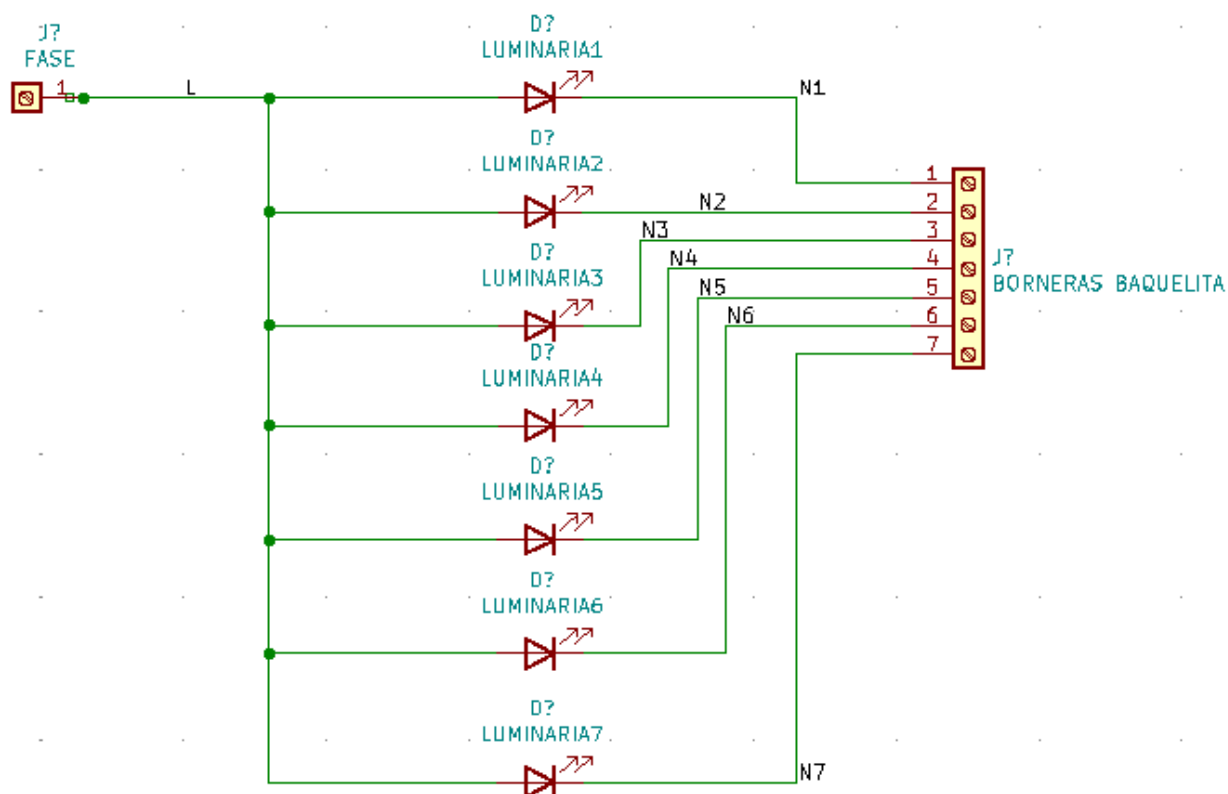


Fig. 63. Conexión luminarias. Fuente: Elaborado por el autor

Esta conexión se observa se realizó en los laboratorios como se observa en la Figura 64, cada luminaria tiene su conexión hacia el sistema y a la entrada de 120 v.

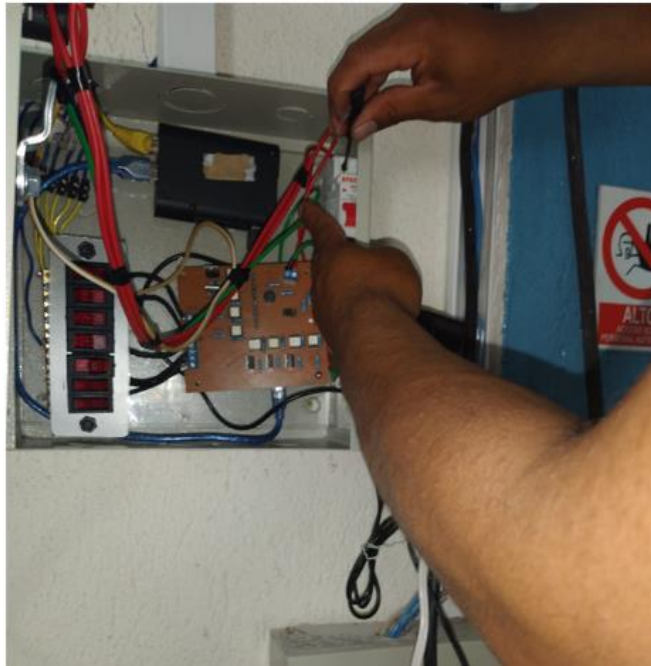


Fig. 64. Conexión eléctrica en los laboratorios. Fuente: Elaborado por el autor

Una vez terminada la conexión se procede a dejar todo empotrado en la pared teniendo como resultado lo que se observa en la Figura 65.



Fig. 65. Conexión eléctrica en los laboratorios. Fuente: Elaborado por el autor

Como se puede observar en la caja se encuentran tanto el Raspberry o módulo de control como el Arduino uno que realiza la parte de distribución de potencia.

### **3.5.2. Módulo de Censado**

Para la ubicación del módulo de censado se procedió a ubicar tanto el motor en la persiana como se muestra en la Figura 66.



Fig. 66. Instalación de servo motor en persiana. Fuente: Elaborado por el autor

Una vez realizada la instalación del motor se requiere instalar la caja de control que lo integran el esp32 el puente H para el motor y los botones para subir y bajar de forma manual, así como remota la persiana, esto se puede observar en la Figura 67.



Fig. 67. Instalación de servo motor en persiana. Fuente: Elaborado por el autor

Los sensores para la iluminación se pueden observar en la Figura 68 y se ubican bajo cada una de las luminarias.



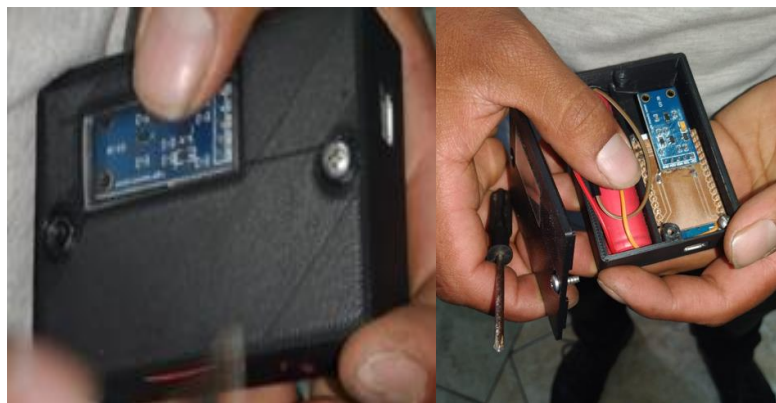



Fig. 68. Sensores de lúmenes. Fuente: Elaborado por el autor

### 3.6. Pruebas de funcionamiento

Una vez implementado el módulo se procede a realizar las siguientes pruebas de funcionamiento:

#### 3.6.1. Prueba #1:

		<b>UNIVERSIDAD TÉCNICA DEL NORTE</b> <b>FACULTAD DE INGENIERÍA</b>
CARRERA	PRÁCTICA Nº	NOMBRE DE LA PRÁCTICA
ELECTRICIDAD	1	Control intensidad lumínica (Modulo distribución de potencia)

#### PROCEDIMIENTO

1) Se procede a conectar el módulo de distribución de potencia al módulo de control como se muestra en la Figura 69, utilizando el cable USB

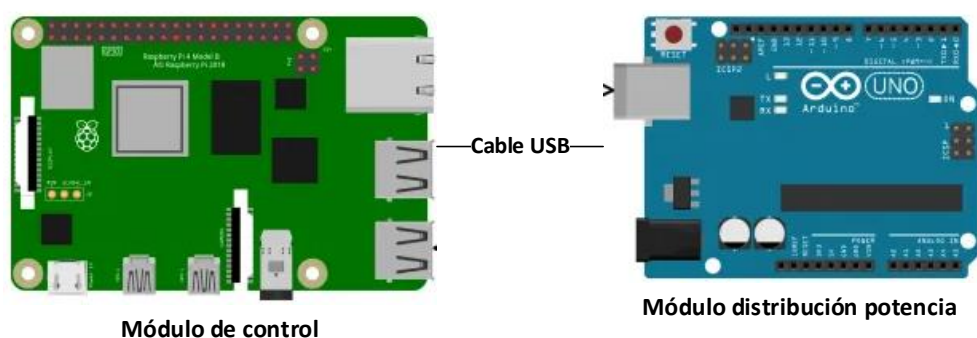





Fig. 69. Conexión Modulo control y Modulo de distribución de potencia. Fuente: Elaborado por el autor

2) Una vez realizada se procede a enviar los datos para el control de las luminarias por medio del puerto serial a una velocidad de 9600 baudios.

3) Se procede a verificar que las luminarias varíen su intensidad en base a los parámetros enviados en el puerto serial.

### 3.6.2. Prueba #2:

		<b>UNIVERSIDAD TÉCNICA DEL NORTE</b> <b>FACULTAD DE INGENIERÍA</b>
CARRERA	PRÁCTICA Nº	NOMBRE DE LA PRÁCTICA
ELECTRICIDAD	2	Censado de intensidad lumínica (módulo de censado)

### PROCEDIMIENTO

1) Se procede a conectar el módulo de censado como se muestra en la Figura 70, utilizando comunicación MQTT

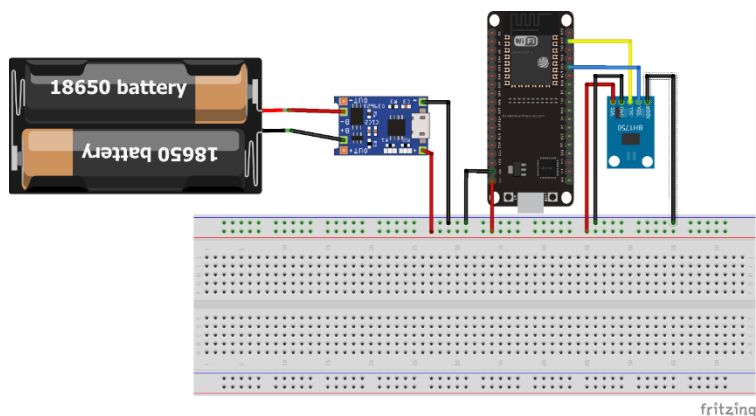



Fig. 70. Conexión Modulo de censado

2) Se procede a cargar el código en el esp32.

3) Por medio del puerto serial se realiza la lectura de los lúmenes detectados en el área seleccionada

### 3.6.3. Prueba #3:

		<b>UNIVERSIDAD TÉCNICA DEL NORTE</b> <b>FACULTAD DE INGENIERÍA</b>
CARRERA	PRÁCTICA N°	NOMBRE DE LA PRÁCTICA
ELECTRICIDAD	3	Envío de datos desde el módulo de sensado hacia el módulo de control por MQTT

#### PROCEDIMIENTO

1) Se procede a conectar el módulo de sensado al módulo de control como se muestra en la Figura 71, utilizando comunicación MQTT

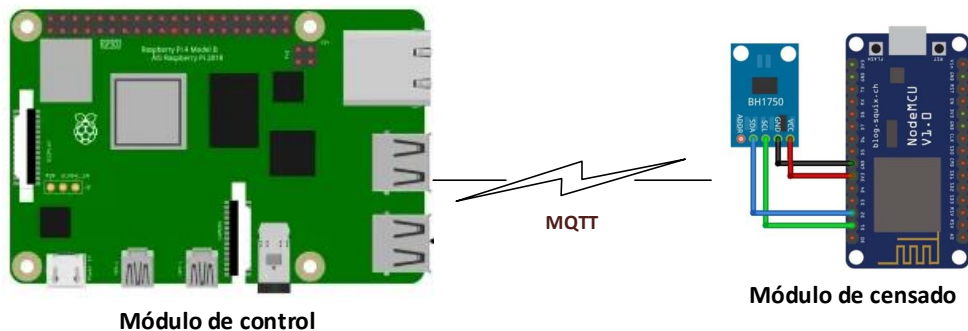


Fig. 71. Conexión Modulo control y Modulo de distribución de sensado

- 2) Se procede a configurar la red inalámbrica.
- 3) Se realiza la conexión del módulo de sensado y módulo de control hacia la red inalámbrica.
- 4) Se procede a configurar los topicos de envío y recepción en el módulo de sensado.
- 5) Se verifica que el módulo central envíe una solicitud a los módulos de sensado y reciba la respuesta con la cantidad de luxes en el ambiente seleccionado para la implementación.

### 3.6.4. Prueba #4:

		<b>UNIVERSIDAD TÉCNICA DEL NORTE</b>  <b>FACULTAD DE INGENIERÍA</b>
CARRERA	PRÁCTICA Nº	NOMBRE DE LA PRÁCTICA
ELECTRICIDAD	4	Control persiana motorizada por medio del módulo de control y MQTT

#### PROCEDIMIENTO

1) Se procede a conectar el control de persiana al módulo de control como se muestra en la Figura 72, utilizando comunicación MQTT

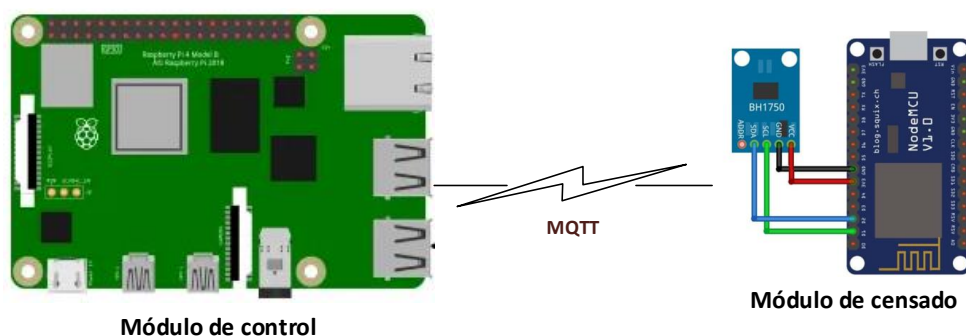



Fig. 72. Conexión Modulo control y control de persiana

- 2) Se procede a conectar el esp32 que controla la persiana a la red inalámbrica
- 3) Se procede a configurar los topics de envío y recepción en MQTT en el esp32 que controla la persiana
- 5) Se verifica que por medio de MQTT se reciba los datos para elevar o bajar la persiana.

### 3.6.5. Prueba #5:

		<b>UNIVERSIDAD TÉCNICA DEL NORTE</b>  <b>FACULTAD DE INGENIERÍA</b>
CARRERA	PRÁCTICA Nº	NOMBRE DE LA PRÁCTICA
ELECTRICIDAD	5	Conexión de todos los módulos por medio de MQTT y comunicación serial.

#### PROCEDIMIENTO

1) Se procede a conectar todos los módulos como se muestra en la Figura 73.

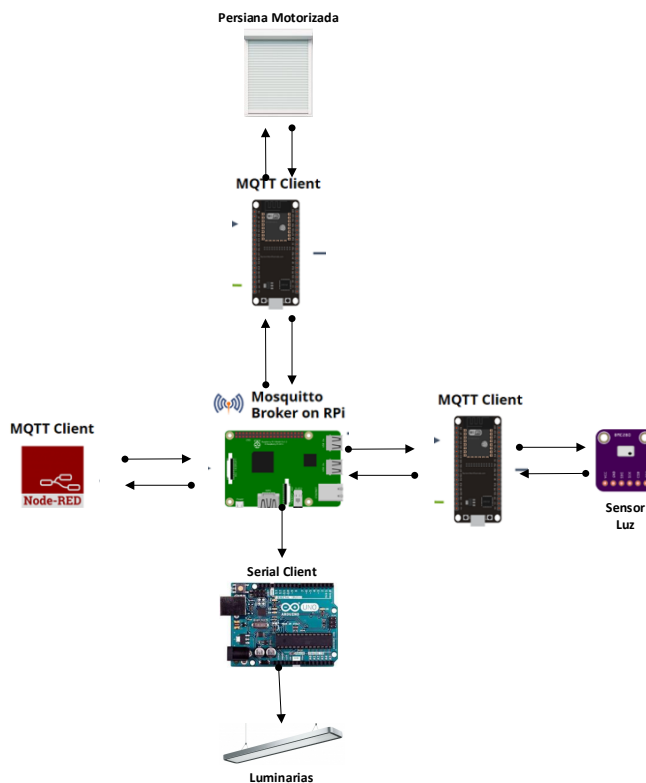


Fig. 73. Conexión Módulos

2) Se procede a realizar las pruebas de recepción y envío de mensajes MQTT

3) Es necesario verificar que todos los datos de los módulos externos al módulo de control envíen datos por MQTT a un tópico.

5) Se realiza una prueba de funcionamiento de sensores para mantener un nivel estable de iluminación en el laboratorio.

Las pruebas de funcionamiento son ideales para alcanzar los objetivos planteados en el proyecto, además es necesario comprobar el desempeño del hardware y el software en tiempo real.

Cada prueba planteada inicia desde la prueba de cada uno de los módulos de forma independiente y termina con las pruebas que integran todo el sistema demostrando la funcionalidad completa de cada proceso y componente.

## Capítulo III: Resultados

El presente capítulo es la conclusión de cada una de las pruebas de funcionamiento en base a la implementación del proyecto, las pruebas de funcionamiento electrónico tanto en hardware como software, así como la depuración de los errores para mejoras continuas del sistema.

### 4.1. Resultados

#### 4.1.1. Prueba #1:

Al momento de realizar la elaboración de las pruebas en base al capítulo anterior se obtuvo los resultados deseados en donde las luminarias se regularon en variación del 30%, 50%, 100%. Demostrando que el circuito regulador construido basado en un circuito de Corte por medio de TRIACs, la variación se puede observar en la Figura



Fig. 74. Intensidad luminaria. Fuente: Elaborado por el autor

Es importante recalcar que la variación se elabora por medio de comandos seriales y que interactúa directamente con el Arduino Uno y la placa de distribución de potencia. Se obtuvo resultados fluctuantes que sirvieron para la calibración y mejora del código.

#### 4.1.2. Prueba #2:

Al elaborar las pruebas se pudo observar que cada uno de los módulos de censado funcionan de forma equivalente en la misma posición, se procedió a usar los distintos módulos de censado en cada uno de los lugares con las mismas condiciones de iluminación y se obtuvo los resultados que se encuentran en la Tabla 20.

TABLA 20  
RESULTADOS MÓDULOS DE CENSADO

LECTURA DE SENSORES	
# LUMINARIAS	VALOR OBTENIDO LÚMENES
1	200
2	198
3	201
4	203
5	205
6	197
7	203

Fuente: Elaborado por el autor

Como se observa en la Tabla 20, se tiene que cada uno de los sensores manejan valores aproximados iguales al tomar las medidas en las mismas condiciones lo que permite comprender que manejan una calibración equivalente.

La forma que se ubicaron los sensores se puede observar en la Figura 75.



Fig. 75. Lectura de sensores. Fuente: Elaborado por el autor

#### 4.1.3. Prueba #3:

Cada uno de los módulos de censado enviaron los datos del censado en el momento que el módulo de control envía la solicitud de manera independiente, también es importante destacar que cada uno de módulos de censado manejan un tópico para envío y recepción de datos que permite independizar las variables recibidas.

Cada uno de los módulos censados se encuentran ubicados en sintonía con cada uno de las luminarias para poder censar los valores requeridos de los laboratorios.

Los valores obtenidos se pueden visualizar en tiempo real en el interfaz de usuario por medio de node red como se muestra en la Figura 76.

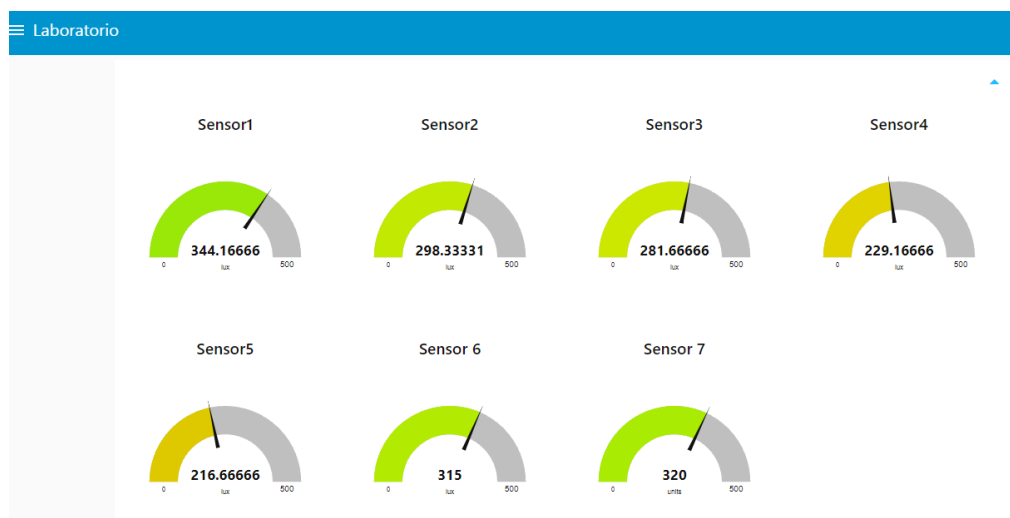


Fig. 76. Lectura de sensores. Fuente: Elaborado por el autor

#### 4.1.4. Prueba #4:

Para las pruebas de funcionamiento del motor se realizó el uso del interfaz web que se observa en la Figura 77 y también los pulsadores instalados en el laboratorio.

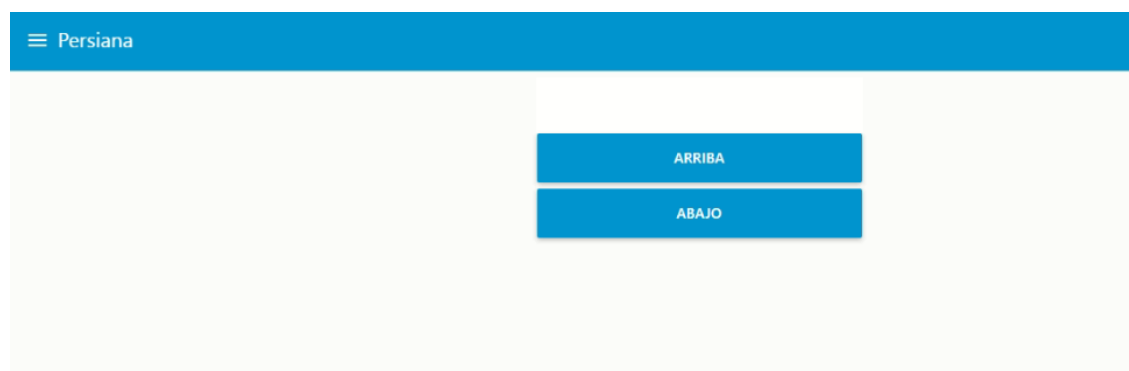


Fig. 77. Botones control de motor. Fuente: Elaborado por el autor

Se pudo observar que en la practica el control de movimiento de la persiana funciona de manera correcta con una velocidad consistente con la capacidad del motor, el tópicos para envío y recepción se maneja de manera independiente.

Las condiciones necesarias para elaborar una conexión con la red inalámbrica en donde se encuentran el resto de los componentes, para manejo de motores todas las partes del circuito y el microcontrolador funcionan de manera correcta.

#### 4.1.5. Prueba #5:

El resultado obtenido en las pruebas se tiene lo siguiente obtenido:

- Conexión de todos los módulos a la red inalámbrica
- Conexión de todos los módulos al servidor MQTT
- Recepción y envío de datos de cada uno de los módulos al servidor MQTT
- Recepción serial por parte del módulo de distribución de potencia



Cada uno de los módulos realizo sus funciones de forma correcta y tuvo resultados positivos en donde la comunicación inalámbrica funciona de forma eficaz.

El resultado de todos los componentes integrados se puede observar por medio de Grafana como se observa en la Figura 78.

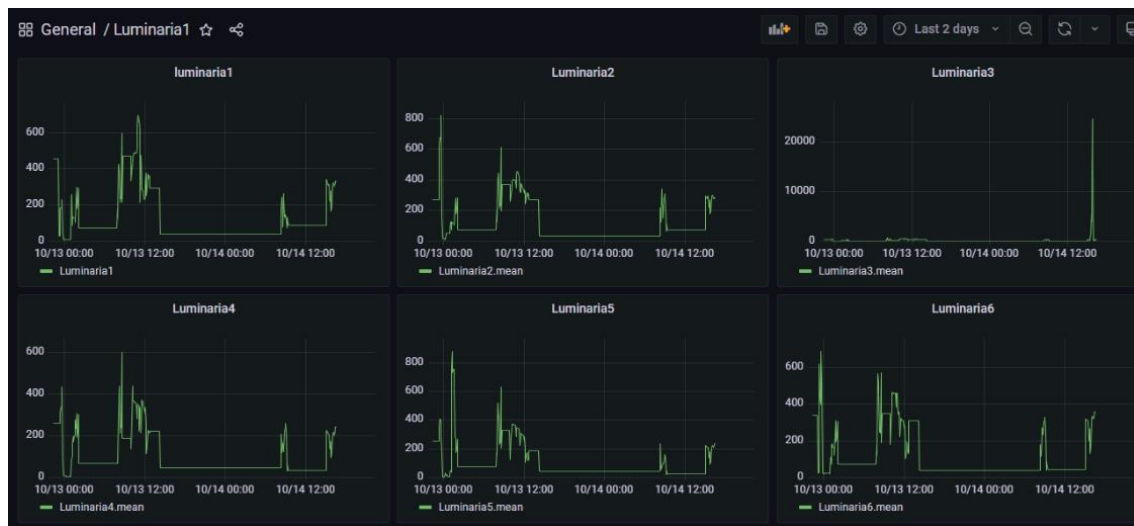


Fig. 78. Botones control de motor. Fuente: Elaborado por el autor

De todo el proceso realizado se puede concluir que cada uno de los módulos que integran el sistema funcionan de manera correcta, permitiendo elaborar un sistema IoT, y cumpliendo el objetivo planteado en el presente proyecto, en la Tabla 21 se observa los resultados de cada una de las prácticas en base a si tuvieron éxito en su ejecución.

TABLA 21  
RESULTADOS PRACTICAS

RESULTADO PRACTICAS	
# PRACTICAS	FUNCIONAMIENTO
1	Correcto
2	Correcto
3	Correcto
4	Correcto
5	Correcto
6	Correcto
7	Correcto

## **Conclusiones**

- En conclusión, toda la teoría referente a IoT y los libros de datos de los componentes electrónicos nos permiten entender el comportamiento del sistema que se busca implementar.
- Se concluye que el diseño tiene varios procesos en donde se selecciona los mejores componentes de acuerdo a los requerimientos, el sitio donde se piensa implementar y el tipo de usuario para el que se requiere el sistema.
- Se puede concluir que al construir el prototipo se mejoran las condiciones de luminancia dentro del laboratorio, mejorando las condiciones de trabajo para estudiantes y docentes.

## **Recomendaciones**

- Además de cada proceso teórico es necesario experimentar los procesos de forma práctica, porque la mayoría de textos realizan las pruebas de componentes en escenarios ideales donde no existen factores externos que afecten la funcionalidad de los sistemas.
- Para realizar un exitoso diseño de un prototipo es necesario empezar desde las necesidades y requerimientos del usuario final y el ámbito en que se busca implementar.
- Al momento de implementar un prototipo es recomendable tomar en consideración la situación inicial del entorno y los cambios que se piensan realizar para afectar en lo más mínimo el área de trabajo y tratando de reutilizar componentes

## Bibliografía

- Andres, M. B. (2018). *Internet de las cosas*. Madrid: Editorial REUS. Obtenido de [www.editorialreus.es](http://www.editorialreus.es).
- Andres, M. B. (2018). Internet de las cosas. En M. B. Andres, *Internet de las cosas* (pág. 126). Madrid: REUS.
- Castro Guaman, M. P., & Posligua Murillo, N. C. (2015). *Diseño de iluminación con luminarios tipo led basado en el concepto eficiencia energética y confort visual, implementacion de estructura para pruebas*. Marzo: Universidad Politecnica Salesiana Sede Guayaquil.
- Crespo, E. (19 de Junio de 2017). *Aprendiendo arduino*. Obtenido de Aprendiendo arduino: <https://aprendiendoarduino.wordpress.com/2017/06/19/arduino-vs-raspberry-pi-3/>
- Damián, J. (4 de Marzo de 2020). *electrogeekshop*. Obtenido de electrogeekshop: <https://www.electrogeekshop.com/arduino-nano-pinout-y-caracteristicas/>
- Fiallos, P. d. (2015). *SISTEMA AUTOMÁTICO DE CONTROL DE ILUMINACIÓN EN LAS SALAS DE PROFESORES DE LA FISEI*. Agosto: UNIVERSIDAD TÉCNICA DE AMBATO.
- Fractal. (9 de Octubre de 2018). *www.fractal.com*. Obtenido de [www.fractal.com](http://www.fractal.com): <https://www.fractal.com/es/blog/2018/10/10/9-aplicaciones-importantes-iot>
- Llamas, L. (30 de Abril de 2021). *www.luisllamas.es*. Obtenido de [www.luisllamas.es](http://www.luisllamas.es): <https://www.luisllamas.es/modelos-de-raspberry-pi/>
- Muelas, C. G. (2018). *Integración de Redes Telemáticas IoT con Raspberry pi*. Cataluña: Universidad de Cataluña.
- Naylamp Mechatronics. (5 de Enero de 2021). *Naylamp Mechatronics*. Obtenido de Naylamp Mechatronics: <https://naylampmechatronics.com/expressif-esp/384-nodemcu-32-esp32-wifi.html>
- prometec. (8 de Febrero de 2021). *prometec*. Obtenido de prometec: <https://prometec.mx/producto/ldr-sensor-de-luz/>
- Shaw, P. (19 de Marzo de 2018). *Digital Lab*. Obtenido de Digital Lab: <http://digitallab.org/2016/03/raspberry-pi-2-pinouts/#.YlxQUbUzblU>
- Teja, R. (15 de Enero de 2021). *Electronics Hub*. Obtenido de Electronics Hub: <https://www.electronicshub.org/arduino-uno-pinout/>
- UIT-T. (2012). *Descripción general de Internet de los objetos*. Union internacional de telecomunicaciones.
- WONKSKNOW LLC. (12 de Enero de 2019). *YoungWonks*. Obtenido de YoungWonks: <https://www.youngwonks.com/blog/Raspberry-Pi-4-Pinout>

## Anexos

### Anexo 1. Código control de cruce por cero Arduino IDE.

```
const byte PinINT_0 = 2; // Se designa como PinINT_0 al pin 2 de Arduino

//Configuración Inicial Arduino
void setup()
{
  Serial.begin(9600); //Se habilita la comunicación RS232 o Serial
  attachInterrupt(0, INTO_Cambio, FALLING); // Se habilita la interrupción 0 en el pin 2 en
                                             // detección Falling o bajada
}

//ciclo de trabajo
void loop()
{
}

//Interrupción
void INTO_Cambio()
{
  Serial.println(1); //Cada que se detecte una interrupción en 0 se imprime 1 por puerto Serial
}
```

## Anexo 2. Código control de fase dimmer Arduino UNO

```
const byte PinINT_0 = 2;
const byte ledPin_0 = 3;

int ValAnalogo = 0;
int ValorTriac = 4000;
int estado=0;
void setup()
{
  pinMode(ledPin_0, OUTPUT);
  Serial.begin(9600);

  attachInterrupt(0, INT0_Cambio, FALLING);
}

void loop()
{
  if(Serial.available(>0)
  {
    estado=Serial.read();

    if(estado=='e') ValorTriac = 6500;
    if(estado=='v') ValorTriac = 5200;
    if(estado=='w') ValorTriac = 5150;
    if(estado=='x') ValorTriac = 4850;
    if(estado=='y') ValorTriac = 4550;
    if(estado=='z') ValorTriac = 3550;
    if(estado=='s') ValorTriac = 2550;
    if(estado=='t') ValorTriac = 2050;
    if(estado=='u') ValorTriac = 1880;

  }

void INT0_Cambio()
{
  delayMicroseconds(ValorTriac);
  digitalWrite(ledPin_0, HIGH);
  delayMicroseconds(100);
  digitalWrite(ledPin_0, LOW);
}
```

### Anexo 3. Código lectura sensor BH1750 con Esp32

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <EEPROM.h>
#include <Wire.h>
#include <BH1750.h>

// Update these with values suitable for your network.

const char* ssid = "Enlace 2";
const char* password = "345.sdf,1";
const char* mqtt_server = "192.168.2.105";
const char* topicEstado = "/casa/comedor/temperatura";
const char* topicDistancia = "/laboratorio/luminaria";
const char* mqttUser = "luminarias";
const char* mqttPassword = "UTN2021";
BH1750 lightMeter(0x23);
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE      (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;
const int EchoPin = 4;
const int TriggerPin = 2;
const int motor=15;
int medida;
int minimo=12;
void setup_wifi() {

  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

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

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
```

```

Serial.print(topic);
Serial.print("] ");
for (int i = 0; i < length; i++) {
  Serial.print((char)payload[i]);
}
Serial.println();

// Switch on the LED if an 1 was received as first character
if ((char)payload[0] == '1') {
  //digitalWrite(BUILTIN_LED, LOW); // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  lumenes();
} else {
  digitalWrite(BUILTIN_LED, HIGH); // Turn the LED off by making the voltage HIGH
}

}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect("ESP8266Client", mqttUser, mqttPassword )) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish(topicEstado, "hello world");
      // ... and resubscribe
      client.subscribe(topicDistancia);
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

void setup() {
  pinMode(BUILTIN_LED, OUTPUT); // Initialize the BUILTIN_LED pin as an output
  Serial.begin(9600);
  Wire.begin();
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  pinMode(TriggerPin, OUTPUT);
  pinMode(EchoPin, INPUT);
  EEPROM.get(0, medida);
  if (lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE)) {
    Serial.println(F("BH1750 Advanced begin"));
  }
}

```



```

} else {
  Serial.println(F("Error initialising BH1750"));
}
//pinMode(motor, OUTPUT);
}

void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

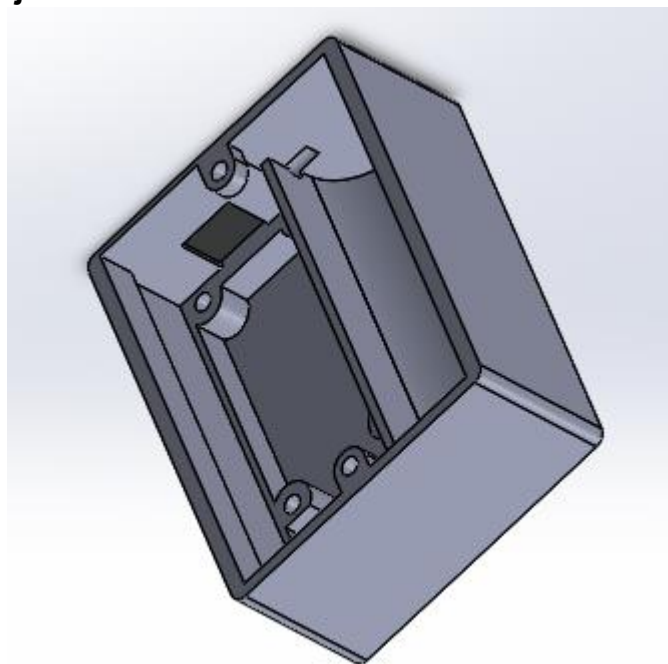
  unsigned long now = millis();
  if (now - lastMsg > 2000) {
    lastMsg = now;
  }

  delay(1000);
}

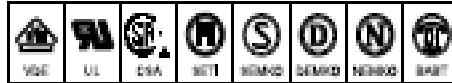
void lumenes (){
  if (lightMeter.measurementReady()) {
    float lux = lightMeter.readLightLevel();
    Serial.print("Light: ");
    Serial.print(lux);
    Serial.println(" lx");
    char Lumenes[5];
    dtostrf(lux, 1, 5, Lumenes);
    client.publish(topicEstado, Lumenes);
  }
}

```

#### Anexo 4. Diseño caja sensor en 3d







## 6-Pin DIP Random-Phase Optoisolators Triac Driver Output (400 Volts Peak)

The MOC3020 Series consists of gallium arsenide infrared emitting diodes, optically coupled to a silicon bilateral switch.

- To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option. They are designed for applications requiring isolated triac triggering.

Recommended for 115/240 Vac(rms) Applications:

- Solenoid/Valve Controls
- Lamp Ballasts
- Interfacing Microprocessors to 115 Vac Peripherals
- Motor Controls
- Static ac Power Switch
- Solid State Relays
- Incandescent Lamp Dimmers

### MAXIMUM RATINGS (T<sub>A</sub> = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
<b>INFRARED EMITTING DIODE</b>			
Reverse Voltage	V <sub>R</sub>	3	Volts
Forward Current — Continuous	I <sub>F</sub>	50	mA
Total Power Dissipation @ T <sub>A</sub> = 25°C Negligible Power in Triac Driver Derate above 25°C	P <sub>D</sub>	100 1.33	mW mW/°C

### OUTPUT DRIVER

Off-State Output Terminal Voltage	V <sub>ORM</sub>	400	Volts
Peak Repetitive Surge Current (PW = 1 ms, 120 pps)	I <sub>TSM</sub>	1	A
Total Power Dissipation @ T <sub>A</sub> = 25°C Derate above 25°C	P <sub>D</sub>	300 4	mW mW/°C

### TOTAL DEVICE

Isolation Surge Voltage <sup>(1)</sup> (Peak ac Voltage, 60 Hz, 1 Second Duration)	V <sub>ISO</sub>	7500	Vac(pk)
Total Power Dissipation @ T <sub>A</sub> = 25°C Derate above 25°C	P <sub>D</sub>	330 4.4	mW mW/°C
Junction Temperature Range	T <sub>J</sub>	-40 to +100	°C
Ambient Operating Temperature Range	T <sub>A</sub>	-40 to +85	°C
Storage Temperature Range	T <sub>stg</sub>	-40 to +150	°C
Soldering Temperature (10 s)	T <sub>L</sub>	260	°C

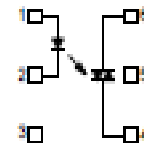
1. Isolation surge voltage, V<sub>ISO</sub>, is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

**MOC3021**  
**MOC3022**  
**MOC3023**



STANDARD THRU HOLE

### SCHEMATIC



- ANODE
- CATHODE
- NC
- MAIN TERMINAL
- SUBSTRATE  
DO NOT CONNECT
- MAIN TERMINAL

## Anexo 6. Hoja de especificaciones TRIAC BT136-600D

### 9. Characteristics

Table 6. Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Static characteristics</b>						
$I_{GT}$	gate trigger current	$V_D = 12\text{ V}; I_T = 0.1\text{ A}; T_2+ G+;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 7</a>	-	2	5	mA
		$V_D = 12\text{ V}; I_T = 0.1\text{ A}; T_2+ G-;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 7</a>	-	2.5	5	mA
		$V_D = 12\text{ V}; I_T = 0.1\text{ A}; T_2- G-;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 7</a>	-	2.5	5	mA
		$V_D = 12\text{ V}; I_T = 0.1\text{ A}; T_2- G+;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 7</a>	-	5	10	mA
$I_L$	latching current	$V_D = 12\text{ V}; I_G = 0.1\text{ A}; T_2+ G+;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 8</a>	-	1.6	10	mA
		$V_D = 12\text{ V}; I_G = 0.1\text{ A}; T_2+ G-;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 8</a>	-	4.5	15	mA
		$V_D = 12\text{ V}; I_G = 0.1\text{ A}; T_2- G-;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 8</a>	-	1.2	10	mA
		$V_D = 12\text{ V}; I_G = 0.1\text{ A}; T_2- G+;$ $T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 8</a>	-	2.2	15	mA
$I_H$	holding current	$V_D = 12\text{ V}; T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 9</a>	-	1.2	10	mA
$V_T$	on-state voltage	$I_T = 5\text{ A}; T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 10</a>	-	1.4	1.7	V
$V_{GT}$	gate trigger voltage	$V_D = 12\text{ V}; I_T = 0.1\text{ A}; T_J = 25\text{ }^\circ\text{C};$ <a href="#">Fig. 11</a>	-	0.7	1	V
		$V_D = 400\text{ V}; I_T = 0.1\text{ A}; T_J = 125\text{ }^\circ\text{C};$ <a href="#">Fig. 11</a>	0.25	0.4	-	V
$I_O$	off-state current	$V_D = 600\text{ V}; T_J = 125\text{ }^\circ\text{C}$	-	0.1	0.5	mA
<b>Dynamic characteristics</b>						
$dV_D/dt$	rate of rise of off-state voltage	$V_{DM} = 402\text{ V}; T_J = 125\text{ }^\circ\text{C}; R_{GT1} = 1\text{ k}\Omega;$ ( $V_{DM} = 67\%$ of $V_{DRM}$ ); exponential waveform	-	5	-	V/ $\mu\text{s}$
$t_{gt}$	gate-controlled turn-on time	$I_{TM} = 6\text{ A}; V_D = 600\text{ V}; I_G = 0.1\text{ A}; dI_G/dt = 5\text{ A}/\mu\text{s}$	-	2	-	$\mu\text{s}$

## Anexo 7. Hoja de especificaciones PC817c

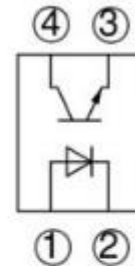
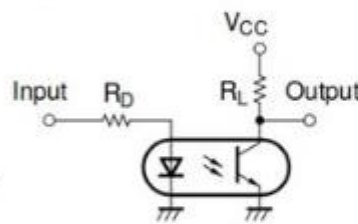
# PC817

## High Density Mounting Type Photocoupler

- Current transfer ratio:  
(CTR: MIN. 50% at  $I_F = 5\text{mA}, V_{CE} = 5\text{V}$ )
- High isolation voltage between input and output:  
 $V_{\text{isolate}} = 5000\text{V}$
- Compact dual-in-line package

### Applications:

- System appliances
- Measuring instruments
- Electric home appliances
- Computer terminals



- ① Anode
- ② Cathode
- ③ Emitter
- ④ Collector

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input	Forward voltage	$V_F$	$I_F = 20\text{mA}$	-	1.2	1.4	V	
	Peak forward voltage	$V_{FM}$	$I_{FM} = 0.5\text{A}$	-	-	3.0	V	
	Reverse current	$I_R$	$V_R = 4\text{V}$	-	-	10	$\mu\text{A}$	
	Terminal capacitance	$C_t$	$V = 0, f = 1\text{kHz}$	-	30	250	pF	
Output	Collector dark current	$I_{CDO}$	$V_{CE} = 20\text{V}$	-	-	$10^{-7}$	A	
	<sup>14</sup> Current transfer ratio	CTR	$I_F = 5\text{mA}, V_{CE} = 5\text{V}$	50	-	600	%	
Transfer characteristics	Collector-emitter saturation voltage	$V_{CE(sat)}$	$I_F = 20\text{mA}, I_C = 1\text{mA}$	-	0.1	0.2	V	
	Isolation resistance	$R_{ISO}$	DC500V, 40 to 60% RH	$5 \times 10^{10}$	$10^{11}$	-	$\Omega$	
	Floating capacitance	$C_f$	$V = 0, f = 1\text{MHz}$	-	0.6	1.0	pF	
	Cut-off frequency	$f_c$	$V_{CE} = 5\text{V}, I_C = 2\text{mA}, R_L = 100\Omega, -3\text{dB}$	-	80	-	kHz	
	Response time	Rise time	$t_r$	$V_{CE} = 2\text{V}, I_C = 2\text{mA}, R_L = 100\Omega$	-	4	18	$\mu\text{s}$
		Fall time	$t_f$		-	3	18	$\mu\text{s}$

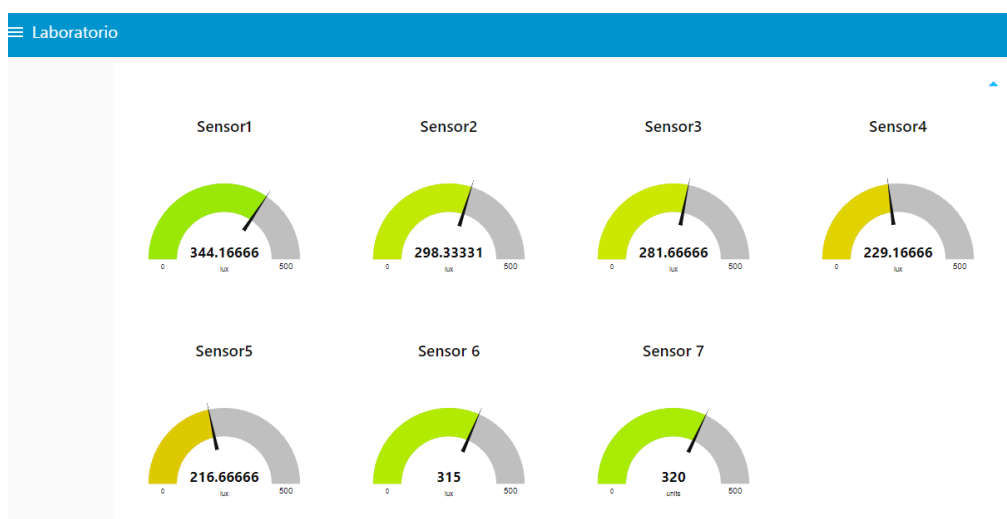
## Anexo 8. Manual de operación y mantenimiento de prototipo

En este apartado se detalla un manual de operación y mantenimiento con la finalidad de dar a conocer el procedimiento para poner en marcha el prototipo encargado de la supervisión y control de las luminarias.

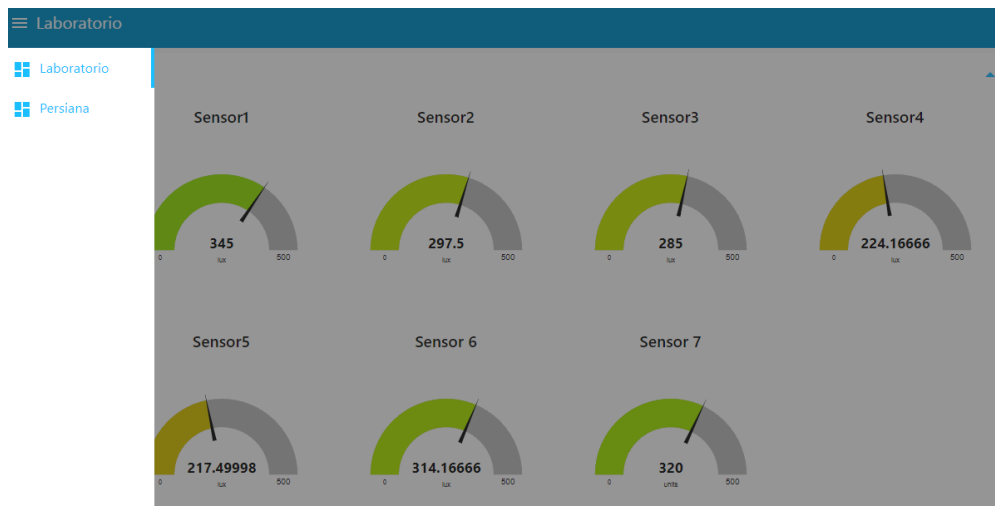
Para poder utilizar el prototipo de manera correcta es necesario seguir las siguientes instrucciones:

1. Conectar el enchufe de alimentación de manera que se respete la conexión Fase – Neutro – Tierra.
2. Encender con el botón ubicado en la caja de paso
3. Verificar luz piloto.
4. Conectarse desde cualquier dispositivo a la red inalámbrica: Luminarias contraseña: UTN2021 .
5. Ingresar mediante un navegador la siguiente dirección para visualizar la interfaz grafica.

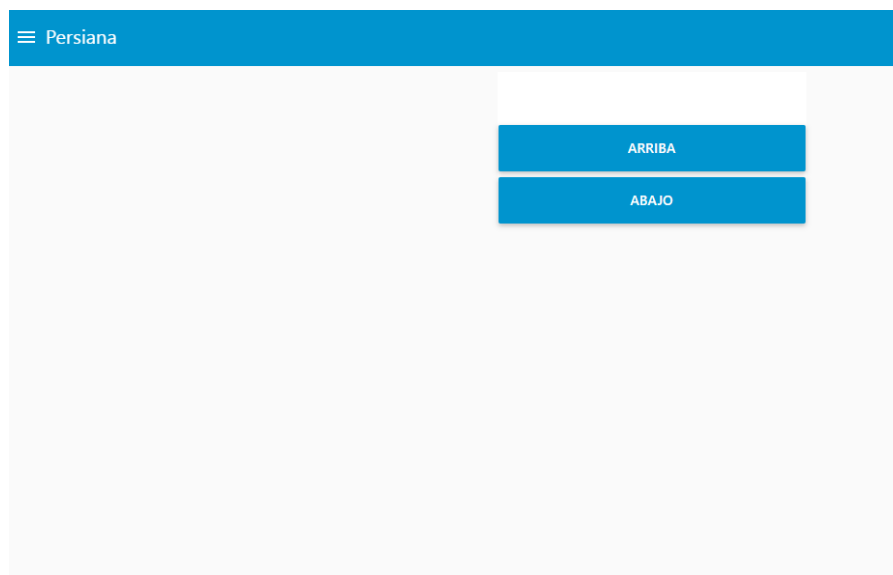
<http://192.168.0.4:1880/ui/#!/0?socketid=xNKDO5gfffU6iudxAAAO>



6. Para el control de persiana desplegar menú y seleccionar persiana.



Se observará los botones de subida y bajada.



## Anexo 9. Código Python pendiente.py

```
from publicar import envio
from lectura import Lectura
import time
from arduinoS import EnvioS
from timeit import default_timer
from promedio import promedio

promedio=promedio()
conexion=envio()
datos=Lectura()
envio=EnvioS()
#time.sleep(0)
setpoint=300
y1=4000
y2=6000
envio.envio_Serial(7000)
time.sleep(1.5)
cliente =conexion.solicitud('/laboratorio/luminaria/1','1')
time.sleep(3)
inicial=promedio.calculo()
print (inicial)
if inicial < setpoint:
    envio.envio_Serial(y2)
    time.sleep(1.5)
    cliente =conexion.solicitud('/laboratorio/luminaria/1','1')
    time.sleep(3)
    x2=promedio.calculo()
    envio.envio_Serial(y1)
    time.sleep(1.5)
    cliente =conexion.solicitud('/laboratorio/luminaria/1','1')
    time.sleep(3)
    x1=promedio.calculo()
    m=(y2-y1)/(x2-x1)
    print(m)
    y=m*(setpoint-x1)+y1
    print(y)
    envio.envio_Serial(y)
    time.sleep(1.5)
    cliente =conexion.solicitud('/laboratorio/luminaria/1','1')
    time.sleep(3)
    print(promedio.calculo())
    final=promedio.calculo()
    set1=setpoint
    while True:

        while final>setpoint:
            set1=set1-10
```



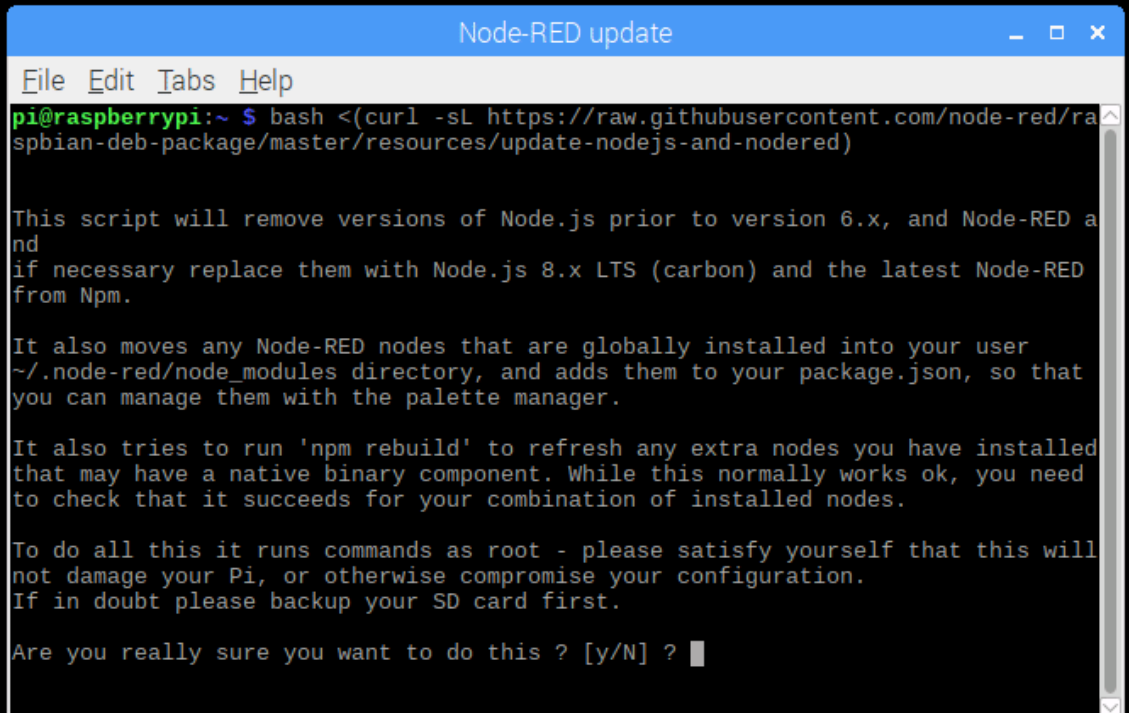
```
y=m*(set1-x1)+y1
print(y)
envio.envio_Serial(y)
time.sleep(1.5)
cliente =conexion.solicitud('/laboratorio/luminaria/1','1')
time.sleep(3)
final=promedio.calculo()
print(promedio.calculo())
time.sleep(10)
cliente =conexion.solicitud('/laboratorio/luminaria/1','1')
time.sleep(3)
final=promedio.calculo()
```

## Anexo 10. Instalación de Node-RED en Raspberry Pi

El sistema viene preinstalado y solo es necesario activarlo por lo cual se procede a introducir el siguiente comando en el terminal de Raspberry.

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

Una vez realizado este proceso en el terminal aparecerá una pregunta de si deseamos activar el componente y actualizarlo donde se debe seleccionar Y y enter.



```
Node-RED update
File Edit Tabs Help
pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)

This script will remove versions of Node.js prior to version 6.x, and Node-RED and
if necessary replace them with Node.js 8.x LTS (carbon) and the latest Node-RED
from Npm.

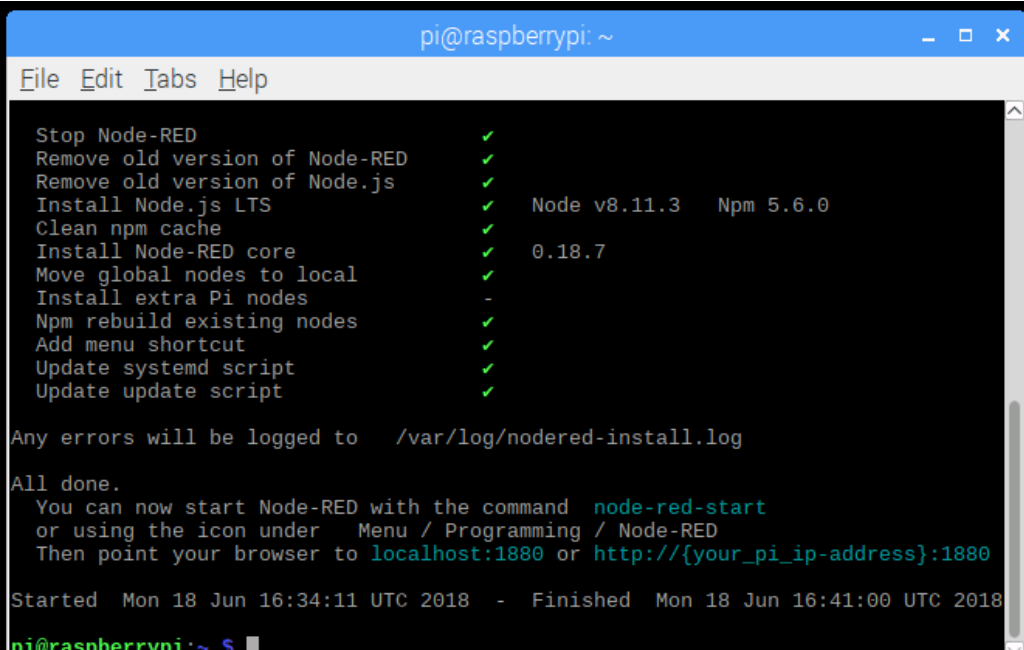
It also moves any Node-RED nodes that are globally installed into your user
~/.node-red/node_modules directory, and adds them to your package.json, so that
you can manage them with the palette manager.

It also tries to run 'npm rebuild' to refresh any extra nodes you have installed
that may have a native binary component. While this normally works ok, you need
to check that it succeeds for your combination of installed nodes.

To do all this it runs commands as root - please satisfy yourself that this will
not damage your Pi, or otherwise compromise your configuration.
If in doubt please backup your SD card first.

Are you really sure you want to do this ? [y/N] ? █
```

Ahora solo se debe esperar y nos aparecerá la ventana en donde se indica que todos los servicios se han activado.



```
pi@raspberrypi: ~
File Edit Tabs Help

Stop Node-RED ✓
Remove old version of Node-RED ✓
Remove old version of Node.js ✓
Install Node.js LTS ✓ Node v8.11.3 Npm 5.6.0
Clean npm cache ✓
Install Node-RED core ✓ 0.18.7
Move global nodes to local ✓
Install extra Pi nodes -
Npm rebuild existing nodes ✓
Add menu shortcut ✓
Update systemd script ✓
Update update script ✓

Any errors will be logged to /var/log/nodered-install.log

All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880

Started Mon 18 Jun 16:34:11 UTC 2018 - Finished Mon 18 Jun 16:41:00 UTC 2018
pi@raspberrypi:~ $ █
```

## Anexo 11. Instalación de Grafana

Es necesario introducir los siguientes comandos:

- `sudo apt-get install -y adduser libfontconfig1`
- `wget https://dl.grafana.com/oss/release/grafana_6.7.2_armhf.deb`
- `sudo dpkg -i grafana_6.7.2_armhf.deb`

Es necesario configurar el servicio con el siguiente mensaje:

- `sudo /bin/systemctl daemon-reload`

Una vez configurado el servicio se procede a habilitarlo:

- `sudo /bin/systemctl enable grafana-server`

Para iniciar el servicio en el arranque:

- `sudo /bin/systemctl start grafana-server`

Una vez realizado se ingresa a la dirección ip 192.168.0.4:3000 para obtener el siguiente resultado:



## Anexo 12. Código Python arduinoS.py

```
import serial
arduino=serial.Serial('/dev/ttyACM0',baudrate=9600, timeout = 1.0)

class EnvioS():
    def envio_Serial(self, dato):
        self.dato=dato
        arduino.write(str(dato)+"\n")
        print("Enviado")
```

## Anexo 13. Código Python lectura.py

```
class Lectura():

    def recuperar_dato(self,ruta):
        self.ruta=ruta
        archi1=open(self.ruta,"r")
        contenido=archi1.readline()
        dato=float(contenido)
        #dato=int(float(contenido))
        #print(dato)
        archi1.close()
        return dato
```

## Anexo 14. Código Python promedio.py

```
from publicar import envio
from lectura import Lectura
import time
from arduinoS import EnvioS
from timeit import default_timer

class promedio():

    def calculo(self):
        datos=Lectura()
        lumenes1=datos.recuperar_dato('/home/pi/luminaria1.txt')
        lumenes2=datos.recuperar_dato('/home/pi/luminaria2.txt')
        lumenes3=datos.recuperar_dato('/home/pi/luminaria3.txt')
        lumenes4=datos.recuperar_dato('/home/pi/luminaria4.txt')
        lumenes5=datos.recuperar_dato('/home/pi/luminaria5.txt')
        lumenes6=datos.recuperar_dato('/home/pi/luminaria6.txt')
        lumenes7=datos.recuperar_dato('/home/pi/luminaria7.txt')
        Promedio_lumenes=int((lumenes1+lumenes2+lumenes3+lumenes4+lumenes
5+lumenes6+lumenes7)/7)
        return Promedio_lumenes
```

## Anexo 15. Código Python publicar.py

```
import paho.mqtt.client as mqtt
from random import randrange, uniform
import time
class envio():

    def solicitud(self,topico,dato):
        self.topico=topico
        self.dato=dato
        client = mqtt.Client("Central")
        client.username_pw_set(username="luminarias",password="UTN2021")
        client.connect('192.168.0.4',1883,60)
        client.publish(self.topico,self.dato)
        client.disconnect()
        return "publicado"
```