



Facultad de Ingeniería
Ingeniería Electrónica

Programa Especial de Titulación

“Desarrollo de un sistema de monitoreo y control
basado en IoT en apoyo a la Oficina de Soporte de
la OGTI del Ministerio Publico Fiscalía de la Nación
– Distrito Cercado de Lima – Perú”

Jonathan Javier Yalle León

Para optar el Título Profesional de
Ingeniero Electrónico

Asesor: Javier Gonzalo Mansilla Yanqui

Lima – Perú

2021

DEDICATORIA

El siguiente trabajo está dedicado a mi Dios y a mi familia por el apoyo permanente.

AGRADECIMIENTO

Agradecer sobre todo a Dios por guiarme por buen camino y a mi querida familia por el apoyo permanente.

RESUMEN

En la Oficina de Soporte de la Oficina General de Tecnologías de la Información del Ministerio Público Fiscalía de la Nación en la Sede Principal Cercado de Lima - Perú, se encuentra el Área de custodia de Bienes en Tránsito, la cual es crítica ya que ahí se encuentran en custodia activos de T.I., sin embargo estos activos de T.I. se ven afectados en pérdida o deterioro a causa de un inadecuado monitoreo y control del acceso de personal autorizado y de una elevada temperatura del área de trabajo que anteriormente ha provocado amago de incendio.

Por consiguiente, el siguiente trabajo desarrolla un sistema de monitoreo y control basado en el Internet de las Cosas (IoT), que utiliza una metodología de investigación de enfoque cuantitativo y de alcance correlacional, empleando la recolección de datos de análisis de contenido, la observación y comparando trabajos con problemáticas similares. Todo esto en apoyo al personal responsable del Área de custodia de Bienes en Tránsito y de la Gerencia de la Oficina de Soporte del Ministerio Público Fiscalía de la Nación.

Se detalla también los aportes, resultados y conclusiones satisfactorios del trabajo siendo los más importantes lograr un control de acceso biométrico auditable, un monitoreo de la temperatura del área de trabajo y el envío de alertas tempranas basadas en IoT que previene de amagos de incendios en el área de trabajo.

ÍNDICE DE CONTENIDO

| | |
|--|-----------|
| DEDICATORIA..... | II |
| AGRADECIMIENTO..... | III |
| RESUMEN..... | IV |
| ÍNDICE DE FIGURAS..... | VII |
| ÍNDICE DE TABLAS..... | VIII |
| INTRODUCCIÓN..... | IX |
| CAPITULO 1 | 10 |
| ASPECTOS GENERALES | 10 |
| 1.1. Definición del Problema | 10 |
| 1.1.1. Descripción del Problema | 10 |
| 1.1.2. Formulación del Problema | 11 |
| 1.2. Definición de Objetivos | 11 |
| 1.2.1. Objetivo General | 11 |
| 1.2.2. Objetivos Específicos | 11 |
| 1.3. Alcances y limitaciones | 12 |
| 1.3.1. Alcances | 12 |
| 1.3.2. Limitaciones | 12 |
| 1.4. Justificación | 12 |
| 1.5. Estado del Arte | 13 |
| CAPITULO 2 | 16 |
| MARCO TEÓRICO | 16 |
| 2.1. Antecedentes | 16 |
| 2.1.1. Antecedentes Nacionales | 16 |
| 2.1.2. Antecedentes Internacionales | 17 |
| 2.2. Fundamento teórico. | 19 |
| 2.2.1. Computador de Placa Única o Reducida (Single-Board Computer) | 19 |
| 2.2.1.1. Raspberry Pi 4 | 19 |
| 2.2.2. Arduino | 21 |
| 2.2.2.1. Arduino Mega | 21 |
| 2.2.3. Sensores y actuadores | 22 |
| 2.2.4. Internet de las cosas (IoT) | 27 |
| 2.2.5. Red de comunicaciones | 28 |
| 2.2.6. Protocolo de comunicación TCP/IP | 28 |

| | |
|--|-----------|
| 2.2.7. Topología de red | 29 |
| 2.3. Marco conceptual | 30 |
| 2.4. Marco Metodológico | 31 |
| 2.4.1. Ciclo de vida del proyecto | 31 |
| 2.4.2. Ciclo de vida del producto | 32 |
| CAPITULO 3 | 34 |
| DESARROLLO DE LA SOLUCIÓN | 34 |
| 3.1. Criterio de selección de componentes..... | 34 |
| 3.1.1. Selección de módulo de monitoreo y sensor de temperatura | 34 |
| 3.1.2. Selección del módulo de control | 36 |
| 3.2. Diagrama de bloques..... | 37 |
| 3.2.1. Diseño del sistema de monitoreo | 38 |
| 3.2.2. Identificación de las entradas y salidas del sistema de monitoreo | 38 |
| 3.2.4. Identificación de las entradas y salidas del sistema de control de acceso | 40 |
| 3.3. Condiciones del Soporte Eléctrico local..... | 40 |
| 3.3.1. Actuador para accionamiento de equipos de enfriamiento local..... | 42 |
| 3.4. Proceso de implementación | 43 |
| 3.4.1. Implementación del Sistema de Monitoreo | 44 |
| 3.4.2. Implementación del Sistema de Control de Acceso | 45 |
| CAPITULO 4 | 48 |
| RESULTADOS | 48 |
| 4.1. Análisis de resultados obtenidos | 48 |
| 4.1.1. Monitoreo al Área de Custodia de Bienes en tránsito de la Oficina de Soporte de la OGTI | 48 |
| 4.1.2. Control de acceso al Área de Custodia de Bienes en tránsito de la Oficina de Soporte de la OGTI | 50 |
| 4.2. Presupuesto..... | 51 |
| CONCLUSIONES | 53 |
| RECOMENDACIONES..... | 54 |
| BIBLIOGRAFÍA..... | 55 |
| ANEXOS | 57 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Raspberry Pi 4 Modelo B | 20 |
| Figura 2. Arduino Mega 2560..... | 21 |
| Figura 3. Módulo lector de huella..... | 24 |
| Figura 4. Módulo I2C conectado a LCD 16x2 | 24 |
| Figura 5. Posicionamiento del eje del servomotor en relación al tiempo de pulso. | 25 |
| Figura 6. Circuito de funcionamiento de RTC DS3231..... | 26 |
| Figura 7. Esquema y distribución de Pines del Módulo Micro SD card..... | 26 |
| Figura 8. Relevador distribución de pines y simbología | 27 |
| Figura 9. Topología estrella | 29 |
| Figura 10. Topología en bus | 30 |
| Figura 11. Topología en anillo..... | 30 |
| Figura 12. Ciclo de vida del proyecto | 32 |
| Figura 13. Ciclo de vida del producto..... | 33 |
| Figura 14. Diagrama en bloques..... | 37 |
| Figura 15. Sistema de monitoreo basado en IoT..... | 38 |
| Figura 16. Sistema de control de acceso. | 40 |
| Figura 17. Diagrama de bloque UPS Interactivo Centralion Blazer 600 | 42 |
| Figura 18. Diagrama de bloque del actuador de enfriamiento..... | 43 |
| Figura 19. Diagrama de bloques de implementación del sistema de monitoreo..... | 44 |
| Figura 20. Diagrama de Flujo de monitoreo | 45 |
| Figura 21. Diagrama de bloques de implementación del sistema de control de acceso...46 | |
| Figura 22. Diagrama de flujo de control | 47 |
| Figura 23. Gráfica de Temperatura (°C) contra el tiempo (formato 24h) de los días 2021-03-05 y 2021-03-06 | 48 |
| Figura 24. Correo electrónico de alerta temprana contra amago de incendio | 49 |
| Figura 25. Plataforma ThingSpeak utilizado por la supervisión de la Oficina de Soporte .50 | |
| Figura 26. Visualización dinámica de temperatura en plataforma Thingspeak. | 50 |
| Figura 27. Gráfica de registro de ingreso contra las horas de ingreso del mes de marzo 2021. | 51 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Comparativo de selección de módulo de monitoreo | 35 |
| Tabla 2. Tabla comparativa de sensor de temperatura | 35 |
| Tabla 3. Comparativo de selección de módulo de control | 36 |
| Tabla 4. Comparativa de selección de submódulo de sensor biométrico | 36 |
| Tabla 5. Características del UPS Interactivo Centralion Blazer 600 | 41 |
| Tabla 6. Descripción básica de equipos de aire acondicionado | 42 |
| Tabla 7. Descripción básica de equipos de aire acondicionado | 43 |
| Tabla 8. Presupuesto del Sistema de Monitoreo y Control basado en IoT | 51 |

INTRODUCCIÓN

Los sistemas desarrollados actualmente a nivel mundial de control y monitoreo basados en IoT se implementan con diversas tecnologías como Single Board Computer (SBC), microcontroladores, sensores de temperatura, de presión, óptico, inalámbricos, entre otros.

Son una alternativa de solución que brindan un mejor rendimiento de los activos o del proceso de negocio que se desea salvaguardar reduciendo de manera significativa los costos e inadecuados usos, permitiendo tomar acción de manera oportuna antes que estos fallen.

Además, la coyuntura global a acelerado de manera drástica la transformación digital por lo cual las entidades privadas y gubernamentales de nuestro país han tomado acción implementando estos sistemas de manera progresiva.

El siguiente trabajo desarrolla un sistema de monitoreo y control basado en IoT que emplea un como modulo principal Raspberry Pi apoyado con un módulo secundario Arduino Mega 3560 y sensores de temperatura y humedad de alta presión, sensor biométrico y otros actuadores, lo que logra aportar un control de acceso auditable, un monitoreo de la temperatura del área de trabajo y el envío de alertas tempranas para evitar amagos de incendio. Todo esto en apoyo al Área de custodia de Bienes en Transito de la Oficina de Soporte que pertenece a la Oficina General de Tecnologías de la Información del Ministerio Publico Fiscalía de la Nación en la Sede Principal Cercado de Lima – Perú.

CAPITULO 1

ASPECTOS GENERALES

1.1. Definición del Problema

1.1.1. Descripción del Problema

En la Oficina de Soporte de la Oficina General de Tecnologías de la Información (OGTI) del Ministerio Público Fiscalía de la Nación (MPFN) ubicada en el piso 10 de la Sede Principal Cercado de Lima - Perú, se encuentra el Área de custodia de Bienes en Tránsito, la cual es una área crítica ya que ahí se ubica en custodia Activos de T.I., como componentes electrónicos, consumibles de equipos de impresión, equipos multifuncionales, equipos de cómputo, equipos de digitalización, entre otros y equipos servidor y pseudo servidor operando 24/7, los cuales permiten brindar el soporte tecnológico de primera línea tanto para el área fiscal como administrativa a nivel nacional del MPFN.

En dicho piso de la sede se ha visto reportados amagos de incendio que han perjudicado el servicio de los sistemas informáticos fiscales y administrativos, causado principalmente por el sobrecalentamiento de los equipos servidores que se alojan ahí, elevando la temperatura de trabajo, que ha llevado a la pérdida de Activos de T.I., es así que fueron noticia en los medios de comunicación.

Adicionalmente se ha evidenciado que en los inventarios de Activos de T.I., bienes no ubicados o deteriorados, la mayoría de casos las responsabilidades no se pueden determinar, puesto que el control de acceso al área en cuestión no era registrado.

Por lo tanto, la problemática, en resumen, es una inadecuada gestión de monitoreo y control de los Activos de T.I. en el Área de custodia de bienes en tránsito de la Oficina de Soporte de la OGTI – MPFN.

1.1.2. Formulación del Problema

De acuerdo al planteamiento del problema se establece el siguiente problema general:

- Inadecuada gestión de monitoreo y control de los Activos de T.I. en el área crítica de Custodia de Bienes en Tránsito de la Oficina de Soporte de la OGTI – MPFN.

Con los siguientes problemas específicos:

- Inadecuado control de acceso de personal.
- Inadecuada auditoría de acceso de personal autorizado.
- Sobrecalentamiento de equipos servidores y pseudo servidores.
- Ineficiente sistema de alerta temprana contra amagos de incendios.

1.2. Definición de Objetivos

1.2.1. Objetivo General

Desarrollar un sistema de monitoreo y control basado en IoT en apoyo al Área de Custodia de Bienes en Tránsito de la Oficina de Soporte de la OGTI – MPFN.

1.2.2. Objetivos Específicos

- Desarrollar un sistema de monitoreo de la temperatura de trabajo en tiempo real.
- Desarrollar un sistema de control de acceso utilizando biometría dactilar.

- Implementar un registro de auditoria de ingreso de personal autorizado.
- Desarrollar un sistema de alerta temprana contra amagos de incendio basado en IoT.

1.3. Alcances y limitaciones

1.3.1. Alcances

El desarrollo del trabajo, permite con éxito preservar y prolongar la vida útil de los activos de T.I. del Área de Custodia de Bienes en Transito a la Oficina de Soporte de la OGTI, aplicando controles de acceso de personal, auditoria, monitoreo de temperatura y alerta temprana contra amago de incendio, ello utilizando como sistema principal un Raspberry Pi 4 modelo B, seleccionado por su alta capacidad de conectividad de red, procesamiento, memoria, programación, numero entradas y salidas, que permite tomar acción de manera oportuna y de remota a través de la internet.

Además, está compuesto por un sistema secundario que utiliza el control de autenticación biométrica de huella dactilar integrado por un Arduino mega 2560 utilizado por su capacidad de memoria suficiente para procesar los sensores y actuadores necesarios, las entradas y salidas, a fin de procesar el registro de huellas dactilares, controlar los módulos de reloj de tiempo real y de almacenamiento de auditoria.

1.3.2. Limitaciones

El sistema de control no utiliza biometría de reconocimiento facial, voz u ocular.

El registro de acceso de personal no se expone de manera externa o hacia la internet, puesto que utiliza un almacenamiento local.

El trabajo se limita su uso para la Gerencia y el personal encargado del Área de Custodia de Bienes Transito de la Oficina de Soporte de la OGTI – MPFN.

1.4. Justificación

El trabajo que se realiza se justifica plenamente por lo siguiente:

Económicamente, por contener componentes de bajo costo en su implementación y permite ahorrar en compra de activos de T.I. ya que prolonga la vida de estos.

Socialmente, puesto que permite una alta disponibilidad de los sistemas informáticos del MPFN en apoyo a la ciudadanía.

Tecnológicamente, puesto que se desarrolla con tecnología a medida para dar solución a la problemática descrita anteriormente, utilizando equipos electrónicos de vanguardia y tendencia en la actualidad.

1.5. Estado del Arte

Actualmente las investigaciones orientadas a control y monitoreo empleando IoT son una alternativa de solución que brinda un mejor rendimiento de los activos o del proceso de negocio que se desea salvaguardar reduciendo de manera significativa los costos e inadecuados usos, permitiendo tomar acción de manera oportuna antes que estos fallen.

Mediante el empleo de dispositivos de tecnología como computador de placa única o reducida, microcontroladores, sensores de temperatura, de presión, óptico, inalámbricos, entre otros, permite obtener resultados particulares que satisfacen el cumplimiento de los objetivos planteados en cada investigación revisada. Utilizando una metodología de enfoque cuantitativo y de alcance correlacional, empleando la recolección de datos de análisis de contenido y de la observación.

Los trabajos de diseño e implementación de monitoreo (Bravo Guillén, G. L., 2016) aporta al sector industrial una alternativa de ahorro de costos en mantenimiento de vehículos industriales tipo montacarga, mediante la medición de los parámetros críticos en dichos vehículos, permitiendo así mantener su operatividad.

El desarrollo de un sistema de seguridad electrónica de control y monitoreo aplicado a empresas privadas o públicas (Sánchez Capistrano, J. B., 2019) a través de circuitos

diseñados a medida y utilizando PLC, logra con éxito prevenir y disuadir de actos delictivos como robo o intrusión contra una determinada empresa.

El desarrollo de un sistema de control y monitoreo utilizando IoT (Valqui Cieza, R., 2019) es un aporte importante ya que permite obtener la información en tiempo real, a fin de evitar escenarios de riesgo como fuga de gases tóxicos en el sector minería.

El monitoreo de las estaciones de vigilancia meteorológicas basado en IoT (D. Pujara, P. Kukreja and S. Gajjar, 2020) midiendo la temperatura, humedad, índice de calidad del aire, entre otros, logra con éxito de manera automatizada tomar acción de manera oportuna ante cambios meteorológicos, a fin de evitar perjuicio a la ciudadanía de un determinado lugar geográfico.

La combinación de tecnologías de monitoreo IoT y clasificación inteligente (C. Chung, I. Peng and J. Yeh, 2020) aporta un valor agregado al desarrollo del trabajo de suficiencia que se realiza, permitiendo el monitoreo, control y la selección de desperdicios en beneficio del ciudadano.

El control y monitoreo de la contaminación ambiental es crítico para determinadas áreas y utilizar IoT (G. Parmar, S. Lakhani and M. K. Chattopadhyay, 2020) es la mejor opción al trabajo que se realiza.

El control de temperatura es el principal aporte de este trabajo (V. Hoang, V. Dao and V. Dang, 2020) con el valor agregado que emplea un protocolo de comunicación de largo alcance (LoRa) local, logrando con éxito tomar acción de manera oportuna ante escenarios de sequía en una determinada área geográfica.

Otro trabajo de monitoreo de la calidad del aire resulta crítico basado en NB-IoT (Y. Cheng, X. Xu, Y. Du, P. Guan, S. Liu and L. Zhao, 2019), utilizando microcontrolador STM32FI03RCT6 para procesar localmente la información recopilada, además puede ser

aplicado a un taller de trabajo con sensores inalámbricos (D. Li, Z. Bao and Y. Yang, 2016) que recogen el estado del ambiente de trabajo óptimo.

El monitoreo y control de las condiciones de trabajo de máquinas eléctricas (H. Barksdale, Q. Smith and M. Khan, 2018) resulta crítico para temas de costos (temperatura, humedad, rendimiento, entre otros) utilizando también IoT.

Otro trabajo considerado importante en este aspecto de control de energía basado en WebAccess (Y. Liu, J. Leng and B. Sun, 2017) resulta interesante su aplicación como desarrollo en el sector energético, permitiendo una administración centralizada y automatizada.

Otro aporte en monitoreo y control es aplicado a las máquinas de soldar (Z. Yu-Long, C. Heng, H. Jing-Feng and W. Jin-Cheng, 2018), utilizando tecnología inalámbrica como WIFI e IoT.

El trabajo de monitoreo y control en el área servidores (A. Roihan, F. Sudarto and T. Cahyo Putro, 2018) es un aporte crítico para la continuidad de los servicios que brinda una determinada entidad u organización.

Por lo cual se determina que los sistemas de monitoreo y control basados en IoT aplicado a la continuidad de negocio de una determinada entidad u organización es fundamental para el crecimiento de la misma permitiendo con éxito reducir los costos y usos inadecuados de los activos de dicha entidad u organización en beneficio de sus clientes o de la sociedad.

CAPITULO 2

MARCO TEÓRICO

2.1. Antecedentes

2.1.1. Antecedentes Nacionales

Saúl Ítalo Vizcarra Cavero (2019). "Diseño e implementación de un sistema de monitoreo inalámbrico de bancos de baterías utilizando Arduino mega 2560", Universidad Tecnológica del Perú, Perú, el cual tiene como objetivo el desarrollo de un sistema basado en IoT que permite monitorear en tiempo real los parámetros de eficiencia de los bancos de baterías de una empresa de telecomunicaciones utilizando como componente principal la plataforma de hardware libre Arduino mega, la tarjeta de Wifi (ESP 8266) como medio de comunicación hacia la internet empleando la plataforma Web Thing Speak y una fuente de alimentación conmutada de 800 watts que alterna con una batería de 12 voltios 12AH en caso de fallas en el fluido eléctrico, concluyendo que logra con éxito el monitoreo de los bancos de baterías de manera remota utilizando la plataforma Web Thing Speak.

Nelson Eduardo Huiman Tocto (2017), "Diseño e implementación de una red de medidores de energía para artefactos domésticos", Pontificia Universidad Católica del Perú, Perú, en el cual tiene como objetivo que mediante módulos clientes utilizando un dispositivo medidor energía (AD71056) y un módulo inalámbrico (ESP12E) como componentes principales para medir y transmitir el consumo eléctrico de los artefactos domésticos, así mismo el

servidor recolecta la información de los clientes y lo muestra al usuario a través de un servicio web utilizando un módulo Ethernet, concluyendo que logra con éxito almacenar los registros de las mediciones de energía eléctrica en tiempo real y poder acceder desde la internet a estos registros de forma fiable y fácil para el usuario, detallando el consumo de energía doméstica en soles o kilo watts hora.

Hansell Valmir Mora Magallanes (2019), “Diseño, desarrollo e implementación de una red de sensores inalámbricos para el control, monitoreo y toma de decisiones aplicado en la agricultura basado en IoT – Caso de estudio cultivo de frijol”, Universidad Ricardo Palma, Perú, el cual tiene como objetivo implementar el monitoreo y control basado en IoT utilizando un módulo Wifi (ESP-12E) como cliente, el cual gestiona la comunicación entre los sensores y la plataforma del servidor implementada en un equipo de cómputo, siendo de arquitectura tipo estrella. Las variables de medición de los sensores fueron de temperatura y humedad del entorno, humedad de la tierra, de existencia de lluvia y de la cantidad de humo, concluyendo que logra con éxito monitorear las variables determinantes del ambiente de agricultura deseado y de conectarse a la internet (IoT), almacenado los registros en la nube de internet permitiendo la toma de decisiones oportuna frente al cultivo del frijol.

2.1.2. Antecedentes Internacionales

Jeysson Emilio Lima Guaqueta y Pedro Leonardo Ospina Fuentes (2018), “Diseño de un sistema de adquisición y visualización de datos basado en la plataforma de sistemas embebidos Raspberry Pi”, Universidad Santo Tomas, Colombia, el cual tiene como objetivo desarrollar un sistema de obtención y presentación de datos que permite mostrar al usuario el confort térmico de su entorno mediante los datos de temperatura y humedad, utilizando como modulo principal la computadora de placa reducida Raspberry Pi el cual hace las veces de servidor que procesa los datos enviándolo a la nube y para medir las variables de temperatura y humedad relativa emplea el DHT11, concluyendo que logra con éxito

estimar la sensación térmica del entorno en tiempo real y mostrarlo de una manera simple a través de una plataforma web al usuario por medio de la internet.

Zareen Choudhury (2019), "Configuración de la conectividad inalámbrica y los límites espaciales para dispositivos de monitorización de IoT", Massachusetts Institute of Technology, USA, el cual tiene como objetivo desarrollar un sistema inalámbrico que permita la configuración de seguridad y brindar los límites de funcionamiento a los dispositivos de monitorización de IoT, utilizando como modulo principal la Placa de desarrollo Nvidia Jetson TK1 con tarjetas Intel y un USB Wi-Fi para la conexión hacia la internet, adicionalmente desarrolla un wifiapp como una aplicación web que evita la dependencia del sistema operativo presentes en las aplicaciones móviles, concluyendo que el sistema desarrollado permite configurar con éxito todos los dispositivos IoT a nivel de seguridad de red y brindar el alcance de conectividad de ellos, ofreciendo una solución escalable y precisa para configurar límites de espacios simples y complejos.

Daniel Alejandro Orellana González (2015), "Implementación y pruebas del estándar de comunicación de servicios de distribución de datos para la aplicación de micro-redes gestionados por computadores de placa única", Universidad de Chile, Chile, el cual tiene como objetivo desarrollar un sistema distribuido para habilitar el crecimiento de micro-redes usando el protocolo de Servicio de Distribución de Datos por medio del monitoreo continuo de parámetros predefinidos, para lo cual utiliza como modulo principal el computador de placa reducida o única Raspberry Pi y equipos de cómputo, para la conexión TCP/IP utiliza red cableada UTP cat. 5 y un switch de comunicación a 100Mbps, concluyendo que logra con éxito la monitorear y controlar la transmisión efectiva de datos de control a través de la red a de todos los dispositivos conectados utilizando el protocolo de Servicio de Distribución de Datos, empleando la computadora de placa única o reducida Raspberry Pi.

2.2. Fundamento teórico.

2.2.1. Computador de Placa Única o Reducida (Single-Board Computer)

Es una computadora de bajo costo y funcional con los componentes necesarios como el microprocesador, entradas, salidas, con una memoria RAM determinada y sin ranuras de expansión de fábrica para periféricos. Este diseño de infraestructura permite una alta fiabilidad a la hora de ejecutar programas a fin de controlar dispositivos complejos.

Mientras que los equipos convencionales de tipo escritorio y portátiles destacan por su uso masificado en entornos empresariales, la estructura simplista de los computadores de placa reducida o única brindan un procesamiento dedicado con menor margen de error aplicado a entornos ágiles de conectividad a internet como en el caso de dispositivos IoT.

Por lo tanto, para ejecutar programas críticos de monitoreo o control son preferidos los computadores de placa reducida o única como en controlar semáforos, sistemas de imágenes médicas, monitoreo del ambiente, etc. Adicionalmente por su bajo consumo de energía es utilizado en entornos de telefonía móvil, electrodomésticos, dispositivos Smart, etc.

2.2.1.1. Raspberry Pi 4

Es un computador de placa reducida o única basado en Linux, empleando el sistema operativo Raspbian de distribución Debian, el cual se instala en una memoria microSD, descargándolo desde el sitio web del fabricante.

Tiene múltiples formas de uso como servidor web, centro multimedia, controlador de robots, monitor de red, sistemas de seguridad, computadora de escritorio básico, entre otros. Siendo de bajo costo y altamente fiable aplicado al internet de las cosas (IoT).

Existen varias generaciones de Raspberry empezando por Raspberry Pi modelo A, modelo B y Zero. Se detalla el Modelo B, el cual es el utilizado en el presente trabajo.

2.2.1.2. Raspberry Pi 4 Modelo B

Es el más reciente de los Raspberry Pi, brindando mayores capacidades de procesamiento, rendimiento, memoria, conectividad a diferencia de la generación anterior, pero manteniendo al mismo tiempo es afín con versiones anteriores y tiene un consumo energético similar.

Los principales atributos de este producto es que incorpora un procesador de 4 núcleos de 64 bits, soporte de doble pantalla con resoluciones de hasta 4K a través de un par de puertos micro-HDMI, hasta 4 GB de RAM, LAN inalámbrica de doble banda de 2,4/5,0 GHz, BT 5.0, Gigabit Ethernet, USB 3.0 y capacidad de PoE, como se puede observar en la Figura 1.

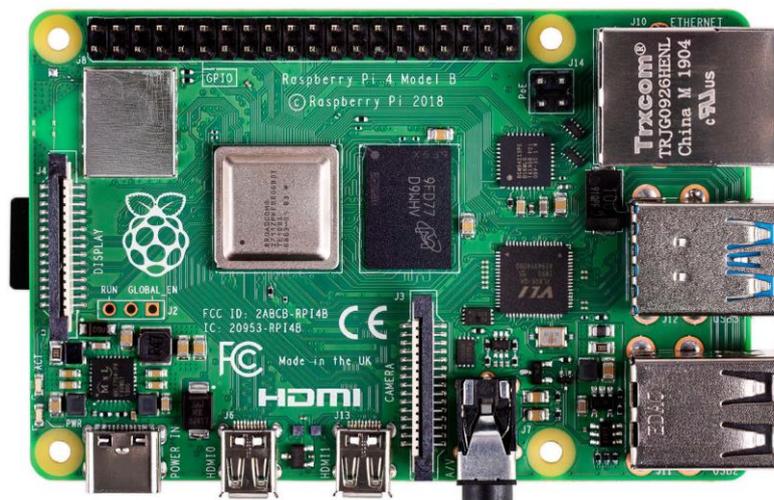


Figura 1. Raspberry Pi 4 Modelo B

Fuente: Raspberry Pi 4 Modelo B recuperado de

<https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>

2.2.1.3. Sistema operativo y lenguaje de programación Raspberry Pi

Como se había mencionado anteriormente el sistema operativo es Linux basado en Debian y es proporcionado por el fabricante.

El lenguaje de programación utilizado por en el Raspberry Pi es Python el cual es independiente de plataforma y orientado a objetos, permitiendo conectar el proyecto desarrollado con IoT.

La sintaxis de Python es ordenada en su programación, reconoce sangrías del código para compilar y las palabras que utiliza son nativas del inglés.

2.2.2. Arduino

El entorno de desarrollo es de código y hardware libre, el cual utiliza un microprocesador ATMEL para su funcionamiento, con entradas y salidas analógicas y digitales, programado en Java y basado en Processing como lenguaje de programación de código abierto. Existen varios modelos de Arduino entre los representativos está el Arduino Uno, Due, Leonardo, Mega, entre otros

2.2.2.1. Arduino Mega

El modelo Arduino Mega emplea el ATmega 2560 como microcontrolador, tiene 54 pines de entradas y salidas digitales, 16 entradas analógicas, un oscilador de cristal de 16 MHz, 4 UARTs, una conexión USB, memoria de 256 KB y un pulsador de reinicio. Este modelo es acorde con la mayor parte de los módulos diseñados para otros modelos de Arduino. La programación se realiza mediante conexión USB compatible con los sistemas operativos más utilizados, se muestra el Arduino mega 2560 en la Figura 2.



Figura 2. Arduino Mega 2560

Fuente: Arduino Mega 2560 recuperado de <https://store.arduino.cc/usa/mega-2560-r3>

2.2.2.2. Sistema y lenguaje de programación Arduino

Como se había mencionado anteriormente el Arduino se programa desde conexión USB mediante la escritura de código y la carga en la placa. Funciona en los sistemas más utilizados como Windows, Linux y Mac OS X. Este software puede ser usado con cualquier placa Arduino.

Dentro de la interface de programación existen librerías que se pueden descargar e integrar para proporcionar mayores funcionalidades, a fin de utilizar diferentes sensores u actuadores. Si se desea ampliar las funcionalidades de la plataforma Arduino, siempre podemos recurrir a una gran variedad de módulos compatibles prácticamente con cualquiera de sus modelos.

2.2.3. Sensores y actuadores

2.2.3.1. Termistor

Es una resistencia fabricada de material de tipo semiconductor sensitivo a las variaciones de temperatura y variando su resistividad en función a los cambios de energía térmica (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2014).

Existen dos grupos uno de coeficiente de temperatura negativo (NTC) y coeficiente de temperatura positivo (PTC).

Adicionalmente en este tipo de sensor la función que describe la resistencia y la temperatura no es lineal, y está dada por la siguiente ecuación:

$$R_t = R_0 e^{\beta \left(\frac{1}{T} - \frac{1}{T_0} \right)}$$

Donde:

R_t : Resistencia del termistor a la temperatura T en Kelvin

T_0 : Temperatura de referencia en Kelvin, 298 Kelvin

R_0 : Resistencia referida a la temperatura T_0 en Kelvin

β : Constante sobre rangos moderados de temperatura que es proporcionado por fabricante del termistor

Los valores de β se encuentran en el rango de los 2000 K y 5000 K, asimismo existe un efecto de consideración por la circulación de la corriente, que es el calentamiento del dispositivo. Ello produce un error de medición de aproximadamente ± 0.3 °C en un rango de 0 a 50 °C y la ecuación que une la potencia consumida con la temperatura del termistor se describe como se muestra:

$$P_{Th} = \delta_d(T_{Th} - T_{amb})$$

Donde:

T_{Th} : Temperatura de trabajo [K]

T_{amb} : Temperatura ambiente [K]

δ_d : Constante de disipación de calor [W/K]

2.2.3.2. Sensor de huella dactilar

Módulo de sensor óptico de huellas dactilares/biométrico con interfaz TTL/UART, tiene la capacidad de almacenar huellas, a fin de escanear y encontrar coincidencias permitiendo identificar la identidad de la persona registrada. Se muestra en la Figura 3 la distribución de los pines y el modelo del sensor.

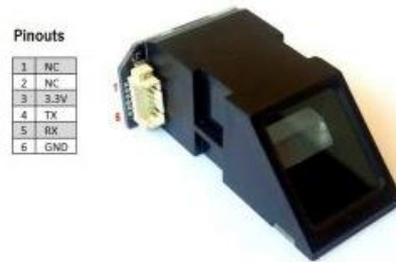


Figura 3. Módulo lector de huella

Fuente: Módulo lector de huella recuperado de <https://artofcircuits.com/product/fpm10a>

2.2.3.3. Módulo I2C y LCD 16x2

La pantalla LCD 16x2 es un periférico que mediante un módulo adaptador I2C permite ser configurado a través del bus serial I2C, como se presenta en la Figura 4.



Figura 4. Módulo I2C conectado a LCD 16x2

Fuente: Modulo I2C LCD 16x2 recuperado de <https://grupoelectrostore.com/wp-content/uploads/2020/08/16x2-i2c-lcd-blackgreen-02.jpg>

2.2.3.4. Servomotor

Es uno de los más utilizados como actuadores incorporados en los sistemas electrónicos. Compuesto por un motor con un reductor de velocidad, un multiplicador de fuerza y un circuito de control, con un ángulo de giro del eje de 180°.

Es conveniente mencionar para el control de este tipo de actuador debe aplicarse un pulso de duración y frecuencia específico, integrado por tres cables, el primero para alimentación Vcc (4.8 a 6v), el segundo para conexión a GND y el tercer utilizado para aplicar una

secuencia de pulsos de control. En la Figura 5 se muestra los posicionamientos del eje del servomotor respecto al ancho del pulso, donde se obtiene 0° , 90° y 180° con pulsos de 0.5, 1.5 y 2.5 ms respectivamente.

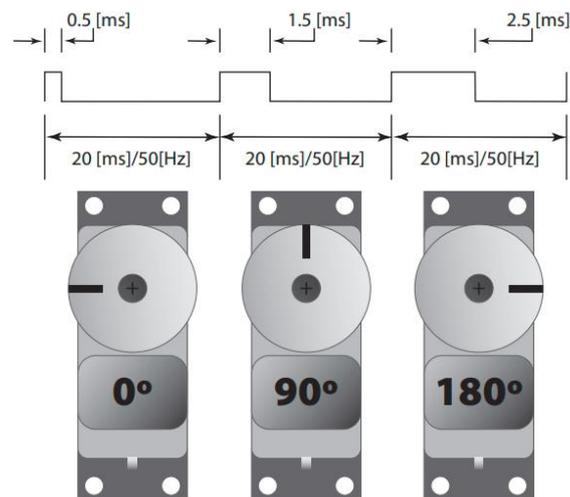


Figura 5. Posicionamiento del eje del servomotor en relación al tiempo de pulso.

Fuente. (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2014)

2.2.3.5. Modulo reloj de tiempo real RTC DS3231

El módulo es de bajo costo y alta confiabilidad con un oscilador de cristal compensado por temperatura (TCXO) y otro cristal incorporado.

Además, incorpora una entrada de batería que mantiene la precisión de la hora en el momento que se interrumpe la alimentación principal del módulo. La incorporación del resonador de cristal mantiene la precisión a largo plazo y decrementa la cantidad de componentes en su fabricación.

El módulo soporta la información de segundos, minutos, horas, día, fecha, mes y año, la fecha final de cada mes se adapta automáticamente para los meses con menos de 31 días, contemplando años bisiestos. Se presenta el circuito de funcionamiento en la Figura 6.

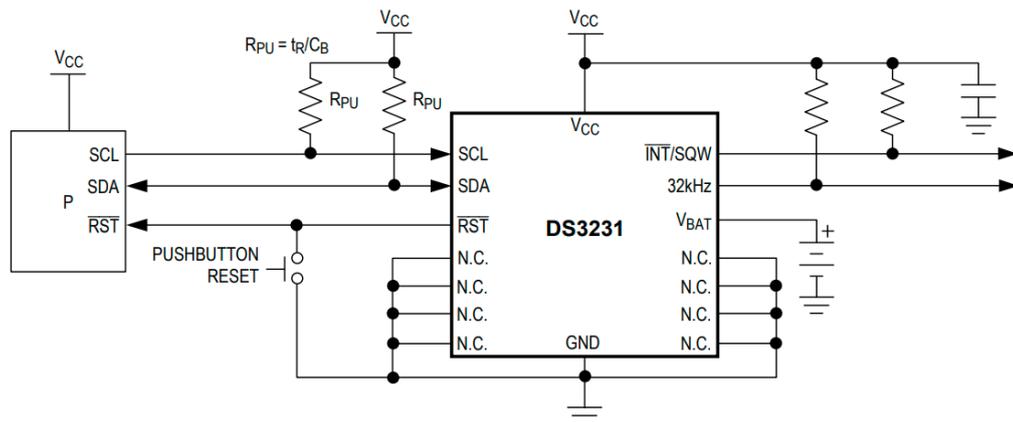


Figura 6. Circuito de funcionamiento de RTC DS3231

Fuente: Circuito de funcionamiento de RTC DS3231 recuperado de

<https://datasheets.maximintegrated.com/en/ds/DS3231-DS3231S.pdf>

2.2.3.6. Modulo micro SD card

Soporta tarjetas micro SD y micro SDHC. El módulo de conversión de voltaje puede conectarse a 3.3v o 5v, la interfaz de comunicación que utiliza es serial por el bus SPI (Serial Peripheral Interface), se presenta el esquema y distribución de pines en la Figura 7.

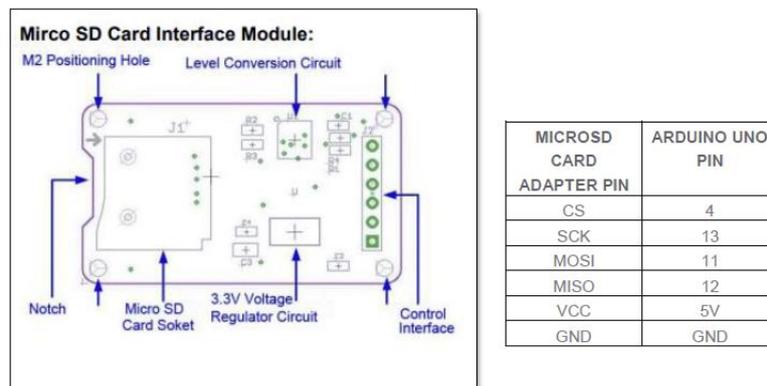


Figura 7. Esquema y distribución de Pines del Módulo Micro SD card

Fuente: Modulo Micro SD card recuperado de

http://synacorp.my/v3/en/index.php?controller=attachment&id_attachment=

2.2.3.7. Relevador

Es un dispositivo electromecánico de conmutación, el cual está compuesto de un interruptor que es accionado por una corriente eléctrica que pasa por una bobina con un núcleo ferromagnético, convirtiéndose en un imán. A más intensidad de corriente y el número de vueltas de la bobina mayor es el efecto del campo magnético producido por el imán. Para el presente trabajo se utiliza un relevador a 5v y 1 canal, el cual necesita 15-20mA de corriente de conductor, se muestra la distribución de pines y la simbología en la Figura 8.

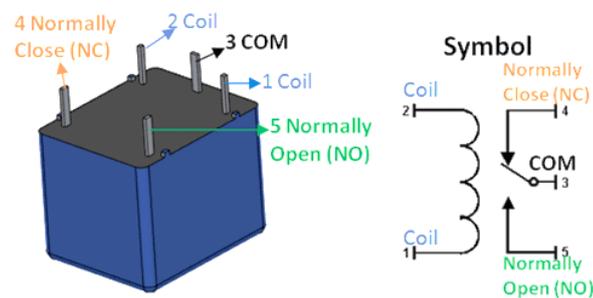


Figura 8. Relevador distribución de pines y simbología

Fuente: Revelador recuperado de <https://1sheeld.com/wp-content/uploads/2018/08/Relay-Pinout.png>

2.2.4. Internet de las cosas (IoT)

Hace referencia a la conexión de dispositivos a internet que intercambian y procesan información sobre su entorno físico, además reconocen cambios o variaciones, y dichos sistemas pueden reaccionar de forma autónoma y adecuada (Barrio Andrés, 2018).

En otras palabras, la IoT es una colección de dispositivos electrónicos que pueden compartir información entre ellos mismos.

Los dispositivos de IoT se llaman a menudo dispositivos "smart" porque tienen sensores y pueden realizar análisis de datos.

Los dispositivos de IoT recogen datos mediante sensores y ofrecen servicios al usuario basados en el análisis de esos datos y de acuerdo con parámetros definidos por el usuario.

2.2.5.Red de comunicaciones

Las redes de comunicación basadas en la transmisión de datos en serie son la plataforma de los sistemas de automatización actuales. Las redes de comunicación ofrecen una serie de ventajas con respecto a los sistemas en los que una línea punto a punto permite que sólo dos participantes se comuniquen entre sí.

Clasificación de las redes de comunicación, dependiendo de la aplicación, es decir, fabricación, proceso, oficina o automatización de edificios, las tareas de comunicación a realizar varían en complejidad y a veces son incluso contradictorias. Por tanto, la utilización de una sola red de comunicación no daría resultados óptimos. Por ello, el mercado ofrece redes y sistemas de bus muy diferentes, más o menos adaptados a una aplicación específica.

Un criterio de clasificación bastante general es la distancia sobre la que la comunicación se establece. Existen redes locales, LAN (Local Area Networks), así como, así como redes ampliamente distribuidas, las WAN (Wide Area Networks)

En el caso de las LAN, se hace hincapié en un intercambio de datos rápido y potente dentro de un área restringida localmente, mientras que la WAN debe ser capaz de transmitir datos en diferentes soportes de datos y a lo largo de varios miles de kilómetros.

2.2.6.Protocolo de comunicación TCP/IP

El protocolo IP (Protocolo de Internet) proporciona mecanismos lógicos de interconexión entre redes de área local y TCP (Protocolo de Control de Transmisión) proporciona mecanismos de control de transmisión de datos entre los extremos de la comunicación.

El modelo TCP/IP le ayuda a determinar cómo debe conectarse un ordenador concreto a Internet y cómo deben transmitirse los datos entre ellos. El objetivo del modelo TCP/IP es permitir la comunicación a grandes distancias.

Las características esenciales del protocolo TCP/IP, pueden ser que brinda un soporte de arquitectura flexible, permite escalabilidad de sistemas, es un protocolo orientado a la conexión, ofrece fiabilidad y garantiza que los datos que llegan fuera de secuencia vuelvan a ponerse en orden, permitiendo implementar el control de flujo, por lo que el emisor nunca sobrecarga a un receptor con datos.

2.2.7. Topología de red

Las topologías de red describen los métodos en los que se mapean todos los elementos de una red. El término topología se refiere tanto a la disposición física como a la lógica de una red.

Esencialmente existen tres grandes topologías posibles, la de tipo estrella, en bus y en anillo.

La topología estrella radica en conectar cada estación de trabajo a un punto central, como se muestra en la Figura 9.

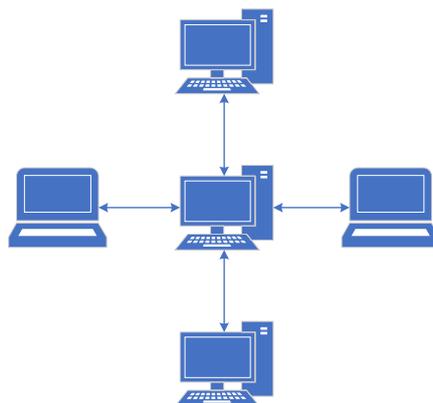


Figura 9. Topología estrella

Fuente: Elaboración propia

La topología en bus se basa en un medio cableado central que conecta todas las estaciones de trabajo, como se muestra en la Figura 10.

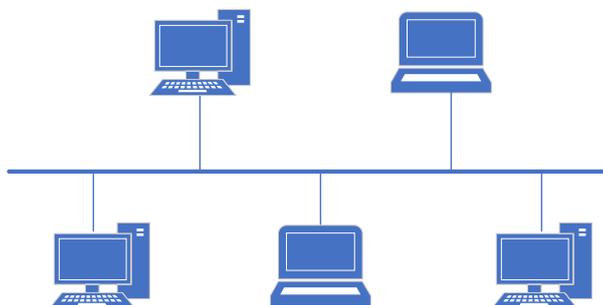


Figura 10. Topología en bus

Fuente: Elaboración propia

La topología en anillo resulta de conectar un grupo de estaciones de trabajo cada una a otras dos formando un anillo, como se muestra en la Figura 11, la comunicación por el anillo es unidireccional.

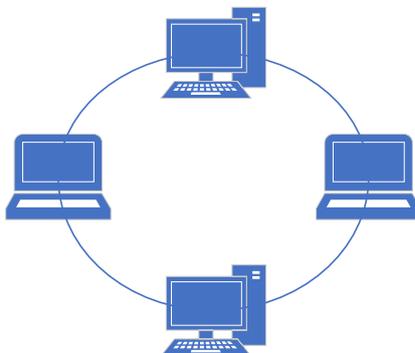


Figura 11. Topología en anillo

Fuente: Elaboración propia

2.3. Marco conceptual

Los sistemas de monitoreo y control utilizados en el presente trabajo están orientado puntualmente al proceso de monitoreo de temperatura y de control de acceso de personal

al área de custodia de bienes en tránsito de la Oficina de Soporte de la Oficina General de Tecnologías de la Información. En ese contexto se utiliza un conjunto de conceptos que se detalla a continuación:

El sistema de monitoreo utilizado, es el que permite obtener datos cuantitativos de un ambiente o área determinado, permitiendo medir la temperatura en tiempo real y respecto a ello tomar acción según la lógica operativa programada.

El sistema de control, es el que permite de acuerdo a los datos cuantitativos obtenidos realizar auditorías a fin de adoptar las medidas pertinentes, con el objetivo de mantener o preservar los Activos de T.I. que se encuentran en un ambiente o área determinada.

Los dispositivos de Internet de las cosas (IoT), permiten la obtención de datos cuantitativos que son procesados por una lógica de programación del sistema desarrollado para luego ser enviados a través de protocolos de comunicación como el TCP/IP hacia la internet, ello con el objetivo de ser supervisados de manera remota y en tiempo real por personal encargado, brindando un valor agregado a la adopción de medidas oportunas.

2.4. Marco Metodológico

2.4.1. Ciclo de vida del proyecto

Se procede a detallar las 5 etapas que comprende el ciclo de vida del proyecto realizado, como se puede apreciar en la figura 12.

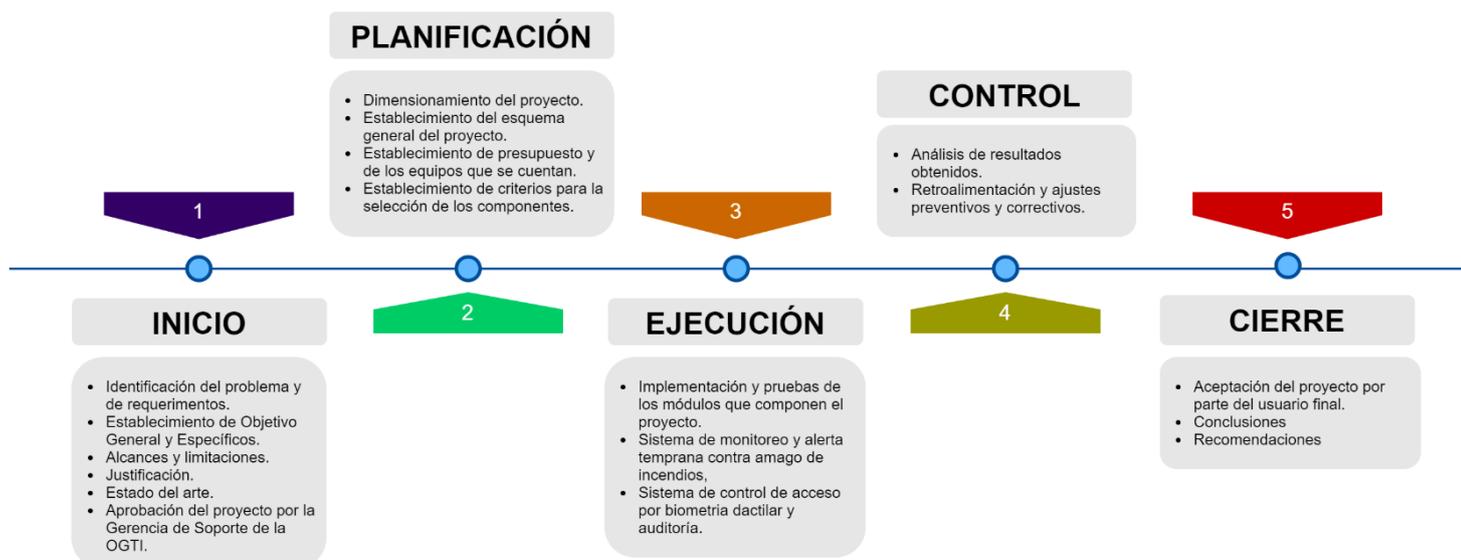


Figura 12. Ciclo de vida del proyecto

Fuente: Elaboración propia

Cabe mencionar que los siguientes capítulos 3 y 4 complementan el detalle de estas etapas, como resultado del análisis comparativo de la tecnología empleada para el proyecto de desarrollo de un sistema de monitoreo y control basado en IoT en apoyo a la Oficina de Soporte de la OGTI del MPFN.

2.4.2. Ciclo de vida del producto

Se procede a detallar el ciclo de vida del producto realizado, como se puede apreciar en la figura 13.

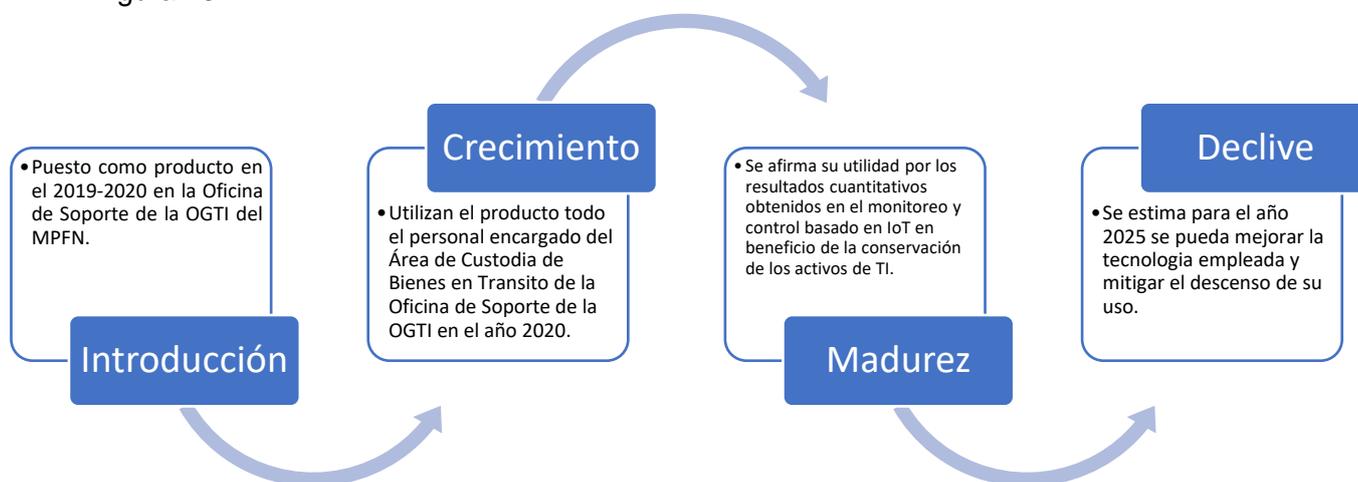


Figura 13. Ciclo de vida del producto

Fuente: Elaboración propia

Como se muestra en la figura 13 el producto se introduce formalmente a finales del 2019 y comienzos del 2020 para el área de custodia de bienes en tránsito de la Oficina de Soporte, por lo que la madurez del producto se sigue desarrollando a la actualidad, sin embargo, se estima debido a la generación de nuevas tecnologías que para el año 2025 pueda decaer el uso de la tecnología empleada debido a su vida útil, indicado por el fabricante.

A pesar de ello, se espera se pueda migrar la tecnología empleada para el 2025 y así extender la vida útil del producto y mitigar el descenso de su uso.

CAPITULO 3

DESARROLLO DE LA SOLUCIÓN

La propuesta de solución se basa en el desarrollo de un sistema de monitoreo y control basado en IoT, para el Área de Custodia de Bienes en Tránsito ubicada en la Oficina de Soporte de la OGTI del MPFN ubicada en el piso 10 de la Sede Principal del MPFN, Av. Abancay cdra. 5 S/N Cercado de Lima, el cual consta de una etapa de monitoreo principal que permite tener acceso desde la internet a la temperatura de trabajo y mediante alertas de correo electrónico informar sobre el aumento de temperatura del área en cuestión, luego una etapa de control de acceso de personal que permite tener la auditoria del registro de ingreso, los cuales son supervisados por la Gerencia de Soporte de la OGTI.

3.1. Criterio de selección de componentes

3.1.1. Selección de módulo de monitoreo y sensor de temperatura

El sistema de monitoreo es modular por lo en su conjunto debe ser un sistema altamente fiable, alta capacidad de procesamiento de datos, preciso, escalable, basado en IoT y de mínimo coste posible.

Para el caso del módulo de monitoreo, se evaluó alternativas como el Raspberry Pi 4 modelo B, Raspberry Pi 3 modelo B+ Orange Pi 3 siendo los más representativos de su categoría.

Por lo cual se realiza la Tabla 1 de comparativa con los aspectos más relevantes para el trabajo que se realiza.

Tabla 1. Comparativo de selección de módulo de monitoreo

| | Raspberry Pi 4 modelo B | Raspberry Pi 3 modelo B+ | Orange Pi 3 |
|------------------|--|--|---|
| Procesador | 1.5GHz Quad core Cortex-A72 (ARM v8) 64-bit | 1.4GHz 64-bit quad-core ARMv8 | Allwinner H6 SoC quad-core 64bit 1.8GHz |
| GPU | VideoCore VI OpenGL ES 3 | VideoCore IV | Mali T720 |
| Memoria | 1 GB , 2 GB, 4 GB LPDDR 4 | 1 GB DDR2 | 2 GB - 1 GB DDR3 |
| Conectividad | Ethernet y 2.4 GHz / 5.0 GHz IEEE 802.11b/g/n/ac | Ethernet y 2.4 GHz / 5.0 GHz IEEE 802.11b/g/n/ac | Ethernet y 2.4 GHz / 802.11b/g/n/ac |
| USB | 2x USB3.0 + 2x USB2.0 + USB-C OTG | 4x USB2.0 | 4x USB3.0 + 1x USB2.0 + micro OTG |
| HDMI | 2x micro HDMI | HDMI | HDMI 2.0a |
| Nivel de Soporte | Alto | Alto | Bajo |

Fuente: Elaboración propia

Por consiguiente, se selecciona el Raspberry Pi Modelo B para el módulo de monitoreo, ya que ofrece mayor capacidad de procesamiento, memoria, conectividad integrada y alto nivel de soporte para la implementación de las aplicaciones que ejecuta dicho módulo.

Para el caso del submódulo sensor de temperatura y humedad, se evaluó alternativas como el DHT11, DHT22 y LM35, como se aprecia en la Tabla 2.

Tabla 2. Tabla comparativa de sensor de temperatura

| | DHT22 | DHT11 | LM35 |
|----------------------|--------------|------------|--------------|
| Rango de temperatura | -40°C a 80°C | 0°C a 50°C | -10°C a 85°C |
| Precisión | ±0.5°C | ±2°C | ±0.5°C |
| Voltaje de operación | 3V a 6V DC | 3V a 5V DC | 4V a 30V DC |

| | | | |
|-------------------|---------|---------|---------------|
| Tipo de Salida | Digital | Digital | Analógica |
| Tiempo de sensado | 2 seg. | 1 seg. | 3 min. Aprox. |

Fuente: Elaboración propia

Por consiguiente, se selecciona el sensor DHT22 para medir la temperatura, puesto que ofrece mayor precisión, salida digital y tiempo de respuesta acorde a lo requerido.

3.1.2. Selección del módulo de control

Para el caso del módulo de control, se evaluó alternativas de hardware libre Arduino, por lo cual se realiza la Tabla 3 de comparativa con los aspectos más relevantes requeridos para el trabajo.

Tabla 3. Comparativo de selección de módulo de control

| | Arduino UNO | Arduino Mega 2560 | Arduino Micro |
|--------------------|------------------|-------------------|---------------|
| Procesamiento | ATmega328 | ATmega2560 | ATmega32U4 |
| Clock Speed | 16MHZ | 16MHZ | 16MHZ |
| Memoria | 32 KB | 256 KB | 32 KB |
| Entradas y salidas | 14 | 54 | 20 |
| Pines analógicos | 6 | 15 | 7 |
| Conectividad USB | Standard A/B USB | Standard A/B USB | Micro-USB |

Fuente: Elaboración propia

Por consiguiente, se selecciona el Arduino Mega 2560 para el módulo de control, puesto que ofrece mayor capacidad de procesamiento, memoria y entradas y salidas para la implementación de los controles que ejecuta dicho modulo. Para el submódulo de sensor biométrico, se evaluó alternativas de lector óptico de huellas dactilares, por lo cual se realiza la Tabla 4 de comparativa con los aspectos más relevantes requeridos para el trabajo.

Tabla 4. Comparativa de selección de submódulo de sensor biométrico

| | AS608 | DY50 | FPM10A |
|------------------------------------|------------|--------------|--------------|
| Voltaje de alimentación | 3.3v DC | 3.6v - 6v DC | 3.6v - 6v DC |
| Corriente de operación | < 60 mA | < 120 mA | < 120 mA |
| Interfaz | USB / UART | USB / TTL | UART / TTL |
| Baud rate | 57600 | 57600 | 57600 |
| Cantidad de huellas de almacenadas | 150 | 200 | 300 |
| Tiempo de adquisición | < 1 s | < 1 s | < 1 s |

Fuente: Elaboración propia

Por consiguiente, se selecciona el sensor óptico de huellas dactilares DY50 como submódulo de sensor biométrico, puesto que ofrece la capacidad de necesaria de almacenamiento de huellas, interfaz, voltaje y corriente de operación para la implementación de los controles de medición que ejecuta dicho módulo.

3.2. Diagrama de bloques

Se presenta los componentes de la solución a través del diagrama de bloque de la Figura 14, el cual permite comprender la relación existente entre los sistemas implementados.

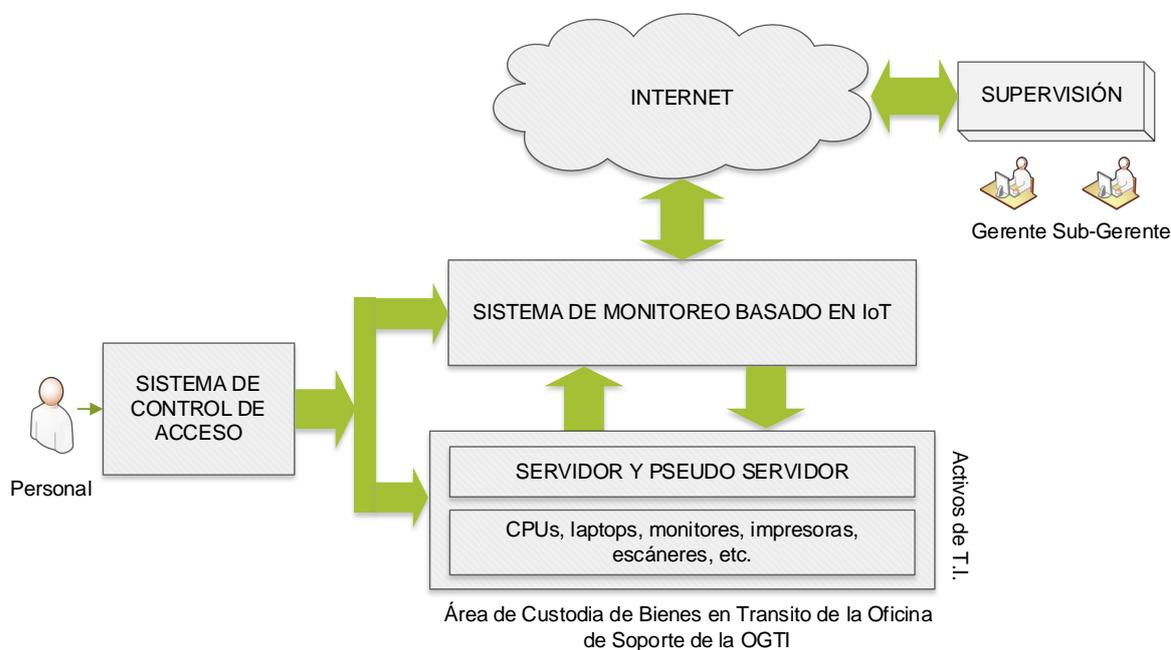


Figura 14. Diagrama en bloques.

Fuente: Elaboración propia

Como se puede apreciar en la Figura 10 el sistema de monitoreo basado en IoT gestiona la conexión hacia la internet, los sensores y actuadores utilizando el Raspberry Pi 4 modelo B y el bloque de control de acceso controla un sensor de biometría dactilar y el registro de acceso de personal utilizando un Arduino Mega 2560. Dichos sistemas brindan el resguardo de los Activos de T.I. en custodia por la Oficina de Soporte de la OGTI.

3.2.1. Diseño del sistema de monitoreo

El diseño del sistema de monitoreo posee un módulo principal de monitoreo, un sub módulo de sensor de temperatura, un actuador, periféricos y la salida de conexión hacia la internet, como se muestra en el diagrama en bloque de la Figura 15.

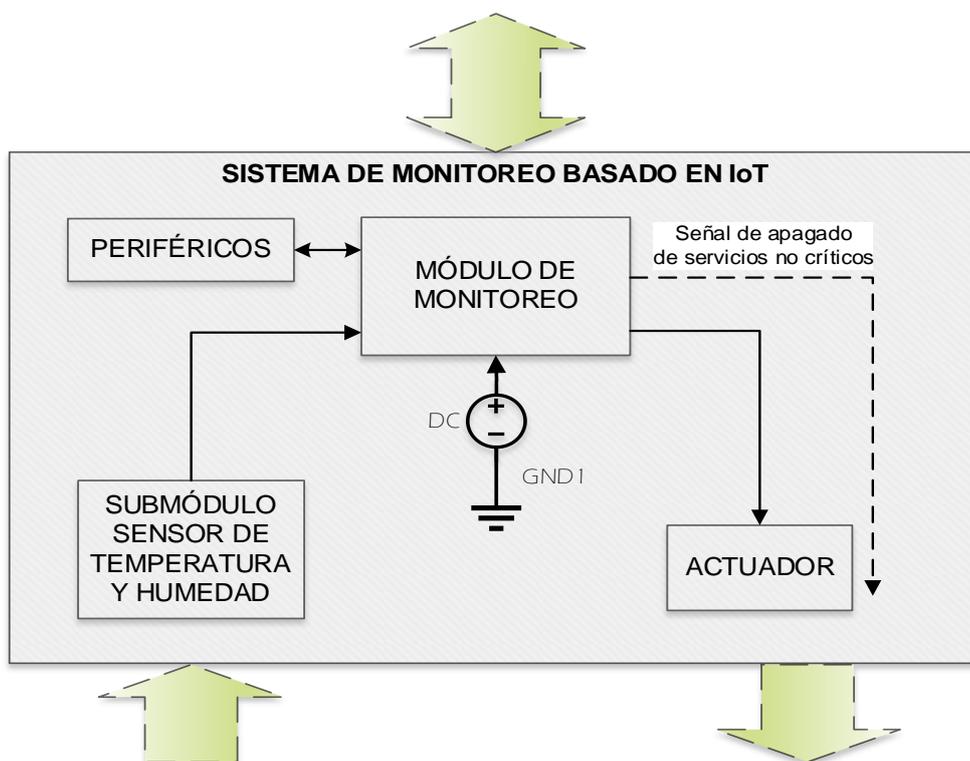


Figura 15. Sistema de monitoreo basado en IoT.

Fuente: Elaboración propia

3.2.2. Identificación de las entradas y salidas del sistema de monitoreo

Para el sistema de monitoreo se identifica las entradas:

- Señal del submódulo de sensor de temperatura y humedad
- Señal de periféricos de programación.
- Señal de voltaje de alimentación.

Para las salidas se identifican:

- Señal de envío de paquetes TCP/IP hacia la internet para la supervisión.
- Señal de periféricos de programación.
- Señal de voltaje para actuador, el cual permite la activación de equipos de refrigeración.
- Señal de apagado de servicios no críticos, mediante ejecución de script remoto en el pseudo servidor.

3.2.3. Diseño del sistema de control de acceso

El diseño del sistema de control posee un módulo de control, un sub módulo de sensor biométrico de huella dactilar, un submódulo de reloj de tiempo real, un submódulo de almacenamiento y un actuador que permiten el control y auditoria de acceso al área de custodia de bienes en tránsito de la Oficina de Soporte de la OGTI, como se muestra en el diagrama en bloque de la Figura 16.

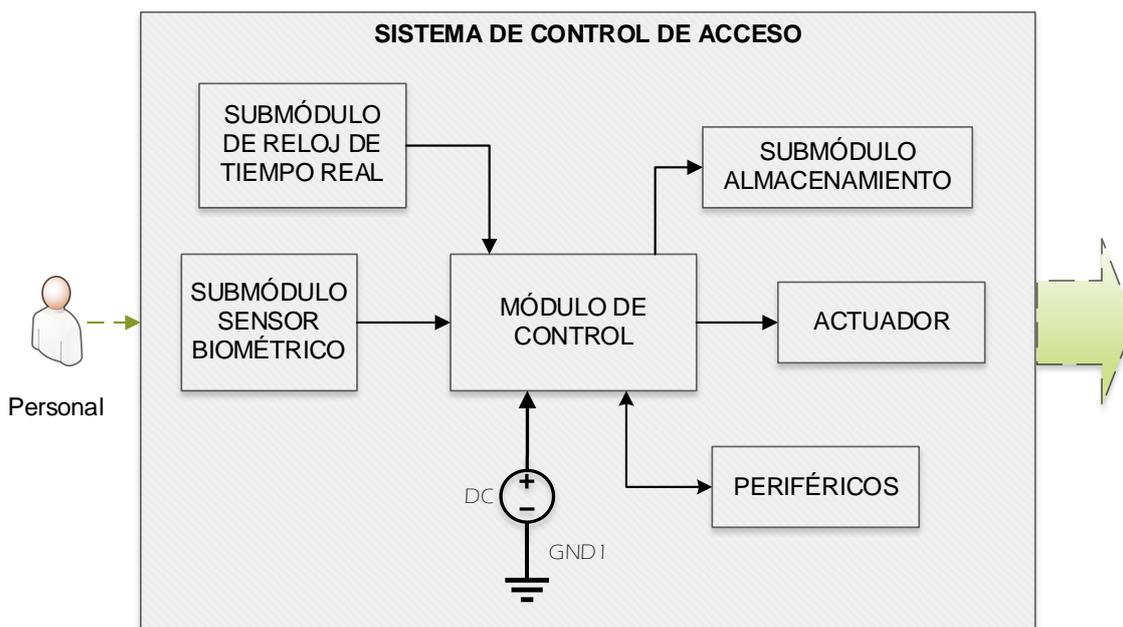


Figura 16. Sistema de control de acceso.

Fuente: Elaboración propia

3.2.4. Identificación de las entradas y salidas del sistema de control de acceso

Para el sistema de control se identifica las entradas:

- Señal de submódulo de sensor biométrico de huella dactilar
- Señal del submódulo de reloj de tiempo real
- Señal de periféricos de programación.

Para las salidas se identifican:

- Señal para activación del actuador, que permite el acceso a personal autorizado.
- Señal de periféricos de programación.
- Señal de registro de personal en submódulo de almacenamiento.

3.3. Condiciones del Soporte Eléctrico local

El Área de Custodia de Bienes en Tránsito de la Oficina de Soporte de la OGTI del MPFN, posee una línea estabilizada de 220VAC a 60 Hz que es administrada por competencia operativa por el Área de Mantenimiento del MPFN.

Sin embargo, la Gerencia de la Oficina de Soporte asigna al proyecto de Sistema de monitoreo y control basado en IoT un Sistema de Alimentación Ininterrumpida (UPS) Interactivo del inventario de activos de TI de marca Centralion modelo Blazer 600, a fin de proporcionar protección y suministro eléctrico ante fallo de red eléctrica al proyecto en cuestión, cabe mencionar que dicho equipo no se considera en el presupuesto del proyecto por ser un activo de TI preexistente.

A continuación, se describe las principales características del UPS Interactivo en la Tabla 5.

Tabla 5. Características del UPS Interactivo Centralion Blazer 600

| | |
|--|--|
| Marca | Centralion |
| Modelo | Blazer 600 |
| Capacidad | 600 VA / 360W |
| Entrada | 220 VAC |
| Salida | 220 VAC / 60Hz |
| Salida Forma de Onda | Senoidal modificada |
| Voltaje de Regulación (Modo Batería) | ± 10% |
| Batería | 12V/7AH |
| Tiempo de respaldo (en una carga de PC con monitor 15") | 15 - 20 min |
| Tiempo de transferencia | 2-6ms |
| Tiempo de carga | De 8 horas a 90% después de la descarga completa |
| Protección eléctrica | Caídas de tensión, sobretensiones, ruido eléctrico y fallo de suministro eléctrico |
| Dimensiones | 330 x 100 x 140 mm |
| Entorno de trabajo | 0-40°C - 0-90% humedad relativa |
| Nivel de ruido | < 40dB |

Fuente: Elaboración propia

Adicionalmente, se presenta el diagrama de bloques del UPS Interactivo en la Figura 17.

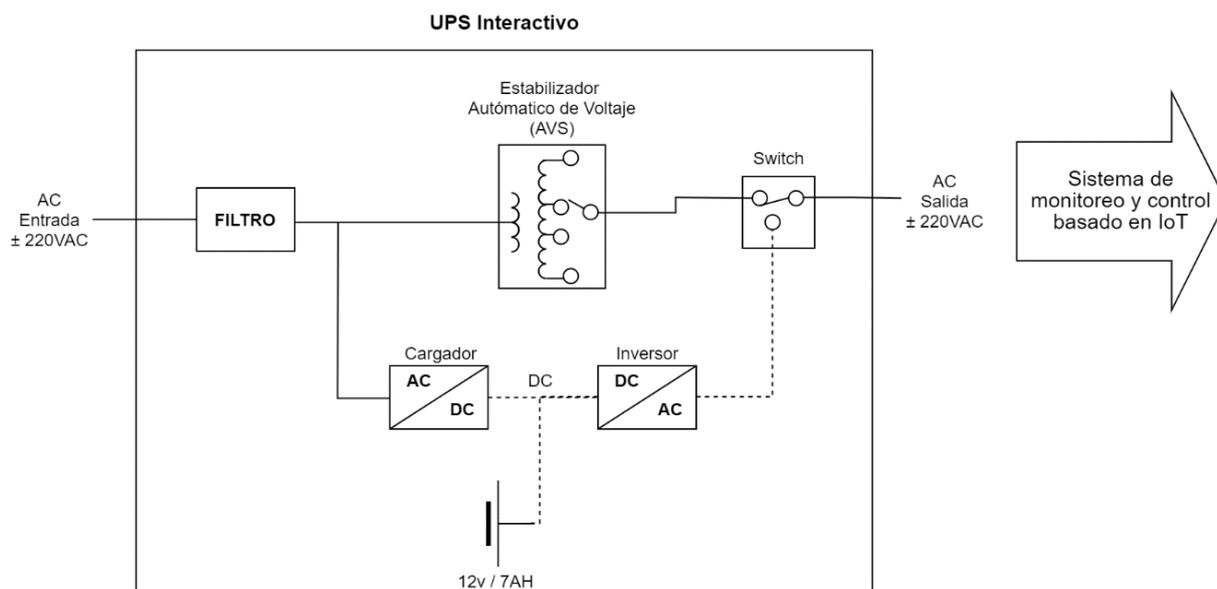


Figura 17. Diagrama de bloque UPS Interactivo Centralion Blazer 600

Fuente: Elaboración propia

3.3.1. Actuador para accionamiento de equipos de enfriamiento local

Los equipos de enfriamiento son equipos de aire acondicionado que cuenta el área de custodia de bienes en tránsito de la Oficina de Soporte de la OGTI del MPFN, estos equipos tienen características generales como se muestra en la Tabla 6.

Tabla 6. Descripción básica de equipos de aire acondicionado

| | Equipo aire acondicionado 1 | Equipo aire acondicionado 2 |
|-----------|-----------------------------|-----------------------------|
| Capacidad | 24000 Btu/h | 36000 Btu/h |
| Corriente | 8.4A | 10A |

Fuente: Elaboración propia

Estos equipos se encuentran a cargo del Área de mantenimiento del MPFN, por ser de su competencia operativa, dichos equipos cuentan con actuadores para su accionamiento los cuales se describe a la tabla 7.

Tabla 7. Descripción básica de equipos de aire acondicionado

| | |
|--------------|--|
| Marca | Songle |
| Modelo | SLA-05VDC-SL-C |
| Descripción | Hasta 30A de conmutación Optocoupler driver relé Módulo, voltaje de bobina 5vDC Montado sobre modulo FC-65 con las protecciones eléctricas para su funcionamiento. |
| Aplicaciones | Se utiliza para la conmutación de potencia, calentador eléctrico, ventilador, aire acondicionado, Refrigeración, automóvil y electrodomésticos. |

Fuente: Elaboración propia

Así mismo se muestra el diagrama de bloque el cual permite apreciar la interacción con el Sistema desarrollado y los equipos de enfriamiento en la figura 18.

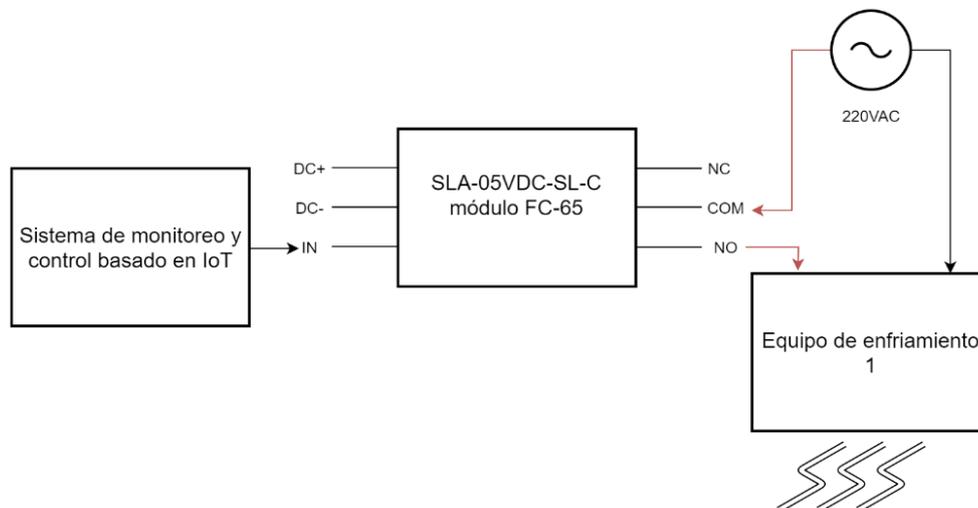


Figura 18. Diagrama de bloque del actuador de enfriamiento

Fuente: Elaboración propia

Cabe mencionar que dicho actuador no se considera en el presupuesto del proyecto por ser un activo de preexistente.

3.4. Proceso de implementación

3.4.1. Implementación del Sistema de Monitoreo

De acuerdo a la selección realizada se procede a detallar las actividades de implementación del sistema de monitoreo, en la figura 19.

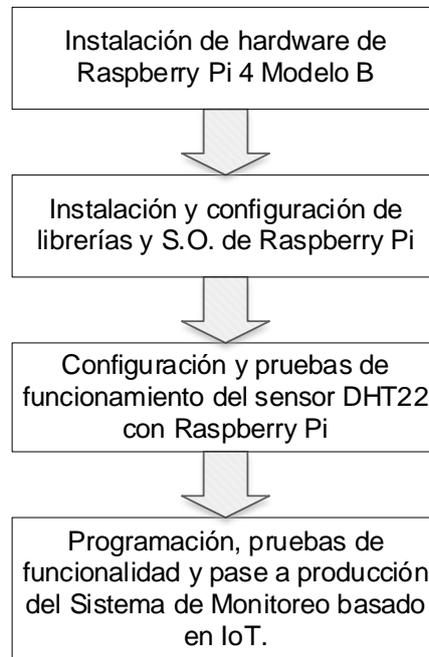


Figura 19. Diagrama de bloques de implementación del sistema de monitoreo

Fuente: Elaboración propia.

- La primera etapa, se instala los disipadores de calor en los circuitos integrados del Raspberry Pi 4, así como la carcasa que incorpora un ventilador, brindando el aislamiento, protección y disipación de calor para un rendimiento óptimo.
- La segunda etapa, se descarga el S.O. del Raspberry Pi desde el sitio web del fabricante en una tarjeta microSD de 32 GB, luego se instala dicha tarjeta en el módulo de Raspberry Pi y se procede a actualizar y configurar el S.O. así como las librerías adecuadas que permiten el funcionamiento del Sistema de Monitoreo.
- La tercera etapa, se configura las librerías y se realiza pruebas de funcionamiento del sensor DHT22 con Raspberry pi.

- d) La cuarta etapa, se desarrolla la lógica de programación del módulo de monitoreo, como se detalla en la figura 20, las pruebas de funcionalidad y finalmente pasar al entorno de producción el Sistema de Monitoreo basado en IoT.

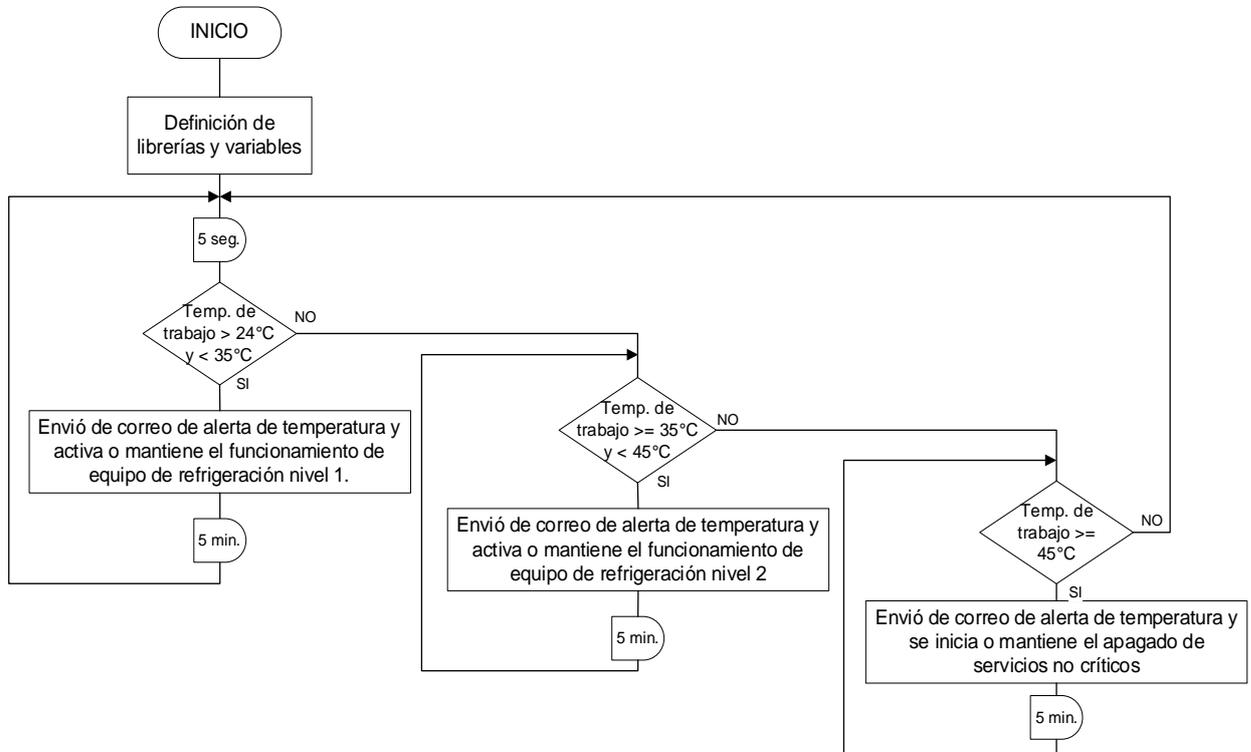


Figura 20. Diagrama de Flujo de monitoreo

Fuente: Elaboración propia.

El código de la programación se encuentra como Anexo.

3.4.2. Implementación del Sistema de Control de Acceso

De acuerdo a la selección realizada se procede a detallar las actividades de implementación del sistema de control, en la figura 21.

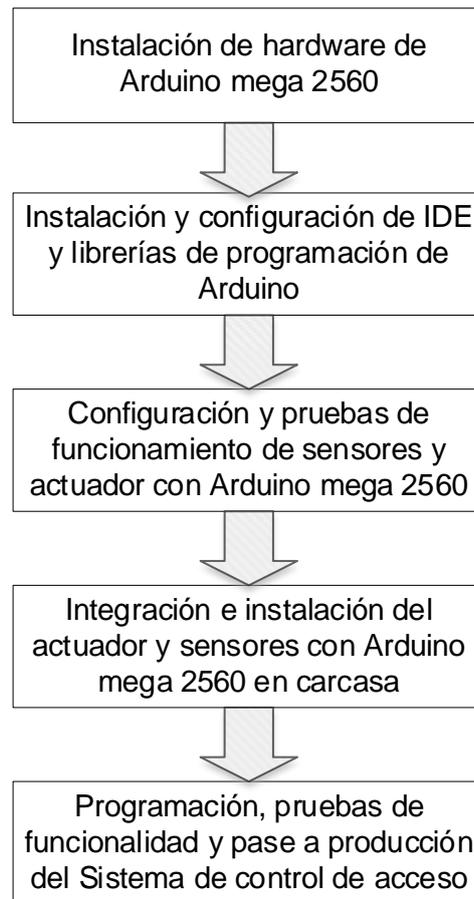


Figura 21. Diagrama de bloques de implementación del sistema de control de acceso

Fuente: Elaboración propia.

- a) La primera etapa, se instala la carcasa de Arduino mega 2560 así como se realiza las pruebas de funcionamiento para un rendimiento óptimo.
- b) La segunda etapa, se descarga y configura IDE de Arduino desde el sitio web del fabricante en computadora personal, luego se instalan las librerías que requieren los submódulos, de modo que, permitan la marcha del sistema de control de acceso.
- c) La tercera etapa, se configura las librerías y se realiza pruebas de funcionamiento de los sensores y del actuador con el Arduino mega 2560.
- d) La cuarta etapa, se realiza la integración e instalación de los sensores en carcasa diseñada especialmente para alojar los componentes modulares del Sistema de Control de acceso.

- e) La quinta etapa, se desarrolla la lógica de programación del módulo de control de acceso, como se detalla en la Figura 22, las pruebas de funcionalidad y finalmente pasar al entorno de producción el Sistema de Control de acceso.

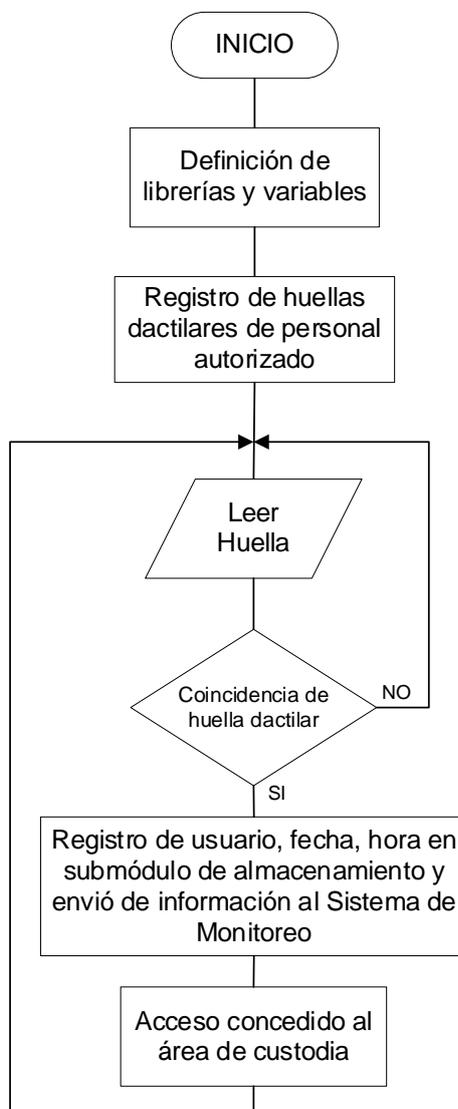


Figura 22. Diagrama de flujo de control

Fuente: Elaboración propia

El código de la programación se encuentra como Anexo.

CAPITULO 4

RESULTADOS

4.1. Análisis de resultados obtenidos

4.1.1. Monitoreo al Área de Custodia de Bienes en tránsito de la Oficina de Soporte de la OGTI

Se adquirió el registro de temperatura de la base de datos de Sistema de monitoreo que se grafica en la Figura 23.

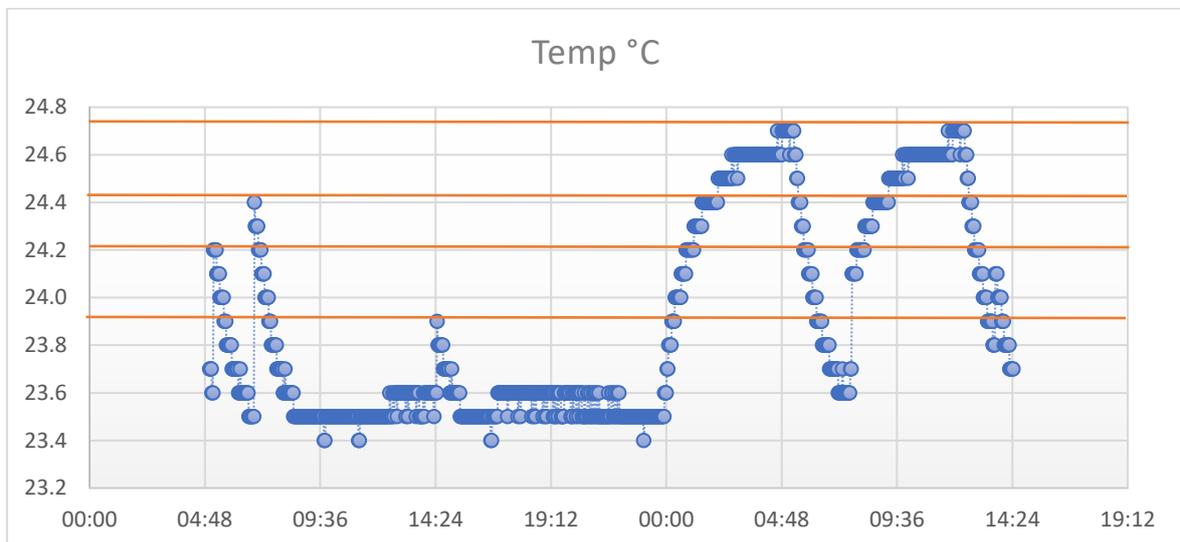


Figura 23. Gráfica de Temperatura (°C) contra el tiempo (formato 24h) de los días 2021-03-05 y 2021-03-06

Fuente: Elaboración propia

Se puede apreciar en la Figura 23 que la temperatura aproximadamente no logra superar el valor de alerta de 25 °C del Área de Custodia de Bienes en Tránsito de la Oficina de Soporte de la OGTI, por lo tanto, el Sistema de Monitoreo logra con éxito mantener la temperatura ideal menor a los 25 °C programados en el módulo de monitoreo Raspberry Pi.

En el caso que la temperatura se aproxime a 25 °C el sistema enviara alertas tempranas contra amago de incendio, a través de correo electrónico a la Gerencia de Soporte de la OGTI, la cual es la encargada de supervisar el área de custodia, como se puede apreciar en la Figura 24.

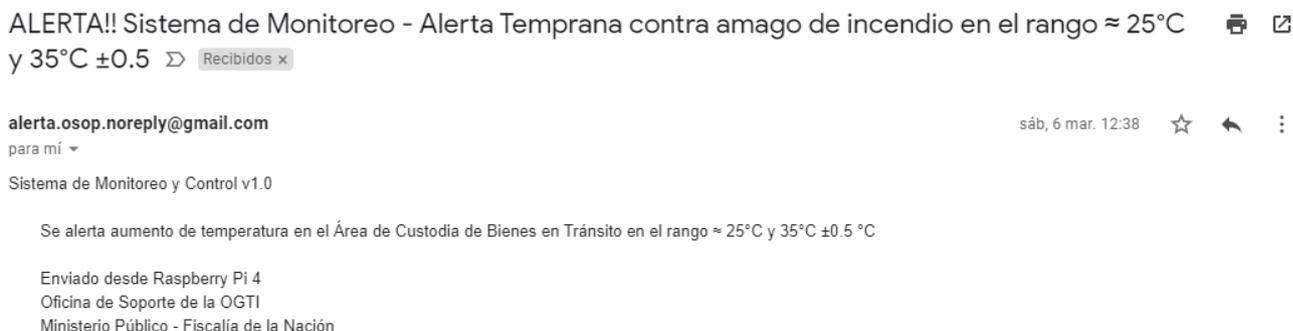


Figura 24. Correo electrónico de alerta temprana contra amago de incendio

Fuente: Elaboración propia

Adicionalmente, a través de la plataforma ThigSpeak, la cual permite almacenar y recuperar datos de dispositivos IoT, informa de manera gráfica a la supervisión de la Oficina de Soporte de la OGTI el estado de la temperatura en tiempo real y la ubicación georreferenciada, como se muestra en la Figura 25.

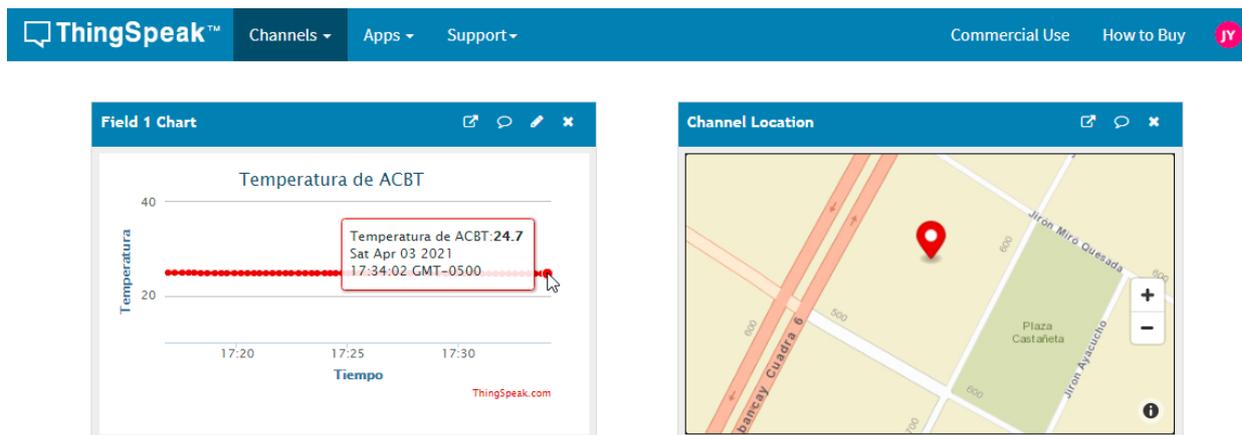


Figura 25. Plataforma ThingSpeak utilizado por la supervisión de la Oficina de Soporte
Fuente: Elaboración propia

Además, dicha plataforma posee herramientas que permiten presentar la información de temperatura de manera más dinámica, como se muestra en la Figura 23.



Figura 26. Visualización dinámica de temperatura en plataforma Thingspeak.
Fuente: Elaboración propia

4.1.2. Control de acceso al Área de Custodia de Bienes en tránsito de la Oficina de Soporte de la OGTI

Se adquirió los registros de control de acceso del Sistema de Control que se grafica en la Figura 27.

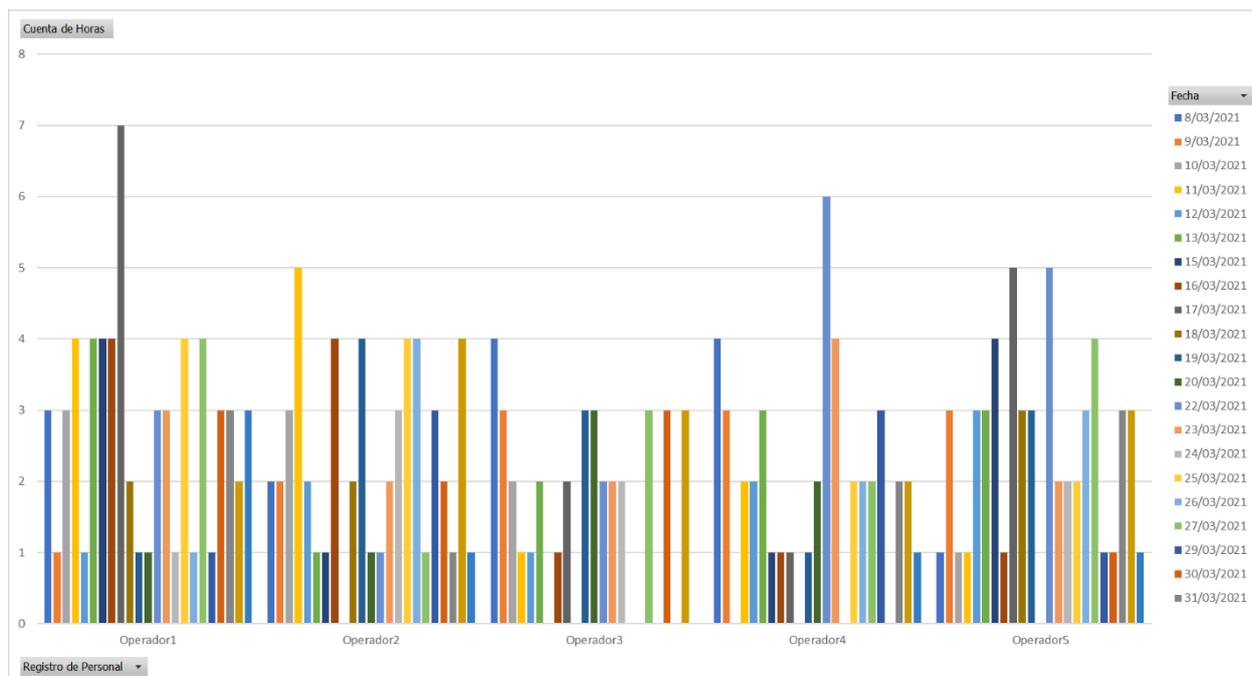


Figura 27. Gráfica de registro de ingreso contra las horas de ingreso del mes de marzo 2021.

Fuente: Elaboración propia

Se puede apreciar en la Figura 27 que el Sistema de Control de Acceso logra con éxito registrar el ingreso del personal autorizado para auditoria, validando su identidad mediante la biometría dactilar.

De igual manera se presenta los registros del Sistema de Control de acceso en Anexo 6.

4.2. Presupuesto

Se detalla los costos aproximados del proyecto implementado en la Tabla 8.

Tabla 8. Presupuesto del Sistema de Monitoreo y Control basado en IoT

| Componentes electrónicos | Cantidad | Precio Unitario | Costo |
|--------------------------|----------|-----------------|----------|
| Raspberry Pi 4 Modelo B | 1 | S/225.00 | S/225.00 |
| Arduino mega 2560 | 1 | S/70.00 | S/70.00 |
| Carcasa Raspberry pi | 1 | S/30.00 | S/30.00 |

| | | | |
|---------------------------------------|---|----------|-----------------|
| Carcasa Arduino mega | 1 | S/18.00 | S/18.00 |
| Modulo RTC | 1 | S/12.00 | S/12.00 |
| Lector óptico de huella dactilar Y50 | 1 | S/120.00 | S/120.00 |
| Sensor de temperatura y humedad DHT22 | 1 | S/18.00 | S/18.00 |
| Motor paso a paso sg90 | 1 | S/9.00 | S/9.00 |
| Fuente de alimentación Raspberry pi | 1 | S/15.00 | S/15.00 |
| Fuente de alimentación Arduino mega | 1 | S/12.00 | S/12.00 |
| Memoria Micro SD | 2 | S/36.00 | S/72.00 |
| Modulo lector de tarjetas micro SD | 1 | S/12.00 | S/12.00 |
| Modulo relé | 1 | S/8.00 | S/8.00 |
| TOTAL GENERAL | | | S/621.00 |

CONCLUSIONES

- 1) El Sistema de Monitoreo y control de temperatura desarrollado permite de manera automatizada corregir el aumento de temperatura, así como evitar el sobrecalentamiento de los servidores y pseudo servidores en el Área de Custodia de Bienes en Transito de la Oficina de Soporte de la OGTI.
- 2) El sistema de Control de acceso desarrollado permite mediante biométrica dactilar registrar el ingreso de personal autorizado, el cual permite conservar y salvaguardar los activos de T.I del área de custodia de la Oficina de Soporte de la OGTI.
- 3) El Sistema de Control implementado permite la auditoria de ingreso de personal autorizado, permitiendo establecer responsabilidades ante incidencias con los Activos de T.I.
- 4) El Sistema de Monitoreo cuenta con un sistema de notificaciones de alerta temprana contra amagos de incendio basado en IoT al identificar un aumento de temperatura mayor a 25 °C.

RECOMENDACIONES

- a) El empleo de una red de sensores inalámbricos basados en IoT para un muestreo más preciso del ambiente de trabajo.
- b) El empleo de otras tecnologías de identificación de la identidad, como facial, ocular y palmar.
- c) La utilización de cámara web para transmitir en tiempo real las imágenes del ambiente a monitorear.
- d) Para futuros trabajos relacionados se puede utilizar la conexión hacia la internet por medio de tecnología 4G o 5G.
- e) La mejora del soporte eléctrico en el piso 10 de la Sede Principal del MPFN, a fin de obtener un Sistema más robusto.
- f) El desarrollo de una plataforma de supervisión remota propia basado en IoT y pueda ser compartido con otras entidades públicas.

BIBLIOGRAFÍA

- A. Roihan, F. Sudarto, & T. Cahyo Putro. (2018). Internet de las cosas en el sistema de supervisión y control en el área de servidores. Semarang, Indonesia.
- Barrio Andrés, M. (2018). *INTERNET DE LAS COSAS*. Madrid.
- Bravo Guillén, G. L. (2016). Diseño y desarrollo de un sistema electrónico para monitoreo remoto del estado y control de velocidad de un montacargas. (*Tesis de grado*). Universidad Tecnológica del Perú, Lima, Perú.
- C. Chung, I. Peng, & J. Yeh. (2018). Sistema inteligente de clasificación y vigilancia del entorno combinado con la tecnología de transmisión inalámbrica LoRa. Taichung, Taiwan.
- Choudhury, Z. (2019). *Configuración de la conectividad inalámbrica y los límites espaciales para los dispositivos de monitorización del IoT*. MASSACHUSETTS INSTITUTE OF TECHNOLOGY, Massachusetts, USA.
- Corona Ramírez, L. G., Abarca Jiménez, G. S., & Mares Carreño, J. (2014). *Sensores y actuadores*. México.
- D. Li, Z. Bao, & Y. Yang. (2016). Diseño de un sistema de monitorización del entorno del taller basado en el Internet de las cosas. Hangzhou, China.
- D. Pujara, P. Kukreja, & S. Gajjar. (2020). Diseño y desarrollo de E-Sense: Sistema de monitorización del entorno basado en IoT. Prayagraj, India.
- G. Parmar, S. Lakhani, & M. K. Chattopadhyay. (2017). Un sistema de vigilancia de la contaminación atmosférica de bajo coste basado en el IoT. Bhopal, India.
- H. Barksdale, Q. Smith, & M. Khan. (2018). Monitorización del estado de las máquinas eléctricas con el Internet de las cosas. St. Petersburg, USA.
- Huiman Tocto, N. E. (2017). *DISEÑO E IMPLEMENTACIÓN DE UNA RED DE MEDIDORES DE ENERGÍA PARA ARTEFACTOS DOMÉSTICOS*. (*Tesis de grado*). PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ, Lima, Perú.
- Lima Guaqueta, J. E., & Ospina Fuentes, P. L. (2018). *DISEÑO DE UN SISTEMA DE ADQUISICIÓN Y VISUALIZACIÓN DE DATOS BASADO EN LA PLATAFORMA DE*

SISTEMAS EMBEBIDOS RASPBERRY PI. (*Tesis de grado*). UNIVERSIDAD SANTO TOMAS, Bogota, Colombia.

- Mora Magallanes, H. V. (2019). DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UNA RED DE SENSORES INALÁMBRICOS (WSN) PARA EL CONTROL, MONITOREO Y TOMA DE DECISIONES APLICADO EN LA AGRICULTURA DE PRECISIÓN BASADO EN INTERNET DE LAS COSAS (IoT). – CASO DE ESTUDIO CULTIVO DE FRIJOL. (*Tesis de grado*). UNIVERSIDAD RICARDO PALMA, Lima, Perú.
- Orellana González, D. A. (2015). IMPLEMENTACIÓN Y PRUEBAS DEL ESTÁNDAR DE COMUNICACIÓN DE SERVICIOS DE DISTRIBUCIÓN DE DATOS PARA APLICACIONES DE MICRO-REDES GESTIONADOS POR COMPUTADORES DE PLACA ÚNICA. (*Tesis de Grado*). UNIVERSIDAD DE CHILE, Santiago de Chile, Chile.
- Sánchez Capistrano, J. B. (2019). Desarrollo de un sistema de seguridad electrónica aplicado a la supervisión y monitoreo en oficinas. (*Tesis de grado*). Universidad Tecnológica del Perú, Lima, Perú.
- V. Hoang, V. Dao, & V. Dang. (2020). Sensor de temperatura para sistemas de vigilancia inteligentes de largo alcance y con eficiencia energética. Hanoi, Vietnam.
- Valqui Cieza, R. (2019). Diseño e implementación de un sistema electrónico de monitoreo de gases tóxicos para minería subterránea con acceso remoto a través de web server. (*Tesis de grado*). Universidad Tecnológica del Perú, Lima, Perú.
- Vizcarra Cavero, S. I. (2019). Diseño e implementación de un sistema monitoreo inalámbrico de bancos de baterías utilizando en Arduino Mega 2560. (*Tesis de grado*). Universidad Tecnológica del Perú, Lima.
- Y. Cheng, X. Xu, Y. Du, P. Guan, S. Liu, & L. Zhao. (2019). Diseño de un sistema de monitorización de la calidad del aire basado en NB-IoT. Shenyang, China.
- Y. Liu, J. Leng, & B. Sun. (2017). Diseño de un sistema de monitorización de la energía basado en WebAccess. Chongqing, China.
- Z. Yu-Long, C. Heng, H. Jing-Feng, & W. Jin-Cheng. (2018). Diseño e implementación de un sistema de monitorización remota de máquinas de soldar basado en la web. Xi'an, China.

ANEXOS

ANEXO 1: Glosario de Términos

T.I.: Tecnologías de la Información

IoT: Internet de las Cosas.

PLC: Controlador Lógico Programable.

NB-IoT: Red del Internet de las Cosas

Data Distribution Service (DDS): Estándar de Servicio de distribución de datos.

Single-Board Computer (SBC): Computador de placa única o reducida.

ANEXO 2: Código de módulo de monitoreo

```

import smtplib
from email.mime.text import MIMEText
import time
import datetime
import board
import adafruit_dht
import requests
from board import *
email_1 = 'alerta.osop.noreply@gmail.com'
pwd = 'QWErty21*'
email_2 = 'jyalle@mpfn.gob.pe'

time1 = datetime.datetime.now()
time2 = datetime.datetime.now()
time3 = datetime.datetime.now()
cincomin = datetime.time (0, 5, 0)
cerotiempo = datetime.time (0, 0, 0)

body = """Sistema de Monitoreo y Control v1.0

        Se alerta aumento de temperatura en el Área de Custodia de Bienes en Tránsito en
        el rango  $\approx 25^{\circ}\text{C}$  y  $35^{\circ}\text{C} \pm 0.5^{\circ}\text{C}$ 

        Enviado desde Raspberry Pi 4
        Oficina de Soporte de la OGTI
        Ministerio Público - Fiscalía de la Nación
        """
body1 = """Sistema de Monitoreo y Control v1.0

```

Se alerta aumento de temperatura en el Área de Custodia de Bienes en Tránsito en el rango $\approx 35^{\circ}\text{C}$ y $45^{\circ}\text{C} \pm 0.5^{\circ}\text{C}$

Enviado desde Raspberry Pi 4
 Oficina de Soporte de la OGTI
 Ministerio Público - Fiscalía de la Nación
 ""

body2 = ""Sistema de Monitoreo y Control v1.0

Se alerta aumento de temperatura en el Área de Custodia de Bienes en Tránsito en el rango $>\approx 45^{\circ}\text{C} \pm 0.5^{\circ}\text{C}$

Enviado desde Raspberry Pi 4
 Oficina de Soporte de la OGTI
 Ministerio Público - Fiscalía de la Nación
 ""

SENSOR_PIN = D4

dhtDevice = adafruit_dht.DHT22(SENSOR_PIN, use_pulseio=False)

while True:

try:

temperatura_c = dhtDevice.temperature

temperature_f = temperatura_c * (9 / 5) + 32

humidity = dhtDevice.humidity

archivo=open("Registro_monitoreo_MPFN.txt",'a')

archivo.write(str(temperatura_c) + ', ' + str(time.strftime("%H:%M:%S"))+ ', ' + str(datetime.date.today()) + '\n')

archivo.close()

if (temperatura_c in range(25, 35)):

tiempo_actual = datetime.datetime.now()

def inicio_refrigeracion_nivel1(1)

```

if tiempo_actual - time1 >= cincomin or tiempo_actual - time1 == cerotiempo:
    time1 = tiempo_actual
    msj = MIMEText(body,'plain')
    msj['Subject'] = 'ALERTA!! Sistema de Monitoreo - Alerta Temprana contra
amago de incendio en el rango  $\approx 25^{\circ}\text{C}$  y  $35^{\circ}\text{C} \pm 0.5^{\circ}\text{C}$ '
    msj['From'] = email_1
    msj['To'] = email_2
    server = smtplib.SMTP('smtp.gmail.com',587)
    server.starttls()
    server.login(email_1,pwd)
    server.sendmail(email_1,email_2,msj.as_string())
    print ("Se envio alerta")
    server.quit()
elif (temperatura_c in range(35, 45)):
    def inicio_refrigeracion_nivel2(2)
    tiempo_actual = datetime.datetime.now()
    if tiempo_actual - time2 >= cincomin or tiempo_actual - time2 == cerotiempo:
        time2 = tiempo_actual
        msj = MIMEText(body1,'plain')
        msj['Subject'] = 'ALERTA!! Sistema de Monitoreo - Alerta Temprana contra
amago de incendio en el rango  $\approx 35^{\circ}\text{C}$  y  $45^{\circ}\text{C} \pm 0.5^{\circ}\text{C}$ '
        msj['From'] = email_1
        msj['To'] = email_2
        server = smtplib.SMTP('smtp.gmail.com',587)
        server.starttls()
        server.login(email_1,pwd)
        server.sendmail(email_1,email_2,msj.as_string())
        print ("Se envio alerta")
        server.quit()
elif (temperatura_c >= 45):
    tiempo_actual = datetime.datetime.now()
    def inicio_apagado_servicios_nocriticos(1)

```

```

if tiempo_actual - time3 >= cincomin or tiempo_actual - time3 == cerotiempo:
    time3 = tiempo_actual
    msj = MIMEText(body2,'plain')
    msj['Subject'] = 'ALERTA!! Sistema de Monitoreo - Alerta Temprana contra
amago de incendio ≈ 45°C ±0.5'
    msj['From'] = email_1
    msj['To'] = email_2
    server = smtplib.SMTP('smtp.gmail.com',587)
    server.starttls()
    server.login(email_1,pwd)
    server.sendmail(email_1,email_2,msj.as_string())
    print ("Se envio alerta")
    server.quit()

print(
    "Temp: {:.1f} F / {:.1f} C  Humidity: {}% ".format(
        temperature_f, temperatura_c, humidity
    )
)
)
subir =
requests.get("https://api.thingspeak.com/update?api_key=H6QTR99S3TEHTPAC&field1=
" + str(temperatura_c) + "&field2=" + str(humidity))
time.sleep(300)

except RuntimeError as error:
    print(error.args[0])
    time.sleep(3.0)
    continue
except Exception as error:
    dhtDevice.exit()
    raise error

time.sleep(5.0)

```

ANEXO 3: Código de módulo de control

```
//Librería para comunicación serie
#include <SoftwareSerial.h>
//Librería para control de LCD
#include <LiquidCrystal_I2C.h>
//Librería para comunicación por bus I2C
#include <Wire.h>
//Librería para control de servomotor
#include <Servo.h>
//Librería para control de lector de huella dactilar
#include <Adafruit_Fingerprint.h>
//Librería para control de SD
#include <SD.h>
//Librería para interaccionar con reloj de tiempo real
#include <DS3231.h>
//Librería para utilizar protocolo de transferencia de datos serie sincronos
#include <SPI.h>

int obthuellavent();
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
File auditoria_1;
LiquidCrystal_I2C lcd(0x27,16,2);
Servo servosop;
SoftwareSerial mySerial(10, 11);

int escaneo_entrada = 4;
int agrega_ide_entrada = 12;
int cierre = 9;
int luz_verde = 8;
int luz_roja = 7;
```

```
int servosop1 = 6;
DS3231 rtc(SDA, SCL);

int usuario_principal = 1;
int puerta_abre = 180;
int puerta_cierra = 0;
String archivo_aud_1 = "auditoria.txt";

bool escaneando_1 = true;
int contador_0 = 0;
bool ide_sum = false;
bool primera_lectura = false;
bool usuario_prin = false;
bool sum_nuevo_ide = false;
bool puerta_bloq = true;

void setup() {
  Serial.begin(57600);
  rtc.begin();
  SD.begin(53);

  //Se comienza a grabar en archivo de auditoria de acceso
  auditoria_1 = SD.open(archivo_aud_1, FILE_WRITE);
  //Se graba primera línea en archivo
  auditoria_1.print("Sistema de cierre ini. ");
  //Se graba hora de inicio de auditoria
  auditoria_1.print(rtc.getTimeStr());
  auditoria_1.print(" y dia ");
  //Se graba fecha de inicio de auditoria
  auditoria_1.print(rtc.getDateStr());
```

```
auditoria_1.println(" ");
//Se cierra archivo de auditoria
auditoria_1.close();

//Se comienza librería para usar LCD
lcd.init();
//Se enciende retroalimentación del LCD
lcd.backlight();
//Se posiciona en la ubicación de inicio
lcd.setCursor(0,0);
//Se envía mensaje a LCD
lcd.print(" Press SCAN ");
//Se posiciona en LCD
lcd.setCursor(0,1);
//Se envía mensaje a LCD
lcd.print(" -Puert cerr- ");

//Se comienza librería para usar LCD
lcd.init();
//Se enciende retroalimentación del LCD
lcd.backlight();
//Se limpia pantalla
lcd.clear();
//Se posiciona en LCD
lcd.setCursor(3,0);
//Se envía mensaje a LCD
lcd.print("ACBT-OSOP");
//Se posiciona en LCD
lcd.setCursor(2,1);
//Se envía mensaje a LCD
lcd.print("Press ENTER");
```

```

delay(1200);

pinMode(escaneo_entrada,INPUT);
pinMode(agrega_ide_entrada,INPUT);
pinMode(cierre,INPUT);
servosop.attach(servosop1);
servosop.write(puerta_abre);
delay(2000);
servosop.write(puerta_cierra);

digitalWrite(luz_roja,HIGH);
digitalWrite(luz_verde,LOW);
finger.begin(57600);
}

void loop() {

if(digitalRead(cierre))
{
  puerta_bloq = true;
  servosop.write(.....);
  digitalWrite(luz_roja,HIGH);
  digitalWrite(luz_verde,LOW);
  //Se posiciona en LCD
  lcd.setCursor(0,0);
  //Se envía mensaje a LCD
  lcd.print(" Puert cerr! ");
  //Se posiciona en LCD
  lcd.setCursor(0,1);
  //Se envía mensaje a LCD

```

```

    lcd.print("-----");
    delay(1800);
    //Se posiciona en LCD
    lcd.setCursor(0,0);
    //Se envía mensaje a LCD
    lcd.print("  Press SCAN  ");
    //Se posiciona en LCD
    lcd.setCursor(0,1);
    //Se envía mensaje a LCD
    lcd.print(" -Puert cerr- ");
    //Se apertura archivo para grabar datos de auditoria
    auditoria_1 = SD.open(archivo_aud_1, FILE_WRITE);
    auditoria_1.print(rtc.getDateStr()); auditoria_1.print(" -- ");
    auditoria_1.print(rtc.getTimeStr());
    auditoria_1.println(" -- Puerta cerr. usu.");
    auditoria_1.close();
}

if(digitalRead(escaneo_entrada) && !lido_sum)
{
    //Se comienza a grabar en archivo de auditoria de acceso
    auditoria_1 = SD.open(archivo_aud_1, FILE_WRITE);
    auditoria_1.print(rtc.getDateStr()); auditoria_1.print(" -- ");
    //Se graba hora para auditoría
    auditoria_1.print(rtc.getTimeStr());
    //Se graba mensaje para auditoría
    auditoria_1.println(" – Intento de ingreso");
    //Se cierra archivo
    auditoria_1.close();
    escaneando_1 = true;
}

```

```
lcd.setCursor(3,0);
lcd.print("ACBT-OSOP ");
//Se posiciona en LCD
lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
lcd.print("SCANNING-----");

while(escaneando_1 && contador_0 <= 60)
{
  getFingerprintID();
  delay(100);
  contador_0 = contador_0 + 1;
  if(contador_0 == 10)
  {
    //Se posiciona en LCD
    lcd.setCursor(3,0);
    //Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
    //Se posiciona en LCD
    lcd.setCursor(0,1);
    //Se envía mensaje a pantalla LCD
    lcd.print("SCANNING -----");
  }

  if(contador_0 == 20)
  {
    //Se posiciona en LCD
    lcd.setCursor(3,0);
    //Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
    //Se posiciona en LCD
```

```
    lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
    lcd.print("SCANNING  ---");
}

if(contador_0 == 40)
{
    //Se posiciona en LCD
    lcd.setCursor(3,0);
//Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
//Se posiciona en LCD
    lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
    lcd.print("SCANNING  -");
}

if(contador_0 == 50)
{
    //Se posiciona en LCD
    lcd.setCursor(3,0);
//Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
//Se posiciona en LCD
    lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
    lcd.print("SCANNING  ");
}

if(contador_0 == 59)
{
//Se posiciona en LCD
    lcd.setCursor(0,0);
```

```

//Se envía mensaje a pantalla LCD
  lcd.print(" Limite tiempo ");
//Se posiciona en LCD
  lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
  lcd.print(" Otra vez! ");
  auditoria_1 = SD.open(archivo_aud_1, FILE_WRITE);
  auditoria_1.print(rtc.getDateStr()); auditoria_1.print(" -- ");
auditoria_1.print(rtc.getTimeStr());
  auditoria_1.println(" -- Scanning timeout!");
  auditoria_1.close();
  delay(1800);
  if(puerta_bloq)
  {
    //Se posiciona en LCD
    lcd.setCursor(3,0);
//Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
//Se posiciona en LCD
    lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
    lcd.print(" -Puert cerr- ");
  }
  else
  {
    //Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
//Se posiciona en LCD
    lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
    lcd.print(" -Puert abie- ");
  }

```

```
}

}
escaneando_1 = false;
contador_0 = 0;

}

uint8_t getFingerprintID()
{
//Se asigna valor a variable
uint8_t dato= finger.getImage();
//Se comienza con función de selección según caso
switch (dato)
{
//Caso correcto ejecuta función
case FINGERPRINT_OK:
//Termina sentencia
break;
//Caso incorrecto retorna
case FINGERPRINT_NOFINGER: return dato;
//Caso incorrecto retorna
case FINGERPRINT_PACKETRECEIVEERR: return dato;
//Caso incorrecto retorna
case FINGERPRINT_IMAGEFAIL: return dato;
//Caso sin coincidencia retorna
default: return dato;
}
```

```
dato= finger.image2Tz();
//Se comienza con función de selección según caso
switch (dato)
{
//Caso correcto ejecuta función
    case FINGERPRINT_OK:
//Termina sentencia
        break;
//Caso incorrecto retorna
        case FINGERPRINT_IMAGEMESS: return dato;
//Caso incorrecto retorna
        case FINGERPRINT_PACKETRECI EVEERR: return dato;
//Caso incorrecto retorna
        case FINGERPRINT_FEATUREFAIL: return dato;
//Caso incorrecto retorna
        case FINGERPRINT_INVALIDIMAGE: return dato;
//Caso sin coincidencia retorna
        default: return dato;
}

dato= finger.fingerFastSearch();
//Se compara variable con estado correcto de lectura de huella
if (dato== FINGERPRINT_OK)
{
    escaneando_1 = false;
    contador_0 = 0;
    if(sum_nuevo_ide)
    {
        if(finger.fingerID == usuario_principal)
        {
            usuario_prin = true;

```

```
    ide_sum = false;
}
else
{
    sum_nuevo_ide = false;
    usuario_prin = false;
    ide_sum = false;
}

}
else
{
servosop.write(puerta_abre);
delay(2000);
servosop.write(puerta_cierra);
digitalWrite(luz_roja,LOW);
digitalWrite(luz_verde,HIGH);
//Se limpia pantalla LCD
lcd.clear();
//Se posiciona en LCD
lcd.setCursor(0,0);
//Se envía mensaje a pantalla LCD
lcd.print(" User match ");
//Se posiciona en LCD
lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
lcd.print(" ID: #");
//Se posiciona en LCD
lcd.setCursor(6,1);
//Se envía valor de variable a pantalla LCD
lcd.print(finger.fingerID);
//Se posiciona en LCD
```

```

    lcd.setCursor(9,1);
//Se envía mensaje a pantalla LCD
    lcd.print("%: ");
//Se posiciona en LCD
    lcd.setCursor(12,1);
//Se envía mensaje a pantalla LCD
    lcd.print("Correcto");

    auditoria_1 = SD.open(archivo_aud_1, FILE_WRITE);
    auditoria_1.print(rtc.getDateStr()); auditoria_1.print(" -- ");
    auditoria_1.print(rtc.getTimeStr());

    auditoria_1.print(" -- Usuario_coninc ID# "); auditoria_1.print(finger.fingerID);

    auditoria_1.print(" coincidencia: "); auditoria_1.print(finger.confidence);
    auditoria_1.println(" - puert. abie.");

    auditoria_1.close();

    puerta_bloq = false;
//tiempo de espera
    delay(4300);
//Se posiciona en LCD
    lcd.setCursor(3,0);
//Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
//Se posiciona en LCD
    lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
    lcd.print(" -Puert abie- ");
//tiempo de espera
    delay(50);
}
}

//Comparando
else if(dato== FINGERPRINT_NOTFOUND)

```

```
{
  escaneando_1 = false;
  ide_sum = false;
  contador_0 = 0;
  //Se posiciona en LCD
  lcd.setCursor(0,0);
  //Se envía mensaje a pantalla LCD
  lcd.print(" No econtra. ");
  //Se posiciona en LCD
  lcd.setCursor(0,1);
  //Se envía mensaje a pantalla LCD
  lcd.print(" Otra vez ");
  //Se asignan valor a las variables
  sum_nuevo_ide = false;
  usuario_prin = false;

  auditoria_1 = SD.open(archivo_aud_1, FILE_WRITE);
  auditoria_1.print(rtc.getDateStr()); auditoria_1.print(" -- ");
  auditoria_1.print(rtc.getTimeStr());
  auditoria_1.print(" -- No existe coincidencia");
  auditoria_1.close();
  delay(2000);
  if(puerta_bloq)
  {
    //Se posiciona en LCD
    lcd.setCursor(3,0);
    //Se envía mensaje a pantalla LCD
    lcd.print("ACBT-OSOP");
    //Se posiciona en LCD
    lcd.setCursor(0,1);
    //Se envía mensaje a pantalla LCD
    lcd.print(" -Puert cerr- ");
  }
}
```

```

}
else
{
//Se posiciona en LCD
  lcd.setCursor(3,0);
//Se envía mensaje a pantalla LCD
  lcd.print("ACBT-OSOP");
//Se posiciona en LCD
  lcd.setCursor(0,1);
//Se envía mensaje a pantalla LCD
  lcd.print(" -Puert abie- ");
}
delay(10);
return dato;
}
}

```

```

int obthuellavent() {
//Se asigna valor a variable
uint8_t dato= finger.getImage();
//Se compara para registro de huella correcta
if (dato!= FINGERPRINT_OK) return -1;
//Se asigna valor a variable
dato= finger.image2Tz();
//Se compara con registro de huella correcta
if (dato!= FINGERPRINT_OK) return -1;
dato= finger.fingerFastSearch();
}

```

```

uint8_t getFingerprintEnroll() {

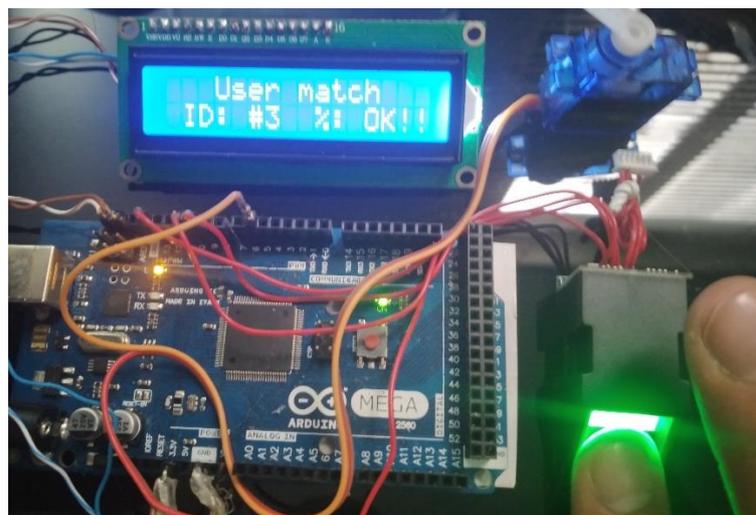
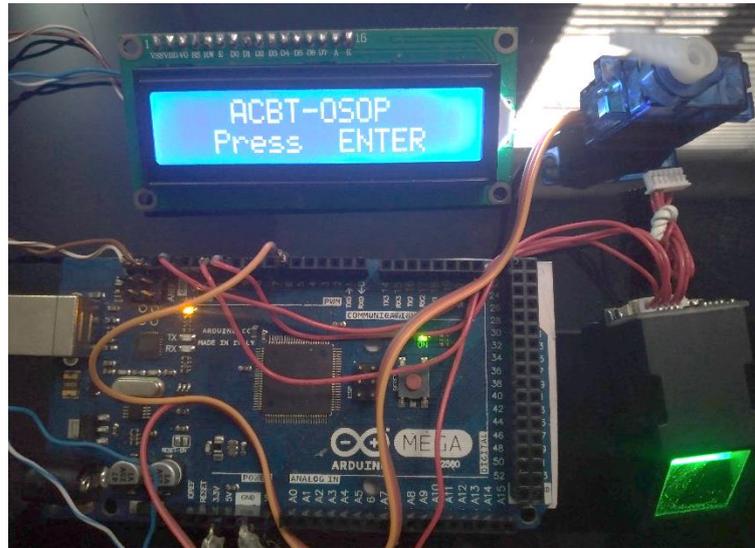
```

```
int dato= -1;
if(!primera_lectura)
{
//Se posiciona en LCD
  lcd.setCursor(0,0);
//Se envía mensaje a LCD
  lcd.print("Sum. nuevo ide# ");
//Se posiciona en LCD
  lcd.setCursor(14,0);
//Se envía mensaje a LCD
  lcd.print(id);
//Se posiciona en LCD
  lcd.setCursor(0,1);
//Se envía mensaje a LCD
  lcd.print(" Col. huella ");
}

while (dato!= FINGERPRINT_OK) {
//Se asigna valor a variable
  dato= finger.getImage();
//Función compara valor por caso
  switch (dato) {
//Caso huella correcta
    case FINGERPRINT_OK:
//Se posiciona en LCD
      lcd.setCursor(0,0);
//Se envía mensaje a LCD
      lcd.print(" Reg. okey ");
//Se posiciona en LCD
      lcd.setCursor(0,1);
      //tiempo de espera
      delay(200);
```

```
    primera_lectura = true;
    break;
//Caso de huella no encontrada
    case FINGERPRINT_NOFINGER:
//Se posiciona en LCD
        lcd.setCursor(0,0);
//Se envía mensaje a LCD
        lcd.print("Sum. nuevo ide# ");
//Se posiciona en LCD
        lcd.setCursor(14,0);
//Se envía variable a LCD
        lcd.print(id);
//Se posiciona en LCD
        lcd.setCursor(0,1);
//Se envia mensaje a LCD
        lcd.print(" Col. huella ");
        break;
//Caso error huella
    case FINGERPRINT_PACKETRECEIVEERR:
//Se posiciona en LCD
        lcd.setCursor(0,0);
//Envío de mensaje a LCD
        lcd.print(" Aviso ");
//Se posiciona en LCD
        lcd.setCursor(0,1);
//Envío de mensaje a LCD
        lcd.print(" ERROR! ");
//tiempo de espera
        delay(2000);
        break;
//Caso imagen huella falla
    case FINGERPRINT_IMAGEFAIL:
```

```
//Se posiciona en LCD
    lcd.setCursor(0,0);
//Envió de mensaje a LCD
    lcd.print("  -Imag.  ");
//Se posiciona en LCD
    lcd.setCursor(0,1);
//Envió de mensaje a LCD
    lcd.print("  ERROR!  ");
//tiempo de espera
    delay(2000);
    break;
default:
//Se posiciona en LCD
    lcd.setCursor(0,0);
//Envió de mensaje a LCD
    lcd.print("  -Descono.  ");
//Se posiciona en LCD
    lcd.setCursor(0,1);
//Envió de mensaje a LCD
    lcd.print("  ERROR!  ");
//tiempo de espera
    delay(2000);
    break;
}
}
```

ANEXO 4: Fotografía del Sistema de control desmontado

ANEXO 5: Fotográfica de Sistema de Monitoreo con carcasa de protección adosable.



ANEXO 6: Registro de ingreso contra las horas de ingreso del mes de marzo 2021

| Hora | Fecha | Registro de Personal |
|----------|------------|----------------------|
| 07:52:00 | 2021-03-05 | Operador1 |
| 09:52:00 | 2021-03-05 | Operador5 |
| 09:56:00 | 2021-03-05 | Operador2 |
| 11:34:00 | 2021-03-05 | Operador3 |
| 11:38:00 | 2021-03-05 | Operador5 |
| 13:30:00 | 2021-03-05 | Operador2 |
| 14:42:00 | 2021-03-05 | Operador5 |
| 14:46:00 | 2021-03-05 | Operador4 |
| 15:22:00 | 2021-03-05 | Operador2 |
| 15:40:00 | 2021-03-05 | Operador3 |
| 15:48:00 | 2021-03-05 | Operador1 |
| 17:20:00 | 2021-03-05 | Operador3 |
| 17:26:00 | 2021-03-05 | Operador2 |
| 17:48:00 | 2021-03-05 | Operador4 |
| 07:52:00 | 2021-03-06 | Operador1 |
| 09:52:00 | 2021-03-06 | Operador5 |
| 09:56:00 | 2021-03-06 | Operador1 |
| 11:34:00 | 2021-03-06 | Operador2 |
| 11:38:00 | 2021-03-06 | Operador1 |
| 13:30:00 | 2021-03-06 | Operador4 |
| 08:23:00 | 2021-03-08 | Operador1 |
| 08:44:00 | 2021-03-08 | Operador2 |
| 09:27:00 | 2021-03-08 | Operador1 |
| 10:02:00 | 2021-03-08 | Operador1 |
| 10:17:00 | 2021-03-08 | Operador4 |
| 10:28:00 | 2021-03-08 | Operador4 |
| 10:40:00 | 2021-03-08 | Operador3 |
| 11:03:00 | 2021-03-08 | Operador3 |
| 13:08:00 | 2021-03-08 | Operador2 |
| 14:17:00 | 2021-03-08 | Operador4 |
| 15:07:00 | 2021-03-08 | Operador4 |
| 15:31:00 | 2021-03-08 | Operador3 |
| 15:50:00 | 2021-03-08 | Operador5 |
| 16:40:00 | 2021-03-08 | Operador3 |
| 07:08:00 | 2021-03-09 | Operador3 |
| 07:54:00 | 2021-03-09 | Operador2 |
| 09:55:00 | 2021-03-09 | Operador5 |
| 10:14:00 | 2021-03-09 | Operador5 |
| 10:21:00 | 2021-03-09 | Operador5 |
| 10:59:00 | 2021-03-09 | Operador3 |
| 12:09:00 | 2021-03-09 | Operador4 |
| 12:25:00 | 2021-03-09 | Operador1 |
| 12:49:00 | 2021-03-09 | Operador3 |
| 14:40:00 | 2021-03-09 | Operador4 |
| 15:26:00 | 2021-03-09 | Operador4 |
| 16:15:00 | 2021-03-09 | Operador2 |
| 08:40:00 | 2021-03-10 | Operador2 |
| 09:16:00 | 2021-03-10 | Operador1 |
| 11:05:00 | 2021-03-10 | Operador1 |

| | | |
|----------|------------|-----------|
| 12:09:00 | 2021-03-10 | Operador1 |
| 13:51:00 | 2021-03-10 | Operador3 |
| 14:15:00 | 2021-03-10 | Operador5 |
| 14:50:00 | 2021-03-10 | Operador2 |
| 15:31:00 | 2021-03-10 | Operador2 |
| 15:50:00 | 2021-03-10 | Operador3 |
| 07:48:00 | 2021-03-11 | Operador1 |
| 09:23:00 | 2021-03-11 | Operador2 |
| 10:15:00 | 2021-03-11 | Operador3 |
| 10:50:00 | 2021-03-11 | Operador2 |
| 11:14:00 | 2021-03-11 | Operador5 |
| 11:23:00 | 2021-03-11 | Operador2 |
| 11:43:00 | 2021-03-11 | Operador1 |
| 13:21:00 | 2021-03-11 | Operador2 |
| 13:46:00 | 2021-03-11 | Operador2 |
| 13:55:00 | 2021-03-11 | Operador4 |
| 15:05:00 | 2021-03-11 | Operador1 |
| 15:14:00 | 2021-03-11 | Operador4 |
| 15:39:00 | 2021-03-11 | Operador1 |
| 07:18:00 | 2021-03-12 | Operador1 |
| 09:05:00 | 2021-03-12 | Operador4 |
| 10:07:00 | 2021-03-12 | Operador5 |
| 11:07:00 | 2021-03-12 | Operador2 |
| 11:58:00 | 2021-03-12 | Operador4 |
| 12:28:00 | 2021-03-12 | Operador5 |
| 13:20:00 | 2021-03-12 | Operador2 |
| 13:45:00 | 2021-03-12 | Operador5 |
| 14:25:00 | 2021-03-12 | Operador3 |
| 07:55:00 | 2021-03-13 | Operador1 |
| 08:41:00 | 2021-03-13 | Operador3 |
| 09:30:00 | 2021-03-13 | Operador4 |
| 09:53:00 | 2021-03-13 | Operador5 |
| 10:14:00 | 2021-03-13 | Operador4 |
| 10:25:00 | 2021-03-13 | Operador1 |
| 10:37:00 | 2021-03-13 | Operador3 |
| 10:58:00 | 2021-03-13 | Operador1 |
| 11:05:00 | 2021-03-13 | Operador1 |
| 12:40:00 | 2021-03-13 | Operador4 |
| 13:29:00 | 2021-03-13 | Operador5 |
| 14:05:00 | 2021-03-13 | Operador2 |
| 14:28:00 | 2021-03-13 | Operador5 |
| 07:03:00 | 2021-03-15 | Operador2 |
| 07:13:00 | 2021-03-15 | Operador4 |
| 07:45:00 | 2021-03-15 | Operador1 |
| 08:08:00 | 2021-03-15 | Operador1 |
| 09:39:00 | 2021-03-15 | Operador5 |
| 10:12:00 | 2021-03-15 | Operador1 |
| 13:45:00 | 2021-03-15 | Operador5 |
| 15:12:00 | 2021-03-15 | Operador5 |
| 15:33:00 | 2021-03-15 | Operador1 |
| 15:50:00 | 2021-03-15 | Operador5 |
| 08:04:00 | 2021-03-16 | Operador2 |
| 08:37:00 | 2021-03-16 | Operador4 |
| 08:58:00 | 2021-03-16 | Operador1 |

| | | |
|----------|------------|-----------|
| 10:02:00 | 2021-03-16 | Operador1 |
| 10:39:00 | 2021-03-16 | Operador2 |
| 12:40:00 | 2021-03-16 | Operador5 |
| 12:58:00 | 2021-03-16 | Operador2 |
| 13:36:00 | 2021-03-16 | Operador2 |
| 13:44:00 | 2021-03-16 | Operador3 |
| 15:15:00 | 2021-03-16 | Operador1 |
| 16:15:00 | 2021-03-16 | Operador1 |
| 07:19:00 | 2021-03-17 | Operador1 |
| 07:42:00 | 2021-03-17 | Operador5 |
| 07:50:00 | 2021-03-17 | Operador1 |
| 08:18:00 | 2021-03-17 | Operador1 |
| 08:48:00 | 2021-03-17 | Operador5 |
| 09:58:00 | 2021-03-17 | Operador1 |
| 10:33:00 | 2021-03-17 | Operador5 |
| 11:25:00 | 2021-03-17 | Operador3 |
| 12:03:00 | 2021-03-17 | Operador1 |
| 12:27:00 | 2021-03-17 | Operador4 |
| 14:05:00 | 2021-03-17 | Operador5 |
| 14:28:00 | 2021-03-17 | Operador3 |
| 15:16:00 | 2021-03-17 | Operador1 |
| 15:41:00 | 2021-03-17 | Operador1 |
| 16:12:00 | 2021-03-17 | Operador5 |
| 08:58:00 | 2021-03-18 | Operador5 |
| 10:11:00 | 2021-03-18 | Operador2 |
| 12:00:00 | 2021-03-18 | Operador2 |
| 13:08:00 | 2021-03-18 | Operador5 |
| 13:51:00 | 2021-03-18 | Operador5 |
| 13:54:00 | 2021-03-18 | Operador1 |
| 15:12:00 | 2021-03-18 | Operador1 |
| 07:23:00 | 2021-03-19 | Operador2 |
| 07:35:00 | 2021-03-19 | Operador3 |
| 08:20:00 | 2021-03-19 | Operador3 |
| 08:45:00 | 2021-03-19 | Operador5 |
| 09:02:00 | 2021-03-19 | Operador5 |
| 11:03:00 | 2021-03-19 | Operador2 |
| 11:59:00 | 2021-03-19 | Operador2 |
| 12:04:00 | 2021-03-19 | Operador2 |
| 13:57:00 | 2021-03-19 | Operador4 |
| 14:15:00 | 2021-03-19 | Operador1 |
| 14:59:00 | 2021-03-19 | Operador5 |
| 15:35:00 | 2021-03-19 | Operador3 |
| 08:49:00 | 2021-03-20 | Operador2 |
| 09:22:00 | 2021-03-20 | Operador1 |
| 09:47:00 | 2021-03-20 | Operador3 |
| 10:11:00 | 2021-03-20 | Operador3 |
| 13:05:00 | 2021-03-20 | Operador4 |
| 14:33:00 | 2021-03-20 | Operador3 |
| 15:10:00 | 2021-03-20 | Operador4 |
| 07:32:00 | 2021-03-22 | Operador4 |
| 07:47:00 | 2021-03-22 | Operador1 |
| 08:45:00 | 2021-03-22 | Operador5 |
| 08:55:00 | 2021-03-22 | Operador1 |
| 09:16:00 | 2021-03-22 | Operador5 |

| | | |
|----------|------------|-----------|
| 09:44:00 | 2021-03-22 | Operador5 |
| 10:45:00 | 2021-03-22 | Operador4 |
| 11:08:00 | 2021-03-22 | Operador4 |
| 11:17:00 | 2021-03-22 | Operador3 |
| 12:08:00 | 2021-03-22 | Operador4 |
| 12:16:00 | 2021-03-22 | Operador1 |
| 12:23:00 | 2021-03-22 | Operador3 |
| 12:29:00 | 2021-03-22 | Operador5 |
| 12:54:00 | 2021-03-22 | Operador2 |
| 13:08:00 | 2021-03-22 | Operador5 |
| 13:20:00 | 2021-03-22 | Operador4 |
| 15:12:00 | 2021-03-22 | Operador4 |
| 08:33:00 | 2021-03-23 | Operador2 |
| 09:30:00 | 2021-03-23 | Operador4 |
| 10:02:00 | 2021-03-23 | Operador4 |
| 10:26:00 | 2021-03-23 | Operador1 |
| 10:51:00 | 2021-03-23 | Operador5 |
| 11:48:00 | 2021-03-23 | Operador3 |
| 13:03:00 | 2021-03-23 | Operador4 |
| 13:15:00 | 2021-03-23 | Operador2 |
| 13:44:00 | 2021-03-23 | Operador3 |
| 14:16:00 | 2021-03-23 | Operador1 |
| 15:04:00 | 2021-03-23 | Operador4 |
| 15:21:00 | 2021-03-23 | Operador1 |
| 15:55:00 | 2021-03-23 | Operador5 |
| 08:17:00 | 2021-03-24 | Operador5 |
| 08:29:00 | 2021-03-24 | Operador2 |
| 09:23:00 | 2021-03-24 | Operador3 |
| 10:58:00 | 2021-03-24 | Operador1 |
| 11:12:00 | 2021-03-24 | Operador5 |
| 11:54:00 | 2021-03-24 | Operador2 |
| 12:04:00 | 2021-03-24 | Operador2 |
| 13:45:00 | 2021-03-24 | Operador3 |
| 07:03:00 | 2021-03-25 | Operador4 |
| 07:12:00 | 2021-03-25 | Operador2 |
| 08:29:00 | 2021-03-25 | Operador2 |
| 09:52:00 | 2021-03-25 | Operador2 |
| 12:25:00 | 2021-03-25 | Operador1 |
| 12:43:00 | 2021-03-25 | Operador2 |
| 13:16:00 | 2021-03-25 | Operador4 |
| 13:54:00 | 2021-03-25 | Operador5 |
| 13:58:00 | 2021-03-25 | Operador1 |
| 14:26:00 | 2021-03-25 | Operador5 |
| 14:34:00 | 2021-03-25 | Operador1 |
| 15:42:00 | 2021-03-25 | Operador1 |
| 07:17:00 | 2021-03-26 | Operador4 |
| 07:32:00 | 2021-03-26 | Operador1 |
| 08:18:00 | 2021-03-26 | Operador2 |
| 10:34:00 | 2021-03-26 | Operador4 |
| 10:48:00 | 2021-03-26 | Operador5 |
| 12:49:00 | 2021-03-26 | Operador2 |
| 12:55:00 | 2021-03-26 | Operador5 |
| 13:04:00 | 2021-03-26 | Operador5 |
| 14:36:00 | 2021-03-26 | Operador2 |

| | | |
|----------|------------|-----------|
| 14:56:00 | 2021-03-26 | Operador2 |
| 07:42:00 | 2021-03-27 | Operador5 |
| 08:16:00 | 2021-03-27 | Operador5 |
| 08:34:00 | 2021-03-27 | Operador5 |
| 08:55:00 | 2021-03-27 | Operador3 |
| 09:47:00 | 2021-03-27 | Operador5 |
| 10:06:00 | 2021-03-27 | Operador4 |
| 11:18:00 | 2021-03-27 | Operador1 |
| 11:22:00 | 2021-03-27 | Operador1 |
| 12:15:00 | 2021-03-27 | Operador3 |
| 13:09:00 | 2021-03-27 | Operador4 |
| 13:26:00 | 2021-03-27 | Operador3 |
| 14:30:00 | 2021-03-27 | Operador2 |
| 15:20:00 | 2021-03-27 | Operador1 |
| 15:51:00 | 2021-03-27 | Operador1 |
| 08:33:00 | 2021-03-29 | Operador4 |
| 08:46:00 | 2021-03-29 | Operador4 |
| 09:52:00 | 2021-03-29 | Operador2 |
| 10:34:00 | 2021-03-29 | Operador5 |
| 11:38:00 | 2021-03-29 | Operador4 |
| 11:56:00 | 2021-03-29 | Operador2 |
| 12:32:00 | 2021-03-29 | Operador1 |
| 14:03:00 | 2021-03-29 | Operador2 |
| 07:01:00 | 2021-03-30 | Operador3 |
| 07:13:00 | 2021-03-30 | Operador2 |
| 07:45:00 | 2021-03-30 | Operador3 |
| 08:47:00 | 2021-03-30 | Operador1 |
| 12:45:00 | 2021-03-30 | Operador3 |
| 12:56:00 | 2021-03-30 | Operador1 |
| 13:59:00 | 2021-03-30 | Operador5 |
| 15:25:00 | 2021-03-30 | Operador1 |
| 15:33:00 | 2021-03-30 | Operador2 |
| 07:24:00 | 2021-03-31 | Operador4 |
| 07:52:00 | 2021-03-31 | Operador4 |
| 09:14:00 | 2021-03-31 | Operador5 |
| 09:33:00 | 2021-03-31 | Operador1 |
| 12:13:00 | 2021-03-31 | Operador5 |
| 12:20:00 | 2021-03-31 | Operador2 |
| 13:45:00 | 2021-03-31 | Operador1 |
| 14:35:00 | 2021-03-31 | Operador5 |
| 15:48:00 | 2021-03-31 | Operador1 |