



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INGENIERÍA,
INDUSTRIA Y CONSTRUCCIÓN**

CARRERA DE INGENIERÍA ELECTRÓNICA

**AUTOMATIZACIÓN DE LOS SISTEMAS DE ACCESO,
ILUMINACIÓN Y MONITOREO DEL LABORATORIO DE
ELECTRÓNICA DE LA UNIVERSIDAD CATÓLICA DE
CUENCA SEDE AZOGUES USANDO TECNOLOGÍA IoT**

**TRABAJO DE TITULACIÓN O PROYECTO DE INTEGRACIÓN
CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRÓNICO**

AUTOR: LUIS ENRIQUE AVILA GALLEGOS

DIRECTOR: ING. PAUL MATA QUEVEDO

AZOGUES - ECUADOR

2020

*Yo me gradué en los
50 años de La Cato!*

©Copyright Luis Enrique Avila G.
All Rights reserved

Dedicatoria

Dedico de forma especial a mi familia, esposa, mi hijo, mis padres y mis hermanos, ya que han sido el cimiento principal para la construcción de mi vida profesional. Ellos han sentado las bases de responsabilidad y deseos de superación.

A Dios por permitirme culminar esta etapa de mi vida profesional, por su ayuda, amor inquebrantable y bendiciones infinitas.

Agradecimiento

Gracias a Dios por permitirme tener y disfrutar de mi familia, gracias a mi familia por apoyarme siempre en cada decisión; gracias a mis tutores, al Ing.D. Luis Ismael Minchala Avila y al Ing.D. Paúl Mata Quevedo por el aporte brindado y a mis padres por permitirme cumplir con excelencia el desarrollo de este trabajo de investigación.

Gracias por creer en mí y gracias a Dios por las bendiciones que todos los días derrama sobre mi familia.

Resumen

Esta propuesta de investigación, requisito para la obtención del título de Ingeniero Electrónico, presenta el diseño de los sistemas de acceso, iluminación y monitoreo del laboratorio de electrónica de la Universidad Católica de Cuenca sede Azogues.

Este sistema integrado de automatización surge de las necesidades que presenta el laboratorio de electrónica y se ha implementado debido que este espacio posee demanda permanente de uso por parte de docentes y alumnos y no cuenta con tecnologías automáticas, para gestión de ingresos, gestión energética y seguridad. Adicionalmente, se ha diseñado un contexto de innovación y tecnología en el laboratorio a través del uso de sensores y equipos de automatización que permiten ofrecer un espacio tecnológico mejorado propiciando, entre otras cosas, la eficiencia energética.

La tecnología de base para la integración de los sistemas de automatización en el laboratorio es el internet de las cosas (IoT, por sus siglas en inglés), este sistema permite la adhesión de todos los equipos mediante la estructura *publisher/suscriber* del protocolo de comunicación MQTT. Con la implementación de este sistema de automatización se espera que los usuarios mejoren su percepción de confort, se ofrezca un espacio de innovación para el mejoramiento en calidad de enseñanza y se pueda inculcar la cultura de eficiencia energética.

Palabras Clave: Diseño, Automatización, IoT, Raspberry Pi, MQTT, Innovación, Tecnología.

Abstract

This research project, a requirement in order to graduate as an electronic engineer, describes the design of the entrance, lightning and supervision systems of the laboratory of electronics of the Catholic University of Cuenca-Azogues.

This automated equipment system is a result on the requirements of the laboratory of electronics and has been deployed due to the fact that this area has a constant use demand by teachers and students and has no automatic devices for managing revenues, energy management and safety. Furthermore, a framework of technology and innovation has been designed in the laboratory by using sensors and automated equipment allowing to provide an enhanced technological space by promoting, for example, energy efficiency.

The main technology for integrating automated systems in the laboratory is based on the Internet of Things (IoT). This technology allows all the equipment to be connected through the publisher/subscriber structure of the MQTT communication protocol. By implementing this automation system, hopefully the users will enhance their perception of comfort, offer a scope for innovation to increase teaching quality and encourage a new culture of energy efficiency.

Keywords: Design, Automation, IoT, Raspberry Pi, MQTT, Innovation, Technology.

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Definición del problema	2
1.3. Línea de investigación	2
1.4. Objeto de estudio	2
1.5. Objetivos	2
1.6. Hipótesis	3
1.7. Contribuciones	3
2. Marco teórico	4
2.1. Fundamentos de automatización en inmuebles	4
2.1.1. Domótica	4
2.1.2. Inmótica	4
2.2. Redes de sensores	5
2.2.1. Internet de las cosas	5
2.2.2. Protocolo de comunicación MQTT	6
2.3. Sistemas de automatización de bajo costo	7
2.3.1. Plataformas de hardware libre	7
2.3.2. Plataformas de software libre	8
2.4. Sistemas biométricos	9
2.4.1. Sensor de huella dactilar (AS608)	10
2.5. Sensor de corriente (SCT-000)	10
2.6. Sensor de movimiento	11
2.7. Módulo NODE MCU ESP 8266	12
3. Diseño construcción e implementación	13
3.1. Introducción	13
3.2. Instalación de openHAB	13
3.3. Diagrama de bloques de automatización	14
3.4. Control de acceso al laboratorio	15
3.4.1. Actualización del firmware del módulo ESP8266	16
3.4.2. Instalación de mosquitto en la raspberry y circuito de conexión	17
3.4.3. Case para el sistema de acceso automático	18

3.5.	Control de los circuitos de iluminación	19
3.5.1.	Seccionamiento del sistema de iluminación	19
3.5.2.	Creación de ítems en openHAB	20
3.5.3.	Integración de equipos LCN para el control de la iluminación	21
3.6.	Sistema de seguridad a través de cámara de monitoreo	22
3.7.	Verificación de presencia a través de sensores de presencia	23
3.8.	Registro del consumo energético del laboratorio	24
3.8.1.	Convertidor Pi16ADC	24
3.8.2.	Circuito acondicionador de señal	25
3.8.3.	Calibración del sensor SCT-000	26
3.8.4.	Circuito de conexión para el consumo energético	27
3.8.5.	Diseño de la interfaz	28
4.	Análisis de resultados	29
4.1.	Control de acceso al laboratorio	29
4.2.	Control de los circuitos de iluminación	31
4.3.	Sistema de seguridad a través de la cámara	33
4.4.	Detección de presencia	35
4.5.	Registro del consumo energético	36
5.	Conclusiones	40
5.1.	Conclusiones	40
5.2.	Trabajo futuro	41
A.	Algoritmos en Arduino	44
A.1.	Algoritmo para registro de la huella	44
A.2.	Algoritmo para lectura de las huellas	48

Índice de tablas

3.1. Items de openHAB	20
3.2. Tabla de valores medidos para la calibración del sensor SCT-000	26

Índice de figuras

2.1. Red de tecnología IoT	6
2.2. Gráfica de comunicación del protocolo MQTT	6
2.3. Microcontrolador Arduino Uno	7
2.4. Microcontrolador Raspberry Pi	8
2.5. Sensor de huellas dactilar	10
2.6. Sensor de corriente SCT-000	11
2.7. Sensor de movimiento PIR	12
2.8. Módulo WiFi ESP8266	12
3.1. Código para tener acceso al repositorio de openHAB	13
3.2. Código para agregar el repositorio estable	13
3.3. Código para sincronización del índice del paquete	14
3.4. Código para la instalación de openHAB	14
3.5. Diagrama de bloques de automatización del laboratorio	14
3.6. Diagrama de bloques de conexión	15
3.7. Circuito de conexión para actualización del firmware	16
3.8. Archivos SDK para la actualización del firmware del módulo ESP8266	16
3.9. Aplicación usada para actualizar el firmware del módulo ESP8266	17
3.10. Actualización completa del firmware	17
3.11. Instalación del protocolo MQTT en la raspberry pi	18
3.12. Circuito de conexión para el acceso al laboratorio	18
3.13. Diseño del <i>case</i> para la implementación del sistema automático	19
3.14. Circuito eléctrico de seccionamiento de iluminación	20
3.15. Código para la creación de ítems	21
3.16. Módulo LCN	21
3.17. Ítems creados para el encendido de la iluminación	21
3.18. Conexión de la cámara en la Raspberry Pi	22
3.19. Case diseñado para la cámara	22
3.20. Hat Pi Ez Connect	23
3.21. Regla creada en OpenHab	23
3.22. Configuración de la regla en la condición <i>when</i>	24
3.23. Configuración de la regla en la condición <i>then</i>	24
3.24. Convertidor analógico digital Pi16ADC	25

3.25. Circuito acondicionador diseñado	25
3.26. Curva de los valores medidos	26
3.27. Curva y ecuación de calibración del sensor SCT-000	27
3.28. Conexión de los componentes para el registro del consumo energético	27
3.29. Creación del canal para la interfaz de consumo energético	28
4.1. Cerradura instalada en la puerta de acceso al laboratorio	29
4.2. Instalación del <i>case</i> y sensor dactilar para la lectura de las huellas	30
4.3. Sensor de proximidad en estado pasivo	30
4.4. Sensor de proximidad en estado activo	31
4.5. Instalación de la pantalla <i>touch</i>	31
4.6. Control de los circuitos de iluminación desde la pantalla <i>touch</i> de raspberry	32
4.7. Control de los circuitos de iluminación desde un teléfono inteligente	32
4.8. Control de los circuitos de iluminación desde la pantalla de un ordenador	33
4.9. Instalación de los pulsantes para el control de la iluminación	33
4.10. Instalación y adecuación de la cámara	34
4.11. Transmisión en tiempo real de la cámara	34
4.12. Transmisión de la cámara desde la plataforma openHAB	35
4.13. Encendido de la iluminación a través del sensor de presencia	35
4.14. Instalación del Hat PiEz Connect	36
4.15. Instalación del sensor SCT-000 en el laboratorio	36
4.16. Instalación del circuito acondicionador diseñado	37
4.17. Instalación del convertidor analógico digital Pi16ADC	37
4.18. Datos obtenidos de la lectura del sensor de corriente en python	38
4.19. Gráfica del registro energético en Thingspeak	38
4.20. Interfaz de automatización del laboratorio de electrónica	39

Capítulo 1

Introducción

1.1. Introducción

Modificar el entorno físico involucra la ingeniosidad y gran parte del tiempo de las personas. Es por ello que existe una predilección a obtener la máxima comodidad con un mínimo esfuerzo en todas las actividades posibles del ser humano.

El controlar el medio que rodea permite un mayor grado de libertad y confort a las personas. Actualmente se ha visualizado una evolución grandiosa de las telecomunicaciones a través de internet. Cantidad de redes existentes funcionan con éxito, lo que los hacen imprescindibles para diversas áreas en la vida cotidiana. Implementar procesos que permiten la automatización en ambientes del hogar y oficinas es muy común en la actualidad.

El control a distancia ha surgido desde hace mucho tiempo atrás y gracias a la innovación tecnológica con la que hoy en día se cuenta se ha logrado robustecer el ambiente de interrelación de las personas con sistemas de control y telecomunicaciones. Gracias a esta importante evolución es como se logra controlar diversidad de dispositivos de forma remota, ya sea mediante internet, desde un teléfono inteligente, con la voz humana o simplemente desde una computadora.

Todas estas nuevas actividades de automatización que pueden ser realizadas por las personas dentro de una oficina o vivienda, como controlar las persianas o el nivel de iluminación desde un teléfono celular se encuentran dentro de un área llamada domótica.

El laboratorio de electrónica de la Universidad Católica de Cuenca sede Azogues, posee demanda permanentemente de uso por parte de docentes y alumnos; el estado en el que se encuentra esta zona de enseñanza y aprendizaje no es eficiente ya que no cuenta con tecnologías automáticas como lo requiere este espacio.

Por esta razón a través del conocimiento del avance de la tecnología se ha diseñado un protocolo de automatización que permite mantener el control de los sistema de acceso, iluminación y monitoreo del laboratorio mediante el uso del internet de las cosas (IoT, por sus siglas). Con la implementación del sistema se prevee ofrecer calidad y confort en la estancia de los usuarios.

1.2. Definición del problema

La automatización de los sistemas de accesos, iluminación y monitoreo del laboratorio de electrónica de la Universidad Católica de Cuenca sede Azogues es una necesidad que nace a través de la demanda continua del uso del laboratorio por parte de docentes y alumnos; además no cuenta con tecnologías automáticas para gestión de ingresos, gestión energética y seguridad, de esta forma la solución para este problema de investigación parte desde tecnología llamada domótica. La domótica es utilizada en los hogares y oficinas con la finalidad de aumentar la comodidad de las personas, la eficiencia en la energía, el ocio y la seguridad. La automatización de los hogares es el impulso al aumento de la calidad de vida y bienestar de las personas [Gutierrez *et al.*, 2018].

El laboratorio de electrónica presenta la necesidad de controlar la eficiencia energética, monitorizar y mantener en alerta la seguridad y de controlar el acceso al laboratorio, ya que en este sector se encuentran equipos de medición de alto costo. Para ello es necesario contar con diversos dispositivos con los cuales se logrará realizar cada proceso. La instalación de sistemas domóticos se encuentra en precios elevados, es por esta razón que este tipo de sistemas se encuentra con frecuencia en hogares de lujo, ya que para la instalación utilizan canales de transmisión de costos elevados, lo que permite obtener una solución robusta como EIB, X10, CeBus, LonWorks, ZigBee, etc [Lledó Emilio *et al.*, 2012].

Hoy en día se utilizan diversos modelos para conectar los dispositivos ya sea intra o inter sistemas domóticos, a través de comunicación inalámbrica o mediante cables, un solo estándar no es suficiente, sino más bien es necesario orillarse por la coexistencia de múltiples interfaces de comunicación ya que con ello se podrían satisfacer diversas demandas de forma única [Zamora-Izquierdo *et al.*, 2010].

Para [Flores *et al.*, 2018], los protocolos de comunicación tienen diferentes capacidades, ya que brindan soluciones accesibles y fiables a la domótica, sin embargo algunos de estos utilizan sus configuraciones de baja a mediana complejidad.

La implementación de este proceso automático y tecnológico es importante ya que brinda a los usuarios del laboratorio confort e impulsa a los mismos a inmiscuirse en la tecnología y el desarrollo de la domótica.

¿En qué medida un sistema de automatización aporta en la seguridad y eficiencia energética del laboratorio de electrónica de la Universidad Católica de Cuenca sede Azogues?

1.3. Línea de investigación

Ciencias exactas, naturales y tecnológicas.

1.4. Objeto de estudio

Control industrial, monitoreo y automatización.

1.5. Objetivos

Objetivo general

- Desarrollar un sistema de automatización de los sistemas de iluminación, acceso y monitoreo del laboratorio de electrónica de la Universidad Católica de Cuenca, sede Azogues.

Objetivos específicos

- Controlar el acceso al laboratorio.
- Controlar los circuitos de iluminación del laboratorio.
- Obtener un sistema de seguridad a través del uso de una cámara de monitoreo.
- Verificar la presencia a través de sensores de presencia.
- Realizar un registro del consumo energético del laboratorio.

1.6. Hipótesis

Un sistema de automatización aporta en gran medida a la seguridad, confort y eficiencia energética, tanto en un laboratorio como en un edificio o residencia ya que facilita la comodidad para controlar diferentes sistemas en los mismos.

1.7. Contribuciones

La aplicación de la metodología de estudio y desarrollo de este trabajo ha generado las siguientes contribuciones:

- Diseño de sistemas de automatización de bajo costo.
- Desarrollo de sistemas domóticos utilizando tecnología IoT.
- Interacción de equipos para el desarrollo de procesos automáticos.
- Lenguajes de programación y estrategias de automatización.

Capítulo 2

Marco teórico

Este capítulo presenta la información necesaria para el desarrollo del trabajo de investigación, enfocandonos en los diferentes dispositivos, tecnologías, protocolos y plataformas existentes previos a ser utilizados.

2.1. Fundamentos de automatización en inmuebles

2.1.1. Domótica

La domótica comienza a inicios de los años 70, como resultado del sin número de investigaciones fue como surgieron los primeros dispositivos destinados a la automatización de edificios basados en la exitosa tecnología de esa época X-10. La domótica se aplica a la ciencia y a los elementos que ofrecen niveles de automatización en el interior de una residencia. También se encuentra asociada en el ámbito de las telecomunicaciones, el cual se denomina hogar digital [Huidobro and Milán Tejedor, 2010].

La domótica surge de la ingeniería y su objetivo se encuentra destinado a zonas domésticas o pequeñas, con la finalidad de automatizar las funciones que tienen los dispositivos de una instalación (iluminación, climatización, seguridad, etc..) y su integración en un sistema que permite un funcionamiento ordenado y coordinado de todas ellas [Sarasúa, 2011]. De esta forma la domótica es la integración e instalación de múltiples redes y dispositivos electrónicos en el hogar u oficina. Para lograr que estos dispositivos trabajen de forma conjunta es necesario que se encuentren conectados a una red interna [Huidobro and Milán Tejedor, 2010]

Hoy en día se utilizan diversos términos de acuerdo al ámbito de la aplicación de estos sistemas de automatización: [Sarasúa, 2011].

- Domótica: usado en el sector doméstico o terciarios pequeños.
- Inmótica: es usado netamente para el sector terciario.
- Urbótica: usado para la ciudad.

2.1.2. Inmótica

La inmótica son las tecnologías aplicadas a la automatización y el control inteligente de los edificios no destinados a vivienda, tales como hoteles, centros comerciales, oficinas de negocio, universidades, escuelas, hospitales, lo que contribuye a una gestión eficiente del uso de la energía, además de brindar confort, seguridad y comunicación

entre el sistema y los usuarios [CEDOM, 2016]. Gran parte de los sistemas eléctricos y electrónicos pertenecientes a edificios terciarios contribuyen un problema, la ineficacia es lo más evidente, ya que el resultado de este es el gasto innecesario y excesivo de diferentes recursos energéticos. Lo cual es un problema para la parte económica y para el medio ambiente. Esta ausencia de control provoca además incomodidades, incapacidades para visualizar desviaciones energéticas, mal gasto de energía y falta de condiciones óptimas para verificar casos de emergencia [Constantino, 2011].

2.2. Redes de sensores

Wireless Sensor Network (WSN) es un sistema de red de sensores que contiene varios dispositivos colocados espacialmente, que utiliza sensores con la finalidad de monitorizar diversidad de condiciones en diversos puntos ya sea temperatura, humedad, movimiento o contaminantes. Estos dispositivos son unidades que tienen un microcontrolador, una fuente de energía, un sensor y un radiotransceptor [Aakvaag and Frey, 2006].

Debido a que la vida de la batería se encuentra limitada, los nodos se contruyen teniendo presente conservar la energía es por ello que pasan mucho tiempo en modo durmiente (sleep). WSN tiene la capacidad de autorestaurarse, esto quiere decir que si se daña algun nodo, la red busca nuevas vías para el envío de los paquetes de datos [Aakvaag and Frey, 2006].

Este sistema forma un importante bus de comunicación entre el mundo virtual y el mundo físico real, la gama de aplicacion de WSN es extensa ya que de forma potencial contribuye a la industria, ciencia, el transporte, la seguridad e infraestructura [Thomson Gale (Firm) and Buitrago, 2013].

En el contexto de IoT, WSN tiene un papel importante para aumentar la ubicuidad de las redes, a los objetos inteligentes que estan constituidos por la red de sensores y actuadores inalámbricos (WSAN), se les asigna direcciones IP, con la finalidad que formen parte de internet y de esta forma aprovechen los servicios de monitorización y control de los dispositivos. Lo que conlleva a la conexión a internet de objetos físicos y digitales. La tecnología inalámbrica forma parte de una vía importante para realizar una comunicación entre los objetos inteligentes y a Internet. La red de sensores inalámbricos (WSN), tiene gran importancia debido a que es la tecnología que permite la escalabilidad del IoT con la funcionalidad para brindar una integración a Internet [Cama *et al.*, 2012].

2.2.1. Internet de las cosas

El IoT ha evolucionado a partir de una serie de tecnologías, que se han desarrollado independientemente desde el origen. IoT, tiene la facilidad de conectarse a internet en cualquier momento y desde cualquier lugar. Esta tecnología consiste en una agrupación de sensores y dispositivos que permanecen conectados a internet mediante redes fijas o inalámbricas.

Debido a que internet se encuentra presente en todas partes, esto facilita la adopción masiva de esta tecnología. Dado el tamaño y costo de los sensores, estos son fácilmente adaptables en oficinas, hogares y lugares públicos, con ello cualquier objeto puede ser conectado a internet y convertirse en una fuentes de datos [Jesús González García *et al.*, 2017].

En la Figura 2.1 se presenta la red de la tecnología IoT.

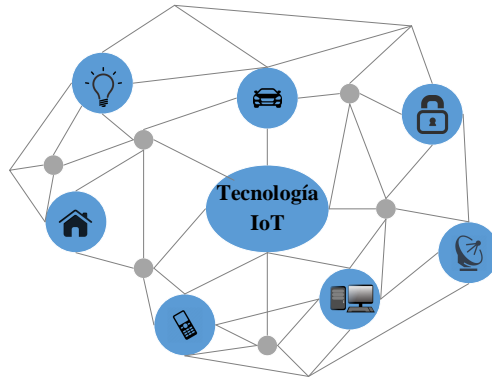


Figura 2.1: Red de tecnología IoT

Fuente: Autor.

2.2.2. Protocolo de comunicación MQTT

MQTT es un protocolo de mensajería de intercambio que utiliza el estándar public-subscribe para enviar mensajes entre un servidor y clientes, este protocolo funciona mediante TCP/IP y se puede ejecutar a través de protocolos de red con servicios ordenados. MQTT es un buen protocolo para establecer la comunicación máquina a máquina y los contextos de IoT, está diseñado para ser ligero, abierto y fácil de implementar, especialmente en contextos donde internet resulta costoso, tiene bajo ancho de banda [Oliveira *et al.*, 2018].

El intercambio de mensajes desde el patrón public-suscribe se realiza mediante un servidor llamado el corredor, éste filtra los mensajes y los distribuye a los clientes según un identificador que cada mensaje tiene [Oliveira *et al.*, 2018].

En la Figura 2.2 se muestra e funcionamiento del protocolo de comunicación MQTT.

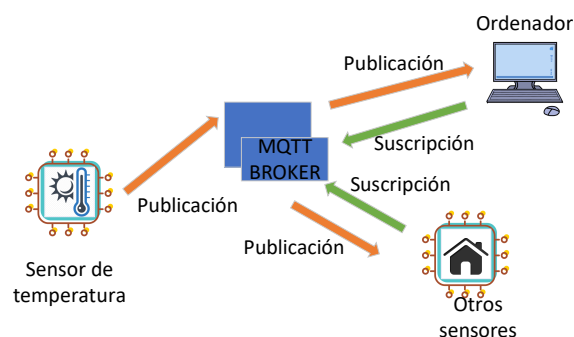


Figura 2.2: Gráfica de comunicación del protocolo MQTT

Fuente: Autor.

2.3. Sistemas de automatización de bajo costo

La automatización de bajo costo se inició en Europa y es uno de los factores mas importantes que esta transformando las tecnologías de automatización. C. Linsky y R. de Groot, fueron quienes trabajando en la Organización de Cooperación y Desarrollo Económico, inculcaron un programa para las pequeñas empresas, con la finalidad que disfrutaran de ventajas de automatización con las que las grandes industrias ya contaban, a través del empleo de equipos económicos faciles de instalar y con un costo accesible[ONUDI, 2011].

2.3.1. Plataformas de hardware libre

En los últimos años debido a su decrecimiento acelerado del costo de fabricación ha incrementado potencialmente el desarrollo de hardware libre y computadoras de placa, lo que conlleva al uso de estas plataformas en numerosas aplicaciones. El hardware libre es de acceso público y puede ser estudiado, recreados sin ninguna restricción por cualquier persona, esto permite experimentar con plataformas emergentes y crear aplicaciones diferentes para una plataforma.

Microcontroladores Arduino

El microcontrolador Arduino es una plataforma de código libre (open source) para prototipos de electrónica, está fomentada en hardware y software fácil de usar. Arduino puede manipular el entorno a través de la recepción de entradas mediante una variedad de sensores y controlar su alrededor con luces, motores entre otros artefactos. El ensamblaje de estas placas se puede realizar a mano, el software para su programación es gratuito[Enríquez Herrador, 2009].

El lenguaje de Arduino está basado en *Wiring*, y su simplicidad hace que su uso sea muy sencillo para cualquier persona, ya que el conocimiento de algoritmos o códigos no son necesarios. Su rápido crecimiento se debe a su naturaleza *open source*, no se requiere ningun tipo de licencia para su desarrollo, por lo que todos los archivos para sus aplicaciones están disponibles. El software de esta plataforma tiene un circuito integrado, el cual necesita ser programado para realizar cualquier proyecto[Casco, 2014].

En la Figura 2.3 se muestra el microcontrolador Arduino Uno.

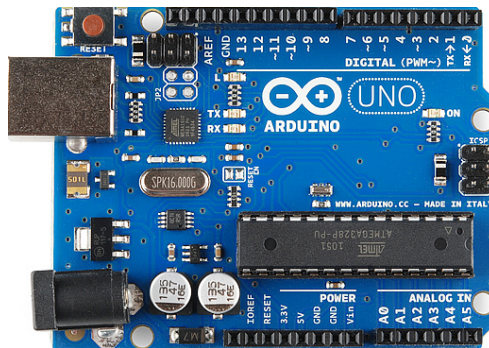


Figura 2.3: Microcontrolador Arduino Uno

Fuente: [Casco, 2014]

Raspberry pi

Esta plataforma se encuentra diseñada a modo de un miniordenador, el lenguaje que utiliza es de alto nivel como Python, C++ y Java. Raspberry Pi surgió por la ineficiencia de estudiantes en detalles técnicos de computación. Siendo una computadora en miniatura de bajo costo de alto desempeño que ayuda a interactuar con los ordenadores de diversas formas[Casco, 2014]. Raspberry Pi posee todos los componentes necesarios para funcionar un sistema operativo. Usa el controlador Broadcom, que es un SoC (System on Chip), el cual tiene un procesador ARM11 que funciona a 700MHz. Raspberry Pi no tiene incorporado un display, sin embargo puede ser usado con un display HDTV, consta de un puerto Ethernet, lo que facilita la conexión a internet, además se puede cargar varios sistemas operativos como Mac, Windows y Linux. La capacidad para correr Linux lo convierte en una opción ideal para realizar servidores web dedicados[Casco, 2014].

En la Figura 2.4 se muestra el microcontrolador Raspberry Pi.



Figura 2.4: Microcontrolador Raspberry Pi

Fuente: [Casco, 2014]

2.3.2. Plataformas de software libre

Python

Python está considerado como un lenguaje de programación potente de fácil utilización y aprendizaje. Tiene estructuras de datos de alto nivel en la programación orientada a objetos. Es ideal para *scripting* y creación de aplicaciones en diversas áreas. El intérprete de este lenguaje de programación se encuentra a disposición para las diferentes plataformas y su distribución es libre. Python no es un lenguaje de propósito general ya que no está orientado a un fin concreto, con python es posible crear páginas sin necesidad de poseer un alto nivel de conocimiento, además se puede crear scripts o software para windows. La programación orientada a objetos es el fuerte de esta plataforma, sin embargo es posible también otros estilos de programación para los usuarios como la programación imperativa o funcional.

Python tiene la posibilidad de ser usado para diversos sistemas operativos como Unix, Linux, Windows y sistemas Mac y Os; además es de tipado dinámico, esto quiere decir que no es necesario indicarle a python que tipos son los

datos (string, int, float), ya que se adaptan a lo que se escribe al ejecutar el programa.

Plataformas IoT

La plataforma de internet de las cosas (IoT, por sus siglas en inglés), es el corazón de las tecnologías, ya que permite la conexión y el intercambio de la información que se requiere transmitir. Con la nueva era del internet, existen diversas plataformas con las que se puede interactuar los proyectos de internet de las cosas, surgiendo una clasificación de acuerdo al costo y el sector al que va orientado cada plataforma. De esta manera existen plataformas orientadas a startups y empresas pequeñas, su uso es de forma gratuita y presenta limitaciones en cuanto al número de dispositivos conectados y mensajes enviados, además existen plataformas que ofertan servicios gratuitos centrado en ofrecer servicios globales a sistemas basados en internet de las cosas y que permite almacenar webs, api para móviles y bases de datos.

Dentro de las plataformas de IoT se encuentran aquellas que las grandes empresas como Google, Amazon, Microsoft ofrecen, las cuales están orientadas a la industria y grandes proyectos de IoT. Existen además plataformas de código abierto que ofrecen acceso al mismo sin restricciones y que se obtienen fácilmente en internet, como las que se presentan a continuación:

- Thingspeak: es una plataforma que toma y almacena datos de sensores en la nube, además permite visualizar datos en matlab y actuar sobre los mismos.
- aREST Framework: es de código abierto y uso gratuito, brinda soporte Arduino y Raspberry Pi, permite crear aplicaciones con servicios RESTful, soporta comunicaciones wifi, Ethernet, Bluetooth y serial.
- Thingier.io: es una plataforma de origen española de código abierto, de programación sencilla y compatible con Arduino, Raspberry Pi y ESP8266. Tiene una consola de administración que permite gestionar y geolocalizar los objetos conectados, la documentación es extensa y libre.
- Cayenne: es sencillo de usar y conectar cualquier dispositivo, dispone de librerías para Arduino de fácil descarga, la plataforma contiene diversos dispositivos pre-configurados.
- Arduino Cloud: es compatible con cualquier microcontrolador Arduino, utiliza el protocolo MQTT para la transmisión de datos.
- Samsung Artik Cloud: es una plataforma creada por Samsung para IoT, se encuentra en la nube y fue creada para dar soporte a dispositivos médicos, soporta IFTTT lo que permite comunicación con cualquier cosa.
- OpenHab: es una plataforma creada para la automatización del hogar de código abierto y tecnología independiente, el cual funciona como el centro del hogar inteligente.

2.4. Sistemas biométricos

Un sistema de identificación personal se realiza a través de la autenticación de una entidad específica relacionada con la persona, mediante un identificador que la persona posee (una llave, una tarjeta, una huella). Un sistema biométrico es un sistema de reconocimiento de identidad de una persona con una determinada característica fisiológica. La autenticación personal se genera a través de algo que la persona es; como un rasgo fisiológico personal, una huella dactilar, el iris. Las características fisiológicas más comunes en las que se basa un reconocimiento biométrico

son: la huella dactilar, la huella palmar, la gemometría de la mano/dedos, la cara, el iris o la retina [Danilo *et al.*, 2003].

2.4.1. Sensor de huella dactilar (AS608)

La huella dactilar es el rasgo biométrico más utilizado por la humanidad para la identificación de personas, contiene todos los algoritmos necesarios para verificar una huella digital, se utilizan comúnmente en cajas de seguridad, equipos checadores y controles de acceso. El sensor de huellas dactilares, es un sistema de reconocimiento biométrico que integra un DSP, basado en el procesamiento digital de una imagen de una huella, la cual se almacena a través de un sensor capacitivo, para posteriormente transmitir a un computador mediante una interfaz de comunicación para la comparación de templates y encontrar a los usuarios enrolados. La conexión del módulo se realiza con una interfaz serial asíncrona con niveles TTL, es posible almacenar 162 huellas en la memoria del lector de huella digital, el sensor posee un led que indica su funcionamiento y asiste en la lectura de la huella [Win and Nyein, 2019].

La Figura 2.5 muestra el sensor dactilar AS608.

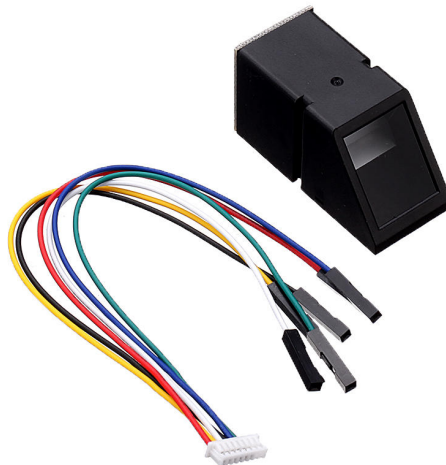


Figura 2.5: Sensor de huellas dactilar
Fuente: [Win and Nyein, 2019]

2.5. Sensor de corriente (SCT-000)

Los sensores de corriente utilizan diferentes métodos de detección, entre las cuales usan un elemento de conversión magnética insertado en un bobinado. El sensor SCT-000 es un sensor de corriente no invasivo, permite medir la corriente atravesada por un conductor, son transformadores de corriente y su medición se obtiene por inducción electromagnética. Dentro de la familia de SCT-000 existen diferentes modelos que brindan la medición como una

salida de corriente o tensión. El transformador de corriente del sensor SCT-000 es similar a un transformador de tensión, el cual se basa en los mismos principios de funcionamiento, la diferencia es que perciben objetivos diferentes. El sensor SCT-000 oferta una precisión de 1-2 %, pero para provocar esto es necesario que el núcleo ferromagnético este correctamente cerrado, a través del sensor de corriente se pretende monitorizar el consumo de energía en diversos puntos [Llamas, 2018].

La Figura 2.6 presenta el sensor de corriente SCT-000.



Figura 2.6: Sensor de corriente SCT-000

Fuente: [Llamas, 2018]

2.6. Sensor de movimiento

Este sensor capta el movimiento y actúa a través de este estado, el sensor PIR funciona con dos elementos detectores separados, de forma que la señal diferencial entre las dos activa la alarma del movimiento. Tiene la función de ajustar los parámetros de detección mediante dos potenciómetros, uno para la distancia y otro para la sensibilidad.

El rango de detección del sensor es ajustable y funciona con alcances de 7 metros con aperturas de 90° a 110°. Este módulo es de bajo costo, fácil de manipular e incorpora tecnologías recientes en sensores de movimiento, entre las especificaciones se encuentran: [Rogers *et al.*, 2018]

- Usa el PIR LH1778 y el controlador BISS0001.
- Voltaje de alimentación: 5 a 12 VDC.
- Consumo promedio: menor a 1 mA.
- Rango de distancia de 3 a 7 metros ajustable.
- Ángulo de detección: cono de 110°.
- Tiempo de inicialización.
- Tiempo de salida inactiva.

La Figura 2.7 muestra el sensor de movimiento.

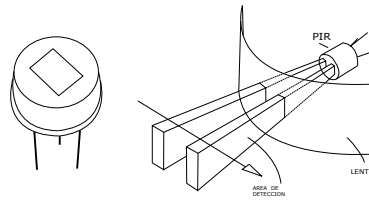


Figura 2.7: Sensor de movimiento PIR

Fuente: Autor.

2.7. Módulo NODE MCU ESP 8266

El módulo ESP8266 es una placa que brinda una conexión entre dispositivos inalámbricos, se encuentra basado en el sistema SOC (Sistema con chip), lo que permite crear una red de internet de las cosas (IoT, por sus siglas en inglés) de forma económica. La Figura 2.8 presenta el módulo de WiFi ESP8266.

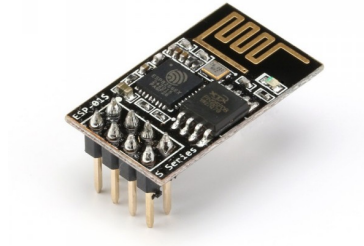


Figura 2.8: Módulo WiFi ESP8266

Fuente: Autor.

Capítulo 3

Diseño construcción e implementación

3.1. Introducción

Este capítulo presenta la estructura para la automatización del laboratorio y los diferentes accesorios utilizados para la implementación de los sistemas. A continuación se describe los métodos y propuestas utilizados para la solución.

3.2. Instalación de openHAB

La automatización del laboratorio de electrónica se realizó con la plataforma openHAB, se decidió usar este software debido a la facilidad de integración de diversos equipos y la independencia de hardware y protocolos de comunicación que se utilice. Además, esta plataforma permite la interoperabilidad lo que representa un fuerte importante para combinar varias soluciones domóticas. La instalación de openHAB se realizó a través del terminal de raspberry pi y el proceso para ello se obtuvo del repositorio de dicha plataforma, lo primero fue agregar la contraseña y permitir que apt utilice el protocolo HTTPS como se muestra en la Figura 3.1

```
wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' | sudo apt-key add -  
sudo apt-get install apt-transport-https
```

Figura 3.1: Código para tener acceso al repositorio de openHAB

Fuente: [openHAB, 2015]

Para obtener una instalación permanente de la plataforma se agregó el repositorio estable en el terminal de la raspberry, el código se muestra en la Figura 3.2

```
echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' | sudo tee /etc/apt/sources.list.d/openhab2.list
```

Figura 3.2: Código para agregar el repositorio estable

Fuente: [openHAB, 2015]

Posteriormente se realizó la sincronización del índice de los paquetes como se observa en la Figura 3.3

```
sudo apt-get update
```

Figura 3.3: Código para sincronización del índice del paquete

Fuente: [openHAB, 2015]

Finalmente, se procedió a instalar openHAB, es importante tener presente que para ello es necesario que la raspberry pi tenga conexión a internet. Con este procedimiento ya fue posible tener acceso a la plataforma. El código usado para la instalación se observa en la Figura 3.4

```
sudo apt-get install openhab2
```

Figura 3.4: Código para la instalación de openHAB

Fuente: [openHAB, 2015]

3.3. Diagrama de bloques de automatización

La Figura 3.5 presenta el diagrama de bloques que conforma los diferentes equipos y accesorios para la automatización del laboratorio.

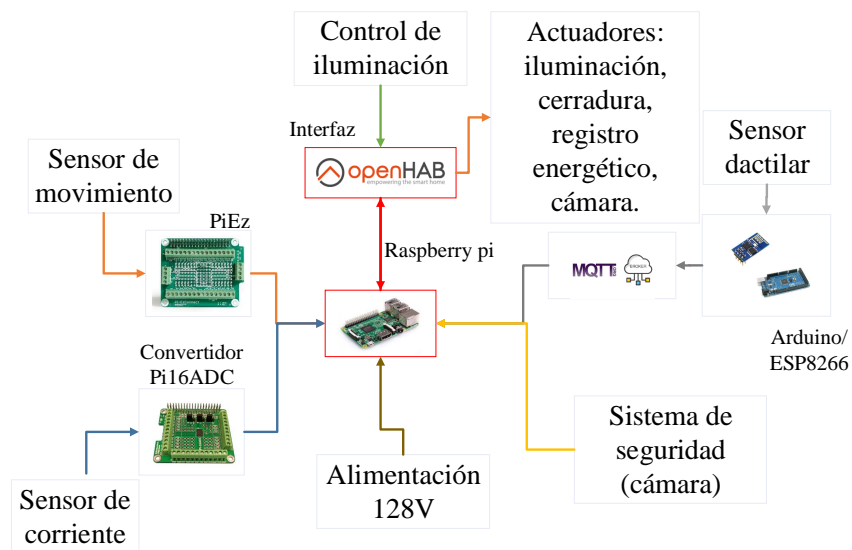


Figura 3.5: Diagrama de bloques de automatización del laboratorio

Fuente: Autor.

3.4. Control de acceso al laboratorio

La solución a la propuesta de investigación comienza por realizar el control de acceso del laboratorio de electrónica. Este proceso se desarrolló a través de un sensor de huellas dactilares digital, que se instaló en la puerta de ingreso del espacio educativo, en este componente se grabaron las huellas mediante la programación del microcontrolador Arduino.

El sensor de huella dactilar tiene la posibilidad de tomar, guardar e identificar la huella de los dedos. Este dispositivo guarda las huellas en su memoria para luego compararlas y reconocerlas con una capacidad de escaneo y almacenamiento de 1 segundo y 128 huellas respectivamente. El sensor posee dos cables para alimentación y dos para la comunicación serial.

La huella de cada persona es diferente por lo que esta opción fue ideal para brindar acceso al laboratorio de forma automática. La comunicación del sistema de acceso se realiza entre Arduino mediante el puerto serial y hacia la Raspberry a través del módulo de wifi ESP8266 y el protocolo de comunicación MQTT. En la Figura 3.6 se muestra el diagrama de bloques del control de acceso.

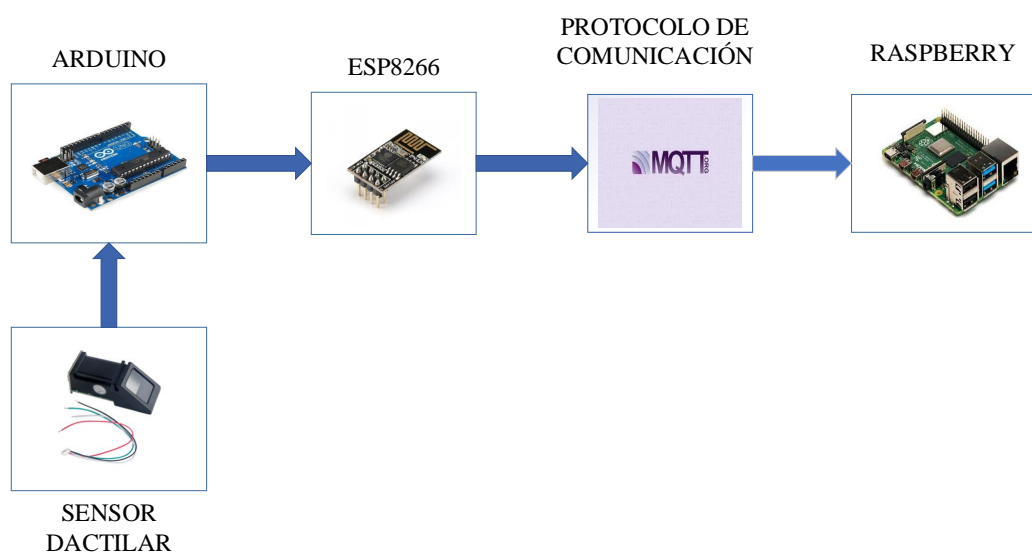


Figura 3.6: Diagrama de bloques de conexión

Fuente: Autor.

El diagrama de bloques presenta la estructura del funcionamiento del sistema de acceso al laboratorio. El sensor dactilar compara la huella registrada a través de la programación en Arduino, si la confirmación es correcta se envía un dato mediante el módulo wifi ESP8266 a través del protocolo de comunicación MQTT hacia la Raspberry Pi con la finalidad de deshabilitar la cerradura magnética y permitir el acceso al laboratorio.

3.4.1. Actualización del firmware del módulo ESP8266

Para establecer la conexión del módulo ESP8266 a la red wifi fue necesario actualizar el firmware del módulo y para ello se realizó una serie de procesos que permitieron obtener esta actualización.

La Figura 3.7 muestra el diagrama de conexión para habilitar el módulo ESP8266 en modo de programación.

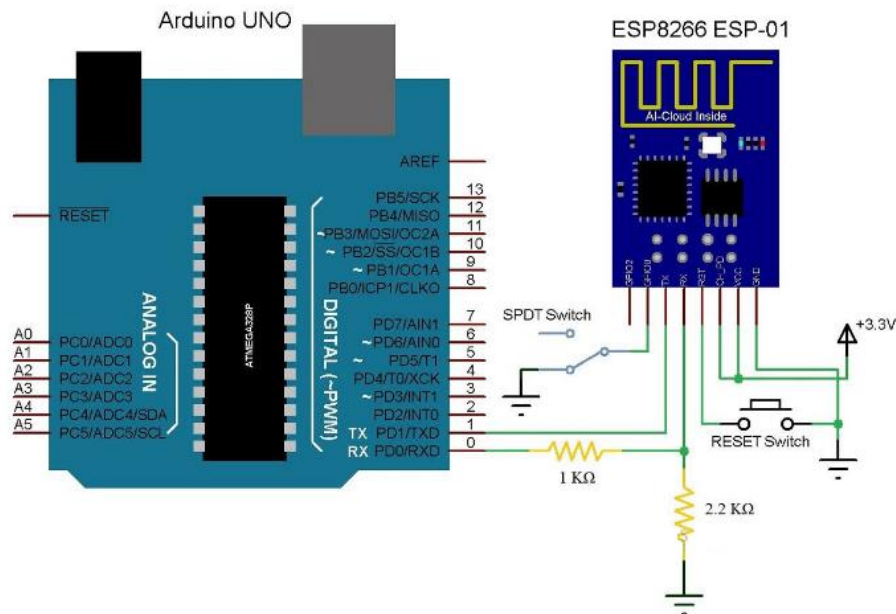


Figura 3.7: Circuito de conexión para actualización del firmware

Fuente: [Alga, 2017].

Los instaladores (SDK) usados para la actualización del firmware del módulo ESP8266 se muestran en la Figura 3.8 y se obtiene de la siguiente página: <https://www.espressif.com/en/products/hardware/esp8266ex/resources>.

ste equipo > Descargas > ESP8266_AT_Bin_V1.7 > bin > at > 1024+1024				Buscar en ...
Nombre	Fecha de modifica...	Tipo	Tamaño	
user1.2048.new.5.bin	23/8/2018 2:09	Archivo BIN	445 KB	
user2.2048.new.5.bin	23/8/2018 2:09	Archivo BIN	445 KB	

Figura 3.8: Archivos SDK para la actualización del firmware del módulo ESP8266

Fuente: Autor.

La aplicación utilizada para la actualización del firmware del módulo ESP8266 se denomina NodeMCU y se muestra en la Figura 3.9.

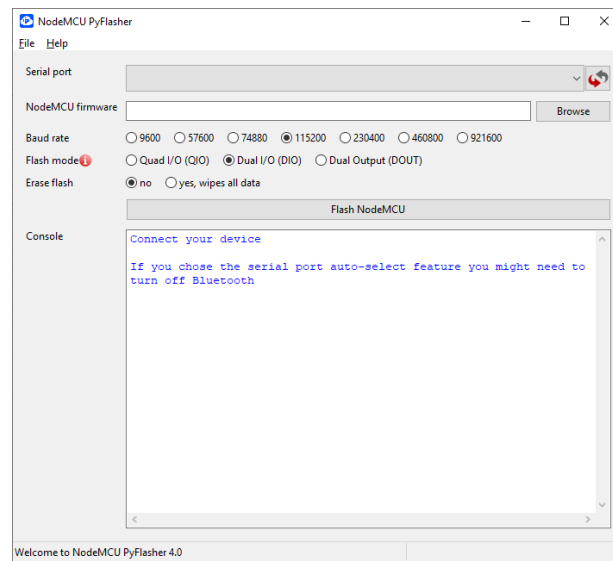


Figura 3.9: Aplicacion usada para actualizar el firmware del módulo ESP8266

Fuente: Autor.

A través de la aplicación y mediante los instaladores SDK, se procedió a realiza la actualización. En la Figura 3.10 se presenta la actualización del firmware completo.

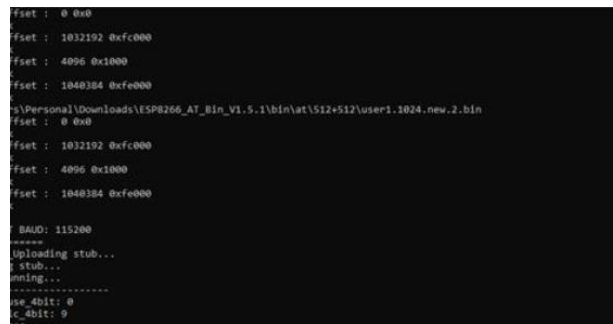


Figura 3.10: Actualización completa del firmware

Fuente: Autor.

3.4.2. Instalación de mosquitto en la raspberry y circuito de conexión

Mosquitto permite la comunicación entre el módulo ESP8266 y la tarjeta Raspberry Pi, su funcionamiento está establecido en un protocolo de mensajería publicación/suscripción. Su uso tiene como objetivo minimizar el ancho de banda y brindar una comunicación bidireccional entre los dispositivos. En la Figura 3.11 se muestra el código utilizado para la instalación de MQTT en la raspberry pi.

```

pi@pi: ~
login as: pi
pi@192.170.1.110's password:
Linux pi 4.19.66-v7+ #1253 SMP Thu Aug 15 11:49:46 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Aug 12 10:08:43 2020 from 192.170.1.102

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@pi:~$ sudo apt-get install mosquitto mosquitto-clients

```

Figura 3.11: Instalación del protocolo MQTT en la raspberry pi
Fuente: Autor.

El circuito de conexión que brinda el acceso automatizado al laboratorio se presenta en la Figura 3.12 ; la tarjeta Raspberry Pi se conecta al módulo ESP8266 a través de la red inalámbrica wifi mediante el protocolo de comunicación MQTT.

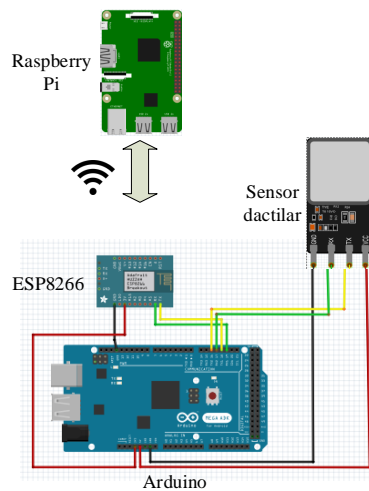


Figura 3.12: Circuito de conexión para el acceso al laboratorio
Fuente: Autor.

3.4.3. Case para el sistema de acceso automático

Para instalar el sistema automático fue necesario diseñar una estructura en la cual se pueda colocar los elementos del circuito de acceso para el registro de los diferentes usuarios. El diseño de la estructura se realizó a través del programa **Thinkercad** y la estructura se obtuvo con la impresora IIP (3D). Thinkercad es un programa que facilita

el diseño e impresión de accesorios en tres dimensiones, a través de este es como se realizó el diseño del case respectivo para la adecuación del sistema de acceso automático, su diseño se presenta en la Figura 3.13

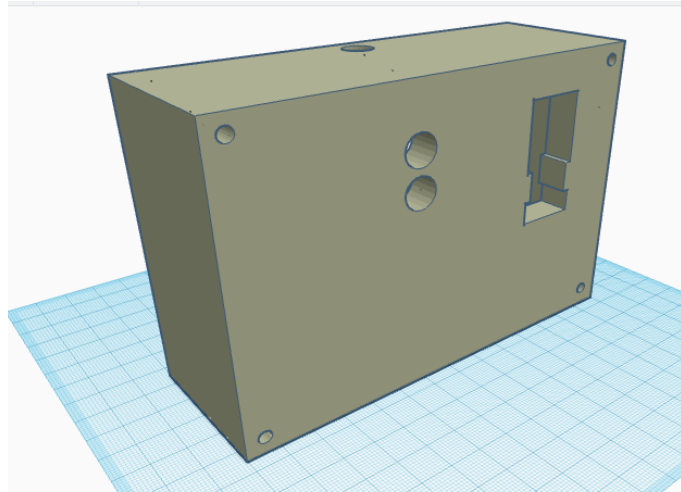


Figura 3.13: Diseño del *case* para la implementación del sistema automático

Fuente: Autor.

3.5. Control de los circuitos de iluminación

El control de la iluminación del laboratorio se realiza a través de una pantalla táctil, la cual representa una interfaz hombre maquina, (HMI, por sus siglas en inglés), para la representación de variables, imágenes y estados de control del sistema de automatización. Además, interviene un software libre llamado openHAB que brinda grandes oportunidades para el diseño de automatización en hogares y variedad para integrar diferentes dispositivos, ya que permite el diseño de una interfaz uniforme ofreciendo la posibilidad de obtener cualquier deseo en la domótica.

La instalación de este software en la Raspberry Pi fue importante debido que es el centro de control para proceder a la operación del encendido y apagado de la iluminación del laboratorio. A través de esta plataforma se realizó la automatización del laboratorio.

3.5.1. Seccionamiento del sistema de iluminación

Se consideró necesario seccionar el sistema de iluminación del laboratorio con la finalidad de presentar una estética mejorada en la automatización, para ello se realizaron dos secciones. A través del seccionamiento se pretende realizar el encendido de la iluminación de diferentes formas; desde un pulsador físico y desde el servidor OpenHab a través de la pantalla de raspberry e incluso desde un ordenador. En la Figura 3.14 se muestra el seccionamiento de la iluminación del laboratorio.

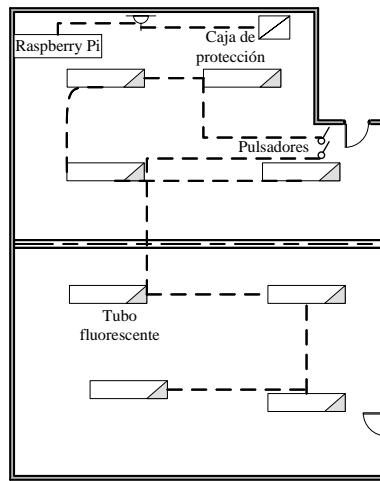


Figura 3.14: Circuito eléctrico de seccionamiento de iluminación

Fuente: Autor.

3.5.2. Creación de ítems en openHAB

Para conectar lo físico con lo digital, openHAB utiliza "ítems", los mismos que utilizan lógica de automatización. Los ítems están compuestos por estados y se utilizan a través de eventos, la Tabla 3.1 muestra los tipos de ítems que OpenHab ofrece.

Tabla 3.1: Ítems de openHAB

Ítem	Descripción	Comando
Color	Información de color	OnOff, IncreaseDecrease, Percent
Contacto	Estado de almacenamiento de ítems	Abrir cerrado
Fecha y hora	Almacena fecha y hora	-
Grupo	Elemento para anidar otros elementos	-
Imagen	Contiene los datos binarios de una imagen	-
Intensidad	Contiene un valor para atenuadores	OnOff, IncreaseDecrease, Percent
Ubicación	Almacena coordenadas GPS	Punto
Número	Almacena valores en formato numérico	Decimal
Persiana	Típicamente se usa en persianas	Encendid y apagado

Los ítems pueden contener cadenas, números o conmutadores, son de tipo básico y pueden leer o escribir un estado con la finalidad de relacionarse con los mismos. Su creación se realiza a través de archivos de extensión (.items); son archivos de texto escritos en la carpeta de Openhab, y cada elemento debe ser único entre todos. La definición del ítem empieza por distinguir el tipo de ítem, el nombre, la etiqueta y el estado del mismo [openHAB, 2015].

Para interactuar con el circuito de iluminación se utilizó el ítem tipo Contacto (Switch); ya que tiene la capacidad de asignar al circuito de iluminación un encendido o apagado. En la Figura 3.15 se muestra el código para crear el

ítem en openHAB; en el gráfico se puede observar que el nombre designado para el ítem es "lab207".

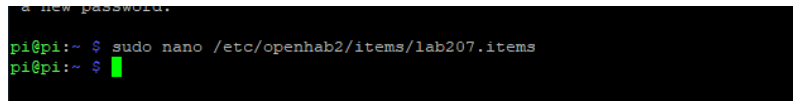


Figura 3.15: Código para la creación de ítems

Fuente: Autor.

3.5.3. Integración de equipos LCN para el control de la iluminación

El módulo LCN-R8H, brinda el control central de los circuitos automatizados. Mediante este módulo se interactúa con los diferentes actuadores del laboratorio ya que su funcionamiento está basado a través de un conjunto de relés. En la Figura 3.16 se presenta el módulo mencionado.



Figura 3.16: Módulo LCN

Fuente: Autor.

Para integrar los equipos de la tecnología LCN con la plataforma openHAB, se requirió adicionar una extensión o binding dentro de la misma. Esto permite interactuar con el servicio propietario de los dispositivos LCN dentro la plataforma web. Para ello se define la configuración con el servicio PCHK dentro de openHAB generando un archivo de extensión .cfg dentro de la ruta: `sudo nano /etc/openhab2/items/lab207.items`.

Efectuada la configuración para actuar con los equipos de LCN se procedió a interactuar con los ítems creados en la plataforma para generar objetos que nos permitan manipular con los distintos dispositivos desplegados en el laboratorio de control. Dichos ítems se muestran en la Figura 3.17.

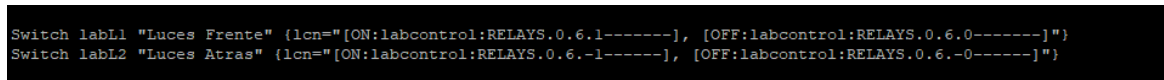


Figura 3.17: Ítems creados para el encendido de la iluminación

Fuente: Autor.

3.6. Sistema de seguridad a través de cámara de monitoreo

El sistema de seguridad presentado en el laboratorio se brinda a través del módulo de cámara de Raspberry Pi; la cual mantiene la transmisión en tiempo real de lo que sucede en el laboratorio. El modelo de Raspberry Pi usado posee un puerto para la conexión del módulo de la cámara, la conexión se realizó mediante un cable plano. La Figura 3.18 presenta la conexión de la cámara para el sistema de seguridad.

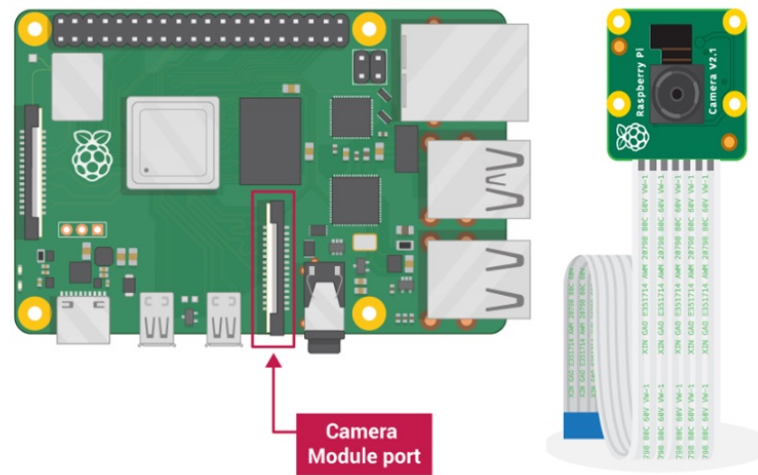


Figura 3.18: Conexión de la cámara en la Raspberry Pi
Fuente: [ECUAROBOT, 2018]

Para instalar la cámara en la Raspberry Pi, se diseñó una estructura que permita adecuar la cámara y con ello facilitar el enfoque y la transmisión hacia el laboratorio. Esta estructura se realizó a través del software **Thinkercad** utilizado con anterioridad y la impresión se obtuvo con la impresora en 3D. El diseño realizado se muestra en la Figura 3.19, en el cual se observa un soporte para adecuar la cámara y orientarla hacia la dirección deseada.

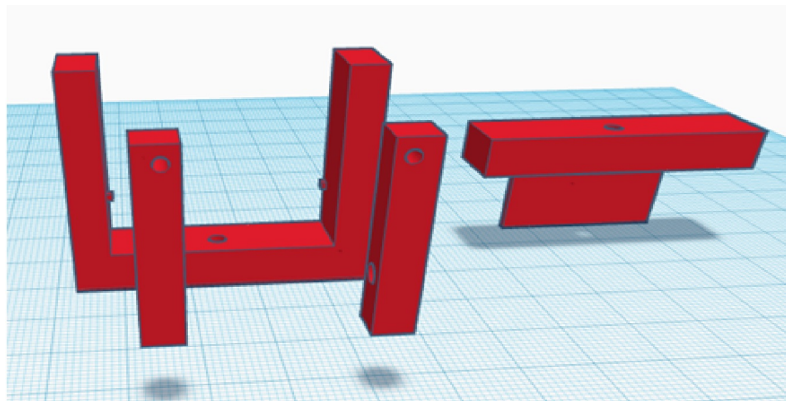


Figura 3.19: Case diseñado para la cámara
Fuente: Autor.

La transmisión en tiempo real de la cámara se realiza a través de un script de programación, el cual establece una conexión a una página web con la dirección IP de la raspberry pi de puerto 8000 (192.170.1.110:8000). Mediante esta dirección se interactúa con openHAB para mostrar la transmisión en la misma.

3.7. Verificación de presencia a través de sensores de presencia

Para verificar la presencia en el laboratorio se usó el sensor de movimiento PIR (passive infrared sensor). Este sensor enciende la iluminación del laboratorio cuando detecta la presencia de usuarios a través de la puerta de ingreso. Para realizar la conexión entre la Raspberry y el sensor de movimiento se utilizó un hat mostrado en la Figura 3.20, denominado Pi Ez Connect, mediante este se hace posible la conexión hacia los pines gpio.

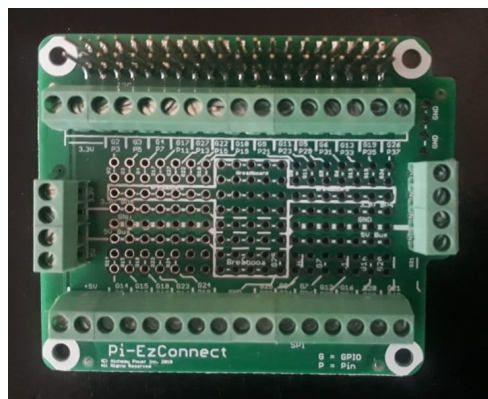


Figura 3.20: Hat Pi Ez Connect

Fuente: Autor.

Posteriormente se realizó la configuración en openHAB que posibilita la verificación de presencia en el laboratorio. Esta configuración fue el diseño de una regla que se utiliza para automatizar los procesos. Al activar ésta se invoca un script que realiza el proceso de verificación para el encendido de la iluminación. Su diseño fue como se muestra en la Figura 3.21.

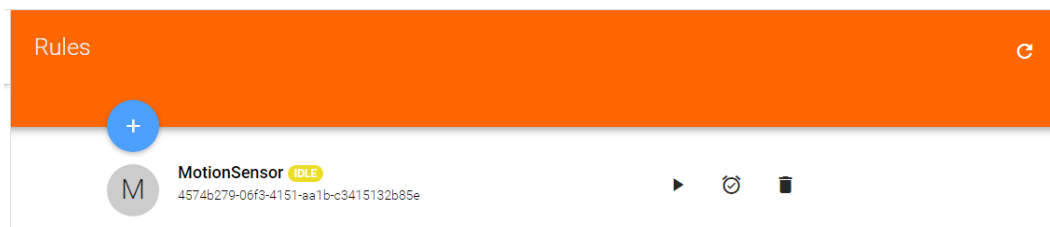


Figura 3.21: Regla creada en OpenHab

Fuente: Autor.

La configuración de las reglas se basa en las condiciones, *when*, *then*, *but only if.*, y se interactúa con dos sintaxis: *item*, que permite consultar un estado y *sendCommand*, que envía un comando para ejecutar una acción. La

configuración realizada se muestra en las Figuras 3.22 y 3.23

Update trigger

Name
an item state changes

Description
This triggers the rule if an item state has changed.

Configurations

Item PIR (MotionSensor) the name of the item	Previous State OPEN the required previous state of the item
--	---

State
CLOSED
the state of the item

DELETE CANCEL OK

Figura 3.22: Configuración de la regla en la condición *when*
Fuente: Autor.

Update action

Name
send a command

Description
Sends a command to a specified item.

Configurations

Item Luces Atras (labL2) the name of the item	Command ON the default command to be sent if none is passed as an input value
---	---

DELETE CANCEL OK

Figura 3.23: Configuración de la regla en la condición *then*
Fuente: Autor.

3.8. Registro del consumo energético del laboratorio

El registro del consumo energético se obtiene a través del uso del sensor de corriente SCT-000, este sensor brinda la posibilidad de medir hasta 100A con una salida de 50mA, es decir con una proporción 100A/50mA. Para obtener un registro del consumo fue necesario disponer de diferentes componentes y circuitos que facilitaron el desarrollo de este objetivo.

3.8.1. Convertidor Pi16ADC

Pi16ADC es un convertidor analógico digital, construido para la Raspberry Pi, contiene 16 puertos, los mismos que capturan información de sensores analógicos. Cada canal analógico puede leer un máximo de 2.5 V pico a pico, además es posible combinar los canales para crear un canal diferencial y de esta forma leer una señal analógica de

máximo -2.5V a +2.5V. La comunicación de esta tarjeta hacia la Raspberry Pi se realizó mediante la conexión I2C. En la Figura 3.24 se muestra el convertidor Pi16ADC usado.

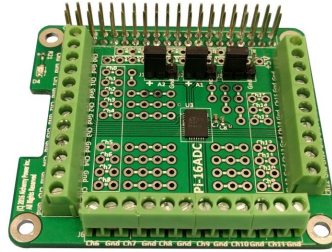


Figura 3.24: Convertidor analógico digital Pi16ADC

Fuente: Autor.

3.8.2. Circuito acondicionador de señal

El sensor SCT-000 ofrece una lectura de señal de corriente alterna, estos valores se encuentran fuera del alcance que ofrece el convertidor analógico digital (0 a +5V), debido que la sección negativa de la señal resultó un inconveniente para obtener el registro del consumo, se rectificó la señal con la finalidad de trabajar con la parte positiva. Se descartó la posibilidad de usar diodos, ya que la caída de voltaje en estos es demasiado grande en comparación al voltaje de la señal; es por ello que se decidió usar un operacional LM356, el cual se ha configurado en un seguidor de voltaje para que trabaje con polaridad positiva y elimine la parte negativa de la señal. Además, fue necesario el uso de una resistencia de carga para obtener una relación ($50\text{mA} * 20\text{ ohm} = 1\text{ V}$) con la finalidad de obtener una lectura con valores de voltaje bajo, que se encuentre dentro del rango de las entradas del sensor SCT-000, el circuito que acondiciona la señal se observa en la Figura 3.25

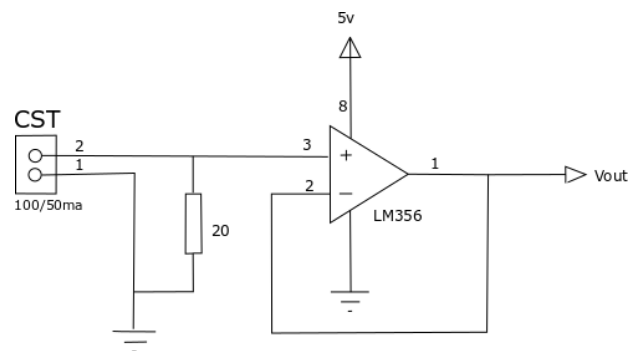


Figura 3.25: Circuito acondicionador diseñado

Fuente: [Llamas, 2018]

3.8.3. Calibración del sensor SCT-000

La calibración de un sensor es una operación que correlaciona las lecturas de un elemento patrón con las lecturas tomadas por el sensor. Para calibrar el sensor SCT-000 se tomó como elemento patrón un amperímetro conectado en serie con el sensor SCT-000 y la lectura de un voltímetro establecido en la salida del circuito acondicionador. La Tabla 3.2 muestra los valores de corriente y voltaje obtenidos que permitieron realizar el ajuste de la recta y la calibración del sensor SCT-000.

Tabla 3.2: Tabla de valores medidos para la calibración del sensor SCT-000

Corriente (I)	Voltaje (V)
0.21	0.01
0.29	0.02
0.37	0.03
0.93	0.05
1.3	0.06
1.6	0.16
2.7	0.52
3.34	0.63
3.74	0.66
4.01	0.73
7.85	1.05
9.65	1.8

Con los valores medidos se generó la gráfica a través del método de dos puntos como se muestra en la Figura 3.26.

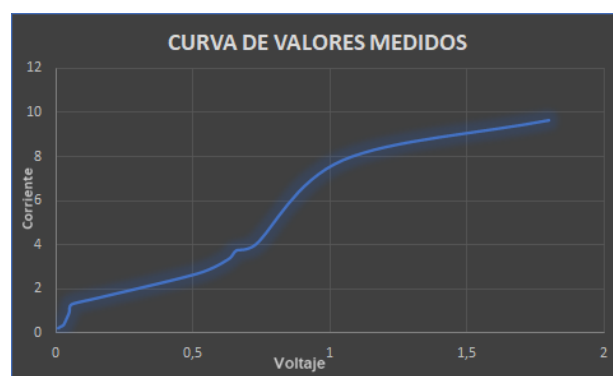


Figura 3.26: Curva de los valores medidos

Fuente: Autor.

Finalmente se obtuvo la línea de tendencia que ajusta los valores medidos para la curva de calibración como se

3.8.5. Diseño de la interfaz

Se procedió a realizar el diseño de la interfaz que muestre el consumo energético del laboratorio y para ello se utilizó una herramienta llamada **ThingSpeak**. Esta plataforma se encuentra creada exclusivamente para el internet de las cosas, a través de este es posible guardar y transmitir todos los datos que el sensor SCT-000 esta tomando y plasmarlos en una interfaz gráfica.

Para obtener la misma se creó una cuenta en la plataforma y con ello se procedió a crear una canal que resulta la interfaz en donde se establecen todas las características que contiene. En la Figura 3.29 se presenta el proceso para la creación del canal.

Vista privada Vista pública Configuraciones de canal Compartir Claves API Importación / Exportación de datos

Configuraciones de canal

Porcentaje completo 50%

Canal ID 942188

Nombre REGISTRO ENERGÉTICO LAB 207

Descripción CONSUMO

Campo 1 POTENCIA ☒

Campo 2 ☐

Campo 3 ☐

Ayuda

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

Configuraciones de canal

- **Porcentaje completo:** calculado en función de los datos ingresados en los diversos campos de un canal. Ingresas el nombre, la descripción, la ubicación, la URL, el video y las etiquetas para completar tu canal.
- **Nombre del canal:** ingresa un nombre único para el canal ThingSpeak.
- **Descripción:** ingresa una descripción del canal ThingSpeak.
- **Campo #:** marque la casilla para habilitar el campo e ingrese un nombre de campo. Cada canal ThingSpeak puede tener hasta 8 campos.
- **Metadatos:** ingrese información sobre los datos del canal, incluidos los datos JSON, XML o CSV.

Figura 3.29: Creación del canal para la interfaz de consumo energético

Fuente: Autor.

Capítulo 4

Análisis de resultados

El presente capítulo ofrece la información acerca de los resultados obtenidos durante el diseño y desarrollo de la investigación para la automatización del laboratorio de electrónica de la Universidad Católica de Cuenca.

4.1. Control de acceso al laboratorio

El control de acceso al laboratorio se produce con la desconexión de la cerradura magnética mediante la lectura de la huella a través del sensor dactilar. La cerradura permanece en estado activo permitiendo tener cerrada la puerta de acceso al laboratorio. El sensor permite que el campo magnético producido por la cerradura se interrumpa y mediante ello se posibilite la apertura de la puerta. La cerradura se instaló en la parte superior interna de la puerta de ingreso al laboratorio como se presenta en la Figura 4.1.



Figura 4.1: Cerradura instalada en la puerta de acceso al laboratorio

Fuente: Autor.

Ademas, se realizó la instalación del sensor dactilar en la parte externa del laboratorio en conjunto con el case previamente diseñado, el microcontrolador arduino y el módulo de wifi ESP8266.

Como se observa en la Figura 4.2 el sensor dactilar está funcionando, lo que posibilita tomar la lectura de las huellas y permitir el acceso al laboratorio.



Figura 4.2: Instalación del *case* y sensor dactilar para la lectura de las huellas

Fuente: Autor.

Adicional a ello se instaló un sensor de proximidad en el interior del laboratorio a un costado de la puerta de ingreso. Este dispositivo tiene la funcionalidad de permitir la apertura de la puerta desde el interior, con la finalidad que las personas que se encuentren dentro de la zona puedan salir sin ningún tipo de inconveniente. Su funcionamiento consiste en acercar la palma de la mano hacia el sensor para que este se active con la intención de cortar la alimentación de la cerradura para la apertura de la puerta. En la Figura 4.3 y 4.4, se observa el funcionamiento del sensor en estado pasivo y activo respectivamente



Figura 4.3: Sensor de proximidad en estado pasivo

Fuente: Autor.



Figura 4.4: Sensor de proximidad en estado activo

Fuente: Autor.

4.2. Control de los circuitos de iluminación

El control de los circuitos de iluminación se realizó a través de la pantalla touch de raspberry pi, un teléfono inteligente y un ordenador mediante el uso del software openHAB. Como se mencionó en el capítulo anterior, se estableció la creación de ítems en la carpeta de openHAB con la finalidad de poder interactuar desde esta plataforma para el control de la iluminación. Cabe mencionar que los módulos LCN son quienes al actuar como relés permiten el encendido y apagado de la iluminación. La pantalla touch en conjunto con la raspberry se instaló a un costado del escritorio del docente, esto con la finalidad de ofrecer el confort para el mismo en el control del laboratorio. En la Figura 4.5, se presenta la instalación de la pantalla.

Figura 4.5: Instalación de la pantalla *touch*

Fuente: Autor.

Se observa también en la Figura 4.6, el control de encendido y apagado de los circuitos de iluminación desde la plataforma openHAB a través de la pantalla touch de raspberry pi. En la imagen se puede apreciar el diseño de dos botones denominados FOCOS ZONA 1 y FOCOS ZONA 2, los cuales permiten el encendido de la iluminación del laboratorio.

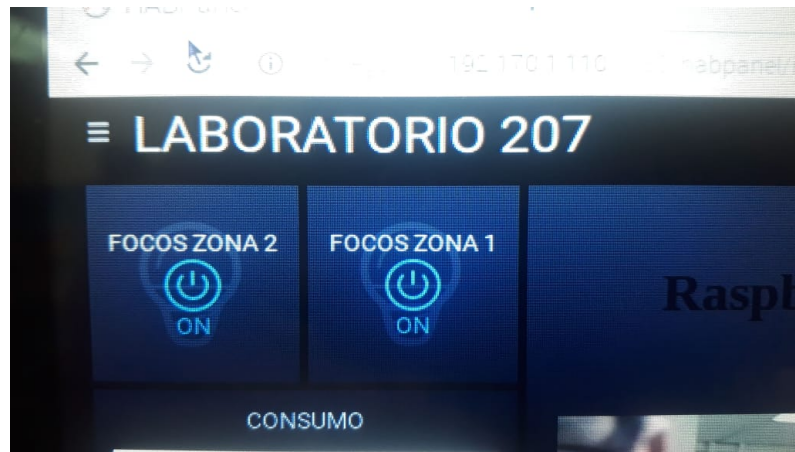


Figura 4.6: Control de los circuitos de iluminación desde la pantalla *touch* de raspberry

Fuente: Autor.

Los circuitos de la iluminación y el sistema de automatización en general, adicionalmente se puede controlar desde un ordenador e incluso desde un teléfono inteligente, accediendo a internet a través de la dirección IP de la raspberry y puerto 8080 con la restricción que estos se encuentren conectados a la misma red. En la Figura 4.7 y 4.8 se presenta la interacción del control de los circuitos de iluminación desde un teléfono inteligente y desde un ordenador.



Figura 4.7: Control de los circuitos de iluminación desde un teléfono inteligente

Fuente: Autor.

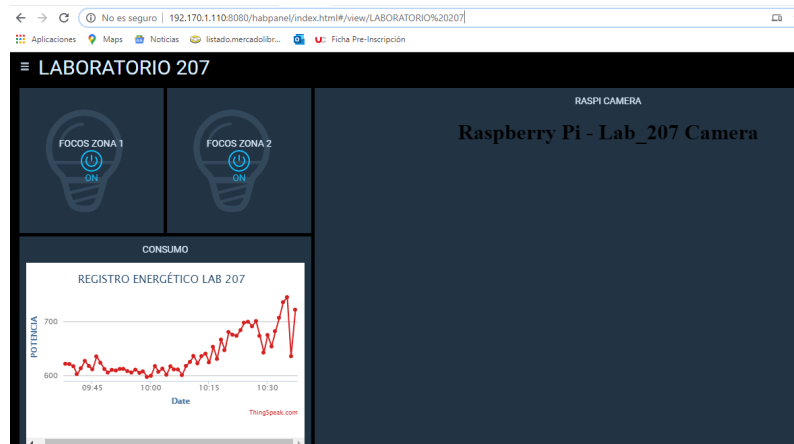


Figura 4.8: Control de los circuitos de iluminación desde la pantalla de un ordenador
Fuente: Autor.

Adicional al proceso de control realizado para la iluminación, se instalaron dos pulsantes en el interior del laboratorio, esto con el objetivo ofrecer el control del sistema de forma manual, cada pulsante corresponde a una sección de iluminación. La instalación de los pulsantes se presenta en la Figura 4.9.



Figura 4.9: Instalación de los pulsantes para el control de la iluminación
Fuente: Autor.

4.3. Sistema de seguridad a través de la cámara

El sistema de seguridad se realizó a través de la cámara de raspberry pi, la cual mantiene una transmisión en tiempo real hacia la plataforma openHAB del laboratorio. Este proceso se obtuvo mediante el programa diseñado

en python y con la conexión de la cámara en el microcontrolador. En la Figura 4.10 se presenta la instalación y adecuación de la cámara con el case diseñado e impreso en 3D.

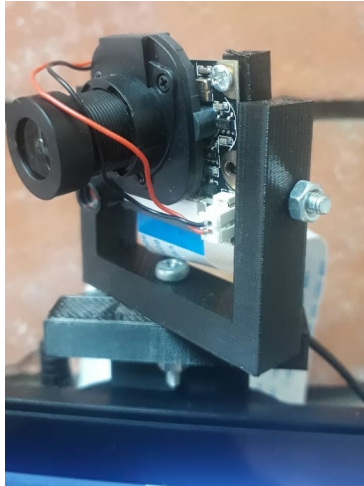


Figura 4.10: Instalación y adecuación de la cámara

Fuente: Autor.

Se accedió a la transmisión de la cámara a través de internet con la siguiente dirección: localhost:8000, el resultado se muestra en la Figura 4.11



Figura 4.11: Transmisión en tiempo real de la cámara

Fuente: Autor.

Posteriormente, se procedió a interactuar con openHAB, para ello se creó un frame en dicho software y se añadió la dirección IP de la transmisión mostrada en la figura anterior. En la Figura 4.12, se observa la transmisión de la cámara desde la plataforma openHAB.



Figura 4.12: Transmisión de la cámara desde la plataforma openHAB
Fuente: Autor.

4.4. Detección de presencia

El control de detección de presencia está basado en activar la iluminación del laboratorio a través del sensor PIR. Este sensor detecta la presencia de usuarios en el laboratorio para proceder al encendido de las luces. En la Figura 4.13 presentada a continuación se observa el funcionamiento del sensor, orientado hacia la puerta de acceso del laboratorio y el encendido de las lámparas.



Figura 4.13: Encendido de la iluminación a través del sensor de presencia
Fuente: Autor.

Además, en la Figura 4.14, se presenta la instalación del Hat Pi Ez Connect en la raspberry pi. El Hat facilita la conexión del sensor hacia los pines GPIO de la raspberry pi para el encendido de la iluminación.

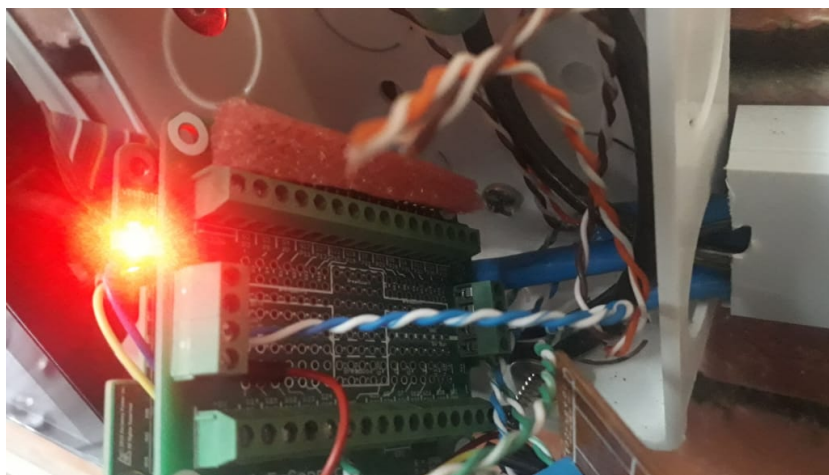


Figura 4.14: Instalación del Hat PiEZ Connect

Fuente: Autor.

4.5. Registro del consumo energético

El registro del consumo energético se obtuvo mediante el diseño de la programación en python y la instalación de los diferentes elementos que posibilitaron el desarrollo del objetivo. En la Figura 4.15 se observa la instalación del sensor SCT-000. El sensor se ubicó en la fase de alimentación del tablero de protección eléctrico con la finalidad de medir el consumo total del laboratorio de electrónica.



Figura 4.15: Instalación del sensor SCT-000 en el laboratorio

Fuente: Autor.

Se realizó la instalación del circuito acondicionador de señal diseñado. Este circuito se conecta de forma directa al sensor SCT-000 y se instaló cerca de la estación de control de la raspberry pi. En la Figura 4.16 se presenta la

instalación del circuito.

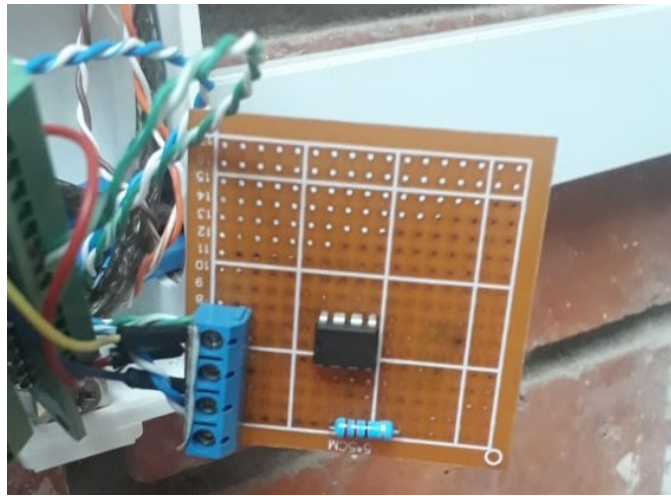


Figura 4.16: Instalación del circuito acondicionador diseñado
Fuente: Autor.

El convertidor analógico digital denominado Pi16ADC establece la conexión entre la raspberry pi y el circuito acondicionador de señal. La instalación se presenta en la Figura 4.17.

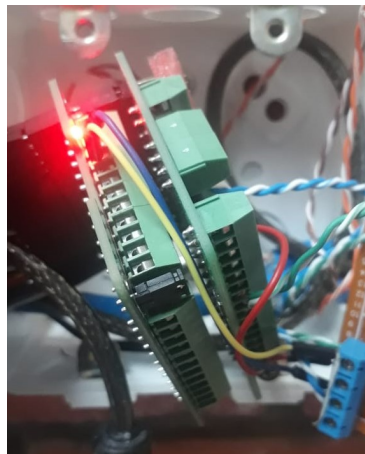
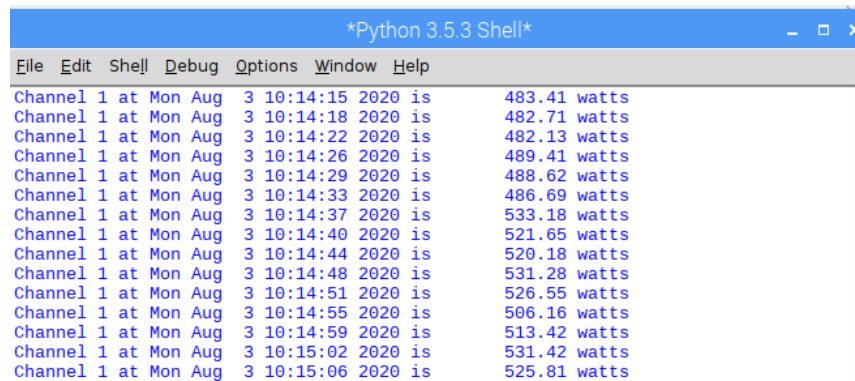


Figura 4.17: Instalación del convertidor analógico digital Pi16ADC
Fuente: Autor.

Con el diseño de la programación realizada en python y la instalación de los componentes se obtuvieron los datos medidos por el sensor SCT-000.

Los datos que se muestran en la Figura 4.18 representan la potencia en watts de consumo del laboratorio de electrónica. Estos fueron tomados en un intervalo de 3 segundos. Además, se visualiza la fecha y hora en que los datos fueron obtenidos.



Channel	1	at	Mon	Aug	3	10:14:15	2020	is	483.41	watts
Channel	1	at	Mon	Aug	3	10:14:18	2020	is	482.71	watts
Channel	1	at	Mon	Aug	3	10:14:22	2020	is	482.13	watts
Channel	1	at	Mon	Aug	3	10:14:26	2020	is	489.41	watts
Channel	1	at	Mon	Aug	3	10:14:29	2020	is	488.62	watts
Channel	1	at	Mon	Aug	3	10:14:33	2020	is	486.69	watts
Channel	1	at	Mon	Aug	3	10:14:37	2020	is	533.18	watts
Channel	1	at	Mon	Aug	3	10:14:40	2020	is	521.65	watts
Channel	1	at	Mon	Aug	3	10:14:44	2020	is	520.18	watts
Channel	1	at	Mon	Aug	3	10:14:48	2020	is	531.28	watts
Channel	1	at	Mon	Aug	3	10:14:51	2020	is	526.55	watts
Channel	1	at	Mon	Aug	3	10:14:55	2020	is	506.16	watts
Channel	1	at	Mon	Aug	3	10:14:59	2020	is	513.42	watts
Channel	1	at	Mon	Aug	3	10:15:02	2020	is	531.42	watts
Channel	1	at	Mon	Aug	3	10:15:06	2020	is	525.81	watts

Figura 4.18: Datos obtenidos de la lectura del sensor de corriente en python

Fuente: Autor.

Los datos obtenidos mediante el sensor SCT-000 se plasmaron en el software ThingSpeak. Mediante esto se obtuvo una gráfica de consumo energético como se observa en la Figura 4.19



Figura 4.19: Gráfica del registro energético en Thingspeak

Fuente: Autor.

Con la dirección IP de la gráfica mostrada en el servidor ThingSpeak, se vinculó con openHAB para visualizar la misma desde esta plataforma. En la Figura 4.20 se presenta la interfaz final de automatización del laboratorio. Allí se detalla la gráfica del registro del consumo energético, la transmisión en tiempo real a través de la cámara de raspberry, y los botones que permiten interactuar con el encendido de la iluminación del laboratorio.

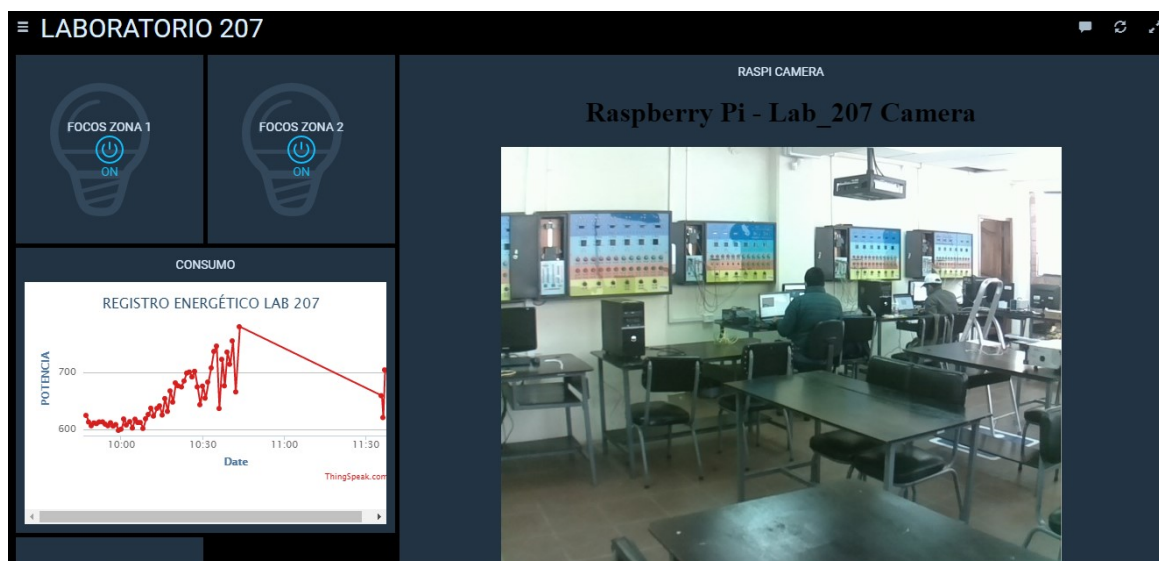


Figura 4.20: Interfaz de automatización del laboratorio de electrónica
Fuente: Autor.

Capítulo 5

Conclusiones

En este capítulo se presentan las conclusiones más importantes surgidas durante el desarrollo del trabajo de investigación. Se da a conocer también algunos análisis sobre los resultados y algunas sugerencias para trabajos futuros que impulsan al crecimiento del sistema de control.

5.1. Conclusiones

Obtener un sistema de control automático conlleva un periodo de estudio, investigación y pruebas. Los resultados obtenidos durante el largo proceso de investigación para la automatización del laboratorio de electrónica de la Universidad Católica de Cuenca han sido alentadores; buscar metodologías que permitan desarrollar sistemas de control automatizado a bajo costo es una oportunidad para el desempeño académico.

Durante el desarrollo del proceso de automatización, fue posible conocer la validez y variedad que la plataforma openHAB ofrece para interactuar y relacionar equipos diferentes con la finalidad de obtener un sistema de control múltiple. El sistema de automatización desarrollado provee la eficiencia energética, así como el cuidado de los elementos y componentes que existen en este espacio.

Poseer un sistema que permite controlar el ingreso a una área determinada es un desarrollo importante para la domótica. En cuanto al control de acceso al laboratorio, openHAB brinda la oportunidad de mantener el control a través de reglas diseñadas y ejecutadas en la plataforma mediante el sensor dactilar. El acceso automatizado al laboratorio de la Universidad brinda la posibilidad de mantener el control de las personas que ingresan al mismo y ofrece a los usuarios el confort necesario.

El control de los circuitos de iluminación del laboratorio posee la capacidad de operarse desde la interfaz diseñada en la plataforma openHAB. Esta interfaz es accesible desde cualquier dispositivo que se encuentre vinculado en la red del microcontrolador. Brindar el control automatizado de los circuitos de iluminación a los usuarios representa la obtención de un laboratorio tecnológico.

El sistema de seguridad realizado a través de la cámara de monitoreo permite obtener una vista de todo lo que sucede en el laboratorio de electrónica a través de la plataforma openHAB. Si bien se observa la existencia de varios modelos de cámara para realizar este proceso, la cámara de raspberry pi es la más opcional debido a su bajo costo y la posibilidad de adaptarla con el microcontrolador. Durante el desarrollo de este objetivo surgió un inconveniente, enfocado en un retardo en la transmisión de video. Sin embargo, esto se solucionó mediante el mejoramiento de la calidad de internet.

Existen diversas formas en las que es posible verificar la presencia de alguna persona en un sector. Verificar la presencia de usuarios en el laboratorio tuvo como principio fundamental brindar comodidad a los mismos. Por esta razón, este proceso se realizó a través del sensor de movimiento (PIR), el cual enciende la iluminación al captar la presencia.

Obtener un registro del consumo energético resulta siempre una necesidad en la vida cotidiana. Mediante este proceso es posible verificar la existencia de alguna anomalía en cuanto al consumo de energía en un sector. Se verificaron diversos retos que surgieron durante el diseño de este objetivo; los valores de salida que el sensor ofrece fue uno de ellos. Si bien se conoce la existencia de diversos sensores que facilitan el desarrollo de este diseño; Se decidió por el SCT-000 debido a su capacidad de lectura y su bajo costo.

Finalmente, el desarrollo de este trabajo de investigación brinda la posibilidad de fusionar el laboratorio de electrónica con la tecnología. Además, permite la expansión y aplicación de conocimientos que en el trayecto de la carrera de ingeniería electrónica se obtuvieron.

5.2. Trabajo futuro

La gran sugerencia que los sistemas de control automatizado poseen en la actualidad se encuentra determinado por el confort y la facilidad de control que este ofrece a los usuarios. La tecnología avanza cada vez más a pasos agigantados y con ello las necesidades de las personas y los requerimientos de las mismas por hacer un mínimo esfuerzo por conseguir algo.

Por esta razón es importante actualizar los procesos de automatización de forma que vayan a la par con el avance de la tecnología. El sistema de control está basado en la automatización del laboratorio, sin embargo el siguiente alcance que el proyecto de investigación debe tener, será automatizar el proyector existente en este espacio para la enseñanza de los docentes. Además, controlar la apertura de las persianas e incluso migrar del sistema de acceso al laboratorio mediante el sensor dactilar por un sistema de reconocimiento facial.

Por otra parte, se realizarán los esfuerzos necesarios por proponer un sistema de automatización con características parecidas al desarrollado, y de esta forma comercializar sistemas de este tipo que beneficien a la sociedad.

Bibliografía

- [Aakvaag and Frey, 2006] Niels Aakvaag and Jan-Erik Frey. Redes de sensores inalámbricos Nuevas soluciones de interconexión para la automatización industrial. Technical report, 2006.
- [Alga, 2017] Miki Alga. Cómo actualizar el firmware Flash ESP8266: firmware oficial AT intermitente, dec 2017.
- [Cama *et al.*, 2012] Alejandro Cama, Emiro De la Hoz, and Dora Pinto. Las redes de sensores inalámbricos y el internet de las cosas. *INGE CUC*, 8(1):163–172, oct 2012.
- [Casco, 2014] Sergio Martín Casco. Raspberry Pi, Arduino y Beaglebone Black Comparación y Aplicaciones. Technical report, 2014.
- [CEDOM, 2016] CEDOM. Qué es Inmótica, 2016.
- [Constantino, 2011] Ingrid Constantino. Domótica e Inmótica: Viviendas y edificios inteligentes. Technical report, Universidad Veracruzana, Veracruz, 2011.
- [Danilo *et al.*, 2003] Autor : Danilo, Simón Zorita, and Javier Ortega García. TESIS DOCTORAL RECONOCIMIENTO AUTOMÁTICO MEDIANTE PATRONES BIOMÉTRICOS DE HUELLA DACTILAR. Technical report, 2003.
- [ECUAROBOT, 2018] ECUAROBOT. Comenzando con el módulo de cámara Raspberry Pi - EcuRobot, 2018.
- [Enríquez Herrador, 2009] Rafael Enríquez Herrador. Guía de Usuario de Arduino. Technical report, 2009.
- [Flores *et al.*, 2018] Pablo Flores, Marlon Moscoso-Martinez, Luis I. Minchala, Andres Vazquez-Rodas, and Darwin Astudillo-Salinas. A simple reliable wireless bus design for home automation systems. In *2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pages 1–4. IEEE, aug 2018.
- [Gutierrez *et al.*, 2018] Sebastian Gutierrez, Carlos Acero, and Pedro M. Rodrigo. Domotic control system for blinds with zigbee communication mobile device. In *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)*, pages 1–4. IEEE, aug 2018.
- [Huidobro and Milán Tejedor, 2010] José Manuel. Huidobro and Ramón Jesús. Milán Tejedor. *Manual de domótica*. Creaciones Copyright, 2010.
- [Jesús González García *et al.*, 2017] Antonio Jesús González García, José López, Vicario Xavi, and Vilajosana Guillen. IoT: Dispositivos, tecnologías de transporte y aplicaciones. Technical report, 2017.

- [Llamas, 2018] Luis Llamas. Sensor de corriente eléctrica no invasivo con Arduino y SCT-013, 2018.
- [Lledó Emilio *et al.*, 2012] Sánchez Lledó Emilio, Sergio Sáez, Vicente Luis, and Atienza Vanacloig. Diseño de un sistema de control domótico basado en la plataforma Arduino. Technical report, 2012.
- [Oliveira *et al.*, 2018] Guilherme M. B. Oliveira, Danielly C. M. Costa, Ricardo J. B. V. M. Cavalcanti, Josiel P. P. Oliveira, Diego R. C. Silva, Marcelo B. Nogueira, and Marconi C. Rodrigues. Comparison Between MQTT and WebSocket Protocols for IoT Applications Using ESP8266. In *2018 Workshop on Metrology for Industry 4.0 and IoT*, pages 236–241. IEEE, apr 2018.
- [ONUDI, 2011] ONUDI. Automatización de bajo costo. Technical report, 2011.
- [openHAB, 2015] Comunidad openHAB. Introduction — openHAB, 2015.
- [Rogers *et al.*, 2018] Felipe; Rogers, Alejandro; Rojas, Sergio; Medina, and Sergio Sobarzo. Sistema de presencia académica a través de sensor de movimiento y control de asistencia a través de huellas dactilares usando un microcontrolador. Technical report, 2018.
- [Sarasúa, 2011] Juan Sarasúa. *Domótica un factor importante para la arquitectura sostenible*. Barranquilla, 2011.
- [Thomson Gale (Firm) and Buitrago, 2013] Diana Milena Archila Thomson Gale (Firm) and Frey Alfonso Santamaría Buitrago. ESTADO DEL ARTE DE LAS REDES DE SENSORES INALÁMBRICOS. *Tecnología Investigación y Academia*, 1(2), dec 2013.
- [Win and Nyein, 2019] Yu Win and Saw Nyein. Wireless Student Attendance System using Fingerprint Sensor. pages 2–5, 2019.
- [Zamora-Izquierdo *et al.*, 2010] M A Zamora-Izquierdo, J Santa, and A F Gomez-Skarmeta. An Integral and Networked Home Automation Solution for Indoor Ambient Intelligence. *IEEE Pervasive Computing*, 9(4):66–77, 2010.

Apéndice A

Algoritmos en Arduino

A.1. Algoritmo para registro de la huella

```
1 //PROGRAMA DE ACCESO A LA MEMORIA DEL SENSOR DACTILAR
2 // ALMACENAMIENTO DE HUELLAS
3 // AUTOR: LUIS ENRIQUE AVILA G.
4 // TRABAJO DE TITULACION – ACCESO AUTOMATICO AL LABORATORIO DE ELECTRONICA
5
6 #include <Adafruit_Fingerprint.h>
7 #include <SoftwareSerial.h>
8
9
10 // #definicion de los puertos serial del arduino
11
12 // pin #2 CONEXION DEL SENSOR (Verde)
13 // pin #3 CONEXION DEL SENSOR (Blanco)
14 // comment these two lines if using hardware serial
15 SoftwareSerial mySerial(2, 3);
16
17 Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
18
19 uint8_t id;
20
21 void setup()
22 {
23   Serial.begin(9600);
24   while (!Serial); //
25   delay(100);
26   Serial.println("\n\nAdafruit Registro de sensor de huella dactilar");
27
28   // establece la velocidad de datos para el puerto serie del sensor
29   finger.begin(57600);
30
31   if (finger.verifyPassword()) {
32     Serial.println("Sensor de huella dactilar encontrado!");
33   } else {
34     Serial.println("Sensor de huella dactilar no encontrado :(");
35     while (1) { delay(1); }
```

```
36     }
37 }
38
39 uint8_t readnumber(void) {
40     uint8_t num = 0;
41
42     while (num == 0) {
43         while (! Serial.available());
44         num = Serial.parseInt();
45     }
46     return num;
47 }
48
49 void loop() // corre el programa varias veces
50 {
51     Serial.println("Listo para registrar una huella digital!");
52     Serial.println("Ingrese el numero de ID (1 a 127) en el que desea guardar la huella digital");
53     id = readnumber();
54     if (id == 0) { // ID #0 no permitido
55         return;
56     }
57     Serial.print("Grabar ID #");
58     Serial.println(id);
59
60     while (! getFingerprintEnroll() );
61 }
62
63 uint8_t getFingerprintEnroll() {
64
65     int p = -1;
66     Serial.print("Esperando una huella valida para inscribirse como #"); Serial.println(id);
67     while (p != FINGERPRINT_OK) {
68         p = finger.getImage();
69         switch (p) {
70             case FINGERPRINT_OK:
71                 Serial.println("imagen capturada");
72                 break;
73             case FINGERPRINT_NOFINGER:
74                 Serial.println(".");
75                 break;
76             case FINGERPRINT_PACKETRECEIVEERR:
77                 Serial.println("Error de comunicacion");
78                 break;
79             case FINGERPRINT_IMAGEFAIL:
80                 Serial.println("Error de imagen");
81                 break;
82             default:
83                 Serial.println("Error desconocido");
84                 break;
85         }
86     }
87
88     // fase de almacenamiento con exito
```



```

89
90 p = finger.image2Tz(1);
91 switch (p) {
92     case FINGERPRINT_OK:
93         Serial.println("Imagen convertida");
94         break;
95     case FINGERPRINT_IMAGEMESS:
96         Serial.println("Imagen demasiado movida");
97         return p;
98     case FINGERPRINT_PACKETRECEIVEERR:
99         Serial.println("Error de comunicacion");
100        return p;
101    case FINGERPRINT_FEATUREFAIL:
102        Serial.println("No se pudieron encontrar las características de huellas digitales");
103        return p;
104    case FINGERPRINT_INVALIDIMAGE:
105        Serial.println("No se pudieron encontrar las características de huellas digitales");
106        return p;
107    default:
108        Serial.println("Error desconocido");
109        return p;
110 }
111
112 Serial.println("Eliminar huella");
113 delay(2000);
114 p = 0;
115 while (p != FINGERPRINT_NOFINGER) {
116     p = finger.getImage();
117 }
118 Serial.print("ID "); Serial.println(id);
119 p = -1;
120 Serial.println("Coloque el dedo nuevamente");
121 while (p != FINGERPRINT_OK) {
122     p = finger.getImage();
123     switch (p) {
124         case FINGERPRINT_OK:
125             Serial.println("Imagen tomada");
126             break;
127         case FINGERPRINT_NOFINGER:
128             Serial.print(".");
129             break;
130         case FINGERPRINT_PACKETRECEIVEERR:
131             Serial.println("Error de comunicacion");
132             break;
133         case FINGERPRINT_IMAGEFAIL:
134             Serial.println("Error de imagen");
135             break;
136         default:
137             Serial.println("Error desconocido");
138             break;
139     }
140 }
141
142 // fase de almacenamiento exitoso

```

```
143
144 p = finger.image2Tz(2);
145 switch (p) {
146     case FINGERPRINT_OK:
147         Serial.println("Imagen tomada");
148         break;
149     case FINGERPRINT_IMAGEMESS:
150         Serial.println("Imagen borrosa");
151         return p;
152     case FINGERPRINT_PACKETRECEIVEERR:
153         Serial.println("Error de comunicacion");
154         return p;
155     case FINGERPRINT_FEATUREFAIL:
156         Serial.println("No se pudieron encontrar las características de huellas digitales");
157         return p;
158     case FINGERPRINT_INVALIDIMAGE:
159         Serial.println("No se pudieron encontrar las características de huellas digitales");
160         return p;
161     default:
162         Serial.println("Error desconocido");
163         return p;
164 }
165
166 // Conversion exitosa!
167 Serial.print("Crear modelo para #"); Serial.println(id);
168
169 p = finger.createModel();
170 if (p == FINGERPRINT_OK) {
171     Serial.println("Impresiones conincidentes!");
172 } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
173     Serial.println("Error de comunicacion");
174     return p;
175 } else if (p == FINGERPRINT_ENROLLMISMATCH) {
176     Serial.println("Las huellas digitales no coinciden");
177     return p;
178 } else {
179     Serial.println("Error desconocido");
180     return p;
181 }
182
183 Serial.print("ID "); Serial.println(id);
184 p = finger.storeModel(id);
185 if (p == FINGERPRINT_OK) {
186     Serial.println("Almacenada!");
187 } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
188     Serial.println("Error de comunicacion");
189     return p;
190 } else if (p == FINGERPRINT_BADLOCATION) {
191     Serial.println("No se pudo almacenar en esa ubicacion");
192     return p;
193 } else if (p == FINGERPRINT_FLASHERR) {
194     Serial.println("Error al escribir en flash");
195     return p;
196 } else {
```

```

197     Serial.println("Error desconocido");
198     return p;
199 }
200 }

```

Listing A.1: Código de registro de huellas

A.2. Algoritmo para lectura de las huellas

```

1  #include <WiFiEsp.h>
2  #include <WiFiEspClient.h>
3  #include <PubSubClient.h>
4  #include "SoftwareSerial.h"
5  #include <Adafruit_Fingerprint.h>
6
7
8  //Conexion a la red wifi: nombre de la red y contraseña
9  #define WIFLAP "LAB.CONTROL"
10 #define WIFLPASSWORD "labcontrol"
11
12 //Nombre o IP del servidor mosquito
13 char server[50] = "192.170.1.104";
14
15 //Inicializamos el objeto de cliente esp
16 WiFiEspClient espClient;
17
18 //Iniciamos el objeto subscriptor del cliente
19 //con el objeto del cliente
20 PubSubClient client(espClient);
21
22 //Conexion serial para el esp8266 con una comunicacion
23 //serial, pines 2: Tx del modulo y 3: Rx del modulo
24 SoftwareSerial soft(2, 3);
25
26 SoftwareSerial mySerial(4,5); //conexion serial para el lector de huella
27
28 Adafruit_Fingerprint huella = Adafruit_Fingerprint (&mySerial);
29
30 int status = WL_IDLE_STATUS;
31
32
33
34
35 void setup() {
36
37     huella.begin(57600);
38
39     if (huella.verifyPassword())
40     {
41         Serial.println("Sensor de huella dactilar encontrado");
42     } else
43

```

```

44     {
45         Serial.println("No se ha encontrado huella digital ");
46         while (1) { delay(1); }
47     }
48
49     huella.getTemplateCount();
50     Serial.print("Se registran "); Serial.print(huella.templateCount); Serial.println("
    huellas en el sensor");
51     Serial.println("Esperando una huella valida.....");
52     // put your setup code here, to run once:
53     //Iniciamos la comunicacion serial para el log
54     Serial.begin(9600);
55     //Iniciamos la conexion a la red WiFi
56     InitWiFi();
57     //Colocamos la referencia del servidor y el puerto
58     client.setServer( server , 1883 ); //define connection to mqtt mosquito server
59
60
61
62
63 }
64
65 void loop() {
66     // put your main code here, to run repeatedly:
67     status = WiFi.status();
68     if(status != WL_CONNECTED) {
69         //Si falla la conexion, reconectamos el modulo
70         reconnectWifi();
71     }
72
73     //Validamos si esta la conexion del servidor
74     if(!client.connected() ) {
75         //Si falla reintentamos la conexion
76         reconnectClient();
77     }
78
79     client.loop(); //keep connection alive with mqtt server
80     getHuellaPrintIDez ();
81     delay (50);
82
83
84 }
85
86
87 void sendDataTopic()
88 {
89     Serial.println("Send message to mqtt");
90     // Prepare a JSON payload string
91     String payload = "Hola desde arduino";
92
93     // Send payload
94     char attributes[100];
95     payload.toCharArray( attributes , 100 );
96     client.publish( "outTopic", attributes );

```

```

97     Serial.println( attributes );
98 }
99
100
101
102 uint8_t getHuellaPrintID ()
103 {
104     uint8_t h = huella.getImage ();
105     switch (h)
106     {
107         case FINGERPRINT_OK:
108             Serial.println("obteniendo imagen");
109             break;
110         case FINGERPRINT_NOFINGER:
111             Serial.println("huella no detectada");
112             return h;
113         case FINGERPRINT_PACKETRECEIVEERR:
114             Serial.println("error de comunicacion");
115             return h;
116         case FINGERPRINT_IMAGEFAIL:
117             Serial.println("error de imagen");
118             return h;
119         default:
120             Serial.println("error desconocido");
121             return h;
122     }
123
124     h = huella.image2Tz();
125     switch (h)
126     {
127         case FINGERPRINT_OK:
128             Serial.println("Imagen convertida");
129             break;
130         case FINGERPRINT_IMAGEMESS:
131             Serial.println("Imagen sucia");
132             return h;
133         case FINGERPRINT_PACKETRECEIVEERR:
134             Serial.println("error de comunicacion");
135             return h;
136         case FINGERPRINT_FEATUREFAIL:
137             Serial.println("no se encontraron las caracteristicas de las huellas digitales");
138             return h;
139         case FINGERPRINT_INVALIDIMAGE:
140             Serial.println("no se encontraron las caracteristicas de las huellas digitales");
141             return h;
142         default:
143             Serial.println("error desconocido");
144             return h;
145     }
146
147     h = huella.fingerFastSearch ();
148     if (h==FINGERPRINT_OK)
149     {
150         Serial.println("Se encontro una coincidencia!!!");

```

```

151 } else if (h== FINGERPRINT.PACKETRECEIVEERR)
152 {
153     Serial.println("Error de comunicacion");
154     return h;
155 }else if(h == FINGERPRINT_NOTFOUND)
156 {
157     Serial.println("No se encontro una coincidencia");
158     return h;
159 }else
160 {
161     Serial.println("error desconocido");
162     return h;
163 }
164
165 Serial.println("huella digital encontrada"); Serial.println(huella.fingerID);
166 Serial.println("grado de confidencialidad de: "); Serial.println(huella.confidence);
167 String payload = String(huella.fingerID);
168
169 // Send payload
170 char attributes[100];
171 payload.toCharArray( attributes , 100 );
172 client.publish( "huellaValida", attributes );
173 return huella.fingerID;
174 }
175
176 int getHuellaPrintIDez ()
177 {
178     uint8_t h = huella.getImage();
179     if (h != FINGERPRINT_OK) return -1;
180
181     h = huella.image2Tz();
182     if (h != FINGERPRINT_OK) return -1;
183
184     h = huella.fingerFastSearch();
185     if (h != FINGERPRINT_OK) return -1;
186
187     //HUELLA ENCONTRADA
188     Serial.print("Huella encontrada "); Serial.print(huella.fingerID);
189     Serial.print("grado de confidencialidad de: "); Serial.println(huella.confidence);
190     digitalWrite(13, HIGH); //encendido de la chapa
191     delay(3000);           //encendido por un tiempo
192     digitalWrite(13, LOW); //apagado de la chapa
193     return huella.fingerID;
194 }
195 //Inicializamos la conexion a la red wifi
196 void InitWiFi()
197 {
198     //Inicializamos el puerto serial
199     soft.begin(9600);
200     //Iniciamos la conexion wifi
201     WiFi.init(&soft);
202     //Verificamos si se pudo realizar la conexion al wifi
203     //si obtenemos un error, lo mostramos por log y denememos el programa
204     if (WiFi.status() == WL_NO_SHIELD) {

```

```

205     Serial.println("El modulo WiFi no esta presente");
206     while (true);
207 }
208 reconnectWifi();
209 }
210
211 void reconnectWifi() {
212     Serial.println("Iniciar conexion a la red WIFI");
213     while(status != WL_CONNECTED) {
214         Serial.print("Intentando conectarse a WPA SSID: ");
215         Serial.println(WIFI_AP);
216         //Conectar a red WPA/WPA2
217         status = WiFi.begin(WIFI_AP, WIFIPASSWORD);
218         delay(500);
219     }
220     Serial.println("Conectado a la red WIFI");
221 }
222
223 void reconnectClient() {
224     //Creamos un loop en donde intentamos hacer la conexion
225     while(!client.connected()) {
226         Serial.print("Conectando a: ");
227         Serial.println(server);
228         //Creamos una nueva cadena de conexion para el servidor
229         //e intentamos realizar la conexion nueva
230         //si requiere usuario y contrasena la enviamos connect(clientId, username,
password)
231         String clientId = "ESP8266Client-" + String(random(0xffff), HEX);
232         if(client.connect(clientId.c_str())) {
233             Serial.println("[DONE]");
234         } else {
235             Serial.print( "[FAILED] [ rc = " );
236             Serial.print( client.state() );
237             Serial.println( " : retrying in 5 seconds]" );
238             delay( 5000 );
239         }
240     }
241 }

```

Listing A.2: Código de lectura de las huellas

Tesis - Luis Avila Rev1

INFORME DE ORIGINALIDAD

8%

INDICE DE SIMILITUD

6%

FUENTES DE
INTERNET

1%

PUBLICACIONES

4%

TRABAJOS DEL
ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

1%

★ issuu.com

Fuente de Internet

Excluir citas

Activo

Excluir coincidencias

Apagado

Excluir bibliografía

Activo

PERMISO DEL AUTOR DE TESIS PARA SUBIR AL REPOSITORIO INSTITUCIONAL

Yo, **LUIS ENRIQUE AVILA GALLEGOS**, portador (a) de la cédula de ciudadanía Nro.0302686019. En calidad de autor/a y titular de los derechos patrimoniales del trabajo de Titulación **"AUTOMATIZACIÓN DE LOS SISTEMAS DE ACCESO, ILUMINACIÓN Y MONITOREO DEL LABORATORIO DE ELECTRÓNICA DE LA UNIERSIDAD CATÓLICA DE CUENCA SEDE AZOGUES, USANDO TECNOLOGÍA IoT"** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de Los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, Así mismo; autorizo a la Universidad para que realice la publicación de éste trabajo de titulación en Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, 31 de agosto de 2020

F: 
LUIS ENRIQUE AVILA GALLEGOS
0302686019

El Bibliotecario de la Sede Azogues

CERTIFICA:

Que: **AVILA GALLEGOS LUIS ENRIQUE**, con cédula de ciudadanía Nro. **0302686019**, de la Carrera de: **INGENIERÍA ELECTRÓNICA**

No adeuda libros, a esta fecha: **31 de agosto del 2020**.



Byron Alonso Torres Romo
Bibliotecario

Biblioteca Universitaria
MONS. "FROILAN POZO QUEVEDO"