



UNIVERSIDAD NACIONAL DEL ALTIPLANO DE PUNO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INTERNET
DE LAS COSAS PARA LA GESTIÓN DEL ALUMBRADO
PÚBLICO EN LA CIUDAD UNIVERSITARIA DE LA
UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO.**

TESIS

PRESENTADA POR:

Bach. FAUSTINO HECTOR ALANOCA CAUNA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2019



DEDICATORIA

A Dios por su infinita bondad, por darme la fe, la esperanza y por estar siempre a mi lado en todo momento protegiéndome y darme la fuerza, sabiduría y guiar mis pasos cada día a pesar de los grandes obstáculos que en el devenir se convirtieron en fortalezas para mi superación profesional.

A mis padres Fausto y Victoria, por darme su apoyo, a mi hija Daenerys Kristel que es mi razón de ser para seguir adelante, y a mi esposa Yanin que sin su paciencia, amor, comprensión y apoyo incondicional no hubiera sido posible este logro, por estar a mi lado sin condiciones y ser mi fortaleza en todo momento. A todos ellos mi eterno reconocimiento.

Faustino Hector Alanoca Cauna



AGRADECIMIENTOS

A la Universidad Nacional del Altiplano de Puno, quien permitió formarme académicamente; a las autoridades y docentes de la Escuela Profesional De Ingeniería Electrónica, por abrirme las puertas para desarrollarme profesionalmente y seguir adelante con mis propósitos.

A mi asesor de investigación de tesis Dr. Eudes Rigoberto Apaza Estaño, quien con sus orientaciones, conocimientos y motivación ha logrado que realice esta investigación.

Agradezco a los miembros del jurado, quien con su sabiduría, conocimiento y apoyo que me brindaron para el desarrollo de la investigación.

Faustino Hector Alanoca Cauna



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTOS

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INDICE DE ACRÓNIMOS

RESUMEN 12

ABSTRACT..... 13

CAPÍTULO I

INTRODUCCIÓN

1.1. FORMULACION DEL PROBLEMA 16

1.2. HIPÓTESIS DE LA INVESTIGACIÓN 16

1.3. OBJETIVOS DE LA INVESTIGACIÓN..... 16

1.3.1. Objetivo general 16

1.3.2. Objetivos específicos 16

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACION 17

2.2. MARCO TEORICO 19

2.2.1. Internet de las cosas (IOT) 19

2.2.2. Microcontroladores 36

2.2.3. Entornos de programación 38

2.2.4. Arduino 40

2.2.3. Sensores..... 43



2.2.4. Actuadores.....	50
------------------------	----

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 MATERIALES.....	52
3.1.1 Hardware	52
3.1.2 Software	54
3.2 MÉTODO.....	55
3.2.1 Diseño de la investigación	55
3.2.2 Nivel de la investigación	55
3.2.3 Población y muestra de la investigación	55
3.2.4 Ubicación y descripción de la investigación.....	57
3.2.5 Técnicas e instrumentos de recolección de datos.....	58
3.2.6 Procedimiento de diseño del prototipo.....	59

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 PRUEBA DE CONECTIVIDAD CON ICMP	84
4.2 RESULTADOS DEL SERVIDOR WEB.....	86
4.3 RESULTADOS DEL FUNCIONAMIENTO	93
4.3.1 Alta iluminación.....	93
4.3.2 Modo nocturno	95
4.4 DETECCIÓN DE MOVIMIENTO EN EL ALUMBRADO	97
V. CONCLUSIONES	100
VI. RECOMENDACIONES.....	101
VII. REFERENCIAS.....	102
ANEXOS.....	105



Área : Telecomunicaciones

Tema : Internet de las Cosa

FECHA DE SUSTENTACIÓN: 30 de diciembre 2019



ÍNDICE DE TABLAS

Tabla 1. Diferentes patrones de uso	25
Tabla 2. Tipos de sensores	46
Tabla 3. Técnicas e instrumentos para recolección de datos	58



ÍNDICE DE FIGURAS

Figura 1. IoT es una pila muy profunda	22
Figura 2. El ciclo Big Data	23
Figura 3. Datos en la niebla	26
Figura 4. Diseño de la arquitectura de IoT	32
Figura 5. Componentes que integran a un microcontrolador	37
Figura 6. Diagrama de la placa de arduino UNO	41
Figura 7. Partes de la fotorresistencia.....	48
Figura 8. Arduino UNO R3.	53
Figura 9. Arduino Ethernet Shield 2.....	54
Figura 10. Unidades de alumbrado en la ciudad universitaria.....	56
Figura 11. conexión grafica del Arduino UNO con el Shield de Ethernet	59
Figura 12. Conexión esquemática del Arduino UNO con el Shield de Ethernet	60
Figura 13. Diagrama grafica de las conexiones de componentes	63
Figura 14. Configuración pull-down	64
Figura 15. Diagrama esquemática de las conexiones de componentes	66
Figura 16. Vista frontal del sensor HCSR501 PIR	69
Figura 17. ARGUS Standard CCTR1P002	70
Figura 18. Rango de detección	71
Figura 19. Vaya LED Uplight	72
Figura 20. MileWide2LED	73
Figura 21. Diagrama esquemática del módulo relé	74
Figura 22. Programación del controlador Arduino.....	75
Figura 23. Medición de la altura de un poste de alumbrado.....	77
Figura 24. Instalación del sensor de presencia en un poste de alumbrado.	78



Figura 25. Triángulo conocido para calcular la distancia entre una persona y el poste.	79
Figura 26. Medición de la distancia entre dos postes de alumbrado.	80
Figura 27. Medición de la distancia entre dos postes de alumbrado.	80
Figura 28. Verificación de la distancia máxima para detectar la presencia de una persona.	81
Figura 29. Topología de la implementación.	83
Figura 30. Prueba de conexión con el web server	87
Figura 31. Captura de la visualización al ingresar al servidor web	88
Figura 32. Inspección de la programación HTML del arduino	89
Figura 33. Petición del cliente desde el ordenador	90
Figura 34. Petición del cliente desde el smartphone.....	91
Figura 35. Momento al pulsar los botones en el cliente desde ordenador	92
Figura 36. Momento al pulsar los botones en el cliente desde smartphone	93
Figura 37. Maqueta con iluminación	94
Figura 38. Gestión en un cliente web cuando exista iluminación	95
Figura 39. Activación de las luminarias por la fotorresistencia	96
Figura 40. Estado en el cliente web al encender las primeras luminarias	97
Figura 41. Detección de movimiento en el alumbrado N°1.....	98
Figura 42. Estado en el cliente web al encender la primera luminaria del alumbrado N°1	99
Figura 43. Diagrama de conexión Ethernet para realizar simulaciones.	127
Figura 44. Dirección MAC del Shield de Ethernet.....	127
Figura 45. Forma de implementación de maqueta con dos leds (uno para la iluminación menos intenso y el otro para alta intensidad).....	128
Figura 46. Ubicación de la fotorresistencia para la implementación de la maqueta. ..	128



Figura 47. Resultado cuando solamente se detecta movimiento el segundo alumbrado.	129
Figura 48. Resultado en el navegador web cuando solamente se detecta movimiento el segundo alumbrado.	129
Figura 49. Resultado cuando se detecta movimiento en sensores de movimiento de ambos alumbrados.	130
Figura 50. Resultado en el navegador web cuando se detecta movimiento en sensores de movimiento de ambos alumbrados.	130
Figura 51. Pruebas en el campus de la universidad.	131
Figura 52. Instalación del sensor de presencia.	132
Figura 53. Instalación del sensor de presencia en el poste.	133
Figura 54. Instalación del sensor de presencia a 6.6 metros sobre el nivel del suelo..	134
Figura 55. Medición de la altura de un poste dentro del campus.	135
Figura 56. Medición de la distancia entre dos postes.	136
Figura 57. Medición de la distancia entre dos postes.	136
Figura 58. Extensión de la herramienta para tomar medidas entre dos postes.	137
Figura 59. Poste de muestra número 1.....	138
Figura 60. Poste de muestra número 2.....	138



INDICE DE ACRÓNIMOS

ACID	: Atomicity, Consistency, Isolation and Durability
BLE	: Bluetooth Low Energy
C	: Lenguaje de programación
CPU	: Central Processing Unit
ENIAC	: Electronic Numerical Integrator And Computer
ETL	: Extract, Transform and Load
GSM	: Global System for Mobile communications
IA	: Inteligencia Artificial
IDE	: Integrated Development Environment
IoT	: Internet of Things
LED	: Light-Emitting Diode
M2M	: Machine to Machine
MCU	: Multipoint Control Unit
MIT	: Massachusetts Institute of Technology
OLAP	: On-Line Analytical Processing
PLC	: Programmable Logic Controller
RAM	: Random Access Memory
SDCC	: Secure Direct Client-to-Client
SQL	: Structured Query Language



RESUMEN

Esta investigación tiene como objetivo general, diseñar e implementar un sistema de Internet de las cosas para la gestión del alumbrado público en la ciudad universitaria de la Universidad Nacional del Altiplano Puno. Se realizó una investigación del nivel exploratorio y el diseño de investigación es experimental. Implicando aspectos como el ahorro energético, la movilidad sostenible, la administración electrónica y la atención a las personas, todos estos mediante la arquitectura IoT gestionándoles remotamente. El sistema esta conformado por un controlador que fue ubicado en las unidades de alumbrado público dentro de la ciudad universitaria, este controlador recibe señales de los sensores, los interpreta en forma digital o analógica y envía los datos remotamente hacia un administrador, junto a esto es capaz de recibir señales remotas del administrador para controlar el estado de las luminarias porque son dos las formas de operación, una de intensidad mínima y otra de una intensidad más fuerte, y también apagarla o encenderla. Aparte de recibir señales de los sensores, también el controlador muestra el estado de las luminarias, si está encendido o apagado. El diseño del sistema de Internet de las cosas para la gestión del alumbrado público en la ciudad universitaria de la Universidad Nacional del Altiplano Puno es de gran avance exploratorio, ya que abre una nueva forma de monitorear y controlar un sistema de iluminación. Al implementar el prototipo resulto posible la gestión del sistema, mediante un navegador web en el cliente, y al implementar el prototipo fue posible determinar el estado del prototipo y también realizar el control del mismo; otro resultado muy favorable se producirá en el momento de realizar el mantenimiento de las unidades de alumbrado, al mismo tiempo que favorece a la economía de la unidad de mantenimiento técnico.

Palabras Clave: Alumbrado público, Gestión, monitoreo, IoT.



ABSTRACT

This research has the general objective of designing and implementing an Internet of Things system for the management of public lighting in the university city of the Universidad Nacional del Altiplano Puno. An investigation of the exploratory level was carried out and the research design is experimental. Involving aspects such as energy saving, sustainable mobility, electronic administration and attention to people, all these through the IoT architecture, managing them remotely. The system is formed by a controller that was located in the public lighting units within the university city, this controller receives signals from the sensors, interprets them in digital or analog form and sends the data remotely to an administrator, along with this is able to receive remote signals from the administrator to control the state of the luminaires because there are two forms of operation, one of minimum intensity and one of a stronger intensity, and also turn it off or on. Apart from receiving signals from the sensors, the controller also shows the status of the luminaires, whether it is on or off. The design of the Internet of things system for the management of public lighting in the university city of the National University of Puno Altiplano is a great exploratory advance, since it opens a new way of monitoring and controlling a lighting system. By implementing the prototype it was possible to manage the system, through a web browser on the client, and by implementing the prototype it was possible to determine the status of the prototype and also to control it; another very favorable result will be produced at the time of maintenance of the lighting units, at the same time that it favors the economy of the technical maintenance unit.

Key Words: Public lighting, Management, monitoring, IoT.



CAPÍTULO I

INTRODUCCIÓN

Los alumbrados públicos en la Universidad Nacional del Altiplano Puno están siendo gestionados individualmente, además por su cantidad el personal encargado no puede percatarse del funcionamiento de todo. Las lámparas a horas de la noche; también las condiciones climatológicas son confusas para los alumbrados, por ejemplo los días nublados que se muestran más en los meses de verano, ya que por esta razón algunos se prenden por el día; otro problema es la cantidad de personas que circulan por las vías de la ciudad universitaria a horas de la noche, ya que la cantidad de personas disminuye a estas horas, también los días no laborables, es decir sábados, domingos y feriados la que la cantidad de estudiantes, docentes y personal administrativo no se encuentran o son una mínima cantidad, por esta razón en algunos alumbrados no es necesario que continúen encendidas cuando no haya circulación de personas. El monitoreo inalámbrico mediante IoT pretende ser una mejor opción ya que será para el un buen funcionamiento y en un correcto momento cuando sea necesario.

El alumbrado público conectado a IoT muestra alta eficiencia energética se convertirá en una ciudad inteligente como algunos proyectos mostrados por las compañías fabricantes de equipos que proporcionan iluminación. Será necesaria una conectividad inalámbrica fiable para que esto suceda. La vinculación de las luces de calle con sensores, dispositivos y sistemas de gestión va más allá de la iluminación, ayudando a que las ciudades sean más eficientes energéticamente.

La investigación se desarrolló en la ciudad universitaria y en el laboratorio de Electrónica General de la Escuela Profesional de Ingeniería Electrónica de la Universidad Nacional del Altiplano.



Desde que surgió el Internet ha ido creciendo de una forma descontrolada, y aún sigue revolucionando la forma en la que concebimos nuestros hogares, nuestras ciudades e incluso nuestra iluminación, para muchos incluso antes de que les haya dado tiempo a aceptar esos cambios. Y es que, gracias a Internet, se han realizado muchos avances como por ejemplo el llamado Internet de las cosas (Internet of things, IoT) Se cree que para el 2020 habrá en el mundo nada menos que unos 250 objetos conectados por segundo, que en total serán unos 50 billones de objetos conectados. La presente investigación puede ya ser un pequeño paso en diferentes ámbitos de las ciudades como una base para ser una ciudad inteligente usando IoT, vamos a centrarnos en aquellos que están relacionados con el campo de la iluminación y que ya son una realidad, como, por ejemplo, poder vincular las luces de las calles con sensores. Esto brindara una gestión y monitoreo de los alumbrados públicos, brindando aspectos como el ahorro energético, la movilidad sostenible, la administración electrónica, la atención a las personas y la seguridad vial.

En la revisión literaria incluirá toda la fundamentación teórica necesaria sobre Internet de las cosas, sensores como el sensor de movimiento y las fotorresistencias, actuadores, al igual que microcontroladores definiendo como es el funcionamiento de un Arduino, al igual que los antecedentes a esta investigación, los objetivos e hipótesis planteadas en la investigación.

En la sección de métodos y materiales indicara el diseño y nivel de investigación, junto a la población dirigida y el muestreo que se realiza, los métodos de recolección de datos, además los materiales usados junto algunas especificaciones y termina con la implementación de la maqueta que simula el sistema

Los resultados se muestran de las pruebas realizadas tanto en las conexiones con los clientes web y el funcionamiento en cada situación, con referencias fotográficas.



1.1.FORMULACION DEL PROBLEMA

¿Cómo realizar el diseño e implementación de un sistema de Internet de las cosas que permitirá la gestión del alumbrado público en la ciudad universitaria de la Universidad Nacional del Altiplano Puno, de esa forma contribuir con ventajas para el mantenimiento de las unidades de alumbrado público en la ciudad universitaria?

1.2.HIPÓTESIS DE LA INVESTIGACIÓN

El diseño e implementación de un sistema de Internet de las cosas permitirá la gestión del alumbrado público en la ciudad universitaria de la Universidad Nacional del Altiplano Puno.

1.3.OBJETIVOS DE LA INVESTIGACIÓN

1.3.1. Objetivo general

Diseñar e implementar un sistema de Internet de las cosas para la gestión del alumbrado público en la ciudad universitaria de la Universidad Nacional del Altiplano Puno.

1.3.2. Objetivos específicos

- Diseñar un sistema de Internet de las cosas para el alumbrado público de la ciudad universitaria de la Universidad Nacional del Altiplano Puno.
- Diseñar e implementar un prototipo que permita la evaluación del sistema de alumbrado en la ciudad universitaria.



CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACION

Se encontraron las siguientes investigaciones relacionadas a internet de las cosas y la gestión de los alumbrados públicos, se muestra los antecedentes a continuación:

Rodríguez (2017), en su investigación llegó a la conclusión que el empleo de controles inteligentes de iluminación en los sistemas de alumbrado público se considera como una alternativa eficiente para obtener un mayor ahorro tanto energético como económico. La implementación de una Plataforma Cloud permite el control de la iluminación de forma remota, facilitando cambios en los niveles de porcentajes de iluminación de las luminarias, si es el caso. Asimismo, permite llevar un control del consumo energético de las luminarias y visualizar el funcionamiento de los sistemas de iluminación en informes semanales y mensuales. El sistema de control de iluminación con luminarias LED implementado en el Campus Huachi UTA permite obtener un significativo ahorro energético y económico. Se calculó un ahorro de hasta 151,36 GWh al año, con un ahorro económico en iluminación pública de hasta 1.740 dólares anualmente.

Navas (2017), en su investigación llegó a la conclusión que los resultados obtenidos de los experimentos realizados con el prototipo fueron satisfactorios, ya que de cada 10 datos adquiridos y procesados por el sistema electrónico y enviados a través de la red inalámbrica 9 fueron almacenados en la base de datos sin inconvenientes y el otro se lo recibió pero con una transmisión realizada por parte del módulo ESP8266-01, lo cual determina que el sistema de red telemétrico tiene un 90% de efectividad en la



adquisición, envío y almacenamiento de la información. Se realizó las pruebas en el prototipo para el encendido y apagado de la lámpara, enviando ordenes desde una aplicación web, obteniendo como resultado que de cada 10 ejecuciones para encender o apagar 8 se realizaron en un tiempo relativamente rápido y 2 se realizaron con un pequeño retardo, pero todas las acciones fueron exitosas.

Zanella *et al.* (2014), concluyen que al analizar las soluciones disponibles actualmente para la implementación de IoTs urbanas. Las tecnologías discutidas están a punto de estandarizarse y los actores de la industria ya están activos en la producción de dispositivos que aprovechan estas tecnologías para permitir las aplicaciones de interés, como las descritas en la Sección II. De hecho, aunque el rango de opciones de diseño para los sistemas IoT es bastante amplio, el conjunto de protocolos abiertos y estandarizados es significativamente menor. Las tecnologías habilitadoras, además, han alcanzado un nivel de madurez que permite la realización práctica de soluciones y servicios IoT, a partir de ensayos de campo que esperamos ayuden a despejar la incertidumbre que aún impide una adopción masiva del paradigma IoT. Una implementación concreta de la prueba de concepto, implementada en colaboración con la ciudad de Padova, Italia, también ha sido descrita como un ejemplo relevante de la aplicación del paradigma IOT a las ciudades inteligentes.

Moenuddin (2017), concluye que el sistema proporciona un sistema de control automático eficiente e inteligente con la tecnología Zigbee. El sistema puede reducir el consumo de energía y los costos de mantenimiento y también ayuda a reducir las actividades delictivas hasta cierto límite. Este sistema de control de alumbrado ayuda en el ahorro de energía, la detección de luces defectuosas y el tiempo de mantenimiento y el aumento de la vida útil del sistema. Otra ventaja obtenida por el sistema de control es la gestión inteligente de los postes de la lámpara mediante el envío de información a una



estación central mediante la comunicación inalámbrica de Zigbee. El mantenimiento del sistema puede planificarse fácil y eficientemente desde la estación central, lo que permite ahorros adicionales. El sistema propuesto es particularmente apropiado para el alumbrado público en zonas urbanas y rurales donde el tráfico es bajo en un intervalo de tiempo dado. La naturaleza independiente de la red de suministro de energía permite realizar el sistema en áreas remotas donde las instalaciones clásicas son prohibitivamente caras. El sistema es siempre flexible, extensible y totalmente ajustable a las necesidades del usuario. La simplicidad de Zigbee, la fiabilidad de los componentes electrónicos, la característica de la red de sensores, la velocidad de procesamiento, los costes reducidos y la facilidad de instalación son las características que caracterizan el sistema propuesto.

2.2.MARCO TEORICO

2.2.1. INTERNET DE LAS COSAS (IOT)

Construir redes telefónicas para servir a los hogares de los consumidores tomó otros 30 o 40 años. Desde la introducción en 1945 de ENIAC, la primera computadora electrónica, hasta que la disponibilidad generalizada de computadoras de escritorio tomó 40 años. La construcción de Internet moderna tomó aproximadamente 30 años.

En cada caso, la adopción se vio frenada por la necesidad de rediseñar los procesos existentes. Las fábricas impulsadas por vapor convertidas a electricidad a través del proceso torpe y lento de reemplazo gradual; cuando falló la maquinaria de vapor, se trajeron máquinas eléctricas, pero la huella de la fábrica siguió siendo la misma. Henry Ford fue el primero en darse cuenta de que el desarrollo, la ingeniería y la producción deberían girar en torno al producto, no a la fuente de energía. Esta idea forzó la convergencia de muchos sistemas vinculados al proceso: diseño de planta, fuente de energía, cadena de suministro, trabajo y distribución, entre otros.



En todos estos casos, se construyeron torres de capacidad y, durante décadas de adopción, las torres se movieron levemente y finalmente convergieron. Podemos predecir que se producirá convergencia entre algunas tecnologías, pero puede ser difícil comprender el momento o la forma del resultado a medida que las diferentes torres verticales comienzan a conectarse entre sí.

Así es con el Internet de las cosas. Muchas torres de tecnología comienzan a inclinarse hacia una arquitectura de referencia de IoT: comunicaciones de máquina a máquina, Big Data, computación en la nube, grandes sistemas distribuidos, redes, dispositivos móviles y telecomunicaciones, aplicaciones, dispositivos inteligentes y seguridad, pero no es predecible. (Dacosta, 2013, p.50)

2.2.1.1.PRECURSORES E INDICADORES PRINCIPALES

La informática empresarial y el control de procesos industriales son los principales antepasados de IoT emergente. El tema general ha sido la descentralización del hardware: la entrega de los sistemas de computación "big iron" construidos para las compañías de seguros, los bancos, la compañía telefónica y el gobierno ha dado paso a servidores, computadoras de escritorio y computadoras portátiles; como los envíos de computadoras directas a los usuarios finales han disminuido, la adopción de dispositivos móviles y la computación en la nube se han acelerado. De manera similar, los sistemas de control de procesos analógicos construidos para controlar fábricas y plantas de energía han pasado por fases de evolución, pero aquí la tendencia ha sido en la otra dirección: centralización de la información: desde medidores de cuadrante, válvulas operadas manualmente y conmutadores neumáticos a sistemas automatizados conectado a sensores integrados. Estas tendencias juegan un papel en IoT pero son al mismo tiempo independientes. El



papel de IoT está conectando estas diferentes tecnologías y tendencias a medida que las torres de tecnología comienzan a converger. (Dacosta, 2013, p.50)

Telecomunicaciones y redes; dispositivos móviles y sus múltiples aplicaciones; dispositivos integrados; sensores; y la nube computa recursos para procesar datos a escala de IoT. Alrededor de este complicado entorno se encuentran mecanismos de identidad y seguridad sofisticados, aunque a veces en conflicto, que permiten que las aplicaciones hablen autoritaria y privada. Estos millones de dispositivos conectados y miles de millones de sensores deben conectarse de forma fiable y segura. Las industrias detrás de cada una de estas tecnologías tienen un punto de vista y un rol a desempeñar en IoT. A medida que la red mundial, los dispositivos móviles, la nube, los datos y las empresas de identidad luchan por la posición, cada uno trata de moldear el mercado para solidificar dónde pueden competir, dónde tienen poder y dónde necesitan colaborar. Además de las tecnologías de conexión, IoT conecta industrias dispares. Las iniciativas inteligentes están en marcha en casi todos los sectores de nuestra economía, desde la atención sanitaria a la automoción, las ciudades inteligentes hasta el transporte inteligente, la energía inteligente y las granjas inteligentes. Cada una de estas industrias separadas depende de toda la tecnología. Por lo tanto, las aplicaciones de IoT se cruzarán a través de la comunicación móvil, la nube, los datos, la seguridad, las telecomunicaciones y las redes, con pocas excepciones. (Asay, 2014, p.30)

IoT es fundamentalmente la conexión de nuestros dispositivos a nuestro contexto, una convergencia imposible antes habilitada ahora por una combinación de computación avanzada, redes generalizadas, computación en la nube centralizada, computación por niebla y tecnologías de bases de datos muy grandes. La seguridad y la identidad contribuyen. Cada una de estas industrias tiene un conjunto complejo de participantes y modelos de negocio, desde jugadores masivos de ecosistemas (Apple, Google) hasta

vendedores de productos (como VoltDB) hasta Amazon. IoT es la mejor combinación entre estos jugadores. IoT no se trata de agregar conectividad a Internet a procesos existentes; se trata de habilitar modelos comerciales innovadores que antes eran imposibles. IoT es una pila muy profunda, como se muestra en la Figura 1.

IoT is a deep stack: Today's focus is data

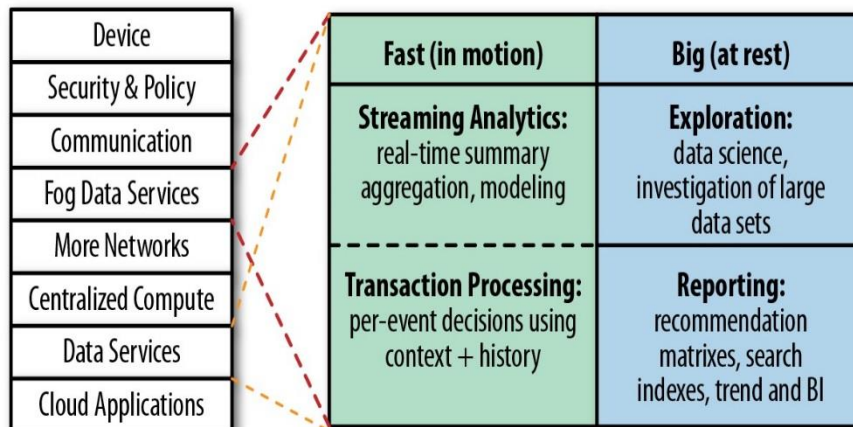


Figura N° 1: IoT es una pila muy profunda

Fuente: Betts (2016).

A medida que continúa esta batalla, está surgiendo una consolidación arquitectónica: una arquitectura de referencia para la gestión de datos en el IoT. Este libro presenta el papel crítico de la base de datos operativa en esa convergencia. (Di Martino y Esposito, 2018)

2.2.1.2. TRANSACCIONES OPERATIVAS Y ANALÍTICAS

Big Data y el IoT están estrechamente relacionados; La similitud es importante porque muchas organizaciones vieron la oportunidad de resolver los desafíos comerciales con Big Data tan recientemente como hace 10 años. Estas empresas pasaron por un ciclo de tratar de resolver grandes problemas de datos. Primero recopilamos una serie de eventos o registraron datos, ensamblándola en un repositorio que les permitió comenzar a explorar los datos recopilados. La exploración fue la segunda parte del ciclo. El proceso de exploración buscó información comercial, por ejemplo, segmentar a los clientes para

descubrir tendencias predictivas o modelos que podrían usarse para mejorar la rentabilidad o la interacción del usuario, lo que ahora denominamos ciencia de datos. Una vez que las empresas encontraron información de la exploración, el siguiente paso del ciclo fue formalizar esta exploración en un proceso analítico repetible, que a menudo implicaba algún tipo de informe, como generar un índice de búsqueda grande o construir un modelo predictivo estadístico.

A medida que las industrias trabajaron en las primeras partes de los ciclos (recoger, explorar, analizar), implementaron y utilizaron diferentes tecnologías, por lo que además del ciclo analítico hay un círculo virtuoso de innovación tecnológica y organizativa. Las nuevas tecnologías conducen a innovaciones organizativas, ya que una mejor comprensión de los datos permite a los líderes de la industria adoptar un modelo operativo impulsado por datos. El ciclo se muestra en la Figura 2. (Di Martino y Esposito, 2018)

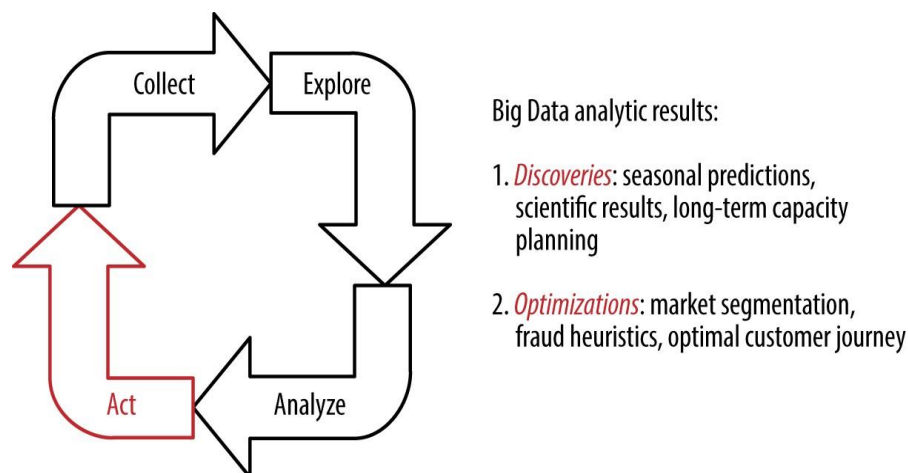


Figura N° 2: El ciclo Big Data
Fuente: Betts (2016).

En la naciente IoT, la fase de recolección del ciclo analítico probablemente desplegó sistemas como Flume o Kafka u otras herramientas orientadas al ingreso. La fase de exploración involucró herramientas estadísticas, así como herramientas de exploración de datos, herramientas gráficas y herramientas de visualización. Una vez que



se identificaron y formalizaron los valiosos informes y análisis, los arquitectos recurrieron a herramientas de informes rápidas y eficientes, como los sistemas OLAP de relación rápida. Sin embargo, hasta este punto, ninguno de los datos, puntos de vista, optimizaciones o modelos recopilados, descubiertos y luego informados se pusieron en uso. Hasta ahora, durante este ciclo, las empresas aprendieron mucho, pero no crearon una aplicación que utilizara ese conocimiento para mejorar los ingresos, la experiencia del cliente o la eficiencia de los recursos. La realización de mejoras operativas a menudo requería una aplicación, y esa aplicación normalmente requería una base de datos operativa que funcionara a velocidades de transmisión.

Las aplicaciones en tiempo real nos permiten tomar conocimiento sobre el comportamiento del cliente o crear modelos que describan cómo podemos interactuar mejor en el mercado. Esto nos permite utilizar la sabiduría histórica, las percepciones analíticas que hemos obtenido, con datos contextuales en tiempo real del feed de datos en vivo, para ofrecer una experiencia de mercado de un usuario móvil, proteger a los clientes del fraude, mejorar ofertas a través de tecnología publicitaria o capacidades de venta, personalice una experiencia o asigne de manera óptima recursos basados en condiciones de tiempo real. (Betts, 2016, p.40)

Tabla N° 1: Diferentes patrones de uso

Tipo	Decisiones en tiempo real	ETL en tiempo real	Análisis en tiempo real / almacenamiento o en caché SQL
Alimentación de entrada	Personalización, solicitudes de puntuación en tiempo real	Datos del sensor, M2M, IoT	Alimentación en tiempo real que se observa para la inteligencia operativa
Metadatos del evento	Parámetros de la política; PDI, perfiles de usuario	Metadatos sobre la infraestructura de los sensores (versiones, ubicaciones, etc.)	
Big data analytic outputs	Rúbricas de puntuación; perfil de segmentación de usuario	Parámetros de interpolación; mín / máx parámetros de validación de umbral	Resultados del informe OLAP en Casos de uso "SQL Caching".
Tipo	Decisiones en tiempo real	ETL en tiempo real	Análisis en tiempo real / almacenamiento o en caché SQL
Respuestas de eventos y Alertas 24/7	Decisiones y resultados de personalización	Alertas / notificaciones sobre eventos excepcionales (o secuencias excepcionales de eventos)	Dashboard y respuestas de consulta de BI. Contadores, tablas de clasificación, agregados y agrupaciones cronológicas para monitoreo operacional
Salida de alimentación	Archivo de flujo de transacciones para análisis históricos	Recurso enriquecido, filtrado, procesado de eventos entregado abajo	

Fuente: Betts, R. (2016). Architecting for the Internet of Things

Estas aplicaciones en tiempo real, adoptadas primero por organizaciones basadas en datos, requieren soporte de bases de datos operativas: una base de datos que permite transacciones ACID para admitir autorizaciones precisas, decisiones precisas de cumplimiento de políticas, asignación correcta de recursos restringidos, evaluación correcta de reglas y opciones de personalización específicas. (Betts, 2016, p.41)

2.2.1.3.COMPUTACIÓN DE NIEBLA, COMPUTACIÓN DE BORDE Y DATOS EN MOVIMIENTO

La intersección de las personas, los datos y los dispositivos IoT está teniendo importantes impactos en la productividad y la eficiencia de la fabricación industrial. Un ejemplo de datos rápidos en IoT industrial es el uso de datos en movimiento, con sensores de temperatura y presión de gas IoT, para mejorar la fabricación de semiconductores.

La eficiencia operativa es un motor primario de la IoT industrial. Presentamos técnicas avanzadas de automatización y gestión de procesos. Con datos rápidos, los fabricantes pueden implementar técnicas de producción más flexibles.

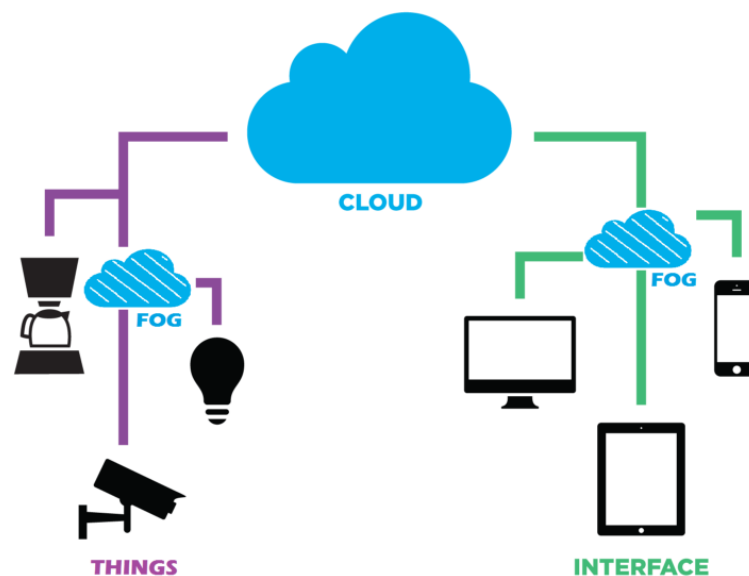


Figura N° 3: Datos en la niebla
Fuente: Betts (2016).



Las organizaciones industriales emplean cada vez más sensores y actuadores para monitorear los entornos de producción en tiempo real, iniciando procesos y respondiendo a anomalías de manera localizada bajo el paraguas de la computación avanzada. Para ampliar esta capacidad a un nivel de planta de producción, es importante tener tecnologías habilitadoras en el nivel de computación niebla. Esto permite reducir los costos operativos generales de los entornos de producción al tiempo que optimiza la productividad y los rendimientos. (Betts, 2016, p.50)

Los sensores avanzados brindan a los dispositivos IoT; **Error! Marcador no definido.** mayores capacidades para monitorear la temperatura, presión, voltaje y movimiento en tiempo real, de modo que la administración pueda ser más consciente de los factores que afectan la eficiencia de la producción. Al incorporar datos rápidos en los procesos de producción, los fabricantes pueden mejorar las eficiencias de producción y evitar posibles retrasos en la fabricación mediante el aprovechamiento eficaz de los datos de producción en tiempo real.

La integración de IoT industrial con datos rápidos permite el uso de análisis y transacciones correlativas en tiempo real en múltiples alimentaciones de datos paralelas desde dispositivos de borde. Los datos rápidos permiten a los desarrolladores capturar y comunicar información precisa sobre los procesos de producción para evitar retrasos en la fabricación y transformar la IoT industrial usando decisiones procesables en tiempo real.

Ya sea que construyamos una arquitectura estilo niebla con sofisticados recursos de almacenamiento y cálculo de bordes o una aplicación centralizada basada en la nube, los requisitos básicos de gestión de datos siguen siendo los mismos. Las aplicaciones continúan requiriendo soporte analítico y operacional ya sea que corran más cerca del



borde o del centro. En IoT industrial, saber qué hay en sus datos y actuar sobre ellos en tiempo real requiere una base de datos operativa que pueda procesar los datos de los sensores tan rápido como sea posible para tomar decisiones y notificar los sensores apropiados de las acciones necesarias de manera prescriptiva. (Betts, 2016, p.52)

2.2.1.4.LAS CUATRO ACTIVIDADES RÁPIDAS DE DATOS

Cuando analizamos los requisitos de las aplicaciones rápidas transaccionales u operativas de datos, vemos cuatro actividades diferentes que deben llevarse a cabo de manera orientada a eventos en tiempo real. A medida que se originan los datos, se analiza el contexto y se presenta a las aplicaciones que tienen efectos secundarios que afectan a los negocios, y luego se capturan para el almacenamiento a largo plazo. Describimos este flujo como ingreso, análisis, decisión y exportación.

Debe poder escalar a las tasas de ingreso de datos entrantes muy rápidos de datos, tal vez datos de registro o datos de sensores, quizás datos de interacción generados por una gran plataforma SaaS o datos de medición en tiempo real desde una red de grillas inteligentes. Debe ser capaz de procesar cientos de miles o incluso millones de eventos por segundo en un flujo orientado a eventos y en la moda operativa antes de que los datos se registren para siempre en un gran almacén de datos para futuras exploraciones y análisis.

Es posible que desee mirar para ver si el evento activa una ejecución de política o tal vez califica a un usuario para una campaña de promoción u oferta. Estas son todas las transacciones que deben producirse contra el feed de eventos en tiempo real. Para tomar estas decisiones, debe ser capaz de combinar las analíticas derivadas del repositorio de datos grandes con el contexto en el análisis en tiempo real generado a partir del flujo entrante de datos.



A medida que se reciben estos datos, debe ser capaz de tomar decisiones en su contra: para apoyar a las aplicaciones que procesan estos eventos en tiempo real. Necesita poder ver los eventos, compararlos con los eventos que se han visto anteriormente y luego proporcionar una habilidad para hacer una decisión a medida que llega cada evento. Desea poder decidir si un evento en particular está en la norma de un proceso, o si es algo que necesita generar una alerta. Una vez que estos datos han sido ingresados y procesados, puede haber un filtrado o proceso de transformación en tiempo real para crear sesiones para extraer los eventos a archivar, o tal vez para reescribirlos en un formato óptimo para históricos analítica. Estos datos luego se exportan al lado de los grandes datos. (Betts, 2016, p.55)

2.2.1.5.FUNCIONES DE UNA BASE DE DATOS EN UNA INFRAESTRUCTURA

DE IOT;Error! Marcador no definido.

Los sistemas heredados de administración de datos no están diseñados para manejar grandes entradas de datos de alta velocidad provenientes de múltiples dispositivos y fuentes. Por lo tanto, administrar y extraer valor de los datos de IoT es un desafío apremiante para los arquitectos y desarrolladores de empresas. Incluso las arquitecturas roll-own-own altamente personalizadas carecen de la consistencia, confiabilidad y escalabilidad necesarias para extraer el valor comercial inmediato de los datos de IoT.

Como se señaló anteriormente, las aplicaciones IoT requieren cuatro capacidades de administración de datos:

Ingreso rápido

Las aplicaciones necesitan ingreso a alta velocidad, rendimiento en memoria y escalabilidad horizontal para proporcionar un solo punto de ingreso para alimentaciones



de datos entrantes de muy alta velocidad. Una base de datos operativa debe tener el rendimiento y la capacidad para ingresar datos de entrada entrantes de muy alta velocidad. Estos podrían ser cientos de miles o incluso millones de eventos por segundo, miles de millones y miles de millones de eventos por día. El sistema necesita el rendimiento para ingresar de forma escalable estos eventos y para poder procesar estos eventos a medida que llegan, discretos unos de otros. Si los eventos están agrupados, debe haber lógica en ese lote. El procesamiento por lotes presenta las preocupaciones del orden del evento, llegada, etc. Cuando los eventos se procesan por evento, el resultado es un sistema más potente y flexible.

Explora y analiza

Debe haber acceso en tiempo real a las aplicaciones y los motores de consulta, lo que permite consultas sobre el flujo de datos entrantes que permiten a los motores de reglas procesar la lógica empresarial. Como estos eventos se reciben, almacenan y procesan en la base de datos operativa, el sistema debe permitir el acceso a eventos para aplicaciones o motores de consulta. Este es un flujo de datos diferente. Los eventos de datos a menudo son un flujo de datos de una sola dirección de información en el sistema operativo. Utilizando un motor de reglas como ejemplo de una aplicación que accede a datos operativos, el flujo de datos es un flujo de datos de solicitud / respuesta, una consulta más tradicional. Se hace una pregunta a la base de datos y debe proporcionar una respuesta a la aplicación o al motor de reglas.

Actuar

Las aplicaciones también requieren la capacidad de activar eventos y tomar decisiones basadas en el flujo de entrada: umbrales, reglas, eventos de procesamiento de políticas y más. Los eventos desencadenados pueden ser actualizaciones de un servicio



de notificación simple o de un servicio de cola simple que se implementan, en función de alguna lógica empresarial que se evalúa en la base de datos operativa. Una base de datos operativa puede almacenar esto en la lógica de la base de datos en forma de procedimientos almacenados en Java. Otros sistemas pueden usar una gran cantidad de aplicaciones de trabajo, pero este tercer requisito es el mismo, independientemente de su implementación. Debe poder proporcionar lógica de negocio a medida que lleguen los eventos para ejecutar esa lógica empresarial y, en muchos casos, empujar un efecto secundario a una cola para un procesamiento posterior.

Exportar

Finalmente, la aplicación necesita la capacidad de exportar datos acumulados, filtrados, enriquecidos o aumentados a sistemas descendentes y almacenes analíticos a largo plazo. A menudo, estos sistemas almacenan datos de forma más permanente. Podrían ser plataformas operativas más grandes pero menos reales. Podrían ser un archivo de datos. En algunas situaciones, vemos a personas que usan componentes operativos para almacenar datos entrada y luego alimenta al final del día a sistemas de facturación de fin de día más tradicionales.

Como estos datos se recopilan en un repositorio entrada en tiempo real o en un sistema operativo, puede comenzar a escribir aplicaciones en tiempo real que rastrean los precios en tiempo real o el consumo en tiempo real, por ejemplo, y luego comienzan a administrar datos o sensores inteligentes o dispositivos de una manera más eficiente que cuando los datos solo están disponibles al final del día.

Una función vital de la base de datos operativa en el IoT es proporcionar acceso en tiempo real a las consultas para que los motores de reglas puedan procesar políticas que deben ejecutarse a medida que pasa el tiempo y los eventos llegan.

2.2.1.6.IOT EN LA VIDA PRÁCTICA

Parte del problema se deriva de la negación. Muchas organizaciones no creen que sean parte del ecosistema IoT, cuando en realidad ya lo son. Para las organizaciones involucradas en campos como el cuidado de la salud, la manufactura, la vivienda, el transporte, la seguridad pública, la generación de energía y la distribución de energía, ese tipo de negación es preocupante.



Figura N° 4: Diseño de la arquitectura de IoT
Fuente: Betts (2016).

Aquí hay un escenario para considerar: usted es responsable de operar los ascensores en un moderno edificio de apartamentos de gran altura en el centro de Manhattan. Estás relativamente seguro de que las conexiones de red entre tus dispositivos de control y los ascensores son seguros, pero no puedes estar absolutamente seguro porque cada elevador tiene miles de partes y subconjuntos fabricados por diferentes proveedores, y algunas de esas partes y subconjuntos "llaman a casa" intermitentemente para informar sobre el estado de su operación. Te guste o no, tus ascensores son parte de la IoT.

"Encontramos que las principales piezas de equipos industriales, junto con los sistemas de control industrial y los PLC han estado expuestos al IoT a través del



crecimiento orgánico de las redes en entornos industriales", dice Paul Rogers, presidente y director ejecutivo. Oficial de Wurldtech , una filial de GE especializada en seguridad para Internet industrial. "En muchos casos, el equipo industrial está en línea y la empresa no lo conoce". En otras palabras, si tiene máquinas hablando con controladores en redes inalámbricas, es parte de la IoT, ya sea que lo sepa o no. Los forajidos del mercado libre podrían describir al IoT como una confederación de sistemas dispares que opera bajo la guía de la mano invisible de Adam Smith. Pero aquí hay una comparación más concisa: el IoT de hoy es Dodge City antes de que llegaran los Marshalls estadounidenses. En lugar de acorralar a un grupo de vaqueros borrachos con seis tiradores, estamos sentando las bases para gobernar una cultura naciente basada en miles de millones de máquinas y dispositivos conectados, incluidos aviones, trenes, automóviles, hogares, juguetes y marcapasos. (Betts, 2016, p.57-68)

2.2.1.7.INFORMÁTICA AMBIENTAL

Una mezcla de redes generalizadas y tecnologías de información interoperables ha producido un nuevo estado de vigilancia, conectividad y análisis permanente. No más sentado en las computadoras portátiles o buscando dispositivos móviles.

Hemos ingresado a Age of Ambient Computing, una convergencia de sensores baratos, conectividad inalámbrica, microchips cada vez más potentes y análisis avanzados que redefinen la naturaleza de los espacios públicos y privados en las economías desarrolladas del mundo.

La buena noticia: la computación ambiental es en gran parte manos libres y sin esfuerzo, al menos desde la perspectiva del ciudadano promedio. Las malas noticias: es como el clima, bueno o malo, no puedes evitarlo. Listos o no, estamos rodeados de sensores que recopilan datos y los envían a través de redes a análisis que convierten los



datos en información para un popurrí de usuarios finales que van desde vendedores corporativos hasta agencias de seguridad del gobierno. Aquí hay algunos ejemplos hipotéticos: entras a una sala de conferencias y una baliza Bluetooth Low Energy (BLE) emite un breve mensaje codificado que anuncia su proximidad. Suponiendo que tiene una aplicación en su teléfono que puede interpretar e identificar la baliza, verá un mensaje de bienvenida en su pantalla de bloqueo, junto con las instrucciones para la sala en la que se inscribió una sesión para asistir, comenzando en cinco minutos. Su teléfono también muestra información sobre cuándo se servirá el almuerzo, y confirma su elección del plato de pollo, pescado o vegano.

Ese es un caso bastante benigno e inocuo. Vayamos un poco más y asuma que los sistemas de computación ambiental de la sala de conferencias también pueden extraer bits de datos personales de sus interacciones con su imagen inteligente. Tal vez reciba una advertencia de alergia basada en su propio perfil de salud personal o en los perfiles de salud de personas similares en su cohorte de edad. O tal vez el sistema lo identificará como VIP y alguien le ofrecerá una copa de champán gratis. En una nota más oscura, el sistema podría identificarlo como una amenaza a la seguridad, y podría ser detenido o arrestado.

Digamos que se está ejecutando en Central Park y su teléfono inteligente detecta una arritmia potencialmente peligrosa en su corazón. Podría lanzar una aplicación que le brinde una alerta en pantalla y sugiera que reduzca la velocidad. O podría omitir la alerta y enviar una ambulancia. Habilitar esos tipos de escenario requerirá mucho trabajo detrás de escena. Los problemas de conectividad de red, seguridad del dispositivo y privacidad personal deben ser descubiertos, discutidos y resueltos. Las empresas deberán desarrollar soluciones prácticas que sean fáciles de usar y que puedan escalarse para una gran variedad de mercados y situaciones.



Todo esto sugiere la necesidad de una guía o marco aproximado para ayudar a los desarrolladores, inversores, proveedores, compradores y usuarios a tener una mejor idea de los componentes y capacidades requeridos para los escenarios de computación ambiental. (Betts, 2016, p.70)

2.2.1.8.MÁS INTELIGENTE, MÁS RÁPIDO Y MÁS POTENTE

El progreso de la computación ambiental dependerá en gran medida de la evolución de la tecnología de microchips. En febrero de 2016, los investigadores del MIT dieron a conocer un nuevo y radicalmente más potente chip diseñado específicamente para redes neuronales. El nuevo chip se llama " Eyeriss " , un acrónimo suelto de "acelerador reconfigurable eficiente de energía para redes neuronales convolucionales profundas", y representa un salto crítico en el desarrollo de la informática ambiental.

Básicamente, Eyeriss es tan eficiente y poderoso que permite a los nodos o dispositivos individuales procesar y manipular datos a velocidades que ahora requieren múltiples nodos trabajando en paralelo. Con Eyeriss, se necesita menos información para fluir a través de la red. Los resultados se producen más rápido y, quizás lo más importante, se preserva la privacidad, ya que la mayoría de la información requerida para realizar una tarea computacional nunca debe abandonar el dispositivo.

"La idea es que los sensores pueden hacer más trabajos de manera independiente", dice YuHsin Chen, miembro del equipo de investigación del MIT que diseñó Eyeriss. En lugar de intercambiar datos sin procesar a través de redes inseguras, los sensores equipados con Eyeriss mantienen los datos "en casa" y solo comparten los resultados de su trabajo.



Las próximas generaciones de autos, trenes, aviones, barcos y robots quirúrgicos conectados dependerán de chips como Eyeriss para realizar cálculos de la vida real a la velocidad del rayo, a la vez que se preservan algunos grados de privacidad y seguridad.

Los nuevos chips también allanarán el camino para la integración de redes neuronales convolucionales en escenarios de computación ambiental. Como las redes neuronales convolucionales son como IA, permitirían el desarrollo de sistemas informáticos ambientales que sean capaces de aprender y evolucionar.

En un futuro no muy lejano, los entornos informáticos ambientales harán algo más que simplemente reconocerlo, comprender sus preferencias y analizar su estado emocional. Serán capaces de cambiar y adaptarse para adaptarse a su gusto y estado de ánimo personales.

Eso, por supuesto, es la parte positiva de la computación ambiental. La desventaja es que esos mismos sistemas también sabrán su puntaje, estado civil, afiliación política, raza, religión y país de origen. No es difícil imaginar escenarios en los que esa información se pueda usar para crear entornos que son poco acogedores u hostiles para algunos grupos de personas.

Está claro que hemos avanzado más allá del "Internet de las cosas" y estamos yendo hacia un camino hacia la conectividad y la computación ubicuas. Solo podemos esperar que nuestros marcos sociales, legales y morales evolucionen también. (Asay, 2014, p.108)

2.2.2.MICROCONTROLADORES

Un microcontrolador (o MCU para unidad de microcontrolador) es una computadora pequeña en un solo circuito integrado. En la terminología moderna, es similar a, pero menos sofisticado que un sistema en un chip o SoC; un SoC puede incluir

un microcontrolador como uno de sus componentes. Un microcontrolador contiene una o más CPU (núcleos de procesador) junto con memoria y periféricos de entrada / salidas programables. La memoria de programa en forma de RAM Ferroeléctrica, Flash NOR u ROM OTP también se incluye a menudo en el chip, así como también una pequeña cantidad de RAM. Los microcontroladores están diseñados para aplicaciones integradas, en contraste con los microprocesadores utilizados en computadoras personales u otras aplicaciones de uso general que consisten en varios chips discretos.

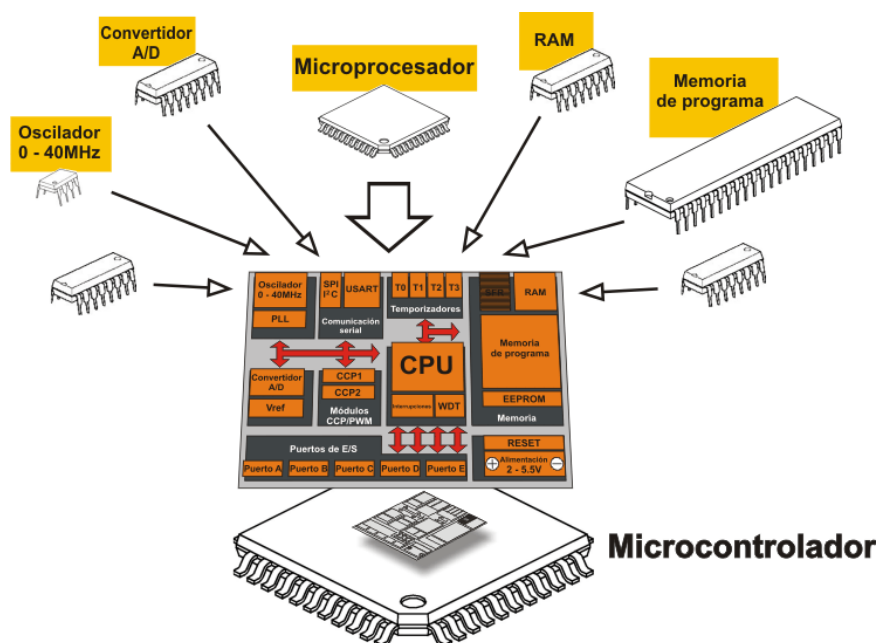


Figura N° 5: Componentes que integran a un microcontrolador
Fuente: Verle, M. (2017)

Los microcontroladores se utilizan en productos y dispositivos controlados automáticamente, como sistemas de control de motores de automóviles, dispositivos médicos implantarles, controles remotos, máquinas de oficina, electrodomésticos, herramientas eléctricas, juguetes y otros sistemas integrados. Al reducir el tamaño y el costo en comparación con un diseño que utiliza un microprocesador, memoria y dispositivos de entrada/salida separados, los microcontroladores hacen que sea económico controlar digitalmente más dispositivos y procesos. Los microcontroladores



de señal mixta son comunes, integrando componentes analógicos necesarios para controlar sistemas electrónicos no digitales. (Verle, 2017, p.90)

Algunos microcontroladores pueden usar palabras de cuatro bits y operar a frecuencias tan bajas como 4 kHz, para un bajo consumo de energía (milivatios o microvatios de un dígito). Por lo general, tendrán la capacidad de conservar la funcionalidad mientras esperan un evento, como presionar un botón u otra interrupción; el consumo de energía mientras duerme (el reloj de la CPU y la mayoría de los periféricos apagados) pueden ser tan solo nanowatts, lo que hace que muchos de ellos sean adecuados para aplicaciones de batería de larga duración. Otros microcontroladores pueden desempeñar roles críticos para el desempeño, donde pueden necesitar actuar más como un procesador de señal digital (DSP), con mayores velocidades de reloj y consumo de energía. (Verle, 2017, p.97)

2.2.3. ENTORNOS DE PROGRAMACIÓN

Los microcontroladores originalmente se programaron solo en lenguaje ensamblador, pero varios lenguajes de programación de alto nivel, como C, Python y JavaScript, ahora también son de uso común para apuntar a microcontroladores y sistemas integrados. Estos lenguajes están diseñados especialmente para cualquiera de los fines, o versiones de lenguajes de propósito general como el lenguaje de programación C. Compiladores para los lenguajes de propósito general, normalmente tendrá algunas restricciones, así como también mejoras para soportar mejor las características únicas de los microcontroladores. Algunos microcontroladores tienen entornos para ayudar a desarrollar ciertos tipos de aplicaciones. Los proveedores de microcontroladores a menudo hacen que las herramientas estén disponibles gratuitamente para facilitar la adopción de su hardware.



Muchos microcontroladores son tan raros que requieren efectivamente sus propios dialectos no estándar de C, como SDCC para el 8051, lo que impide utilizar herramientas estándar (como bibliotecas de códigos o herramientas de análisis estáticas) incluso para códigos no relacionados con las características del hardware. A menudo, los intérpretes se utilizan para esconder esas peculiaridades de bajo nivel.

El firmware del intérprete también está disponible para algunos microcontroladores. Por ejemplo, BASIC en los primeros microcontroladores Intel 8052; BASIC and FORTH en el Zilog Z8, así como algunos dispositivos modernos. Por lo general, estos intérpretes son compatibles con la programación interactiva.

Los simuladores están disponibles para algunos microcontroladores. Estos permiten que un desarrollador analice cuál debería ser el comportamiento del microcontrolador y su programa si estuvieran usando la parte real. Un simulador mostrará el estado del procesador interno y también el de las salidas, además de permitir la generación de señales de entrada. Si bien, por un lado, la mayoría de los simuladores estarán limitados por la imposibilidad de simular mucho otro hardware en un sistema, pueden ejercer condiciones que, de lo contrario, podrían ser difíciles de reproducir a voluntad en la implementación física y pueden ser la forma más rápida de depurar y analizar problemas.

Los microcontroladores recientes a menudo se integran con los circuitos de depuración en un chip que, cuando se accede mediante un emulador en circuito (ICE) a través de JTAG, permiten depurar el firmware con un depurador. Un ICE en tiempo real puede permitir la visualización y/o manipulación de estados internos mientras se ejecuta. Un ICE de rastreo puede registrar un programa ejecutado y estados de MCU antes / después de un punto de activación. (Verle, 2017, p.180)



2.2.4. ARDUINO

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activar un motor, encender un LED y publicar algo en línea. Puede decirle a su tablero qué hacer enviando un conjunto de instrucciones al microcontrolador en el tablero. Para hacerlo, usa el lenguaje de programación Arduino (basado en Cableado) y el Software Arduino (IDE), basado en Procesamiento.

A lo largo de los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Una comunidad mundial de fabricantes, estudiantes, aficionados, artistas, programadores y profesionales, se ha reunido en torno a esta plataforma de código abierto, sus contribuciones han agregado una cantidad increíble de conocimiento accesible que puede ser de gran ayuda para novatos y expertos por igual.

Arduino nació en el Ivrea Interaction Design Institute como una herramienta fácil para prototipos rápidos, dirigido a estudiantes sin experiencia en electrónica y programación. Tan pronto como llegó a una comunidad más amplia, la placa Arduino comenzó a cambiar para adaptarse a las nuevas necesidades y desafíos, diferenciando su oferta de tableros simples de 8 bits a productos para aplicaciones IoT, portátiles, impresión 3D y entornos integrados. Todas las placas Arduino son completamente de código abierto, lo que permite a los usuarios construirlas de forma independiente y eventualmente adaptarlas a sus necesidades particulares. El software también es de código abierto y está creciendo gracias a las contribuciones de los usuarios de todo el mundo. (Arduino, 2017)

2.2.4.1. POR QUÉ ARDUINO

Gracias a su experiencia de usuario simple y accesible, Arduino se ha utilizado en miles de proyectos y aplicaciones diferentes. El software Arduino es fácil de usar para principiantes, pero lo suficientemente flexible para usuarios avanzados. Funciona en Mac, Windows y Linux. Los maestros y los estudiantes lo usan para construir instrumentos científicos de bajo costo, para probar los principios de la química y la física, o para empezar a usar la programación y la robótica. Diseñadores y arquitectos construyen prototipos interactivos, músicos y artistas lo usan para instalaciones y para experimentar con nuevos instrumentos musicales. Los fabricantes, por supuesto, lo utilizan para construir muchos de los proyectos exhibidos en el Maker Faire, por ejemplo. Arduino es una herramienta clave para aprender cosas nuevas. Cualquier persona (niños, hobbistas, artistas, programadores) puede comenzar a jugar simplemente siguiendo las instrucciones paso a paso de un kit.

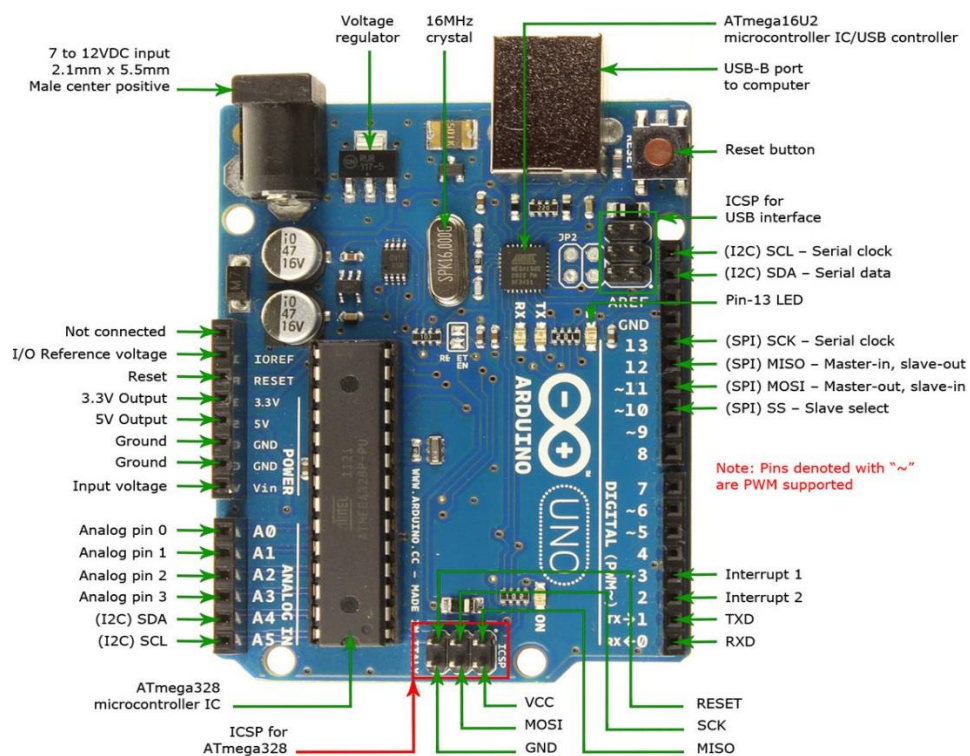


Figura N° 6: Diagrama de la placa de arduino UNO
Fuente: Arduino (2017)



Existen muchos otros microcontroladores y plataformas de microcontroladores disponibles para la informática física. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, el Handyboard de MIT y muchos otros ofrecen una funcionalidad similar. Todas estas herramientas toman los detalles desordenados de la programación del microcontrolador y la envuelven en un paquete fácil de usar. Arduino también simplifica el proceso de trabajo con microcontroladores, pero ofrece cierta ventaja para los profesores, estudiantes y aficionados interesados sobre otros sistemas:

- **Barato:** Las placas Arduino son relativamente baratas en comparación con otras plataformas de microcontroladores. La versión menos costosa del módulo Arduino se puede ensamblar manualmente, e incluso los módulos Arduino premontados cuestan menos de \$ 50
- **Multiplataforma:** El software de Arduino (IDE) se ejecuta en Windows, Macintosh OS X, y Linux. La mayoría de los sistemas de microcontroladores están limitados a Windows.
- **Entorno de programación simple y clara:** el software Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible para que los usuarios avanzados también lo puedan aprovechar. Para los profesores, está convenientemente basado en el entorno de programación de Procesamiento, por lo que los estudiantes que aprenden a programar en ese entorno estarán familiarizados con cómo funciona el Arduino IDE.
- **Software de código abierto y extensible:** El software Arduino se publica como herramientas de código abierto, disponibles para su extensión por programadores experimentados. El lenguaje se puede expandir a través de las bibliotecas de C++, y las personas que quieran comprender los detalles técnicos pueden hacer el



salto de Arduino al lenguaje de programación AVR C en el que se basa. De manera similar, puede agregar el código AVR-C directamente en sus programas Arduino si lo desea.

- **Hardware de código abierto y extensible:** los planos de los paneles de Arduino se publican bajo una licencia de Creative Commons, por lo que los diseñadores experimentados de circuitos pueden crear su propia versión del módulo, ampliándola y mejorándola. Incluso los usuarios relativamente inexpertos pueden construir la versión del módulo para comprender cómo funciona y ahorrar dinero. (Arduino, 2017)

2.2.3.SENSORES

Un sensor es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos aparatos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas. Por ejemplo: existen sensores que se instalan en los vehículos y que detectan cuando la velocidad de desplazamiento supera la permitida; en esos casos, emiten un sonido que alerta al conductor y a los pasajeros.

Otro tipo de sensor muy habitual es aquel que se instala en la puerta de entrada de las viviendas y reacciona ante el movimiento. Si una persona se acerca al sensor, éste emite una señal y se enciende una lámpara. La utilización de estos sensores está vinculada a la seguridad, ya que evitan que alguien aproveche la oscuridad para ocultarse e ingresar en la casa sin ser advertido. El termómetro también es un tipo de sensor que aprovecha la capacidad del mercurio para reaccionar ante la temperatura y, de este modo, permite detectar si una persona tiene fiebre.

Los sensores, en definitiva, son artefactos que permiten obtener información del entorno e interactuar con ella. Así como los seres humanos apelan a su sistema sensorial



para dicha tarea, las máquinas y los robots requieren de sensores para la interacción con el medio en el que se encuentran.

Cuando se desarrollan computadoras capaces de responder a órdenes de voz, por ejemplo, se las provee de micrófonos, que son sensores capaces de captar las ondas sonoras y transformarlas. Si estos sensores están conectados con otros circuitos, la máquina podrá reaccionar al estímulo de acuerdo a lo requerido por el usuario. (Komninos *et al.* 2013)

2.2.3.1. CARACTERÍSTICAS DE UN SENSOR

Muchas de las características de los sensores dependen de la variable a medir, pero otras son comunes a todos los sensores. Algunos de los aspectos a tener en cuenta en el momento de seleccionar un sensor son los siguientes:

- *Exactitud*: especifica la diferencia entre el valor medido y el valor real de la variable que se está midiendo.
- *Conformidad o repetitividad*: el grado con que mediciones sucesivas difieren unas de las otras.
- *Resolución*: Es el cambio más pequeño que se puede medir.
- *Precisión*: Se compone de las características de conformidad y resolución.
- *Sensibilidad*: viene dado por el mínimo valor de la variable medida que produce un cambio en la salida.
- *Error*: es la desviación entre valor verdadero y valor medido.
- *Linealidad*: nos indica que tan cerca está la correlación entre la entrada y la salida a una línea recta.



- *Rango*: es la diferencia entre el mayor valor y el menor valor que se puede medir.
- *Rapidez de respuesta*: capacidad del instrumento de seguir las variaciones de la entrada.

Existen otros aspectos a toma en consideración, por ejemplo, la alimentación del sensor, el tipo de salida, el tipo de conexión (a dos hilos o a tres hilos) de la salida. Los sensores de proximidad, que es uno de los más usados, son de varios tipos: ópticos, inductivos, capacitivos, ultrasónicos y finales de carrera. Estos aspectos serán tratados más adelante en otras entradas.

Un sensor es un tipo de transductor que transforma la magnitud que se quiere medir o controlar, en otra, que facilita su medida. Pueden ser de indicación directa (e.g. un termómetro de mercurio) o pueden estar conectados a un indicador (posiblemente a través de un convertidor analógico a digital, un computador y un visualizador) de modo que los valores detectados puedan ser leídos por un humano. Por lo general, la señal de salida de estos sensores no es apta para su lectura directa y a veces tampoco para su procesado, por lo que se usa un circuito de acondicionamiento, por ejemplo un puente de Wheatstone, amplificadores y filtros electrónicos que adaptan la señal a los niveles apropiados para el resto de los circuitos. (Schaffers *et al.*, 2013, p.130)

2.2.3.2. TIPOS DE SENSORES

En la siguiente tabla se indican algunos tipos y ejemplos de sensores electrónicos.

Tabla N° 2: *Tipos de sensores*

Magnitud	Transductor	Característica
Posición lineal y angular	Potenciómetro	Analógica
	Encoder	Digital
	Sensor Hall	Digital
Desplazamiento y deformación	Galga extensiométrica	Analógica
	Magnetostrictivos	A/D
	Magnetorresistivos	Analógica
	LVDT	Analógica
Velocidad lineal y angular	Dinamo tacométrica	Analógica
	Encoder	Digital
	Detector inductivo	Digital
	Servo-inclinómetros	A/D
	RVDT	Analógica
	Giróscopo	
Aceleración	Acelerómetro	Analógico
	Servo-accelerómetros	
Fuerza y par (deformación)	Galga extensiométrica	Analógico
	Triaxiales	A/D
Presión	Membranas	Analógica
	Piezoeléctricos	Analógica
	Manómetros Digitales	Digital
Caudal	Turbina	Analógica
	Magnético	Analógica
Temperatura	Termopar	Analógica
	RTD	Analógica
	Termistor NTC	Analógica
	Termistor PTC	Analógica
	Bimetal - Termostato	I/O
Sensores de presencia	Inductivos	I/O
	Capacitivos	I/O
	Ópticos	I/O y Analógica
Sensores táctiles	Matriz de contactos	I/O



	Piel artificial	Analógica
Visión artificial	Cámaras de video	Procesamiento digital
	Cámaras CCD o CMOS	Procesamiento digital
Sensor de proximidad	Sensor final de carrera	
	Sensor capacitivo	Analógica
	Sensor inductivo	Analógica
	Sensor fotoeléctrico	Analógica
Sensor acústico (presión sonora)	micrófono	Analógica
Sensores de acidez	ISFET	
Sensor de luz	fotodiodo	Analógica
	Fotorresistencia	Analógica
	Fototransistor	Analógica
	Célula fotoeléctrica	Analógica
Sensores captura de movimiento	Sensores inerciales	

Fuente: Schaffers *et al.* (2013)

Algunas magnitudes pueden calcularse mediante la medición y cálculo de otras, por ejemplo, la velocidad de un móvil puede calcularse a partir de la integración numérica de su aceleración. La masa de un objeto puede conocerse mediante la fuerza gravitatoria que se ejerce sobre él en comparación con la fuerza gravitatoria ejercida sobre un objeto de masa conocida (patrón). (Komninos *et al.*, 2013)

2.2.3.3.FOTORRESISTENCIA

Es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Puede también ser llamado concha de day coronel fotorresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz, cuyas siglas, LDR, se originan de su nombre en inglés light-dependent resistor. Su cuerpo está formado por una célula o celda y dos patillas.

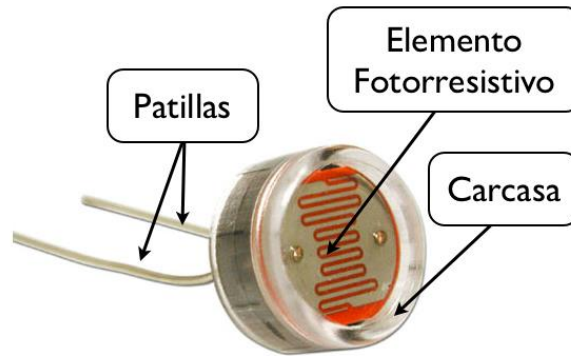


Figura N° 7: Partes de la fotorresistencia
Fuente EcuRed.co (2017)

Un fotorresistor está hecho de un semiconductor de alta resistencia como el sulfuro de cadmio, CdS. Si la luz que incide en el dispositivo es de alta frecuencia, los fotones son absorbidos por la elasticidad del semiconductor dando a los electrones la suficiente energía para saltar la banda de conducción. El electrón libre que resulta, y su hueco asociado, conducen la electricidad, de tal modo que disminuye la resistencia. Los valores del enlace típico varían entre $1\text{ M}\Omega$, o más, en la oscuridad y $100\ \Omega$ con luz brillante. Las células de sulfuro del cadmio se basan en la capacidad del cadmio de variar su resistencia según la cantidad de luz que incide la célula. Cuanta más luz incide, más baja es la resistencia. Las células son también capaces de reaccionar a una amplia gama de frecuencias, incluyendo infrarrojo (IR), luz visible, y ultravioleta (UV). La variación del valor de la resistencia tiene cierto retardo, diferente si se pasa de oscuro a iluminado o de iluminado a oscuro. Esto limita a no usar los LDR en aplicaciones en las que la señal luminosa varía con rapidez. El tiempo de respuesta típico de un LDR está en el orden de una décima de segundo. Esta lentitud da ventaja en algunas aplicaciones, ya que se filtran variaciones rápidas de iluminación que podrían hacer inestable un sensor (ej. tubo fluorescente alimentado por corriente alterna). En otras aplicaciones (saber si es de día o es de noche) la lentitud de la detección no es importante. Se fabrican en diversos tipos y



pueden encontrarse en muchos artículos de consumo, como por ejemplo en cámaras, medidores de luz, relojes con radio, alarmas de seguridad o sistemas de encendido y apagado del alumbrado de calles. (EcuRed, 2017).

2.2.3.4.SENSOR DE MOVIMIENTO

También llamado sensor de presencia, es un dispositivo electrónico equipado de sensores que responden a un movimiento físico. Se encuentran generalmente en sistemas de seguridad o en circuitos cerrados de televisión.

El sistema puede estar compuesto, simplemente, por una cámara de vigilancia conectada a un ordenador que se encarga de generar una señal de alarma o poner el sistema en estado de alerta cuando algo se mueve delante de la cámara. Aunque, para mejorar el sistema se suele utilizar más de una cámara, multiplexores y grabadores digitales. Además, se maximiza el espacio de grabación, grabando solamente cuando se detecta movimiento.

Existen diferentes aplicaciones para un sensor de movimiento: seguridad, entretenimiento, iluminación, comodidad. Por ejemplo, en las tiendas se tienen sensores que detectan cuando una persona va a entrar y se abren las puertas automáticamente.

También hay varios tipos de sensores:

- *Sensores activos.* Este tipo de sensores inyectan luz, microondas o sonido en el medio ambiente y detectan si existe algún cambio en él.
- *Sensores pasivos.* Muchas alarmas y sensores utilizados usan la detección de ondas infrarrojas. Estos sensores son conocidos como PIR (pasivos infrarrojos). Para que uno de estos sensores detecte a los seres humanos se debe de ajustar la sensibilidad del sensor para que detecte la temperatura del cuerpo humano.



- *Acelerómetro.* El acelerómetro es uno de los transductores más versátiles, siendo el más común el piezoeléctrico por compresión. Este se basa en el principio de que cuando se comprime un retículo cristalino piezoeléctrico, se produce una carga eléctrica proporcional a la fuerza aplicada.
- *Giroscopio mecánico.* Es un dispositivo para medir orientación o mantenerla. Consiste en un disco giratorio que puede tomar cualquier orientación, la cual cambia por las fuerzas externas causadas por el movimiento. El primero fue construido en el año 1810 en Alemania por Bohnenberg y en 1852 el físico francés Leon Foucault demostró que un giroscopio puede detectar la rotación de la tierra. (Barlow, 2016, p.123)

2.2.4. ACTUADORES

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control, como por ejemplo una válvula. Son los elementos que influyen directamente en la señal de salida del automatismo, modificando su magnitud según las instrucciones que reciben de la unidad de control. Existen varios tipos de actuadores como son:

- Electrónicos
- Eléctricos

Los actuadores hidráulicos, neumáticos y eléctricos son usados para manejar aparatos mecatrónicos. Por lo general, los actuadores hidráulicos se emplean cuando lo que se necesita es potencia, y los neumáticos son simples posicionamientos. Sin embargo, los hidráulicos requieren mucho equipo para suministro de energía, así como de



mantenimiento periódico. Por otro lado, las aplicaciones de los modelos neumáticos también son limitadas desde el punto de vista de precisión y mantenimiento. (AUMA, 2017, p.5).



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 MATERIALES

3.1.1 Hardware

- Ordenador para programación (laptop)

Modelo: HP 250

Procesador: Intel(R) Core(TM) i5-2430 2.40GHz

Memoria instalada (RAM): 4.00GB de RAM.

Adaptador de red 8002.11n Bradcom.

Tipo de sistema: Sistema Operativo de 64 bits Windows 10

- SmartPhone para cliente.

Modelo: Sony Xperia Z3

Procesador: Qualcomm Snapdragon 801 Quad-core a 2.5 GHz

Memoria: 4.00GB de RAM.

Tipo de sistema: Android 6.0.1

- Arduino UNO R3

Microcontrolador ATmega328.

Voltaje de entrada 7-12V.

14 pines digitales de I/O (6 salidas PWM).

6 entradas análogas.

32k de memoria Flash.

Reloj de 16MHz de velocidad. Servidor (computadora)

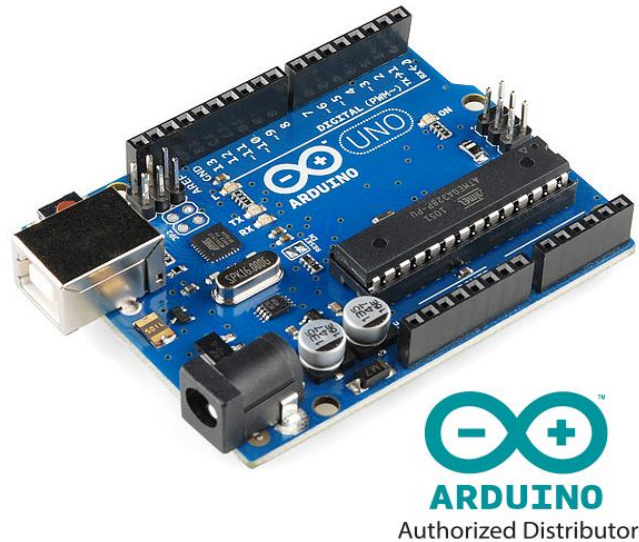


Figura N° 8: Arduino UNO R3.
Fuente: Arduino (2017)

- Arduino Ethernet Shield 2

Voltaje de funcionamiento 5V (suministrado por la placa Arduino)

Controlador Ethernet: W5500 con buffer interno 32K

Velocidad de conexión: 10 / 100Mb

Conexión con Arduino en el puerto SPI

Cumple con IEEE802.3af



Figura N° 9: Arduino Ethernet Shield 2
Fuente: Arduino (2017)

- Potenciómetros
- Resistencias
- Luminarias LED (para prototipo)
- Pulsadores (para prototipo)
- Fotorresistencia (LDR)
- Sensor de presencia PIR.

3.1.2 Software

- Sistema Operativo de 64 bits Windows 10.
- Software para la compilación Arduino IDE.
- Navegador Google Chrome.
- Navegador Google Chrome Móvil.



3.2 MÉTODO

3.2.1 Diseño de la investigación

El método deductivo es el que se usa en esta investigación, que consiste en analizar la totalidad de las reglas y procesos, con cuya ayuda es posible deducir conclusiones finales a partir de unos enunciados supuestos llamados premisas. Se utiliza también un diseño de investigación para resolver problemas cotidianos e inmediatos; también envuelve la transformación y mejora de una realidad (social, educativa, administrativa, etc.). De hecho, se construye de ésta.

3.2.2 Nivel de la investigación

El nivel de investigación se refiere a la profundidad del conocimiento que se busca lograr con la investigación, por tanto, el nivel de la presente investigación es exploratoria, señalando que las investigaciones exploratorias buscan abrir nuevos caminos en el desarrollo del conocimiento humano. Y la presente investigación siendo un prototipo busca abrir un camino para un nuevo método de monitoreo y control de alumbrados públicos de la Universidad Nacional Del Altiplano Puno, mediante la arquitectura IoT.

3.2.3 Población y muestra de la investigación

3.2.3.1. Población:

Siendo la población el conjunto de las entidades, o cosas respecto a las cuales se basa las conclusiones de una investigación, para nuestro caso la población está definida por todos los componentes del alumbrado público de la ciudad universitaria de la Universidad Nacional del Altiplano Puno, y consta de 98 unidades para esta investigación.

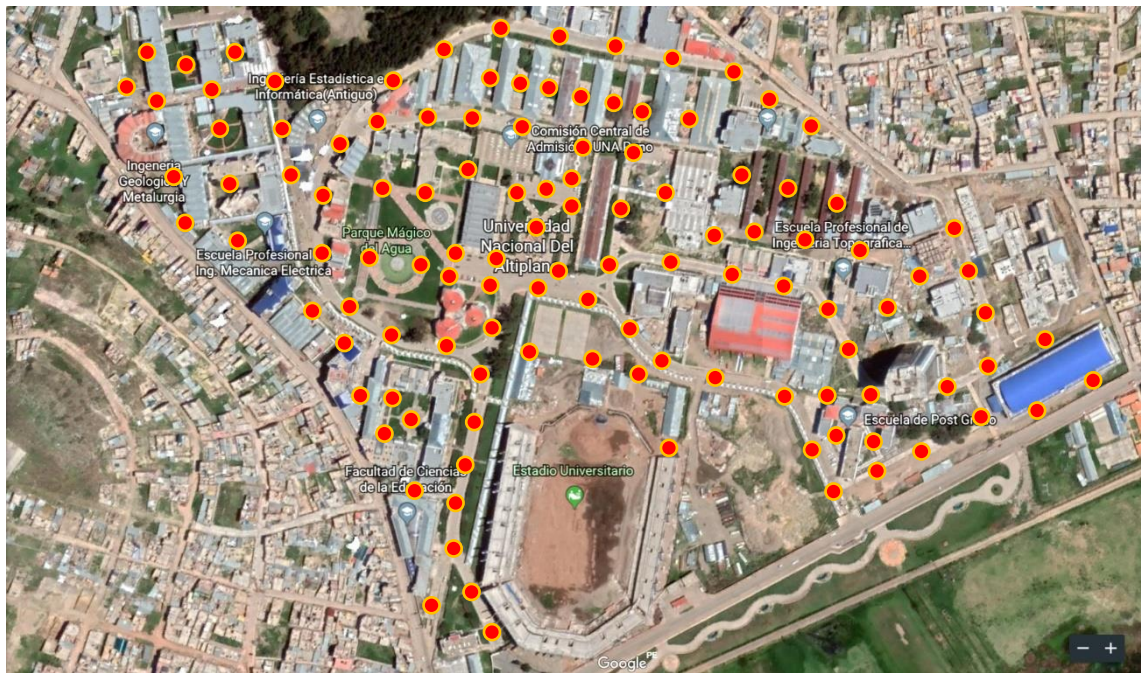


Figura N° 10: Unidades de alumbrado en la ciudad universitaria.
Elaboración propia.

3.2.3.4 Muestra

Se define la muestra como parte que se estudia y es representativa de la población, es decir un segmento que tiene las características y propiedades de la población. El universo finito o población está dado por el número total de unidades, es decir 98. Por lo tanto, para hallar la muestra con un error del 5% (0.05), con un nivel de confianza del 90%, una probabilidad de ocurrencia o éxito de 0.9 y una probabilidad de no ocurrencia o fracaso de 0.1; se procede a hallar la muestra con la fórmula para calcular el tamaño de muestra cuando se conoce el tamaño de la población y es la siguiente:

$$n = \frac{N \times Z_a^2 \times p \times q}{d^2 \times (N - 1) + Z_a^2 \times p \times q}$$

En donde:

N = tamaño de la población

Z_a = nivel de confianza



p = probabilidad de éxito o proporción esperada

q = probabilidad de fracaso

d = precisión (Error máximo admisible en términos de proporción).

Si tenemos los valores $N=98$, $Z_a=1.645$ (nivel de confianza 90%), $p=0.9$, $q=0.1$ y $d=0.05$. La muestra resulta con el valor de 49.11. Es decir que sería necesario realizar la investigación por lo menos en 50 unidades una vez implementado el sistema y obtener los resultados deseados. Sin embargo, se considerará un muestreo no probabilístico al tener todas las unidades las mismas características generales en cuanto al hardware y en cuanto a la forma de gestión, que hace que se tome como caso estudio el prototipo desarrollado. La muestra no servirá para hacer generalizaciones, pero sí para el estudio exploratorio, se ha elegido a los individuos utilizando diferentes criterios relacionados con las características de la investigación y está determinado por el autor.

3.2.4 Ubicación y descripción de la investigación

3.2.4.1 Ubicación

La investigación se desarrollará en la ciudad universitaria y en el laboratorio de Electrónica General de la Escuela Profesional de Ingeniería Electrónica de la Universidad Nacional del Altiplano.

3.2.4.2 Descripción de la investigación

La presente investigación demostrará la implementación y diseño de un sistema de un sistema con IoT para la gestión de alumbrado público en la Universidad Nacional Del Altiplano Puno, esta investigación consta de tres fases que se describen a continuación.

- **Fase 1:** Esta fase trata esencialmente búsqueda de información, consulta bibliográfica y sitios web, sobre plataformas de desarrollo, elección de

componentes y técnicas o métodos implementación del sistema de gestión con arquitectura IoT.

- **Fase2:** Esta fase comprende el diseño del sistema en la parte física y pruebas preliminares.
- **Fase2:** ya realizado el diseño físico, esta fase comprende de la programación del Arduino y pruebas finales.
- **Fase 3:** Comprende la obtención y el proceso de verificación, que consiste en correlacionar los resultados, para luego analizar los resultados, ya que es necesario para ver el buen funcionamiento y su posterior corrección.

3.2.5 Técnicas e instrumentos de recolección de datos

La recolección de datos se refiere a cómo y qué medios se usan para la obtención de la información que será de utilidad para la corroboración de nuestras hipótesis, por lo tanto, resumimos este apartado de la siguiente forma:

Tabla 3: *Técnicas e instrumentos para recolección de datos*

Técnicas	Instrumentos
Consultas bibliográficas y de bases de datos.	Papers, foros, blogs, libros, video-tutoriales y más fuentes de información publicadas en Internet
Observación de pruebas finales	Capturas a la pantalla, navegador web, navegador web móvil, software de Arduino y fotos.
Elaboración propia	

3.2.6 Procedimiento de diseño del prototipo

3.2.6.1 Diseño de la maqueta

Se desarrollo una maqueta en la cual se diseña el sistema de gestion. La maqueta muestra de alguna manera todo los procesos que realizara el sistema. El diseño de la maqueta cuenta con el Arduino UNO R3, junto al Shield de Ethernet, ademas led que simularan a las luminarias en los alumbrados publicos, pulsadores que tambien simularan las señales enviadas desde el sensor de movimiento, lo mas importante cuenta con sensores de luminosidad (fotorresistencia) y finalmente tambien cuenta con resistencias variables; las funciones se estaran explicando posteriormente. Esta maqueta esta realizada sobre un protoboard, para ser visualizado de mejor manera.

Para iniciar con la implementacion se juntaron el Arduino UNO encima el Shield de Ethernet esto ya que es la forma de coneccion predeterminada y el Shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.. La figura N° 11 se visualiza una vista de encima del arduino.

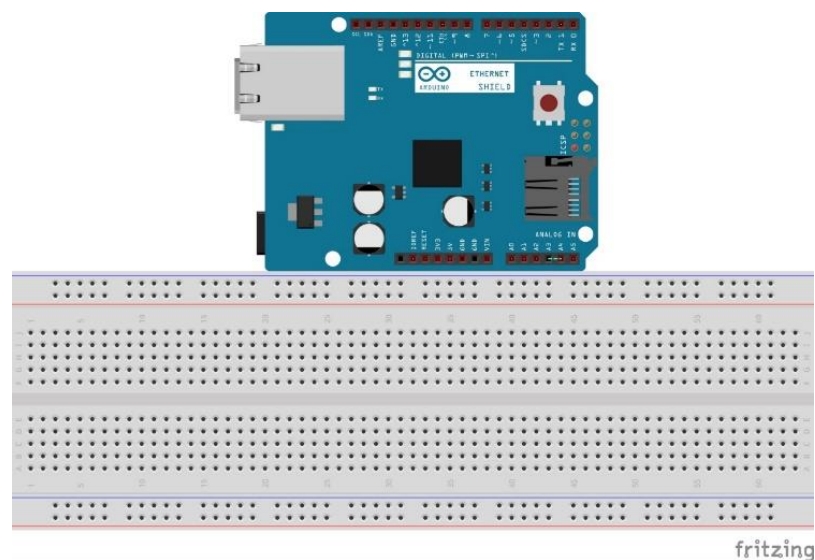


Figura N° 11: conexión grafica del Arduino UNO con el Shield de Ethernet
Elaboración propia

En la figura N° 12 esta una vista de las conexiones en la forma esquemática de ambos componentes, en esta figura se muestra la conexión que hay al realizar el empalme entre ambos componentes.

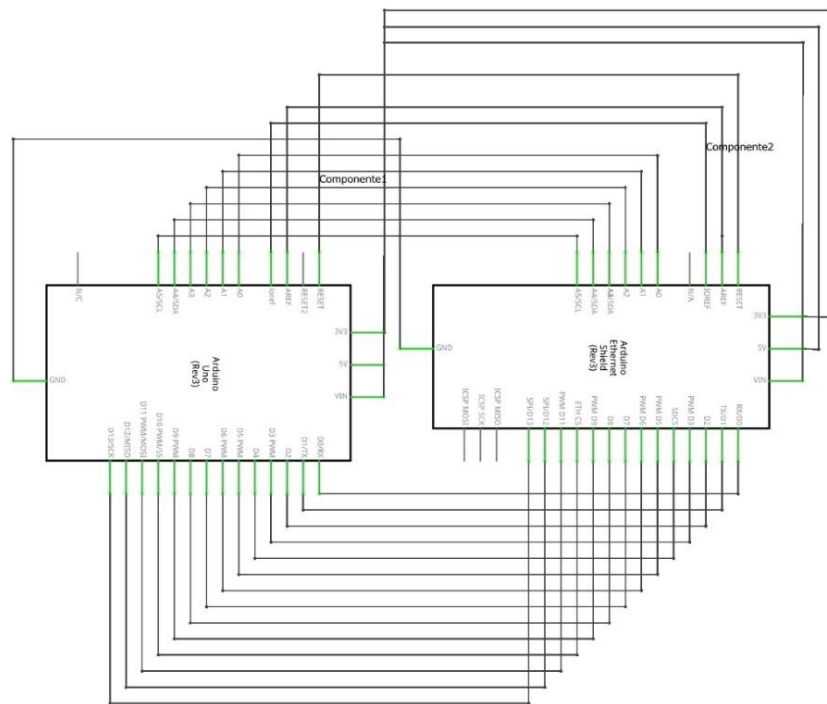


Figura N° 12: Conexión esquemática del Arduino UNO con el Shield de Ethernet
Elaboración propia

El Arduino Ethernet Shield nos da la capacidad de conectar un Arduino a una red Ethernet. Es la parte física que implementa la pila de protocolos TCP/IP. Está basada en el chip Ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de soportar TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para leer y escribir los flujos de datos que pasan por el puerto Ethernet.

El shield provee un conector ethernet estándar RJ45. Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la Ethernet Shield. Estos pines no pueden ser usados para e/s genéricas. También se toma las siguientes indicaciones:



- El botón de reset en la Shield resetea ambos, el W5100 y la placa Arduino.
- La Shield contiene varios LEDs para información:
- ON: indica que la placa y la Shield están alimentadas
- LINK: indica la presencia de un enlace de red y parpadea cuando la Shield envía o recibe datos
- 100M: indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- RX: parpadea cuando el Shield recibe datos
- TX: parpadea cuando el Shield envía datos

El jumper soldado marcado como “INT” puede ser conectado para permitir a la placa Arduino recibir notificaciones de eventos por interrupción desde el W5100, pero esto no está soportado por la librería Ethernet. El jumper conecta el pin INT del W5100 al pin digital 2 de Arduino. Para usar la Ethernet Shield solo hay que montarla sobre la placa Arduino. Para cargar los sketches a la placa con el Shield, conectarla al ordenador mediante el cable USB como se hace normalmente. Luego conectar la Ethernet a un ordenador, a un switch o a un router utilizando un cable Ethernet standard (CAT5 o CAT6 con conectores RJ45). La conexión al ordenador puede requerir el uso de un cable cruzado (aunque muchos ordenadores actuales, pueden hacer el cruce de forma interna).

Puntos a recordar del Ethernet Shield:

- Opera a 5V suministrados desde la placa de Arduino
- El controlador Ethernet es el W5100 con 16K de buffer interno. No consume memoria.



- El Shield se comunica con el microcontrolador por el bus SPI, por lo tanto, para usarlo siempre debemos incluir la librería SPI.h:
<http://arduino.cc/en/Reference/SPI>
- Soporta hasta 4 conexiones simultáneas
- Usar la librería Ethernet para manejar el Shield:
<http://arduino.cc/en/Reference/Ethernet>
- El Shield dispone de un lector de tarjetas micro-SD que puede ser usado para guardar ficheros y servirlos sobre la red. Para ello es necesaria la librería SD:
<http://arduino.cc/en/Reference/SD>
- Al trabajar con la SD, el pin 4 es usado como SS.

Arduino UNO se comunica con W5100 y la tarjeta SD usando el bus SPI a través del conector ICSP. Por este motivo los pines 10, 11, 12 y 13 en el UNO. Los pines 10 y 4 se usan para seleccionar el W5100 y la tarjeta SD. El Ethernet y el SD no pueden trabajar simultáneamente y debemos tener cuidado al usar ambos de forma conjunta, pero para este diseño no es usado el slot de tarjeta SD.

Siguiendo la implementación se realizó con las conexiones de los componentes; la figura 13 muestra las conexiones de forma gráfica realizadas sobre un protoboard.

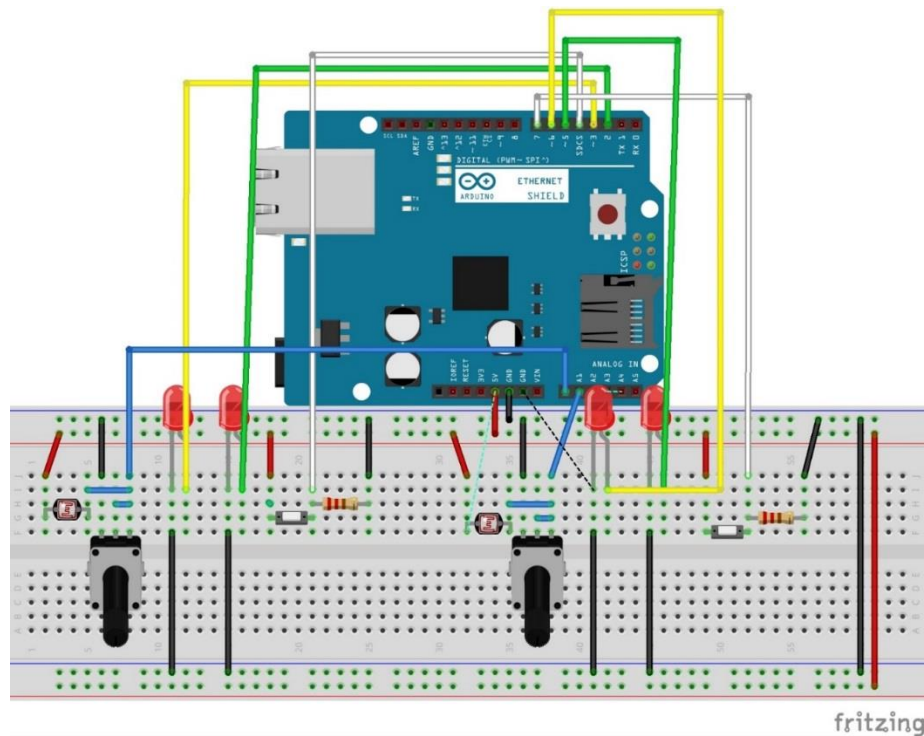


Figura N° 13: diagrama grafica de las conexiones de componentes
Elaboración propia

El diagrama esquemático está en la figura 14 donde se indica el nombre de cada componente junto a su valor la descripción o el motivo de la conexión de cada componente se realizará posteriormente; excepto del Arduino y el Shield.

Las funciones de cada componente son:

- **SENSOR DE MOVIMIENTO1:** Este pulsador simulara los datos que enviara el sensor de movimiento; es decir que cuando el sensor enví un 0 lógico indica que no se detectó movimiento y si es enviado un 1 lógico indicara que se produjo algún movimiento cerca al alumbrado. La conexión de un extremo va hacia la alimentación positiva y la pata sobrante es conectada junto a la resistencia R1, también al pin para entrada digital D4; este tipo de conexión ayuda para que se emita pulsos lógicos hacia el Arduino.

- FOTORRESISTENCIA 1: La fotorresistencia ayudará para enviar datos analógicos al Arduino para ver la cantidad de iluminación existente sobre el área del alumbrado, gracias a esto se determinará si encender las luminarias o mantenerlas apagadas. La forma de conexión mostrada en la figura 14 indica la conexión hacia la alimentación de positiva mientras el extremo sobrante es conectado a la entrada analógica A0 y a la resistencia variable POT1 realizando un puente en un lado de la resistencia junto a la pata del medio.
- R1: La resistencia se conoce como resistencia *pull-down*. Pull en inglés es estirar. Ahora estamos en el caso en que la resistencia estira hacia abajo, de esta manera lo que sucede es que la tensión que llega al pin es la que viene de GND, ya que la entrada de 5v está a la otra parte del interruptor y aún no es pulsado. Cuando pulsamos, lo que ocurre es que la corriente es muy vaga y busca el camino fácil. El camino fácil es que busque a 5V, ya que por el otro camino está la resistencia; mostrado en la figura 14.

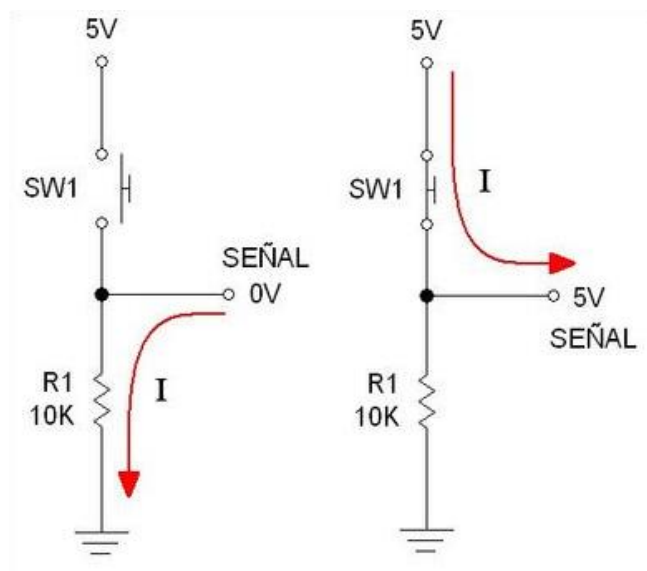


Figura N° 14: Configuración pull-down
Elaboración propia



- POT1: este componente ayuda para realizar un divisor de tensión. Un divisor de tensión es un circuito muy utilizado cuando se quiere regular el voltaje que le llega a un componente, circuito, etc. Está compuesto generalmente por dos resistencias en serie, conectadas entre una fuente de tensión y tierra. Cuando conectas entre ambas resistencias el elemento sobre el que deseas obtener un voltaje específico, basta con que encuentres los valores de ambas resistencias que te proporcionan dicha tensión. La idea en este caso es que una de esas resistencias sea tu LDR. Como se muestra, sin embargo, en función de dónde se coloque el LDR y del valor de la otra resistencia, los resultados variarán.
- Se conecta el LDR con la fuente de voltaje, cuanto más luz incida sobre la fotorresistencia, menor será la diferencia de potencial (caída de voltaje) que se tendrá entre la fuente y el Arduino, con lo que placa de Arduino *leerá* un valor mayor. De la misma forma, cuanto mayor sea la otra resistencia (que en este caso va conectada a tierra) mayor será la caída de voltaje que tendrás entre el Arduino y tierra. Como en tierra (GND) tienes 0V y eso no varía, cuanto mayor sea la resistencia, mayor será el valor de tensión que llegará a la placa. En el caso de la corriente el planteamiento es similar. Si incide una gran cantidad de luz sobre el LDR, su resistencia se reducirá y la corriente que llegará a tu microcontrolador será mayor. Si la otra resistencia (la que está a tierra) *es grande, dificultará el paso de la corriente a través de ella, y esta pasará por tu placa Arduino.*
- LED-LDR1: Este led simula la luminaria que se encenderá cuando la FOTORRESISTENCIA1 indique que hay baja iluminación en el área del alumbrado, esta luminaria es de menor intensidad.

- LED-SENS.MOV.1: Este led al igual que en el anterior simula la luminaria que se encenderá, pero cuando el SENSOR DE MOVIMIENTO 1 mande el 1 lógico. Esta luminaria es de mayor intensidad.

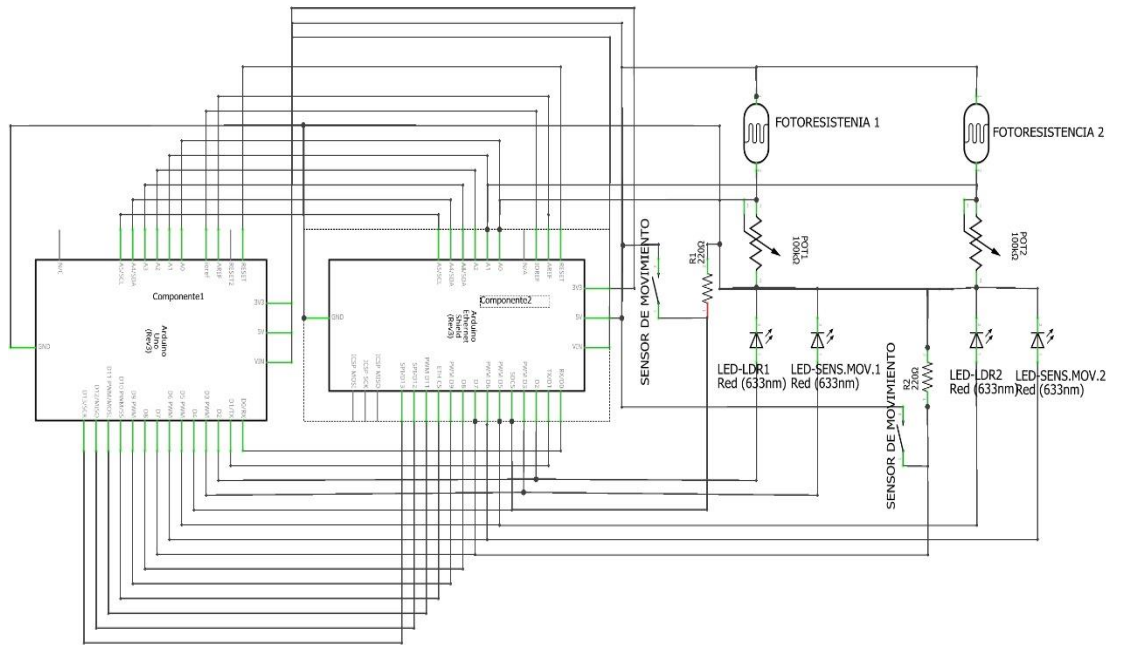


Figura N° 15: diagrama esquemática de las conexiones de componentes
Elaboración propia

- SENSOR DE MOVIMIENTO2: Cumple con la misma función que el primero, la diferencia es que pertenece al segundo alumbrado.
- FOTORRESISTENCIA 2: Es un componente para el segundo alumbrado y su función es la misma que en el primer alumbrado.
- R2: La función se describe en R1, la diferencia es que esta resistencia realiza el pull-down con el pulsador de sensor de movimiento 2
- POT2: Es para la división de tensión con la fotodiodo 2.
- LED-LDR2: La función es idéntica a LED-LDR1, con la diferencia que será activado por la FOTORRESISTENCIA 2.



- LED-SENS.MOV.2: Activado por el segundo sensor de movimiento y funciones idénticas a LED-SENS.MOV.1.

Los componentes como los leds y los pulsadores son indicadores para la verificación de la maqueta. El pulsador indica un sensor de movimiento para lo cual puede ser usado el detector de movimiento ya sea para pruebas o implementación en este caso tenemos el sensor PIR HCSR501 que es un módulo para Arduino, puede ser usado para pruebas o implementar en zonas externas, pero no se garantiza rendimiento en ambientes externos; en otro caso se puede usar el sensor marca Schneider el modelo ARGUS Standard CCTR1P002 diseñado para exteriores, las especificaciones técnicas se detallarán posteriormente. Los LEDs representan a las luminarias para el alumbrado, en esta investigación se usan dos, el primero que se active a falta de luminosidad es de menor luminosidad y el segundo cuando se produzca movimiento por esa área debe ser de alta intensidad. Para el caso del alumbrado de menor intensidad es la luminaria Vaya LED Uplight de la marca Philips y para el caso de alta luminosidad estaría el modelo MileWide2LED también de Philips las especificaciones se muestran a continuación.

HCSR501 PIR

HC-SR501 se basa en tecnología infrarroja, módulo de control automático, con diseño de sonda importada LHI778 de Alemania, alta sensibilidad, alta confiabilidad, modo de operación de ultra bajo voltaje, ampliamente utilizado en diversos equipos eléctricos de autodetección, especialmente para batería automática productos controlados. Las especificaciones para el funcionamiento son las siguientes:

- Voltaje: 5V - 20V
- Consumo de energía: 65mA
- Salida TTL: 3.3V, 0V



- Tiempo de retardo: Ajustable (.3-> 5min)
- Tiempo de bloqueo: 0.2 seg
- Métodos de activación: L - desactiva el disparador de repetición, H habilita el disparador de repetición
- Rango de detección: menos de 120 grados, dentro de los 7 metros
- Las características que cumple este sensor son amplias tales como:
- Inducción automática: para entrar en el rango de detección de la salida es alta, la persona deja el rango de detección de la demora automática de alta, salida baja.
- El control fotosensible (opcional, no configurado de fábrica) puede establecerse como control fotosensible, día o intensidad de luz sin inducción.
- Compensación de temperatura (opcional, restablecimiento de fábrica): en verano, cuando la temperatura ambiente aumenta de 30 ° C a 32 ° C, la distancia de detección es ligeramente más corta, la compensación de temperatura se puede utilizar para la compensación del rendimiento.
- Activado de dos maneras: (puente seleccionable)
- Disparador no repetible: la salida del sensor es alta, el tiempo de retardo ha terminado, la salida cambia automáticamente de nivel alto a nivel bajo;
- Activador repetible: la salida del sensor alta, el período de retardo, si hay actividad humana en su rango de detección, la salida siempre permanecerá alta hasta que la gente se quede después de que el nivel alto sea bajo (el módulo del sensor detecta un período de retraso) se extenderá automáticamente cada actividad humana, y el punto de inicio para el tiempo de demora hasta el último evento del tiempo).

- Con el tiempo de bloqueo de inducción (la configuración predeterminada: 2.5s bloqueó el tiempo): módulo del sensor después de cada salida del sensor (alto a bajo), seguido de un período de bloqueo establecido, durante este período el sensor no acepta ninguna señal del sensor. Esta característica se puede lograr el tiempo de salida del sensor "y el intervalo de" tiempo de bloqueo "entre el trabajo se puede aplicar a los productos de detección de intervalo; Esta función puede inhibir una variedad de interferencias en el proceso de conmutación de carga. (Este tiempo puede establecerse en cero segundos, unas pocas decenas de segundos).
- Amplio rango de tensión de funcionamiento: voltaje predeterminado DC4.5V-20V.
- Consumo de Micropower: corriente estática <50 microamperios, particularmente adecuado para productos de control automático alimentados por batería.
- Señal de salida alta: acoplamiento fácil de alcanzar con los diversos tipos de circuito. (Karvinen, 2014)

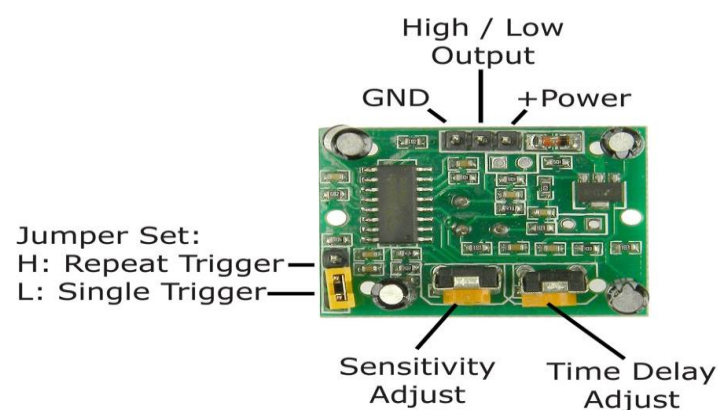


Figura N° 16: Vista frontal del sensor HCSR501 PIR
Fuente: All Electronics (2016)

ARGUS STANDARD CCTR1P002

Es un detector de movimiento estándar de 360° para interiores y exteriores (denominado en lo sucesivo detector de movimiento) que detecta las fuentes de calor móviles en el área de detección (tecnología de infrarrojos pasiva) y, por ejemplo, enciende una lámpara durante la detección de movimiento. Este periodo predefinido se reinicia cada vez que se detecta un movimiento (función de disparo). El detector de movimiento tiene un umbral de luminosidad ajustable, de modo que solamente se encuentra activo cuando la luminosidad ambiente está por debajo de un determinado nivel. (Schneider, 2017, p.1)



Figura N° 17: ARGUS Standard CCTR1P002
Fuente: Schneider (2017)

Las especificaciones técnicas importantes son:

- Corriente máx. de conmutación: 6 A
- Tensión de alimentación: 220-240 V CA, 50 Hz
- Área de detección: 360° / \varnothing 7,90 m

- Umbral de luminosidad: de 5 lx a 2000 lx
- Tipo de protección IP 20

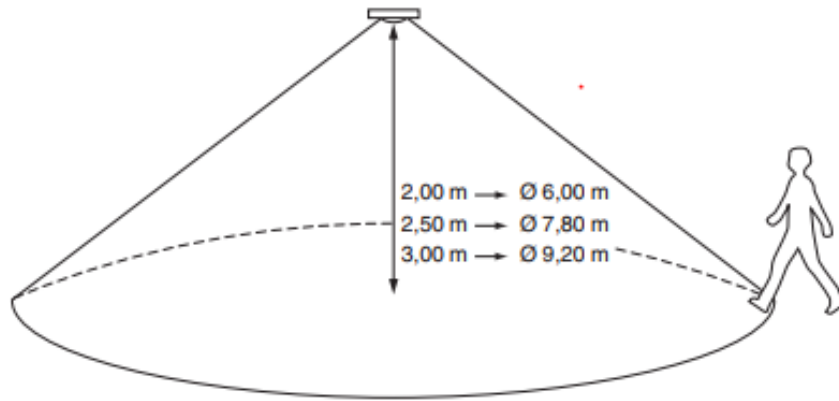


Figura N° 18: Rango de detección
Fuente: Schneider (2017)

VAYA LED UPLIGHT

Los municipios de las ciudades cada vez más necesitan crear un agradable entorno, una imagen urbana atractiva, sin poner en peligro la calidad de la iluminación y a un costo razonable. Vaya LED Uplight es una solución accesible y de gran fiabilidad que minimiza la inversión inicial y ofrece una opción de iluminación atractiva para transformar estos paisajes urbanos, creando diferentes efectos con la posibilidad de color de luz blanca en 3000°K y 4000°K o monocromática de color. Diseño robusto de alta confiabilidad LEDs high-power, de alta calidad lumínica. (Philips, 2017, p.112)

- Tipo: BBP330
- Fuente de luz: Integral Led module
- Potencia: 15,8W
- Flujo luminoso: 140Lm
- Eficacia lumínica: AM 9Lm/W

- Vida útil (L70): 25000hrs a 35°C
- Grados de protección: IP67 / IK10



Figura N° 19: Vaya LED Uplight
Fuente: Philips (2017)

MILEWIDE2LED

Diseño y eficiencia Muchas ciudades se están uniendo a los beneficios de la iluminación exterior contemporánea. Esto incluye un más atractivo entorno urbano y una mayor comodidad, y seguridad para el público. Por no hablar de lo positivo. Philips ha desarrollado MileWide2 y MileWide2 Large, el sucesor del MileWide, su diseño puro y limpio se integra perfectamente con el paisaje urbano de hoy y del mañana. Con su alta calidad de acabados y alto nivel de rendimiento de la iluminación, MileWide2 es adecuado para una amplia gama de aplicaciones, que van desde el tráfico de carreteras a los centros urbanos. La óptica de múltiples capas proporciona una uniforme distribución de luz y brillo reducido, lo que permite directa sustitución de las soluciones HID sin sacrificar el espaciamiento, altura o calidad de luz de montaje. (Philips, 2017, p.65)

- Tipo: BPP435
- Fuente de luz: integral LED module

- Potencia: 14 a 86W (según configuración)
- Flujo lumínico: 1750 a 9000Lm
- Eficacia: hasta 125Lm/W
- Vida útil (L80F10): 100.000hrs
- Alimentación: 220-240V / 50-60Hz
- Altura de montaje entre 6,00 y 8.00mts
- Protección: IP66 / IK08



Figura N° 20: MileWide2LED
Fuente: Philips (2017)

Las salidas de las placas Arduino son perfectamente útiles para controlar cargas que no consuman demasiada corriente, como un Led, pero son insuficientes para cargas mayores. Para realizar la conexión de las luminarias sera necesario conectar un componente en la salida, reemplazando a los led. Este componenete es el modulo de reles de arduino, el esquemático se muestra en la figura 19.

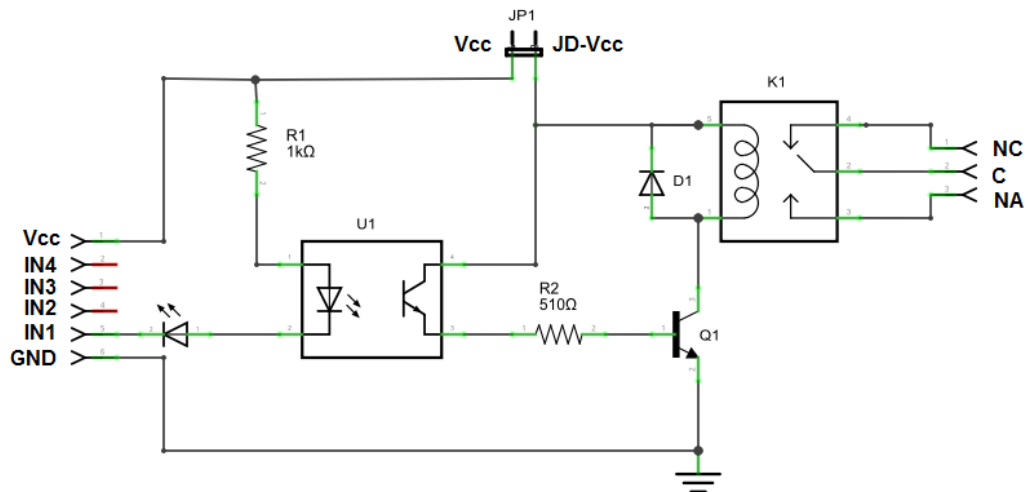


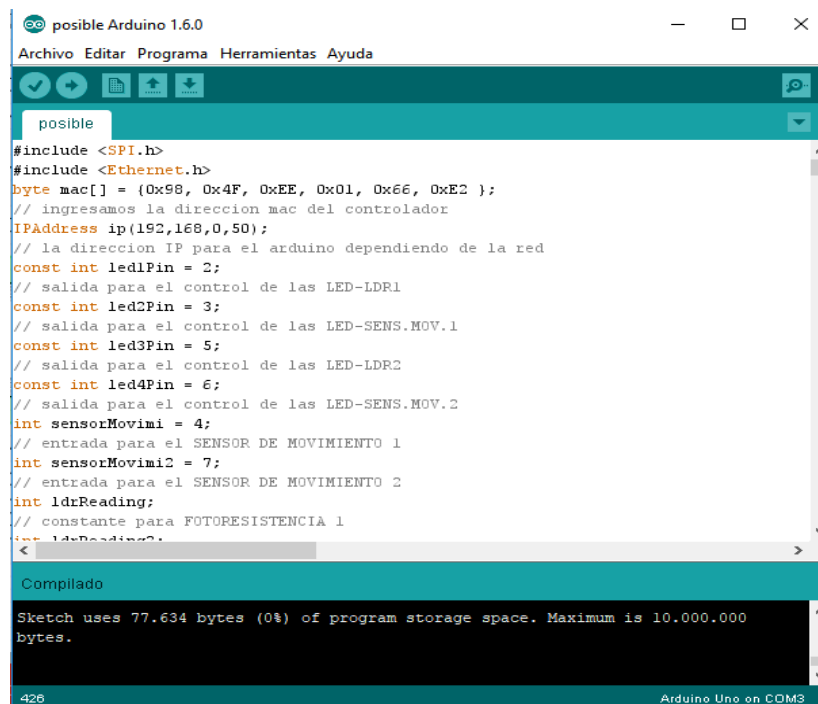
Figura N° 21: Diagrama esquemática del módulo relé
Elaboración propia

El modulo trata de un módulo de 4 relés (o relays) que funcionan a 5 Voltios, capaces de manejar cargas de hasta 10 Amperes en 250 Voltios, convenientemente aislados mediante optoacopladores de las entradas, las que cuentan con leds individuales que sirven como indicadores de estado. la placa tiene un conector de entradas (IN1 a IN4) y alimentación (GND es masa o negativo y Vcc es el positivo), cuatro leds que indican el estado de la entradas, un jumper selector para la alimentación de los relés , cuatro optoacopladores del tipo FL817C, cuatro diodos de protección, cuatro relés marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V [6] y cuatro borneras, con tres contactos cada una (Común, Normal abierto y Normal cerrado), para las salidas de los relés. Se pierde la aislación eléctrica que brindan los optoacopladores, lo que aumenta la posibilidad de daño al Arduino si hay algún problema con las cargas de los relés. La corriente consumida por las bobinas de los relés debe ser provista por la placa Arduino. Cada bobina consume unos 90 mA y las cuatro juntas suman 360 mA. Si a esto le sumamos los consumos que pueden tener otras salidas, estamos muy cerca de los 500 mA que puede suministrar un puerto USB. En este

caso se debería alimentar al Arduino con una fuente externa, lo que aumenta el límite de corriente a 1A (en el caso de la Arduino UNO).

3.2.6.2 Programación de la placa de Arduino

La programación realizada consta de 426 líneas incluyendo las líneas de comentario, el tamaño es de 77,634 bytes. Para cargar la programación al Arduino se conecta al COM3 de la computadora, tal como indica en la figura 20. La dirección IP utilizada es 192.168.0.50, la cual se puede cambiar y el Shield tiene una dirección MAC: 98-4F-EE-01-66-E2.



```
posible Arduino 1.6.0
Archivo Editar Programa Herramientas Ayuda
posible
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {0x98, 0x4F, 0xEE, 0x01, 0x66, 0xE2 };
// ingresamos la direccion mac del controlador
IPAddress ip(192,168,0,50);
// la direccion IP para el arduino dependiendo de la red
const int led1Pin = 2;
// salida para el control de las LED-LDR1
const int led2Pin = 3;
// salida para el control de las LED-SENS.MOV.1
const int led3Pin = 5;
// salida para el control de las LED-LDR2
const int led4Pin = 6;
// salida para el control de las LED-SENS.MOV.2
int sensorMovimi = 4;
// entrada para el SENSOR DE MOVIMIENTO 1
int sensorMovimi2 = 7;
// entrada para el SENSOR DE MOVIMIENTO 2
int ldrReading;
// constante para FOTORESISTENCIA 1
int ldrReading2;

Compilado
Sketch uses 77.634 bytes (0%) of program storage space. Maximum is 10.000.000 bytes.

426 Arduino Uno on COM3
```

Figura N° 22: Programación del controlador Arduino
Elaboración propia

La línea de comandos escrita en el Arduino se muestra en el [ANEXO A](#) donde se explica de forma resumida en los espacios donde inicia el carácter “//”. En esta sección se explicará de manera detallada las acciones que toma el Arduino cuando ya se esté ejecutando la programación. Las acciones son las siguientes:



- LED-LDR1 y LED-LDR2 son encendidos cuando el de las fotorresistencias envían valores analógicos mayores a 2 y su estado será visualizado por el supervisor en el servidor web como “ENCENDIDA”. Si no son encendidos mediante la fotorresistencia hay la posibilidad de ser encendida ingresando a la dirección IP del servidor y pulsando el botón de ON.
- Las fotorresistencias son colocadas en las entradas analógicas y son visualizadas en el servidor web.
- LED-SENS.MOV1 y LED-SENS.MOV2 son activadas para ser encendidas mediante la fotorresistencia. Se enciende cuando el sensor de movimiento manda el 1 lógico; al igual que en LED-LDR es posible encender desde un navegador web e ingresando a la dirección IP del controlador.
- El estado de los sensores de movimiento se muestra cuando se ingresa al servidor web e indican si la calle existe circulación o no.
- Para la visualización en el navegador web se envían códigos de HTML a los clientes conectados, donde incluyen los botones para activar las salidas y visualizar el estado de las entradas.

La gestión de los alumbrados será individual, pero ingresando a un solo servidor web.

3.2.6.3 Implementación del prototipo

Para implementar el prototipo es necesario determinar la medida de la altura de los postes de alumbrado público dentro del campus de la Universidad Nacional del Altiplano. Debido a la gran cantidad de factores que intervienen en la iluminación de lugares públicos (deslumbramiento, características de los pavimentos, condiciones

meteorológicas, etc.) y en la percepción de estas, el cálculo del alumbrado público es una tarea complementaria en esta investigación, sin embargo, no es el objetivo de la investigación. Los cálculos se pueden enfocar a determinar unas condiciones de iluminancia sobre la calzada que proporcionarían una buena visibilidad dentro de los márgenes establecidos por los organismos competentes y también los cálculos se pueden orientar hacia la determinación de luminancias.

3.2.6.4 Determinar la ubicación del sensor PIR

El sensor de presencia (PIR) que usamos para esta investigación es el HC-SR501 y su rango de detección es de menos de 120 grados y dentro de los 7 metros, por eso es necesario determinar la altura del poste de alumbrado público dentro del campus de la universidad.



Figura N° 23: Medición de la altura de un poste de alumbrado.
Elaboración propia.

El valor obtenido de la altura de un poste de alumbrado tomado como muestra dentro del campus de la universidad es de 6.6 metros; este valor es menor a los 7 metros de rango del sensor de presencia, por lo tanto, se determina que el sensor de presencia puede estar ubicado en cualquier parte del poste, pero de preferencia lo más cerca a la luminaria para integrar todo en un solo módulo.

De esta forma se procede a realizar pruebas del funcionamiento del sensor de presencia HC-SR501 para determinar si detecta la presencia de una persona que está caminando y pasa por debajo del sensor, se procede a instalar el sensor en el poste con un pequeño circuito de prueba que indique si el sensor está detectando la presencia de una persona. El resultado fue que a 6.6 metros de altura con respecto al suelo, si puede detectar sin problemas la presencia de una persona.



Figura N° 24: Instalación del sensor de presencia en un poste de alumbrado.
Elaboración propia.

3.2.6.5 Determinar la distancia entre poste y una persona

Es importante conocer la distancia entre los postes de alumbrado público dentro del campus de la universidad porque se puede determinar a qué distancia del poste de alumbrado el sensor de presencia detectará a la persona que se aproxima. El sensor de presencia (PIR) que usamos para esta investigación es el HC-SR501 y su rango de detección es de menos de 120 grados y dentro de los 7 metros.

Si el ángulo de apertura del sensor de presencia es de 120 grados para un solo lado del poste se tendrán 60 grados, por lo que se tiene un triángulo rectángulo conocido de 60 y 30 grados, como se muestra en la figura siguiente.

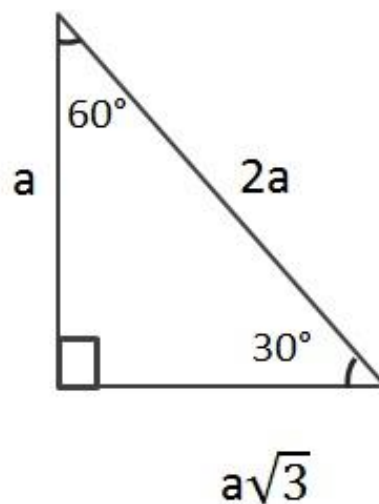


Figura N° 25: Triángulo para calcular la distancia entre una persona y el poste.
Elaboración propia.

Se puede determinar que la altura del poste de alumbrado es 'a' y la distancia entre una persona que se aproxima al poste de alumbrado y el mismo poste es 'a' multiplicada por la raíz cuadrada de 3. Si sabemos que 'a' es 6.6 metros entonces la distancia entre el poste y una persona que se aproxima a él, es de 11.43 metros. Esto quiere decir que cuando una persona se aproxime al poste de alumbrado, el sensor de presencia detectará a la persona a 11.43 metros o menos y de esa forma encender la luminaria en su máxima intensidad.

Se realizó la medición de la distancia entre dos postes de alumbrado público dentro del campus y se obtuvo un valor de 21 metros.



Figura N° 26: Medición de la distancia entre dos postes de alumbrado.
Elaboración propia.



Figura N° 27: Medición de la distancia entre dos postes de alumbrado.
Elaboración propia.

Si la distancia entre dos postes es de 21 metros, entonces se tiene que el punto medio entre los dos postes es de 10.5 metros, la distancia máxima para que un sensor de presencia pueda detectar la presencia de una persona es de 11.43 metros, lo que supera por casi un metro al punto medio entre los dos postes (10.5 metros).

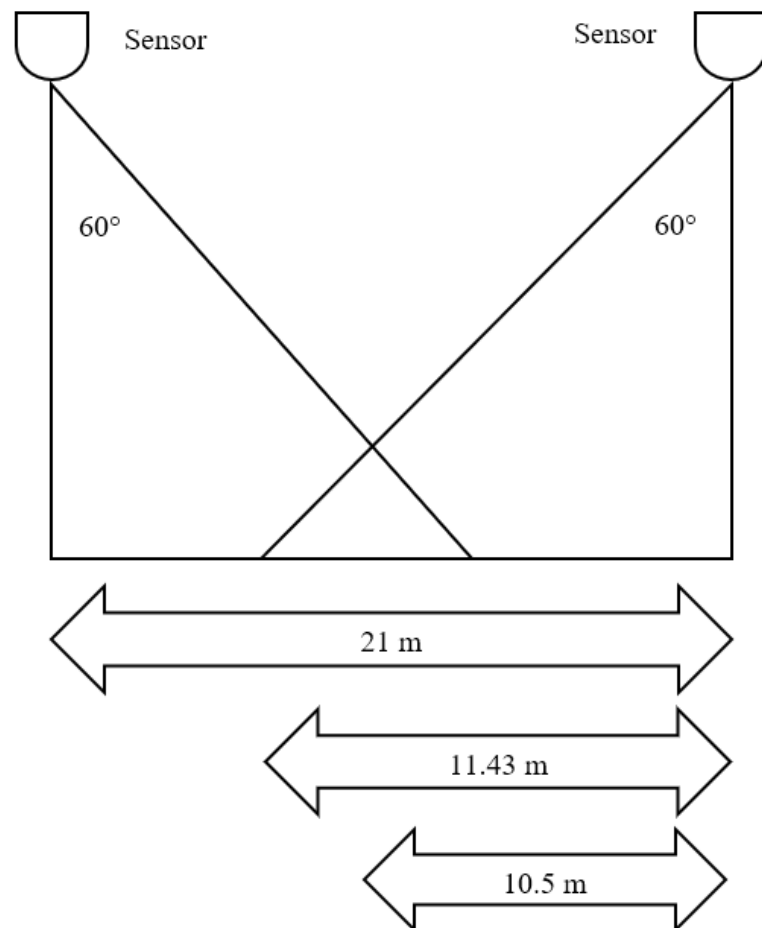


Figura N° 28: Verificación de la distancia máxima para detectar la presencia de una persona.
Elaboración propia.

Esto quiere decir que, si una persona se aleja de un poste y se aproxima al siguiente, la luz del poste siguiente se encenderá a su máxima potencia mientras que la luz del poste anterior también continúa en su máxima potencia, de esa forma se garantiza que no haya un momento en que una persona se quede sin iluminación con una potencia máxima. Por lo tanto, se comprueba que el sensor de presencia que estamos usando



cumple con los requerimientos mínimos al momento de la implementación. De esta forma existe una intersección de cobertura de los sensores entre postes adyacentes.

3.2.6.6 Topología

La topología que será implementada cuenta con un administrador y 98 controladores para las luminarias, todos con una interfaz Ethernet para su conectividad en red. Se define controladores de red a los dispositivos IoT que gestionan la iluminación de cada uno de los postes de alumbrado público dentro del campus. Por lo tanto, se podría usar 3 switches de 48 puertos que en total suman 144 puertos para cubrir la demanda de los 98 controladores para luminarias, o 5 switches de 24 puertos para cubrir la misma demanda de dispositivos IoT, pero se ha especificado N switches, donde N es un número necesario para permitir la conectividad de todos los controladores a la red sin superar los 100 metros de distancia máxima permitida entre un dispositivo final (controlador IoT o computadora) y un switch; o entre switches. Esto se hace porque en las mediciones se ha determinado que, por ejemplo, la distancia entre dos postes de muestra fue de 21 metros, por lo tanto, basta con contabilizar 5 postes para llegar a una distancia de 105 metros aproximadamente, allí entonces se necesitaría otro switch adicional que actúe como repetidor o regenerador de señal.

El administrador usa un navegador web para supervisar, monitorear y configurar los controladores, cada dispositivo en la red, incluyendo controladores, switches y la computadora de administración, tienen una dirección IP única en la red.

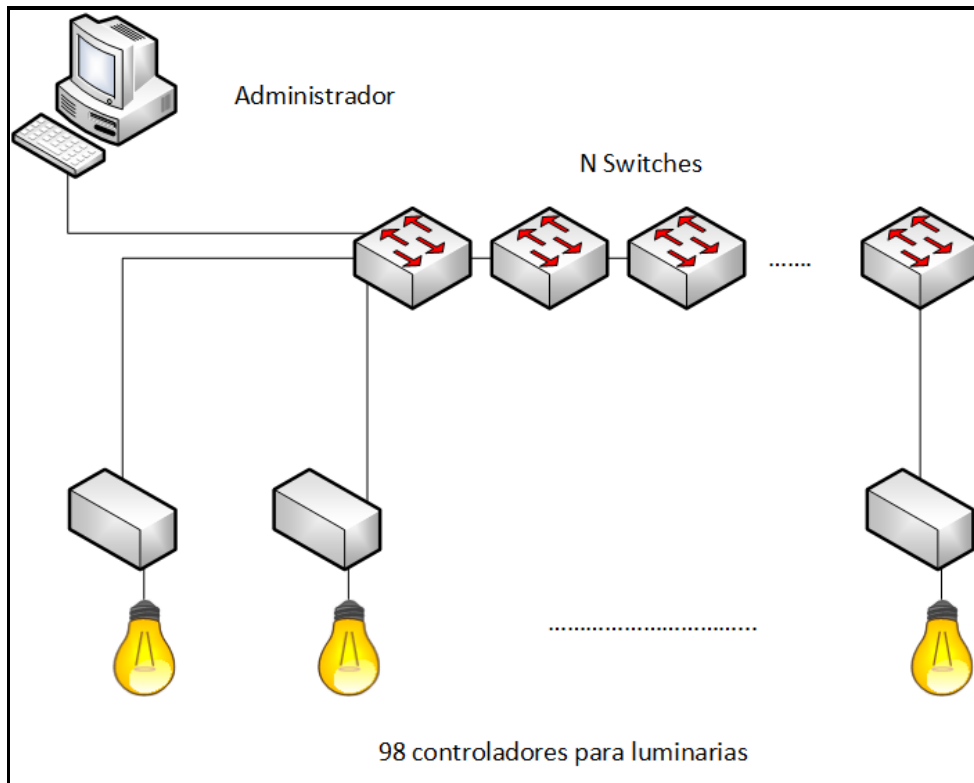


Figura N° 29: Topología de la implementación.
Elaboración propia.



CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1 PRUEBA DE CONECTIVIDAD CON ICMP

ICMP es el protocolo de mensajes de control en Internet, permite realizar estadísticas de conectividad IP entre un origen y un destino, trabaja en la capa 3 del modelo OSI y puede dar información importante como el tiempo de respuesta, el número de saltos y resolver el nombre de dominio a una dirección IP. Para las pruebas de rigor de esta investigación se ha determinado realizar 1000 pruebas de ping entre la computadora de administración y un controlador de luminaria (dispositivo IoT).

Para realizar la prueba de ping se ingresa el comando seguido de la opción -n que indica una repetición de 1000 pruebas, al final del comando se ingresa la dirección IP de destino.

```
C:\Users\ADMIN>ping -n 1000 192.168.0.50
```

Haciendo ping a 192.168.0.50 con 32 bytes de datos:

```
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=9ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=247ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
```



Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=5ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=11ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=12ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=88ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=5ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=16ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255

Estadísticas de ping para 192.168.0.50:

Paquetes: enviados = 1000, recibidos = 1000, perdidos = 0
(0% perdidos),

Tiempos aproximados de ida y vuelta en milisegundos:

Mínimo = 0ms, Máximo = 247ms, Media = 2ms

En la parte final después de realizar las pruebas de ping se observa los resultados estadísticos, pero antes de hacer el análisis, es importante señalar que se está usando el



tamaño por defecto de los paquetes para la prueba de ping, que es de 32 bytes, adicionalmente el tiempo de vida es de 255 (TTL).

Los paquetes enviados son 1000 y se recibieron 1000 con éxito, esto quiere decir que existe un 0% de paquetes perdidos. El tiempo aproximado de ida y vuelta tiene como mínimo un valor de 0ms, un valor máximo de 247ms y una media o promedio de 2ms. El promedio de 2ms es un valor aceptable para poder aplicar otros servicios como HTTP (web), y en la siguiente sección se describe.

4.2 RESULTADOS DEL SERVIDOR WEB

Para verificar los resultados del servidor web hacemos una prueba hacia la dirección IP configurada en el controlador, la cual es 192.168.0.50. La prueba con un ping resulta ser un éxito ya que de cuatro datos enviados se reciben también cuatro. Para que nos muestre si el controlador esta con esta dirección IP y no es otro que está utilizando, realizaremos una consulta “arp -a”. El protocolo ARP (Address Resolution Protocol) tiene un importante papel entre los protocolos de comunicaciones relacionados con TCP/IP. Su principal objetivo es conocer la dirección física (MAC) de una tarjeta de interfaz de red correspondiente a una dirección IP (Internet Protocol). De ahí viene su nombre: Protocolo de Resolución de Dirección (Address Resolution Protocol). ARP muestra las direcciones ya sea de tablet, smartphone, Arduino, Raspberry Pi, etc. Toda la documentación del protocolo ARP está en el RFC 826. Al introducir el comando se muestra que la dirección IP 192.168.0.50 está relacionada con la dirección MAC 98-4F-EE-01-66-E2 la cual es la misma que se muestra encima del puerto del Shield y es configurada en la programación.

Realizando la prueba con el “ping” y comprobando si esta la placa conectada en esta dirección con “arp -a”, se muestra que la conexión es viable para la prueba mediante el navegador.

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.15063]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\libre>PING 192.168.0.50

Haciendo ping a 192.168.0.50 con 32 bytes de datos:
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=64
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=64
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=64

Estadísticas de ping para 192.168.0.50:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 2ms, Media = 1ms

C:\Users\libre>ARP -A

Interfaz: 192.168.0.10 --- 0x9
Dirección de Internet      Dirección física      Tipo
192.168.0.1                00-00-ca-11-22-33    dinámico
192.168.0.3                f0-43-47-d6-6a-0b    dinámico
192.168.0.50              98-4f-ee-01-66-e2    dinámico
192.168.0.255             ff-ff-ff-ff-ff-ff    estático
224.0.0.2                 01-00-5e-00-00-02    estático
224.0.0.22                01-00-5e-00-00-16    estático
224.0.0.251               01-00-5e-00-00-fb    estático
224.0.0.252               01-00-5e-00-00-fc    estático
239.2.0.252               01-00-5e-02-00-fc    estático
239.192.152.143          01-00-5e-40-98-8f    estático
239.255.255.250          01-00-5e-7f-ff-fa    estático
255.255.255.255          ff-ff-ff-ff-ff-ff    estático

C:\Users\libre>
```

Figura N° 30: Prueba de conexión con el web server
Elaboración propia

Los resultados posteriores se mostrarán en capturas de pantalla en el navegador web del cliente ya sea mediante un ordenador portátil o un Smartphone. Como se muestra en la figura 23 muestra la captura del navegador ingresando a la dirección IP configurada en la programación del Arduino junto al Shield. Se puede mostrar la gestión para dos alumbrados donde indican los estados en las entradas de las fotorresistencias, el estado del alumbrado ya sea que este encendido o apagado las luminarias; los LEDs de LED-LDR1 y 2 son mostrados aquí como las primeras luminarias, ya sea porque siempre serán encendidas cuando la iluminación sea escasa y los LED-SENS.MOV.1 y 2 son de

nominados segunda luminaria, puesto que se activan cuando el sensor de movimiento envíe el 1 lógico. El sensor de movimiento indica “DESOLADO” si no se detecta movimiento por esa área, caso contrario se muestra “EXISTE CIRCULACIÓN”. Los botones de “ON” y “OFF” son para el control remoto cada uno de estos controla a la luminaria descrita encima de estos botones.

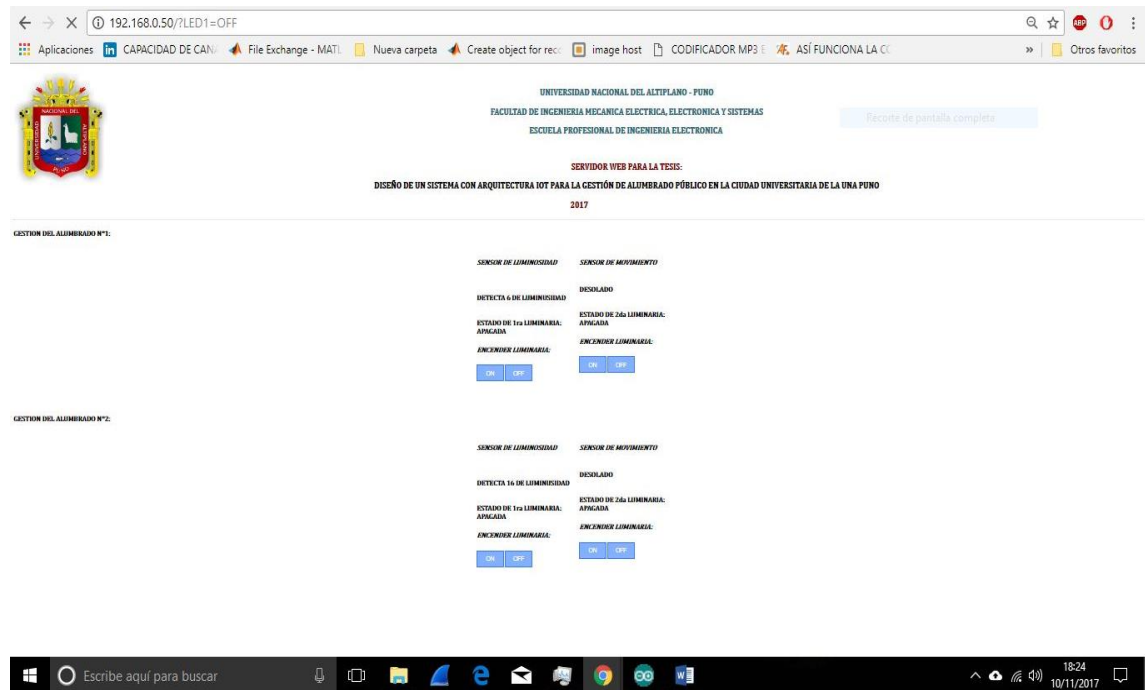


Figura N° 31: Captura de la visualización al ingresar al servidor web
Elaboración propia

Para verificar que los comandos enviados por el controlador hacia los clientes de HTML se utilizaran la opción de inspeccionar, esta opción está disponible en todos los navegadores. En el navegador Google Chrome se ingresa mediante el clic derecho y seleccionar “inspeccionar” o de forma más rápida gracias la atajo, presionando las teclas Ctrl+Mayus+I. Mostrado ya la ventana se visualiza la línea de comandos HTML, hay listas que resumen la programación, pero son expandibles en las cuales indican toda la programación recibida por el servidor; expandiendo una parte de la línea de comandos se observa que es similar a la escrita en la programación del Arduino.

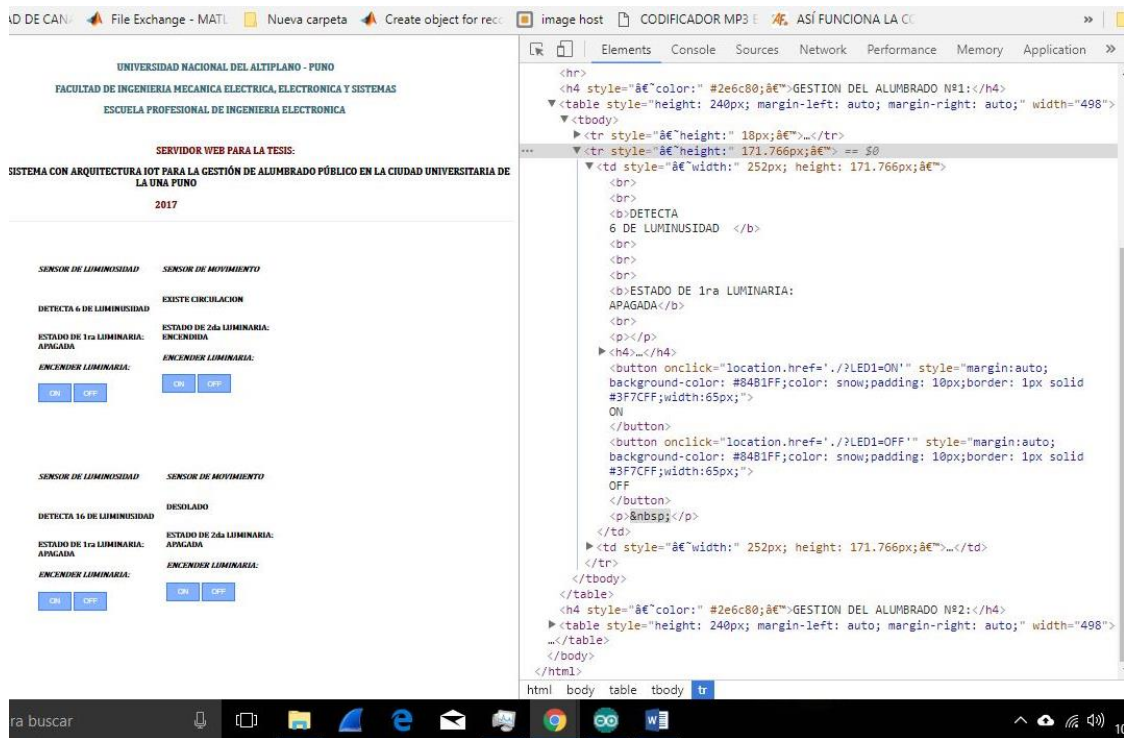
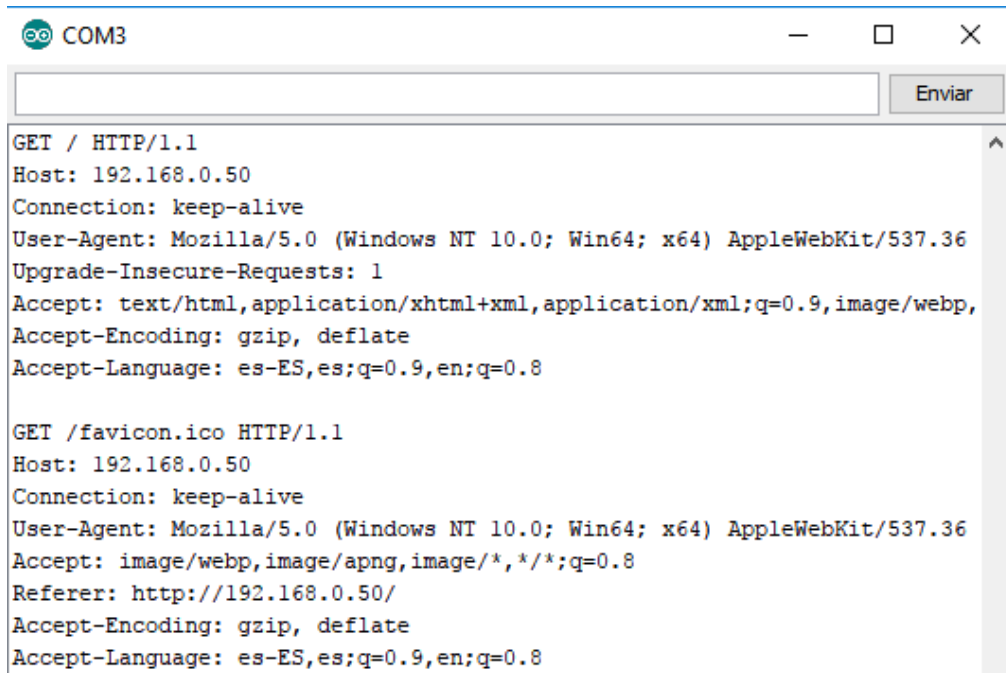


Figura N° 32: Inspección de la programación HTML del arduino
Elaboración propia

Despues de ver que la comunicacion es correcta en el cliente web revisaremos la comunicacion serial que envia el controlador respecto a lo sucedido cuando se conectan los clientes y activar los botones. En las figuras 26 y 27 se muestran las peticiones conectarse con el servidor. Las peticion desde el ordenador es visualizado en la figura 24 inicia pidiendo la conexión indicando el protocolo de presentacion que es HTTP la version 1.1; luego indica la forma de conexión en este caso es “keep-alive”, esto indica mantener la conexión. El agente de usuario es por Mozilla 5.0 para el controlador pero en el cliente se muestra que el sistema operativo es Windows 10.0. tambien se muestra que solo hay una solicitud de parte del cliente web.



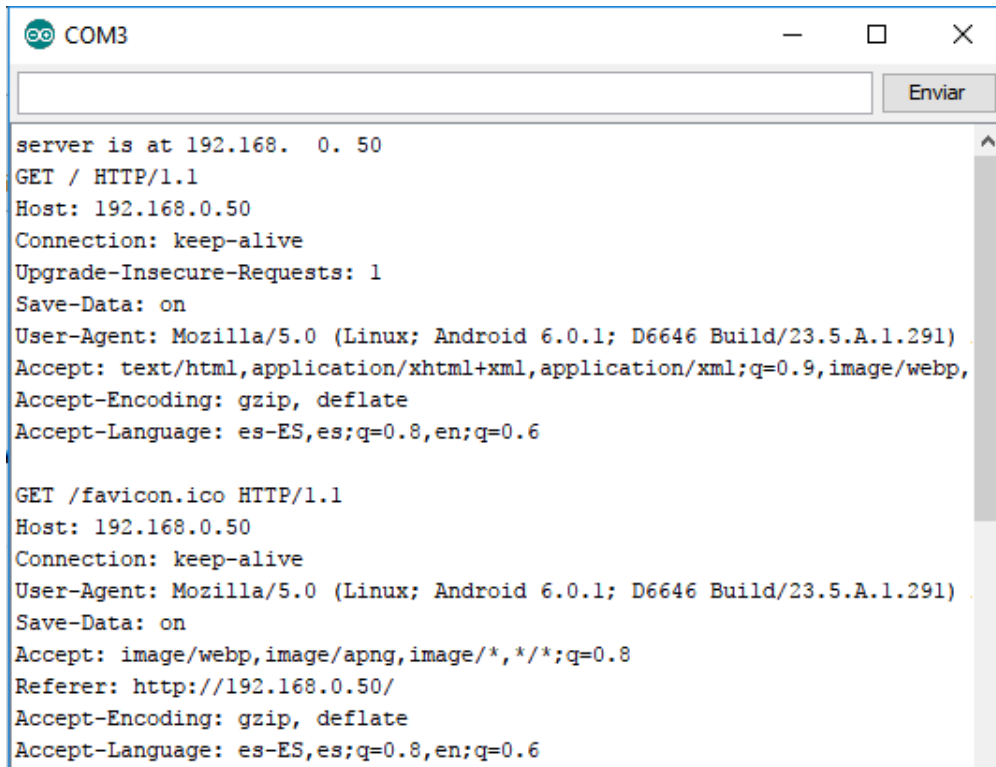
```
COM3
Enviar
GET / HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8

GET /favicon.ico HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://192.168.0.50/
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8
```

Figura N° 33: Petición del cliente desde el ordenador
Elaboración propia

Ya explicado anteriormente en la figura 26 muestra la conexión con el Smartphone en la línea de “user-agent” se describe el sistema operativo Android 6.0.1 y el modelo que es D6646 junto la versión de compilación del sistema.

El espacio de GET/favicon.ico es enviado a los clientes. Un favicon, del inglés *favorites icon* (icono de favoritos), también conocido como icono de página, es una pequeña imagen asociada con una página o sitio web en particular. Los navegadores gráficos suelen mostrar el favicon de la página visitada, si ésta lo posee, en la barra de direcciones y en el encabezado de la pestaña correspondiente. Asimismo los utilizan para identificar más fácilmente las páginas, presentándolos junto a sus respectivos títulos, en las listas de marcadores (navegadores web) o favoritos, en los historiales (navegador web) de páginas visitadas y de navegación reciente, y toda vez que deban hacer referencia a ellas.

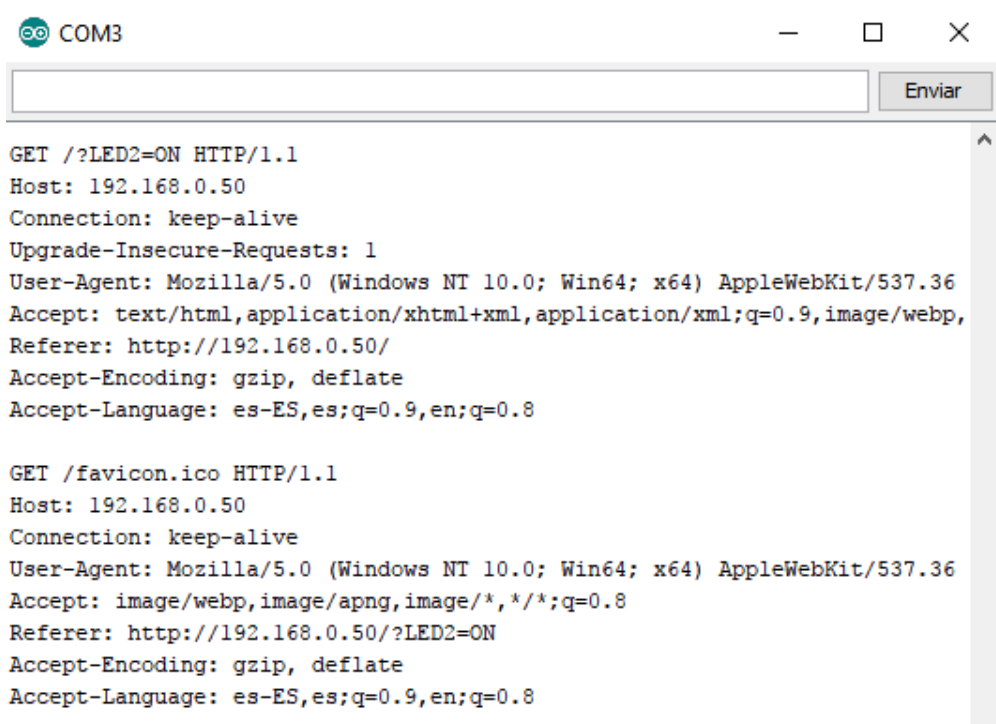


```
COM3
server is at 192.168. 0. 50
GET / HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Save-Data: on
User-Agent: Mozilla/5.0 (Linux; Android 6.0.1; D6646 Build/23.5.A.1.291)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.8,en;q=0.6

GET /favicon.ico HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 6.0.1; D6646 Build/23.5.A.1.291)
Save-Data: on
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://192.168.0.50/
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.8,en;q=0.6
```

Figura N° 34: Petición del cliente desde el smartphone
Elaboración propia

Después de la conexión con el cliente web se procede a la ejecución de órdenes desde el navegador del cliente web, en la figura 28 se muestra la petición desde el ordenador donde solicita cambiar el estado de LED2 a ON, los espacios sobrantes son idénticos a la anterior solicitud de conexión. En la segunda parte donde se muestra la respuesta de la solicitud indica que la petición se realizó.



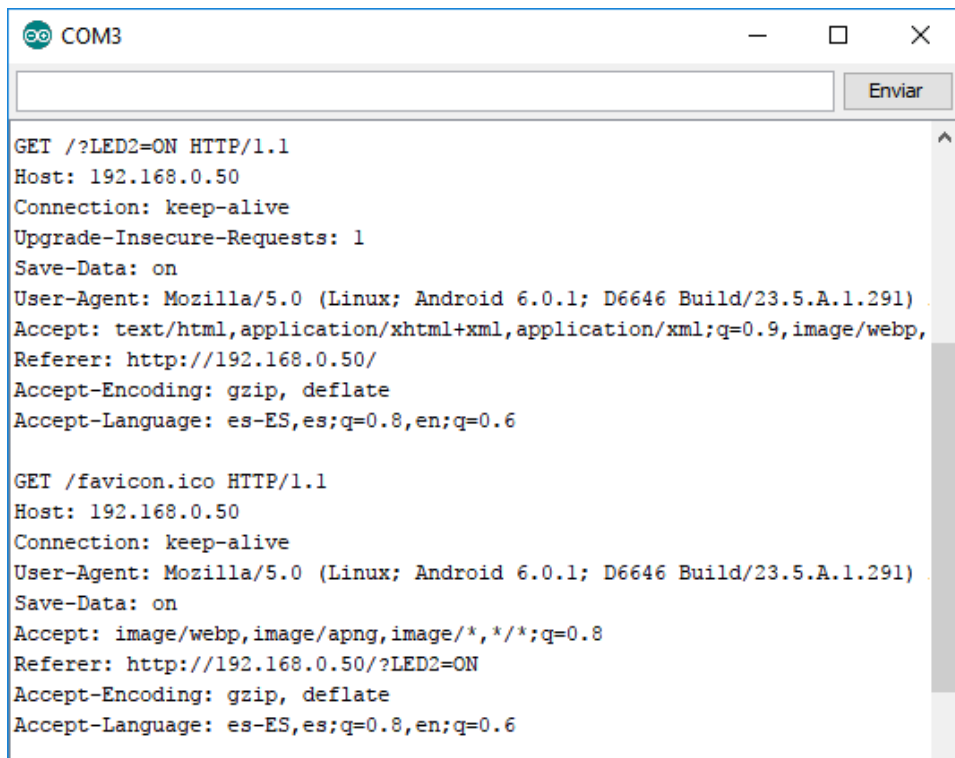
```
COM3
Enviar

GET /?LED2=ON HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
Referer: http://192.168.0.50/
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8

GET /favicon.ico HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://192.168.0.50/?LED2=ON
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en;q=0.8
```

Figura N° 35: Momento al pulsar los botones en el cliente desde ordenador
Elaboración propia

En la figura N° 29 se muestra la petición desde el smartphoner donde solicita cambiar el estado de LED2 a ON, los espacios sobrantes son idénticas a la anterior solicitud de conexión. En la segunda parte donde se muestra la respuesta de la solicitud indica que la petición se realizó. Este proceso se repite en cada petición de cambio de estado, en la actualización en el navegador o en una nueva conexión de otro dispositivo.



```
COM3
[Input field] [Enviar]
GET /?LED2=ON HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Save-Data: on
User-Agent: Mozilla/5.0 (Linux; Android 6.0.1; D6646 Build/23.5.A.1.291)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
Referer: http://192.168.0.50/
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.8,en;q=0.6

GET /favicon.ico HTTP/1.1
Host: 192.168.0.50
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 6.0.1; D6646 Build/23.5.A.1.291)
Save-Data: on
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://192.168.0.50/?LED2=ON
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.8,en;q=0.6
```

Figura N° 36: Momento al pulsar los botones en el cliente desde smartphone
Elaboración propia

4.3 RESULTADOS DEL FUNCIONAMIENTO

Terminado con la conexión del servidor web con los clientes web, se mostrará el funcionamiento de la maqueta con toda la programación.

4.3.1 Alta iluminación

En la Figura 37 se muestra la maqueta donde se observa que la iluminación exterior es visible por esta razón la fotorresistencia indicara un valor mayor en la comparación realizada en la programación, por lo cual las luminarias estarán apagadas hasta que la fotorresistencia mande un valor menos, es decir cuando ya la iluminación esta mínima.



Figura N° 37: Maqueta con iluminación
Elaboración propia

En la figura 31 se comprueba los valores monitoreadas por las fotorresistencias las cuales son mayores. Como la activación de las luminarias aún no se realizó, el sensor de movimiento tampoco es activado ya que por esta razón se muestra “DESOLADO” debajo de “SENSOR DE MOVIMIENTO” esto en ambas luminarias. Al igual que las luminarias están desactivadas también los botones para el control están deshabilitadas para activar las luminarias. Mientras la fotorresistencia mande valores altos a la comparación no se podrán activar las luminarias ya sea automáticamente (por la programación) o mediante los botones mostrados en el navegador del cliente web.

CIUDAD UNIVERSITARIA DE LA UNA PUNO

2017

GESTION DEL ALUMBRADO N°1:

<i>SENSOR DE LUMINOSIDAD</i>	<i>SENSOR DE MOVIMIENTO</i>
DETECTA 5 DE LUMINUSIDAD	DESOLADO
ESTADO DE 1ra LUMINARIA: APAGADA	ESTADO DE 2da LUMINARIA: APAGADA
<i>ENCENDER LUMINARIA:</i>	<i>ENCENDER LUMINARIA:</i>
<input type="button" value="ON"/> <input type="button" value="OFF"/>	<input type="button" value="ON"/> <input type="button" value="OFF"/>

GESTION DEL ALUMBRADO N°2:

<i>SENSOR DE LUMINOSIDAD</i>	<i>SENSOR DE MOVIMIENTO</i>
DETECTA 15 DE LUMINUSIDAD	DESOLADO
ESTADO DE 1ra LUMINARIA: APAGADA	ESTADO DE 2da LUMINARIA: APAGADA
<i>ENCENDER LUMINARIA:</i>	<i>ENCENDER LUMINARIA:</i>
<input type="button" value="ON"/> <input type="button" value="OFF"/>	<input type="button" value="ON"/> <input type="button" value="OFF"/>

Figura N° 38: Gestión en un cliente web cuando exista iluminación
Elaboración propia

4.3.2 Modo nocturno

En el modo nocturno indica que la fotorresistencia envía valores mínimos, lo cual indicara que la iluminación es baja o nula, por lo cual se encenderán las luminarias de baja intensidad, como se muestra en la figura 32. Puesto que se activaron las luminarias de baja intensidad también ya comienza a activarse los sensores de movimiento, los cuales activaran a las luminarias de alta intensidad posteriormente.



Figura N° 39: Activación de las luminarias por la fotorresistencia
Elaboración propia

En la figura N° 33 se muestra por el navegador web que el estado de las primeras luminarias es “ENCENDIDA” ya que el valor enviada por la fotorresistencia es 0, lo que de acuerdo a la programación es menor al valor de comparación. Si las luminarias no son encendidas hay la posibilidad de encenderlas por el botón “ON” o apagarlas con “OFF”, puesto que ya se activó su funcionamiento, por el mismo motivo del encendido de las primeras luminarias. Ya activado los botones de control de la primera luminaria, de la misma manera también se activaron el funcionamiento de los botones de la segunda luminaria; es decir que se puede apagar y encender la segunda luminaria desde el navegador, independientemente de lo enviado desde el sensor de movimiento.



Figura N° 40: Estado en el cliente web al encender las primeras luminarias
Elaboración propia

4.4 DETECCIÓN DE MOVIMIENTO EN EL ALUMBRADO

Cuando los sensores detectan el movimiento cerca del alumbrado procederán a activar a la segunda luminaria. La segunda luminaria al ser de más intensidad da mayor visibilidad para los transeúntes cuando estén cerca del alumbrado o mientras el sensor a un detecte el movimiento. En la figura 34 muestra que la intensidad del alumbrado N°1 ha aumentado mientras que del otro aun esta con la iluminación de menor intensidad, debido a que la segunda luminaria (activada con la detección de movimiento) no está activada. El sensor de movimiento solo detecta movimiento cerca del alumbrado N°1 y no en el N°2.



Figura N° 41: Detección de movimiento en el alumbrado N°1
Elaboración propia

Cuando se enciende la segunda luminaria en el alumbrado N°1, en el monitoreo por el navegador web indica “EXISTE CIRCULACIÓN”, refiriéndose que existe movimiento por el área de detección del sensor de movimiento de este alumbrado. En el alumbrado N°2 muestra “DESOLADO” ya que no se detectó movimiento por en el rango del sensor. Como se indica el estado en la figura 35, también puede ser controlado con sus respectivos botones en el navegador web.

Estos resultados se repetirán cuando se detecte movimiento cerca al alumbrado N°2, con la diferencia que se activará la segunda luminaria de este alumbrado y en la ventana del navegador se mostrará que “EXISTE CIRCULACION” en la parte de alumbrado N°2 (ver figura B. 5 y figura B. 6 del ANEXO B). La posibilidad que en ambos alumbrados enciendan a la segunda luminaria es posible. (ver figura B. 7 y figura B. 8 del ANEXO B)

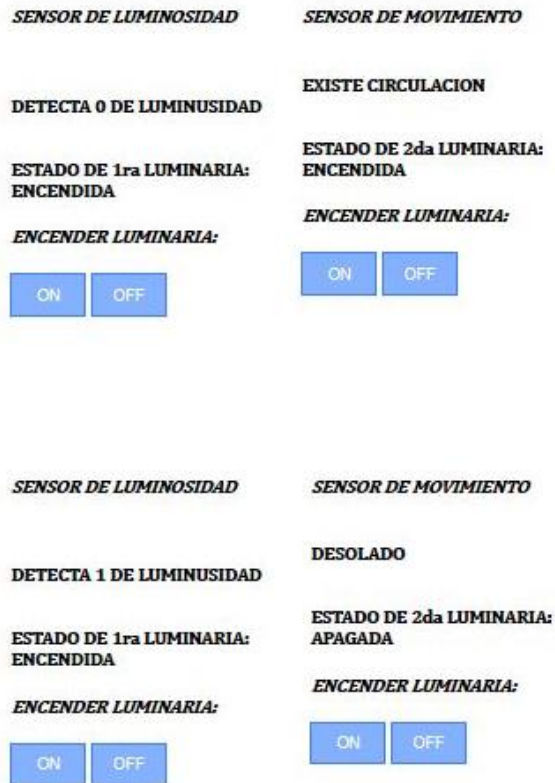


Figura N° 42: Estado en el cliente web al encender la primera luminaria del alumbrado N°1

Elaboración propia



V. CONCLUSIONES

Primero: El diseño e implementación del sistema de Internet de las cosas para la gestión del alumbrado público en la ciudad universitaria de la Universidad Nacional del Altiplano en la ciudad de Puno es de gran avance exploratorio, ya que abre una nueva forma de monitoreo y control del sistema. Al implementar el prototipo resulto sencilla la gestión del sistema, mediante un navegador web en el cliente, que ya implementado estaría a cargo del técnico supervisor o personal encargado.

Segundo: Al diseñar un sistema de Internet de las cosas para el alumbrado público de la ciudad universitaria de la Universidad Nacional del Altiplano Puno es posible porque los resultados no mostraron fallas de conectividad ni de funcionalidad. El sistema comprende una parte en hardware conformada por las conexiones y una parte de software conformada por la programación, en la cual, todos los requisitos explicados en la implementación lograron un funcionamiento adecuado al momento de hacer el cambio de los modos de operación.

Tercero: La implementación del prototipo permitió la evaluación del sistema de alumbrado que será implementado en la ciudad universitaria, mediante un navegador web mostrando los estados de las luminarias, si están encendidas o apagadas, al igual el estado enviado por el sensor de movimiento y también la cantidad de luminosidad detectada por la fotorresistencia; además ofrece la opción de activar las luminarias desde el navegador web.

En cuanto al funcionamiento y rendimiento se obtuvieron buenos resultados, como el ahorro de consumo de energía eléctrica



VI. RECOMENDACIONES

1. Esto permitirá el ahorro de energía cuando se instale en alumbrados públicos o en interiores ya sea en laboratorios u hogares. Para implementar en otros lugares como interiores u hogares es necesario hacer un estudio mínimo de la situación del ambiente para luego realizar algunos cambios e implementarlo.
2. Se puede integrar con un sistema de paneles solares que permitan un mayor ahorro, esto es posible ya que la cantidad de energía necesaria no es excesiva, debido al uso de luminarias LED.
3. Se puede usar un servidor general que gestione todos los clientes, es decir que un servidor central o un ordenador pueda gestionar a los controladores Arduino. Gestionar ya sea controlándolos o monitoreándolos, al igual que la función del Arduino en este sistema.



VII. REFERENCIAS

- Arduino (2017). Arduino.cc. Recuperado de:
<https://www.arduino.cc/en/Guide/Introduction>
- Asay. (27 de Junio de 2014). *ReadWrite. Obtenido de The Internet of Things Will Need Millions of Developers By 2020.* Recuperado de:
<http://readwrite.com/2014/06/27/internet-of-thingsdevelopers-jobs-opportunity/>
- AUMA (2017). *ACTUADORES ELÉCTRICOS para la automatización de válvulas industriales.* Recuperado de:
http://www1.auma.com/uploads/media/sp_import2/prospekte/pb_modular_range_es.pdf
- Curran, K. (2011). Ubiquitous developments in ambient computing and intelligence. *Hershey, PA: Information Science Reference.*
- Dacosta, F. (2013). Rethinking The Internet Of Things, A Scalable Approach to Connecting Everything. *Appress Open.*
- DeLoach, D., Berthelsen, E. and Elrifai, W. (2017). The Future of IoT: Leveraging the Shift to a Data Centric World. Paperback. *Singapore.*
- Wikipedia (2017). *Detector de movimiento.* Recuperado de:
https://es.wikipedia.org/wiki/Detector_de_movimiento
- Di Martino, B., Li, K., Yang, L. y Esposito, A. (2018). Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives. *Springer Nature Singapore Pte Ltd.*



- EcuRed. (2017). *Fotorresistencia*. Recuperado de:
<https://www.ecured.cu/Fotorresistencia>
- Faverón, O. (2017). *La Internet de las Cosas y el Hogar del Futuro* (Spanish Edition). 30th ed. *Enhanced Typesetting*.
- Gartner (2013). *Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020*. Recuperado de: <http://www.gartner.com/newsroom/id/2636073>
- Guillen, S. (2015). *Beehive, una plataforma Open-Source para el Internet de las Cosas* (Tesis de Pregrado). Universidad Mayor de San Andres, La Paz, Bolivia.
- Herzberg, C. (2017). *Smart Cities, Digital Nations: Building Smart Cities in Emerging Countries and Beyond*. Hardcover: Cisco.
- H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, y A. Oliveira. (2011). Smart cities and the future internet: Towards cooperation frameworks for open innovation. *The Future Internet, Lect. Notes Comput. Sci.*, vol. 6656, pp. 431–446, 2011.
- IncubaWeb (2012). *El internet de las cosas y sus mayores riesgos*. Recuperado de:
<https://www.incubaweb.com/internet-de-las-cosas-mayores-riesgos/>
- Karvinen, K. y Karvinen, T. (2014). *Getting Started with Sensors: Measure the World with Electronics, Arduino, and Raspberry Pi 1st Edition*. Maker Media, Inc. Estados Unidos de América.
- Komninos, N., Pallot, M. y Schaffers, H. J Knowl Econ (2013). *Special Issue on Smart Cities and the Future Internet in Europe*. Recuperado de:
<https://doi.org/10.1007/s13132-012-0083-x>



- Mohsin, M. (2017). High Efficiency Intelligent Street Lighting System Using a Zigbee Network and GSM. *International Journal for Research in Applied Science and Engineering Technology*, V(II), pp.510-516.
- Melnick, S. and Barraza, J. (2017). Internet de las Cosas (IoT) Web 3.0 y la revolución móvil: El acceso a la nueva mente tecnológica colectiva (Spanish Edition). *Enhanced Typesetting*.
- Microcontroller. (2017). *En.wikipedia.org*. Recuperado de: https://en.wikipedia.org/wiki/Microcontroller#Programming_environments
- Navas, P. (2017). *Sistema de Red Telemétrico de Alumbrado Público para la Provincia de Santa Elena* (tesis de grado). Universidad Estatal Península De Santa Elena.
- Rodríguez, K. (2017). *Plataforma Cloud/GIS de Gestión, Operación y Control de Sistemas Fotovoltaicos de Iluminación en el Alumbrado Público en la Universidad Técnica de Ambato* (tesis de grado). Universidad Técnica De Ambato.
- Uckelmann, D., Harrison, M., y Michahelles, F. (2011). *Architecting the internet of things*. Berlin: Springer.
- Verle, M. (2017). *Introducción al mundo de los microcontroladores / Microcontroladores PIC – Programación en C con ejemplos*. Recuperado de: <https://learn.mikroe.com/ebooks/microcontroladorespicc/chapter/introduccion-al-mundo-de-los-microcontroladores/>
- Zanella, A., Bui, N., Castellani, A., Vangelista, L. and Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1), pp.22-32.



ANEXOS

Anexo 1: Programación realizada en Arduino

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {0x98, 0x4F, 0xEE, 0x01, 0x66, 0xE2 };
// ingresamos la dirección mac del controlador
IPAddress ip(192,168,0,50);
// la dirección IP para el arduino dependiendo de la red
const int led1Pin = 2;
// salida para el control de las LED-LDR1
const int led2Pin = 3;
// salida para el control de las LED-SENS.MOV.1
const int led3Pin = 5;
// salida para el control de las LED-LDR2
const int led4Pin = 6;
// salida para el control de las LED-SENS.MOV.2
int sensorMovimi = 4;
// entrada para el SENSOR DE MOVIMIENTO 1
int sensorMovimi2 = 7;
// entrada para el SENSOR DE MOVIMIENTO 2
int ldrReading;
// constante para FOTORESISTENCIA 1
int ldrReading2;
// constante para FOTORESISTENCIA 2
int ldrPin = A0;
// entrada analogica para el FOTORESISTENCIA 1
int ldr2Pin = A1;
// entrada analogica para el FOTORESISTENCIA 2
String estado1;
// variable para visualizar estado de LED-LDR1 en el cliente HTML
String estadomov;
// variable para visualizar estado de LED-SENS.MOV.1 en el cliente
HTML
String estado2;
// variable para visualizar estado de LED-LDR2 en el cliente HTML
String estadomov2;
// variable para visualizar estado de LED-SENS.MOV.2 en el cliente
HTML
String estadoperso;
// variable para visualizar estado de la area del alumbrado 1, en
el cliente HTML
String estadoperso2;
// variable para visualizar estado de la area del alumbrado 1, en
el cliente HTML
char c;
// variable para inicio del cliente HTML
int movState = 0;
```



```
// variable para la comparacion con estadomov
int movState2 = 0;
// variable para la comparacion con estadomov2
// Inicializar la biblioteca del servidor Ethernet
// con la dirección IP y el puerto que quieres usar
// (el puerto 80 es el predeterminado para HTTP):
EthernetServer server(80);
void setup() {
  // Abra las comunicaciones serie y espere a que se abra el
  puerto:
  Serial.begin(9600);
  while (!Serial) {
    ; // esperando la comunicacion serial
  }
  //asignacion de entradas y salidas
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
  pinMode(led4Pin, OUTPUT);
  pinMode(sensorMovimi, INPUT);
  pinMode(sensorMovimi2, INPUT);
  // inicia la coneccion Ethernet y el servidor:
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}
void loop() {
  // asignacion variables de estado para posterior comparacion
  int est;
  int est1;
  int est2;
  int est3;
  //declaracion de entradas analogicas de las fotoresistencias
  ldrReading = analogRead(ldrPin);
  ldrReading2 = analogRead(ldr2Pin);
  // escuchado clientes web
  EthernetClient client = server.available();
  if (client) {
    // si ha llegado al final de la línea (recibió un nuevo
    carácter
    // de línea) y la línea está en blanco, la solicitud http ha
    finalizado,
    // para que pueda enviar una respuesta
    boolean currentLineIsBlank = true;
    //creamos la cadena para crear los botones en el servidor web
    y recibir los datos enviados por el cliente web
    String cadena = "";
    String cadena2 = "";
```



```
while (client.connected()) {
  if (client.available()) {
    c = client.read();
    Serial.write(c);
    cadena.concat(c);
    int posicion3=cadena.indexOf("LED3=");
    //Guardamos la posición de la instancia "LED3=" a la
variable 'posicion'
    if(cadena.substring(posicion3)=="LED3=ON")
    //Si a la posición 'posicion3' hay "LED3=ON"
    {
      digitalWrite(led3Pin,HIGH);
      //se encendera LED-LDR2
      est2=1;
      //estado de LED-LDR2 sera 1
    }
    if(cadena.substring(posicion3)=="LED3=OFF")
    //Si a la posición 'posicion3' hay "LED3=OFF"
    {
      digitalWrite(led3Pin,LOW);
      //se apagara LED-LDR2
      est2=0;
      //estado de LED-LDR2 sera 0
    }

    int posicion=cadena.indexOf("LED4=");
    //Guardamos la posición de la instancia "LED4=" a la
variable 'posicion'

    if(cadena.substring(posicion)=="LED4=ON")
    //Si a la posición 'posicion' hay "LED4=ON"
    {
      digitalWrite(led4Pin,HIGH);
      //se encendera LED-SENS.MOV.2
      est3=1;
      //estado de LED-SENS.MOV.2 sera 1
    }
    if(cadena.substring(posicion)=="LED4=OFF")
    //Si a la posición 'posicion' hay "LED4=OFF"
    {
      digitalWrite(led4Pin,LOW);
      //se apagara LED-SENS.MOV.2
      est3=0;
      //estado de LED-SENS.MOV.2 sera 0
    }

    cadena2.concat(c);
    int posicion1=cadena2.indexOf("LED1=");
```



```
//Guardamos la posición de la instancia "LED1=" a la variable
'posicion'

    if(cadena2.substring(posicion1)== "LED1=ON")
    //Si a la posición 'posicion1' hay "LED1=ON"
    {
        digitalWrite(led1Pin,HIGH);
        //se encendera LED-LDR1
        est=1;
        //estado de LED-LDR1 sera 1
    }
    if(cadena2.substring(posicion1)== "LED1=OFF")
    //Si a la posición 'posicion1' hay "LED=OFF"
    {
        digitalWrite(led1Pin,LOW);
        //se apagara LED-LDR1
        est=0;
        //estado de LED-LDR1 sera 0
    }

    int posicion2=cadena.indexOf("LED2=");
    //Guardamos la posición de la instancia "LED2=" a la
variable 'posicion2'

    if(cadena2.substring(posicion2)== "LED2=ON")
    //Si a la posición 'posicion2' hay "LED2=ON"
    {
        digitalWrite(led2Pin,HIGH);
        //se encendera LED-SENS.MOV.2
        est1=1;
        //estado de LED-SENS.MOV.2 sera 1
    }
    if(cadena.substring(posicion2)== "LED2=OFF")
    //Si a la posición 'posicion2' hay "LED2=OFF"
    {
        digitalWrite(led2Pin,LOW);
        //se apagara LED-SENS.MOV.2
        est1=0;
        //estado de LED-SENS.MOV.2 sera 0
    }

    // si ha llegado al final de la línea (recibió una nueva
línea)
    // character y la línea está en blanco, la solicitud http
ha finalizado,
    // para que puedas enviar una respuesta
    if (c == '\n' && currentLineIsBlank) {
        // enviar un encabezado de respuesta http estándar
        // inicio del código http
```




```
//linea
    client.println("<h4 style='color: #2e6c80;'>GESTION DEL
ALUMBRADO N&ordm;1:</h4>");
// indica tamaño y color del texto: GESTION DEL ALUMBRADO PUBLICO
client.println("<table style='height: 240px; margin-left: auto;
margin-right: auto;' width='498'>");
client.println("<tbody>");
client.println("<tr style='height: 18px;'>");
client.println("<td style='width: 252px; height: 18px; text-
align: center;'>");
client.println("<h4><span style='color: #0000ff;'><em>SENSOR DE
LUMINOSIDAD&nbsp;</em></span></h4>");
client.println("</td>");
client.println("<td style='width: 252px; height: 18px; text-
align: center;'>");
client.println("<h4><span style='color: #0000ff;'><em>SENSOR DE
MOVIMIENTO</em></span></h4>");
client.println("</td>");
client.println("</tr>");
client.println("<tr style='height: 171.766px;'>");
client.println("<td style='width: 252px; height: 171.766px;'>");
//creacion de pulsadores para controlar LED-LDR1;
    client.println("<br /><br />");
    client.println("<b>DETECTA  ");
    client.print(ldrReading);
    client.println(" DE LUMINUSIDAD  </b><br />");
    client.println("<br /><br />");
    client.println("<b>ESTADO DE 1ra LUMINARIA:  ");
    client.print(estado1);
    client.println("</b><br />");
    client.println("<p></p>");
    client.println("<h4><span
                                style='color:
#0000ff;'><em>ENCENDER LUMINARIA:</em></span></h4>");
    client.println("<button
onClick=location.href='./?LED1=ON\'
style='margin:auto;background-color:
                                #84B1FF;color:
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");
    client.println("ON");
    client.println("</button>");
    client.println("<button
onClick=location.href='./?LED1=OFF\'
style='margin:auto;background-color:
                                #84B1FF;color:
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");
    client.println("OFF");
    client.println("</button>");
    client.println("</b></body>");
    client.println("</head>");
client.println("<p>&nbsp;</p>");
client.println("</td>");
```



```
client.println("<td style='width: 252px; height: 171.766px;'>");
//creacion de pulsadores para controlar LED-
SENS.MOV.1
    client.println("<b>");
    client.print(estadoperso);
    client.println("</b><br />");
    client.println("<br /><br />");
    client.println("<b>ESTADO DE 2da LUMINARIA: ");
    client.print(estadomov);
    client.println("</b><br />");
    client.println("<p></p>");
    client.println("<h4><span                    style='color:
#0000ff;'><em>ENCENDER LUMINARIA:</em></span></h4>");
    client.println("<button
onClick=location.href='./?LED2=ON\'
style='margin:auto;background-color:                    #84B1FF;color:
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");
    client.println("ON");
    client.println("</button>");
    client.println("<button
onClick=location.href='./?LED2=OFF\'
style='margin:auto;background-color:                    #84B1FF;color:
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");
    client.println("OFF");
    client.println("</button>");
    client.println("</b></body>");
client.println("<p>&nbsp;</p>");
client.println("</td>");
client.println("</tr>");
client.println("</tbody>");
client.println("</table>");
//creacion de pulsadores para controlar LED-LDR2;
client.println("<h4 style='color: #2e6c80;'>GESTION DEL ALUMBRADO
N&ordm;2:</h4>");
client.println("<table style='height: 240px; margin-left: auto;
margin-right: auto;' width='498'>");
client.println("<tbody>");
client.println("<tr style='height: 18px;'>");
client.println("<td style='width: 252px; height: 18px; text-
align: center;'>");
client.println("<h4><span style='color: #0000ff;'><em>SENSOR DE
LUMINOSIDAD&nbsp;</em></span></h4>");
client.println("</td>");
client.println("<td style='width: 252px; height: 18px; text-
align: center;'>");
client.println("<h4><span style='color: #0000ff;'><em>SENSOR DE
MOVIMIENTO</em></span></h4>");
client.println("</td>");
client.println("</tr>");
```



```
client.println("<tr style='height: 171.766px;'>");
client.println("<td style='width: 252px; height: 171.766px;'>");
//client.println("<div style='text-align:center;'>");
    client.println("<br /><br />");
    client.println("<b>DETECTA  ");
    client.print(ldrReading2);
    client.println(" DE LUMINUSIDAD  </b><br />");
    client.println("<br /><br />");
    client.println("<b>ESTADO DE 1ra LUMINARIA:  ");
    client.print(estado2);
    client.println("</b><br />");
    client.println("<p></p>");
    client.println("<h4><span                                style='color:
#0000ff;'><em>ENCENDER LUMINARIA:</em></span></h4>");
        client.println("<button
onClick=location.href='./?LED3=ON\'
style='margin:auto;background-color:                                #84B1FF;color:
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");
            client.println("ON");
            client.println("</button>");
        client.println("<button
onClick=location.href='./?LED3=OFF\'
style='margin:auto;background-color:                                #84B1FF;color:
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");
            client.println("OFF");
            client.println("</button>");
        client.println("</b></body>");
        client.println("</head>");
client.println("<p>&nbsp;</p>");
client.println("</td>");
client.println("<td style='width: 252px; height: 171.766px;'>");
    //creacion de pulsadores para controlar LED-SENS.MOV.2
    client.println("<b>");
    client.print(estadoperso2);
    client.println("</b><br />");
    client.println("<br /><br />");
    client.println("<b>ESTADO DE 2da LUMINARIA:  ");
    client.print(estadomov2);
    client.println("</b><br />");
    client.println("<p></p>");
    client.println("<h4><span                                style='color:
#0000ff;'><em>ENCENDER LUMINARIA:</em></span></h4>");
        client.println("<button
onClick=location.href='./?LED4=ON\'
style='margin:auto;background-color:                                #84B1FF;color:
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");
            client.println("ON");
            client.println("</button>");
```




```
        client.println("<button  
onClick=location.href='./?LED4=OFF\  
style='margin:auto;background-color:           #84B1FF;color:  
snow;padding: 10px;border: 1px solid #3F7CFF;width:65px;'>");  
        client.println("OFF");  
        client.println("</button>");  
        client.println("</b></body>");  
client.println("<p>&nbsp;</p>");  
client.println("</td>");  
client.println("</tr>");  
client.println("</tbody>");  
client.println("</table>");  
        client.println("</html>");  
        break;  
        //TERMINO DEL ENVIO DEL CODIGO HTTP  
    }  
    if (c == '\n') {  
        // Estás comenzando una nueva línea  
        currentLineIsBlank = true;  
    }  
    else if (c != '\r') {  
        // Has conseguido un personaje en la línea actual  
        currentLineIsBlank = false;  
    }  
} }  
} }  
// Darle tiempo al navegador web para recibir los datos  
delay(1);  
// CERRAR LA COEXION  
client.stop();  
}  
//si estado de LED-LDR1 es 0 (apagado)  
if (est==0){  
    //si el valor de entrada de la fotorresistencia es mayor a 2  
    if (ldrReading > 2){  
        digitalWrite(led1Pin, LOW);  
        //apagar LED-LDR1  
        estado1="APAGADA";  
        //el estado para mostrar al cliente sera 'APAGADA' del  
LED-LDR1  
    }  
    //de lo contrario si la fotoresistencia mayor a 2  
    else{  
        digitalWrite(led1Pin, HIGH);  
        //encender LED-LDR1  
        estado1="ENCENDIDA";  
        //el estado para mostrar al cliente sera 'ENCENDIDA'  
del LED-LDR1  
    }  
}
```



```
}  
//de lo contrario si estado de LED-LDR1 es 1 (encendido)  
else{  
    digitalWrite(led1Pin, HIGH);  
    //mantener encendido LED-LDR1  
    estado1="ENCENDIDA";  
    //el estado de LED-LDR1 para mostrar al cliente sera  
'ENCENDIDA'  
    }  
    // se realiza la lectura digital de SENSOR DE MOVIMIENTO  
movState = digitalRead(sensorMovimi);  
//si estado de LED-SENS.MOV.1 es 0 (apagado)  
if (est1==0){  
    //si el valor de entrada del SENSOR DE MOVIMIENTO esta en alto  
(1 logico)  
    if (movState == HIGH) {  
        digitalWrite(led2Pin, HIGH);  
        //encender LED-SENS.MOV.1  
        estadomov="ENCENDIDA";  
        // mostrar al cliente que LED-SENS.MOV.1 esta 'ENCENDIDA'  
        estadoperso="EXISTE CIRCULACION";  
        // indicar que 'EXISTE CIRCULACION'  
    }  
    //de lo contrario si el valor de entrada del SENSOR DE MOVIMIENTO  
esta en bajo (0 logico)  
    else {  
        digitalWrite(led2Pin, LOW);  
        //apagar LED-SENS.MOV.1  
        estadomov="APAGADA";  
        // mostrar al cliente que LED-SENS.MOV.1 esta 'APAGADA'  
        estadoperso="DESOLADO";  
        // indicar 'DESOLADO'  
    }  
    }  
    //de lo contrario si estado de LED-SENS.MOV.1 es 1 (encendido)  
else {  
    digitalWrite(led2Pin, HIGH);  
    //mantener encendido LED-SENS.MOV.1  
    estadomov="ENCENDIDA";  
    // mostrar al cliente que LED-SENS.MOV.1 esta 'ENCENDIDA'  
    estadoperso="EXISTE CIRCULACION";  
    // indicar que 'EXISTE CIRCULACION'  
    }  
//si estado de LED-LDR2 es 0 (apagado)  
if (est2==0){  
    //si el valor de entrada de la fotorresitecia 2 es mayor a 2  
    if (ldrReading2 > 2){  
        digitalWrite(led3Pin, LOW);  
        //apagar LED-LDR2
```



```
        estado2="APAGADA";
        //el estado de LED-LDR2 para mostrar al cliente sera
'APAGADA'
    }
    //de lo contrario si la fotoresistencia mayor a 2
    else{
        digitalWrite(led3Pin, HIGH);
        //encender LED-LDR1
        estado2="ENCENDIDA";
        //el estado de LED-LDR2 para mostrar al cliente sera
'ENCENDIDA'
    }
}
else{
    digitalWrite(led3Pin, HIGH);
    //mantener encendido LED-LDR2
    estado2="ENCENDIDA";
    //el estado de LED-LDR2 para mostrar al cliente sera
'ENCENDIDA'
}
// se realiza la lectura digital de SENSOR DE MOVIMIENTO 2
movState2 = digitalRead(sensorMovimi2);
//si estado de LED-SENS.MOV.2 es 0 (apagado)
if (est2==0){
    //si el valor de entrada del SENSOR DE MOVIMIENTO 2 esta en
alto (1 logico)
    if (movState2 == HIGH) {
        digitalWrite(led4Pin, HIGH);
        //encender LED-SENS.MOV.2
        estadomov2="ENCENDIDA";
        // mostrar al cliente que LED-SENS.MOV.2 esta 'ENCENDIDA'
        estadoperso2="EXISTE CIRCULACION";
        // indicar que 'EXISTE CIRCULACION' por ese alumbrado
    }
    //de lo contrario si el valor de entrada del SENSOR DE
MOVIMIENTO 2 esta en bajo (0 logico)
    else {
        digitalWrite(led4Pin, LOW);
        //apagar LED-SENS.MOV.2
        estadomov2="APAGADA";
        // mostrar al cliente que LED-SENS.MOV.2 esta 'APAGADA'
        estadoperso2="DESOLADO";
        // indicar 'DESOLADO'
    }
}
////de lo contrario si estado de LED-SENS.MOV.2 es 1 (encendido)
else {
    digitalWrite(led4Pin, HIGH);
    //mantener encendido LED-SENS.MOV.2
```



```
estadomov2="ENCENDIDA";  
// mostrar al cliente que LED-SENS.MOV.2 esta 'ENCENDIDA'  
estadoperso2="EXISTE CIRCULACION";  
// indicar que 'EXISTE CIRCULACION'  
}  
}
```



Anexo 2. Prueba de conectividad con ICMP

```
C:\Users\ADMIN>ping -n 1000 192.168.0.50
```

```
Haciendo ping a 192.168.0.50 con 32 bytes de datos:  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=9ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=247ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=5ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=11ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=12ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=5ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=38ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=20ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255  
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
```




Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=3ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=3ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=4ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=3ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=6ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255



Respuesta desde 192.168.0.50: bytes=32 tiempo=27ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=9ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=29ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=30ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=9ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=66ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255



Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=2ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=109ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=65ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=7ms TTL=255



Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=3ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=33ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=33ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=7ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=67ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=51ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=15ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=3ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255



```
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=88ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=5ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo=16ms TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=255
```

Estadísticas de ping para 192.168.0.50:

Paquetes: enviados = 1000, recibidos = 1000, perdidos = 0
(0% perdidos),

Tiempos aproximados de ida y vuelta en milisegundos:

Mínimo = 0ms, Máximo = 247ms, Media = 2ms

Anexo 3. Capturas de pantalla y fotos

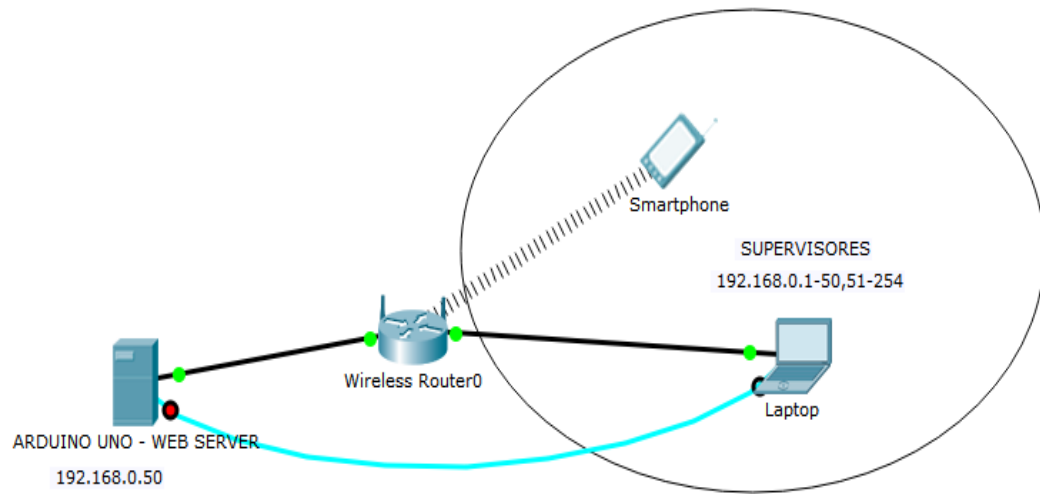


Figura N° 43: Diagrama de conexión Ethernet para realizar simulaciones.
Elaboración propia



Figura N° 44: Dirección MAC del Shield de Ethernet
Elaboración propia.



Figura N° 45: Forma de implementación de maqueta con dos leds (uno para la iluminación menos intenso y el otro para alta intensidad)
Elaboración propia.



Figura N° 46: Ubicación de la fotorresistencia para la implementación de la maqueta.
Elaboración propia.



Figura N° 47: Resultado cuando solamente se detecta movimiento el segundo alumbrado.

Elaboración propia.



Figura N° 48: Resultado en el navegador web cuando solamente se detecta movimiento el segundo alumbrado.

Elaboración propia.



Figura N° 49: Resultado cuando se detecta movimiento en sensores de movimiento de ambos alumbrados.
Elaboración propia.

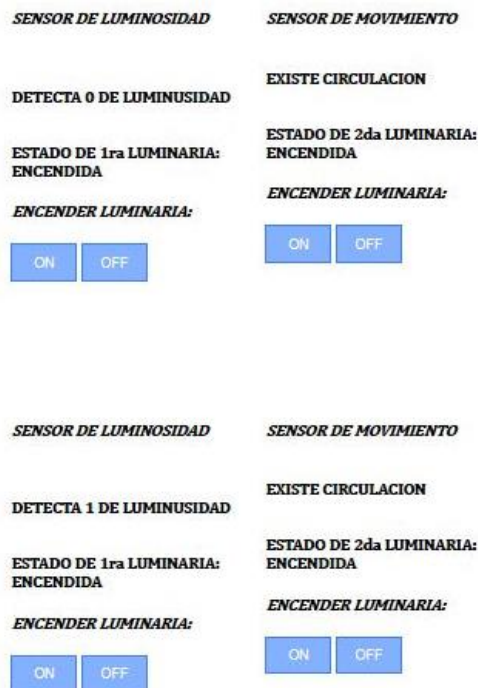


Figura N° 50: Resultado en el navegador web cuando se detecta movimiento en sensores de movimiento de ambos alumbrados.
Elaboración propia.

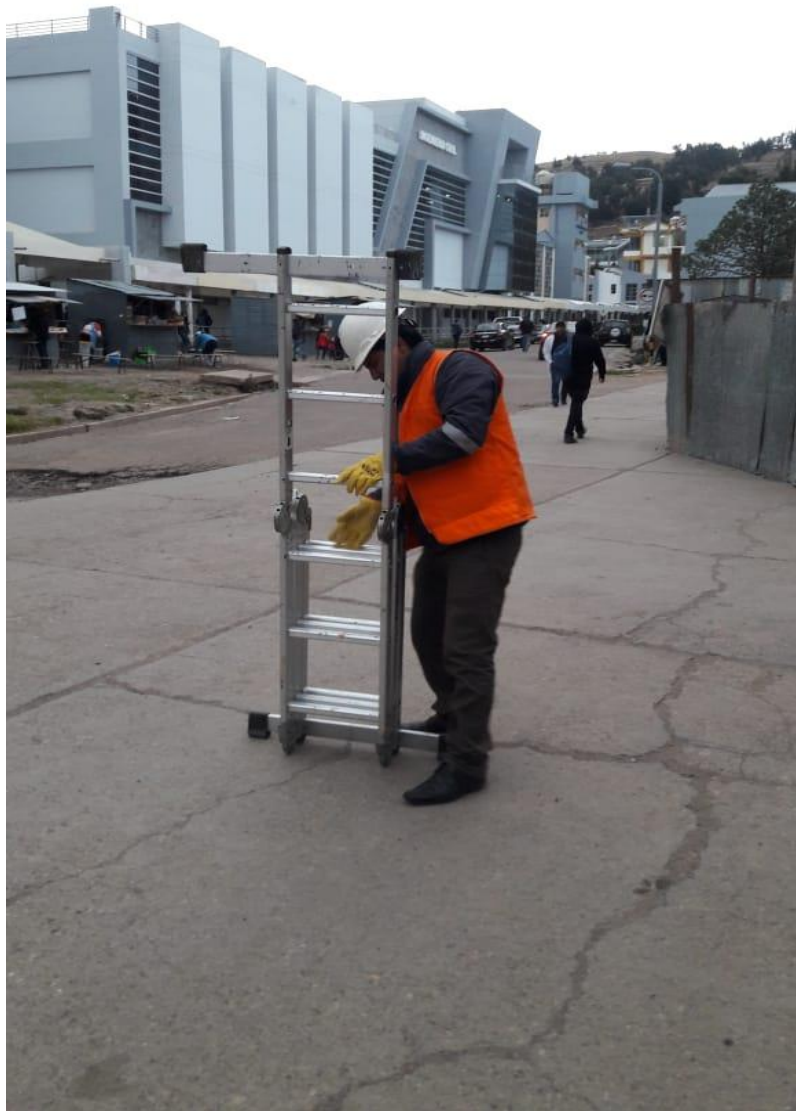


Figura N° 51: Pruebas en el campus de la universidad.
Elaboración propia.



Figura N° 52: Instalación del sensor de presencia.
Elaboración propia.



Figura N° 53: Instalación del sensor de presencia en el poste.
Elaboración propia.



Figura N° 54: Instalación del sensor de presencia a 6.6 metros sobre el nivel del suelo.
Elaboración propia.



Figura N° 55: Medición de la altura de un poste dentro del campus.
Elaboración propia.



Figura N° 56: Medición de la distancia entre dos postes.
Elaboración propia.



Figura N° 57: Medición de la distancia entre dos postes.
Elaboración propia.



Figura N° 58: Extensión de la herramienta para tomar medidas entre dos postes.
Elaboración propia.



Figura N° 59: Poste de muestra número 1.
Elaboración propia.



Figura N° 60: Poste de muestra número 2.
Elaboración propia.