



UNL

Universidad
Nacional
de Loja

Universidad Nacional de Loja

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

Maestría en Ingeniería en Software

Diseño de un Sistema ANDON basado en un aplicativo web utilizando PHP
y dispositivo Arduino en una red ethernet aplicando la metodología de
desarrollo ICONIX

Trabajo de Titulación, previo a la
obtención del Título de Magíster en
Ingeniería en Software

AUTOR:

Edison Xavier Benítez Chica

DIRECTOR:

Ing. Cristian Ramiro Narvárez Guillen, Mg. Sc.

Loja - Ecuador

2025

Certificación

Loja, 13 de febrero de 2025

Ing. Cristian Ramiro Narváez Guillen, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Titulación denominado: **Diseño de un Sistema ANDON basado en un aplicativo web utilizando PHP y dispositivo Arduino en una red ethernet aplicando la metodología de desarrollo ICONIX**, previo a la obtención del título de **Magíster en Ingeniería en Software**, de autoría del estudiante **Edison Xavier Benítez Chica**, con cédula de identidad Nro. **1716752744**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Cristian Ramiro Narváez Guillen, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Edison Xavier Benítez Chica**, declaro ser autor del Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación del Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de Identidad: 1716752744

Fecha: 13/02/2025

Correo electrónico: xavidison@hotmail.com

**Carta de autorización por parte del autor, para consulta, reproducción parcial o total
y/o publicación electrónica de texto completo, del Trabajo de Titulación**

Yo, **Edison Xavier Benítez Chica**, declaro ser autor del Trabajo de Titulación denominado: **Diseño de un Sistema ANDON basado en un aplicativo web utilizando PHP y dispositivo Arduino en una red ethernet aplicando la metodología de desarrollo ICONIX** como requisito para optar el título de **Magíster en Ingeniería en Software**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

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

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

Para constancia de esta autorización, suscribo en la ciudad de Loja, a los trece días del mes de febrero de dos mil veinticinco.

Firma:

Autor: Edison Xavier Benítez Chica

Cédula de identidad: 1716752744

Dirección: Briceño OE167 y Av. Guayaquil, Quito - Ecuador

Correo electrónico: edison.benitez@unl.edu.ec

Teléfono: 0984509813

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Cristian Ramiro Narváez Guillen Mg. Sc.

Dedicatoria

Dedico este trabajo a todas aquellas personas que, con paciencia y comprensión, aceptaron mi ausencia durante este proceso. A quienes esperaron con cariño en los momentos en los que no pude estar presente, les agradezco profundamente por su apoyo incondicional y por entender que este esfuerzo era un sueño que requería toda mi atención.

Cada instante que no pude compartir con ustedes está lleno de gratitud por su amor, su paciencia y su fe en mí. Este logro también les pertenece, porque sin su comprensión, esta meta no habría sido posible.

Gracias por estar ahí, incluso cuando parecía que había desaparecido del mapa. Prometo que después de esto, habrá más tiempo para ustedes... y menos excusas de “estoy con la tesis”

De corazón este trabajo está dedicado para todos ustedes.

Edison Xavier

Agradecimiento

Todos los días tengo presente a Dios y a la Virgen Santísima y en mis oraciones siempre les pido me iluminen, me den la sabiduría necesaria y que cada decisión que tome sea acertada. También les pido protección y no puedo dejar pasar por alto esta oportunidad para agradecer todo lo que me proveen. Siempre serán misteriosos los designios de Dios.

Lo bonito de poder agradecer, no solo es ser recíproco, a veces el agradecimiento sale de corazón en cada gesto y cada palabra.

A mi familia, por su amor incondicional y apoyo constante, que siempre ha sido mi mayor fortaleza y motivación. A mis amigos, quienes con sus palabras de aliento y momentos de alegría me han ayudado a mantener el equilibrio y la perspectiva. A mis profesores y mentores, por compartir su conocimiento y por inspirarme a superar mis propios límites.

También quiero agradecer, en estas pocas palabras a todas las personas que, de una forma u otra hacen que el tiempo a lo largo de este proceso, sirva para reconocer que cada esfuerzo realizado, esfuerzo que, por más pequeña que sea, es un peldaño que nos acerca a nuestros sueños, y que la verdadera satisfacción reside no solo en lograrla, sino en valorar el camino recorrido y a quienes lo hicieron significativo.

Gracias también a ti y a mí, a ti por ver siempre en mí que todo es posible, siempre que se ponga un granito de arena en buscar conseguirlo, gracias por levantarte día a día y al vernos frente a frente en el espejo, decir con firmeza, los buenos somos pocos, y aún con adversidades elegir para cada nuevo día elegir ser feliz.

Finalmente, invito a todos los que lean este trabajo a establecer sus propias metas y a disfrutar de cada paso del proceso, recordando siempre agradecer a quienes caminan junto a ellos. Es en la constancia, el esfuerzo y la gratitud donde encontramos la verdadera esencia de nuestros logros.

Con humildad y emoción, agradezco a todos por ser parte de nuevo logro cumplido.

Edison Xavier

Índice de contenidos

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas	xi
Índice de figuras	xii
Índice de anexos	xiv
1. Título	1
2. Resumen	2
Abstract	3
3. Introducción	4
4. Marco teórico	7
4.1. Sistemas ANDON	7
4.1.1. Una revisión conceptual.....	7
4.1.2. ¿Qué es un sistema ANDON y cómo funciona?	7
4.1.3. Características principales	8
4.1.4. Componentes básicos de un sistema ANDON.....	8
4.2. Metodología de Desarrollo ICONIX	9
4.2.1. Orientación a objetos	9
4.2.2. Reutilización de componentes	9
4.2.3. Integración de sistemas	10
4.2.4. Ciclo de Vida del Desarrollo de Software en ICONIX.....	10
4.3. Tecnologías: PHP y Arduino	11
4.3.1. PHP.....	11
4.3.2. Arduino	11
4.3.3. Combinando PHP y Arduino	12
4.4. Integración de Tecnologías.....	13
4.5. Relación con la problemática de investigación.....	13
5. Metodología	15

5.1.	Metodología General	15
5.1.1.	Localización.....	15
5.1.2.	Método de estudio	15
5.1.3.	Tipo de investigación.....	15
5.1.4.	Población y muestra.....	15
5.2.	Metodología Específica.....	16
5.3.	Análisis de requerimientos	17
5.3.1.	Requerimiento Funcionales	19
5.3.2.	Requerimientos No Funcionales	21
5.4.	Casos de Uso.....	22
5.4.1.	Caso de Uso 1: Generar Alerta Manual	24
5.4.2.	Caso de Uso 2: Clasificar Alertas.....	24
5.4.3.	Caso de Uso 3: Registrar Alertas Automáticas.....	25
5.4.4.	Caso de Uso 4: Monitorear Estado en Tiempo Real.....	25
5.4.5.	Caso de Uso 5: Enviar Notificaciones Automáticas.....	26
5.4.6.	Caso de Uso 6: Configurar Usuarios, Roles y Permisos	26
5.4.7.	Caso de Uso 7: Agregar o Eliminar Estaciones de Trabajo.....	27
5.4.8.	Caso de Uso 8: Almacenar Historial de Alertas.....	28
5.4.9.	Caso de Uso 9: Generar Reportes Exportables	28
5.4.10.	Caso de Uso 10: Monitorear Conectividad de Dispositivos.....	29
5.5.	Diagrama de Casos de Uso UML.....	30
5.6.	Diagramas de Secuencia.....	31
5.6.1.	Diagrama de Secuencia Generar Alerta Manual.....	31
5.6.2.	Diagrama de Secuencia Clasificar Alertas	32
5.6.3.	Diagrama de Secuencia Visualizar Alertas	33
5.6.4.	Diagrama de Secuencia Monitorear Estado en Tiempo Real	34
5.6.5.	Diagrama de Secuencia Configurar Usuarios y Roles.....	35
5.6.6.	Diagrama de Secuencia Agregar o Eliminar Estaciones.....	36
5.6.7.	Diagrama de Secuencia Generar Reportes Exportables.....	37
5.6.8.	Diagrama de Secuencia Monitorear Conectividad	38
5.6.9.	Diagrama de Secuencia Almacenar Historial de Alertas	39
5.6.10.	Diagrama de Secuencia Registrar Alertas Automáticas.....	40
5.7.	Diagrama de Clases	41
5.7.1.	Ajuste Diagrama de Clases.....	41
5.8.	Diagrama de Componentes.....	42

5.8.1.	Definición de los componentes clave (Nodos)	42
5.8.1.1.	Nodo Usuario Final.....	43
5.8.1.2.	Nodo Frontend.....	43
5.8.1.3.	Nodo Backend (PHP).....	43
5.8.1.4.	Nodo Base de Datos (MySQL).....	43
5.8.1.5.	Nodo Dispositivo Arduino.....	43
5.8.1.6.	Nodo Correo Electrónico (Alerta).....	44
5.8.1.7.	Relaciones entre Componentes	44
5.9.	Diseño de la base de datos.....	45
5.10.	Desarrollo del backend PHP.....	47
5.11.	Desarrollo de la interfaz web	47
5.12.	Implementación del Arduino	49
5.13.	Pruebas de integración	52
5.13.1.	Prueba: Generar Alerta Manual	52
5.13.2.	Prueba: Registrar Alertas Automáticas	53
5.13.3.	Prueba: Monitorear Estado en Tiempo Real.....	54
5.13.4.	Prueba: Generar Reportes Exportables	54
5.13.5.	Prueba: Notificaciones Automáticas.....	55
5.13.6.	Prueba: Configuración de Usuarios y Roles.....	56
5.14.	Implementación y pruebas en campo.....	56
5.14.1.	Prueba de Conectividad.....	57
5.14.2.	Prueba de Funcionamiento de Alertas	58
5.14.3.	Prueba de Notificaciones.....	59
5.14.4.	Prueba de Reportes	60
5.14.5.	Prueba de Rendimiento	62
5.14.6.	Prueba de Usabilidad	64
6.	Resultados	66
6.1.	Cumplimiento de Requerimientos Funcionales	66
6.2.	Pruebas de Integración	67
6.3.	Desempeño del Sistema	67
6.4.	Experiencia del Usuario	68
6.5.	Análisis de Reportes Generados.....	68
6.6.	Limitaciones Identificadas.....	68
7.	Discusión	69
8.	Conclusiones	73

9. Recomendaciones	74
10. Bibliografía	76
11. Anexos	78

Índice de tablas

Tabla 1. Requerimientos Funcionales	19
Tabla 2. Requerimientos No Funcionales	21
Tabla 3. Descripción de Casos de Uso.....	22
Tabla 4. Caso de Uso 1: Generar Alerta Manual	24
Tabla 5. Caso de Uso 2: Clasificar Alertas	24
Tabla 6. Caso de Uso 3: Registrar Alertas Automáticas.....	25
Tabla 7. Caso de Uso 4: Monitorear Estado en Tiempo Real	25
Tabla 8. Caso de Uso 5: Enviar Notificaciones Automáticas	26
Tabla 9. Caso de Uso 6: Configurar Usuarios, Roles y Permisos.....	27
Tabla 10. Caso de Uso 7: Agregar o Eliminar Estaciones de Trabajo	27
Tabla 11. Caso de Uso 8: Almacenar Historial de Alertas	28
Tabla 12. Caso de Uso 9: Generar Reportes Exportables	28
Tabla 13. Caso de Uso 10: Monitorear Conectividad de Dispositivos	29

Índice de figuras

Figura 1. Diagrama de Casos de Uso UML.....	30
Figura 2. Generar Alerta Manual.....	31
Figura 3. Clasificar Alerta.....	32
Figura 4. Visualizar alertas	33
Figura 5. Monitorear estado en tiempo real	34
Figura 6. Configurar Usuario Roles.....	35
Figura 7. Agregar eliminar estaciones	36
Figura 8. Generar Reportes Exportables	37
Figura 9. Monitorear Conectividad.....	38
Figura 10. Almacenar historias alertas.....	39
Figura 11. Registrar alertas automáticas	40
Figura 12. Diagrama de clases	41
Figura 13. Ajuste Diagrama de Clases.....	42
Figura 14. Diagrama de componentes	45
Figura 15. Modelo de la Base de Datos.....	46
Figura 16. Fragmento de código PHP.....	48
Figura 17. Interface para visualización de alertas.....	49
Figura 18. Interface para visualizar las actividades del supervisor / administrador	49
Figura 19. Fragmento de código para Arduino.....	50
Figura 20. Software Arduino, placa base.....	51
Figura 21. Proceso de la Prueba de conectividad	58
Figura 22. Proceso de la Prueba de Funcionamiento de Alertas	59
Figura 23. Proceso de la prueba de notificaciones.....	60
Figura 24. Proceso de la prueba de reportes	61
Figura 25. Proceso de la prueba de rendimiento.....	62

Figura 26. Captura de pantalla de la generación de reportes	63
Figura 27. Reporte generado, descargado y abierto con el programa de ofimática Excel.....	63
Figura 28. Proceso de la prueba de usabilidad.....	65

Índice de anexos

Anexo A. Módulo RFID RC522.....	78
Anexo B. Módulo Ethernet.....	79
Anexo C. Certificado de traducción Resumen - Abstract	80

1. Título

**Diseño de un Sistema ANDON basado en un aplicativo web utilizando PHP y dispositivo Arduino en una red ethernet aplicando la metodología de desarrollo
ICONIX**

2. Resumen

El presente estudio muestra el diseño e implementación de un sistema ANDON basado en un aplicativo web y dispositivos Arduino, orientado a optimizar la gestión de incidencias en entornos industriales. El objetivo principal fue desarrollar un sistema que permita la generación de alertas en tiempo real, la categorización de incidencias, la notificación automática a responsables y la visualización centralizada del estado de las estaciones de trabajo. Se utilizó la metodología ICONIX para el desarrollo del sistema, apoyándose en diagramas UML para estructurar los casos de uso, secuencia y clases. El aplicativo fue implementado en PHP para el backend y Arduino para la gestión de hardware, utilizando el protocolo HTTP/1.1 para la transmisión de datos.

Los resultados obtenidos incluyen una reducción significativa en los tiempos de respuesta ante incidencias, gracias a la implementación de un sistema de monitoreo en tiempo real con códigos de color y notificaciones automáticas. El sistema soportó hasta 50 conexiones simultáneas sin degradación de rendimiento y demostró ser escalable para entornos de alta exigencia. Además, los reportes generados permitieron analizar tendencias y mejorar la toma de decisiones en la gestión operativa. Entre las limitaciones identificadas se encuentra la dependencia de redes estables para garantizar la conectividad de los dispositivos Arduino, lo que podría mejorarse mediante la integración de protocolos adicionales como MQTT.

En conclusión, el sistema ANDON desarrollado representa una solución accesible y eficiente para la mejora continua de procesos industriales, ofreciendo un modelo flexible y escalable que puede ser adaptado a diversas necesidades. Este proyecto sienta las bases para futuras investigaciones en la integración de tecnologías emergentes como inteligencia artificial y blockchain para potenciar la automatización y trazabilidad del sistema.

Palabras clave: sistemas ANDON, Arduino, monitoreo en tiempo real, protocolo HTTP, gestión de alertas, aplicaciones web.

Abstract

This paper presents the design and implementation of an ANDON system based on a web application and Arduino devices, with the objective of optimizing incident management in industrial environments. The main objective was to develop a system that enables real-time alert generation, incident categorization, automatic notification to responsible parties, and centralized visualization of workstation statuses. The ICONIX methodology was used for system development, leveraging UML diagrams to structure use cases, sequences, and classes. The application was implemented in PHP for the backend and Arduino for hardware management, utilizing the HTTP/1.1 protocol for data transmission.

The results obtained include a significant reduction in response times to incidents, thanks to the implementation of a real-time monitoring system with color-coded indicators and automatic notifications. The system supported up to 50 simultaneous connections without performance degradation and proved scalable for larger environments. Additionally, the generated reports allowed for trend analysis and improved decision-making in operational management. Among the identified limitations was the dependence on stable networks to ensure Arduino device connectivity, which could be improved by integrating additional protocols such as MQTT.

In conclusion, the developed ANDON system represents an accessible and efficient solution for the continuous improvement of industrial processes, offering a flexible and scalable model adaptable to various needs. This project lays the ground work for future research in integrating emerging technologies such as artificial intelligence and blockchain to enhance the system's automation and traceability.

Keywords: ANDON systems, Arduino, real-time monitoring, HTTP protocol, alert management, web applications.

3. Introducción

El sistema ANDON, ampliamente utilizado en entornos de manufactura y producción, es una herramienta clave dentro de los principios del Lean Manufacturing, ya que permite identificar y responder rápidamente a problemas operativos. El término ANDON se refiere a un sistema de señalización visual y/o auditiva que notifica a los trabajadores y supervisores sobre incidencias en tiempo real, contribuyendo a la mejora continua de los procesos. Este proyecto aborda el diseño e implementación de un sistema ANDON basado en un aplicativo web y dispositivos Arduino, integrando tecnologías accesibles y funcionales que optimicen la gestión de alertas en entornos industriales.

En la actualidad, muchas empresas enfrentan desafíos relacionados con la falta de comunicación eficiente y la demora en la identificación de incidencias en sus procesos de producción. Esto puede resultar en interrupciones, pérdidas económicas y desperdicio de recursos. Según Liker (2004), la implementación de herramientas Lean, como el sistema ANDON, no solo mejora la comunicación, sino que también contribuye a reducir los tiempos de inactividad y los costos asociados. Este estudio responde a dicha necesidad mediante el uso de tecnologías de bajo costo, como Arduino y PHP, que ofrecen una solución escalable y adaptable a las necesidades de pequeñas y medianas empresas (Banzi y Shiloh, 2014).

El objetivo principal del proyecto es desarrollar un sistema ANDON que permita gestionar alertas en tiempo real desde estaciones de trabajo, optimizando la comunicación y mejorando los tiempos de respuesta. Los objetivos específicos incluyen: diseñar un aplicativo web que permita visualizar el estado de las estaciones en tiempo real; integrar dispositivos Arduino para la generación automática de alertas; configurar notificaciones automáticas para

los responsables; y desarrollar funcionalidades administrativas para la configuración de usuarios, roles y reportes.

La importancia del sistema radica en los beneficios directos para el sector industrial, como la reducción de tiempos improductivos, el aumento en la eficiencia de los procesos y la mejora en la toma de decisiones basada en datos históricos. Además, el uso de tecnologías accesibles contribuye a democratizar el acceso a herramientas avanzadas, permitiendo que empresas de menor tamaño puedan implementar soluciones tecnológicas eficaces.

Este estudio también se fundamenta en desarrollos previos que han demostrado la efectividad de los sistemas ANDON en la mejora de procesos industriales. Estudios recientes destacan la integración de tecnologías de bajo costo como Arduino y el uso de protocolos de comunicación HTTP/1.1 para la transmisión de datos en tiempo real. Aunque estos avances han permitido un progreso significativo, aún existen desafíos en la conectividad y escalabilidad que este proyecto busca abordar.

El alcance de este estudio se centra en el diseño e implementación de un prototipo funcional que gestione alertas en un entorno industrial simulado, considerando las limitaciones de recursos y conectividad típicas en pequeñas y medianas empresas. Aunque el sistema se probó en escenarios controlados, sus principios y diseño son fácilmente adaptables a entornos más complejos y de mayor escala.

En resumen, este estudio busca llenar un vacío en la implementación de sistemas ANDON accesibles y adaptables, proponiendo una solución que combina tecnologías modernas, facilidad de uso y funcionalidad. Con ello, se espera no solo mejorar los procesos

operativos de las empresas, sino también sentar las bases para investigaciones futuras en la mejora continua de sistemas de gestión industrial.

4. Marco teórico

El presente marco teórico se centra en el diseño y desarrollo de un sistema ANDON, una herramienta visual que permite comunicar en tiempo real el estado de una línea de producción. En este caso, se busca abordar la problemática de la gestión de la falta de materia prima y los mantenimientos programados en plantas de producción. Para ello, se propone la implementación de un sistema ANDON basado en un aplicativo web utilizando PHP y Arduino, siguiendo la metodología de desarrollo ICONIX.

4.1. Sistemas ANDON

4.1.1. Una revisión conceptual

Los sistemas ANDON han evolucionado desde simples paneles de luces hasta sofisticadas plataformas digitales que integran diversas tecnologías. Estos sistemas cumplen un papel fundamental en la mejora continua de los procesos productivos, al permitir una rápida identificación de problemas y una respuesta ágil por parte del personal de producción.

4.1.2. ¿Qué es un sistema ANDON y cómo funciona?

Según Lean Construction México [1] y SafetyCulture [2], el sistema ANDON es una herramienta clave dentro del Lean Manufacturing que permite la notificación visual y auditiva de problemas en la producción. Como destacan Liker y Hoseus [3], el éxito de Toyota radica en su enfoque en la comunicación visual efectiva y la resolución de problemas en tiempo real, donde ANDON juega un papel clave.

El sistema ANDON se alinea con los principios del Lean Manufacturing, proporcionando una herramienta eficaz para la identificación y eliminación de desperdicios en el proceso de producción. Su implementación mejora la eficiencia operativa al reducir los

tiempos de respuesta ante fallos en la producción y optimizar la coordinación entre los equipos de trabajo (Lean Construction México, 2023).

Además, investigaciones recientes han destacado la importancia del uso de sistemas ANDON en la industria moderna, donde la integración con tecnologías digitales y plataformas IoT permite una gestión más efectiva de las alertas en tiempo real, mejorando la trazabilidad y el control del proceso productivo (Martínez-Hernández et al., 2020). Estos avances han permitido a las empresas manufactureras reducir defectos y minimizar tiempos de inactividad en la producción, contribuyendo así a una mayor eficiencia y rentabilidad.

4.1.3. Características principales

El sistema ANDON ha evolucionado en los últimos años con la incorporación de nuevas tecnologías digitales y métodos de optimización industrial. Según Koen [4], la aplicación de principios Lean en la industria moderna no solo mejora la eficiencia operativa, sino que también optimiza la comunicación entre trabajadores y sistemas automatizados, aumentando la productividad y reduciendo el desperdicio.

4.1.4. Componentes básicos de un sistema ANDON

Estudios recientes como el de Martínez-Hernández et al. [5] destacan que la implementación de sistemas ANDON ha reducido tiempos de respuesta y mejorado la eficiencia operativa en líneas de producción. Entre los componentes básicos que podemos destacar en un sistema ANDON, están los siguientes:

- **Luces de señalización:** Cada estación de trabajo o área de producción tiene una luz que indica su estado: verde (funcionando correctamente), amarillo (problema menor) o rojo (parada de emergencia).
- **Botones o pulsadores:** Los trabajadores activan las luces de señalización al presionar un botón cuando detectan un problema.

- **Panel de control:** Un panel centralizado muestra el estado de todas las estaciones y permite a los responsables tomar decisiones rápidas.
- **Sistema de comunicación:** En sistemas más avanzados, se utilizan pantallas digitales, alarmas sonoras y sistemas de comunicación internos para notificar a los equipos de mantenimiento, calidad o ingeniería.

4.2. Metodología de Desarrollo ICONIX

ICONIX es una metodología orientada a objetos que facilita la documentación del desarrollo de software, asegurando la coherencia en la transición entre requerimientos y diseño [6]. Según Pressman y Maxim [7], metodologías basadas en UML como ICONIX proporcionan un marco detallado para capturar requisitos, definir diagramas y garantizar una implementación alineada con los objetivos del sistema, facilitando la gestión de proyectos de software de manera eficiente. Sus principales características son:

4.2.1. Orientación a objetos

Modela los elementos del sistema como objetos con atributos y comportamientos.

- **Modelado del mundo real:** ICONIX permite modelar los elementos del sistema real como objetos en el software, lo que facilita la comprensión y el mantenimiento del código.
- **Encapsulación:** Los objetos agrupan datos (atributos) y comportamiento (métodos) en una sola entidad, promoviendo la modularidad y la ocultación de información.
- **Herencia:** Los objetos pueden heredar características de otros objetos, lo que fomenta la reutilización de código y la creación de jerarquías de clases.
- **Polimorfismo:** Permite que objetos de diferentes clases puedan responder de manera diferente al mismo mensaje, aumentando la flexibilidad del sistema.

4.2.2. Reutilización de componentes

Promueve la reutilización de componentes software para acelerar el desarrollo.

- **Bibliotecas de componentes:** ICONIX promueve la creación y el uso de bibliotecas de componentes preconstruidos, lo que acelera el desarrollo y reduce el tiempo de salida al mercado.
- **Patrones de diseño:** Se utilizan patrones de diseño reconocidos para resolver problemas comunes en el desarrollo de software, asegurando soluciones probadas y eficientes.

4.2.3. Integración de sistemas

Arduino es una de las plataformas más utilizadas en la automatización de procesos industriales debido a su facilidad de integración con sensores y actuadores. Como señala Hossain [8], su flexibilidad y compatibilidad con IoT permiten desarrollar sistemas escalables, económicos y de alto rendimiento, facilitando la implementación de monitoreo en tiempo real en diversas aplicaciones industriales.

- **Interfaces bien definidas:** ICONIX enfatiza la definición clara de interfaces entre componentes, lo que facilita la integración de sistemas heterogéneos.
- **Adaptabilidad:** Los sistemas desarrollados con ICONIX son más adaptables a cambios en los requisitos y a la integración de nuevas tecnologías.

4.2.4. Ciclo de Vida del Desarrollo de Software en ICONIX

Proceso iterativo que permite a los equipos de desarrollo ajustar y refinar el software a medida que avanzan.

1. **Análisis de Requisitos:** Identificación clara y detallada de los requisitos funcionales y no funcionales del sistema.
2. **Diseño:** Creación de modelos de objetos que representan el sistema, utilizando diagramas UML (Unified Modeling Language).
3. **Implementación:** Codificación de los objetos y componentes del sistema, utilizando un lenguaje de programación orientado a objetos.
4. **Pruebas:** Verificación de que el sistema cumple con los requisitos y funciona correctamente.
5. **Implementación:** Despliegue del sistema en el entorno de producción.

6. **Mantenimiento:** Actualización y corrección de errores del sistema a lo largo de su ciclo de vida.

4.3. Tecnologías: PHP y Arduino

4.3.1. PHP

PHP (Pre-Procesador de Hipertexto - Hypertext Preprocessor), es un lenguaje de programación de propósito general, pero ha encontrado su nicho principal en el desarrollo web del lado del servidor. Esto significa que el código PHP se ejecuta en un servidor web (como Apache o Nginx) y genera el HTML (Lenguaje de Marcas de Hipertexto - HyperText Markup Language) que se envía al navegador del usuario, ampliamente utilizado para el desarrollo de aplicaciones web. Ofrece una sintaxis sencilla y una gran comunidad de desarrolladores.

4.3.1.1. Características Clave de PHP

- **Facilidad de Aprendizaje:** Su sintaxis es similar a la del lenguaje C, lo que lo hace accesible para programadores con experiencia en otros lenguajes.
- **Amplia Comunidad:** Cuenta con una gran comunidad de desarrolladores que contribuyen a su crecimiento y ofrecen soporte.
- **Extensibilidad:** A través de extensiones (o módulos), PHP puede integrarse con una amplia variedad de bases de datos (MySQL, PostgreSQL, etc.), sistemas de gestión de contenidos (WordPress, Drupal) y otros servicios.
- **Eficiencia:** PHP es conocido por su velocidad y eficiencia en la generación de contenido dinámico para sitios web.
- **Aplicaciones Típicas:**
 - Desarrollo de sitios web dinámicos.
 - Creación de aplicaciones web.
 - Desarrollo de sistemas de gestión de contenido (CMS).
 - Desarrollo de servicios web (APIs).

4.3.2. Arduino

Es una plataforma de hardware de código abierto basada en una placa sencilla con un microcontrolador y un entorno de desarrollo integrado (IDE) fácil de usar. Arduino es

ampliamente utilizado en proyectos industriales debido a su flexibilidad y bajo costo [9], [10]. Aunque se originó como una herramienta para enseñar electrónica, ha evolucionado hasta convertirse en una plataforma popular para una amplia gama de proyectos, desde la domótica hasta la robótica. Es ideal para la adquisición de datos de sensores y el control de actuadores.

4.3.2.1. Características Clave de Arduino

- **Hardware de Bajo Costo:** Las placas Arduino son relativamente económicas, lo que las hace accesibles para estudiantes, aficionados y profesionales.
- **Software Libre:** El IDE de Arduino es gratuito y de código abierto, lo que permite a los usuarios personalizarlo y ampliar sus funcionalidades.
- **Lenguaje de Programación Sencillo:** El lenguaje de programación de Arduino se basa en C++, pero simplificado para facilitar su aprendizaje.
- **Gran Ecosistema:** Existe una amplia variedad de sensores, actuadores, shields y bibliotecas disponibles para Arduino, lo que facilita la creación de proyectos personalizados.
- **Aplicaciones Típicas:**
 - Control de dispositivos electrónicos.
 - Automatización de tareas.
 - Robótica.
 - Internet de las Cosas (IoT).

4.3.3. Combinando PHP y Arduino

Para lograr una comunicación eficiente entre Arduino y aplicaciones web, se emplean APIs REST, las cuales permiten intercambiar datos en tiempo real mediante el protocolo HTTP. White [11] explica que las APIs RESTful han revolucionado la forma en que los dispositivos IoT se comunican con servidores y aplicaciones web, brindando escalabilidad y flexibilidad en la transmisión de datos en la nube.

- PHP es ideal para la creación de interfaces web y la gestión de datos en el servidor.
- Arduino es excelente para la interacción con el mundo físico, el control de dispositivos y la adquisición de datos.
- Al combinar estas dos tecnologías, se pueden crear aplicaciones inteligentes y personalizadas que interactúan con el mundo físico a través de Internet.

La combinación de PHP y Arduino en aplicaciones IoT es una tendencia en el desarrollo de sistemas industriales debido a su adaptabilidad y escalabilidad [12]. Por ejemplo:

- **Control Remoto de Dispositivos:** Un sitio web desarrollado con PHP puede enviar comandos a un Arduino para controlar dispositivos electrónicos en un hogar o en una fábrica.
- **Recopilación de Datos:** Sensores conectados a un Arduino pueden recopilar datos (temperatura, humedad, etc.) y enviarlos a un servidor PHP para su almacenamiento y análisis.
- **Interfaces Gráficas para Proyectos Arduino:** PHP puede utilizarse para crear interfaces web interactivas que permiten controlar y monitorear proyectos basados en Arduino.

4.4. Integración de Tecnologías

La integración de PHP y Arduino en un sistema ANDON permite:

- **Adquisición de datos en tiempo real:** Los sensores conectados a Arduino recolectan datos sobre el estado de la línea de producción (por ejemplo, disponibilidad de materia prima, estado de las máquinas).
- **Procesamiento de datos:** PHP procesa los datos recibidos de Arduino y los presenta en una interfaz web intuitiva.
- **Visualización:** El aplicativo web muestra información relevante sobre el estado de la línea, como alarmas, tiempos de ciclo y eficiencia.
- **Control remoto:** Los usuarios pueden interactuar con el sistema a través de la interfaz web para realizar ajustes y tomar decisiones.

4.5. Relación con la problemática de investigación

La implementación de un sistema ANDON basado en PHP y Arduino permitirá:

- Identificar de manera temprana las paradas de producción causadas por la falta de materia prima y los mantenimientos programados.
- Reducir los tiempos de respuesta ante eventos no planificados, minimizando las pérdidas económicas.

- Mejorar la comunicación entre los diferentes actores involucrados en el proceso productivo.
- Facilitar la toma de decisiones basadas en datos en tiempo real.

5. Metodología

5.1. Metodología General

5.1.1. Localización

El estudio se llevará a cabo en el contexto de una planta de producción industrial, donde se identificarán los procesos susceptibles de ser optimizados mediante la implementación de un sistema ANDON. La elección de la planta específica dependerá de la disponibilidad de datos y la colaboración de la empresa.

5.1.2. Método de estudio

Se emplea un enfoque mixto, combinando elementos cuantitativos y cualitativos. El enfoque cuantitativo permitirá medir el impacto del sistema ANDON en variables como el tiempo de ausencia de materia prima o tiempos de para por mantenimientos programados. Por otro lado, el enfoque cualitativo permitirá comprender en profundidad las percepciones de los operadores y la gestión sobre el sistema, así como identificar las barreras y oportunidades para su mejora.

5.1.3. Tipo de investigación

Se realizará un estudio de caso, centrándose en la implementación y evaluación del sistema ANDON en una planta piloto. Este diseño permitirá analizar en profundidad el proceso de implementación y los resultados obtenidos.

5.1.4. Población y muestra

La población de estudio estará compuesta por los operadores de la línea de producción y el personal de mantenimiento. La muestra se seleccionará de manera no probabilística,

teniendo en cuenta criterios como la experiencia y el nivel de conocimiento de los procesos productivos.

5.2. Metodología Específica

La metodología específica enmarca los siguientes procedimientos:

- **Diseño del sistema:**

Arquitectura del sistema: Definir la arquitectura del sistema, incluyendo los componentes hardware (Arduino, sensores, red Ethernet) y software (PHP, base de datos).

Diseño de la interfaz de usuario: Crear una interfaz web intuitiva para visualizar el estado de los procesos, las alertas y los históricos de datos.

Diseño de la base de datos: Diseñar una base de datos para almacenar los datos recopilados por el Arduino, como los niveles de inventario, las fechas y horas de las alertas.

- **Desarrollo del sistema:**

Programación del Arduino: Programar el Arduino para leer los datos de los sensores y enviarlos al servidor PHP a través de la red Ethernet.

Desarrollo del aplicativo web: Desarrollar el aplicativo web en PHP para recibir los datos del Arduino, almacenarlos en la base de datos y presentarlos en la interfaz de usuario.

Integración: Integrar el sistema con los sistemas existentes de la empresa, si es necesario.

- **Implementación y pruebas:**

Instalación: Instalar el hardware y software en la línea de producción.

Configuración: Configurar los sensores, la red Ethernet y el servidor web.

Pruebas unitarias: Realizar pruebas unitarias del código PHP y Arduino.

Pruebas de integración: Verificar que los diferentes componentes del sistema funcionen correctamente juntos.

Pruebas de usuario: Realizar pruebas con los usuarios finales para evaluar la usabilidad de la interfaz.

- **Análisis de datos:**

Estadística descriptiva: Calcular medidas como la media, mediana y desviación estándar de los niveles de inventario y los tiempos de respuesta a las alertas.

Análisis de varianza: Utilizar el análisis de varianza para identificar si existen diferencias significativas en los niveles de inventario entre diferentes productos o líneas de producción.

Visualización de datos: Utilizar gráficos y diagramas para identificar patrones y tendencias en los datos.

5.3. Análisis de requerimientos

Gestión de alertas de ANDON

- El sistema debe permitir generar alertas manuales desde las estaciones de trabajo en la planta.
- Las alertas deben clasificarse según el tipo de incidencia (falta de materia prima o mantenimiento programado).
- El sistema debe permitir registrar y visualizar alertas activadas automáticamente desde el dispositivo Arduino.

Visualización de estado en tiempo real

- El aplicativo web debe mostrar el estado de cada estación de trabajo en tiempo real.
- Las alertas deben visualizarse con códigos de color (por ejemplo, rojo para iniciar alerta, amarillo para resolver alerta y verde para verificación de correcto funcionamiento).
- Debe haber una visualización centralizada del estado general de la planta. Pantalla de monitoreo.

Notificaciones automáticas

- El sistema debe enviar notificaciones automáticas por correo electrónico cuando se active una alerta.

Integración con Arduino

- El sistema debe recibir señales de dispositivos Arduino conectados a estaciones de trabajo para activar o desactivar alertas.
- Panel de administración para la configuración de usuarios, roles y permisos.
- El sistema debe permitir agregar o eliminar estaciones de trabajo o líneas de producción desde el panel de administración.

Historial de alertas y reportes

- El sistema debe almacenar un historial de todas las alertas generadas, incluyendo detalles como la hora de activación, la estación de trabajo afectada, el tipo de alerta y el tiempo de resolución.
- El sistema debe generar reportes sobre los tiempos de cada alerta y los tiempos de respuesta.
- Los reportes deben poder exportarse en formato PDF o Excel.

Interfaz amigable y responsiva

- El sistema debe tener una interfaz intuitiva, accesible desde distintos dispositivos (PC, tablets, smartphones).
- Debe ser responsivo y adaptarse a diferentes tamaños de pantalla.
- Debe ofrecer una experiencia de usuario fácil, con un diseño minimalista y centrado en la usabilidad para el personal de planta.

Comunicación en red Ethernet

- El sistema debe garantizar una comunicación estable y segura a través de la red Ethernet entre los dispositivos Arduino y el servidor PHP.
- Debe existir chequeos de conectividad y avisos en caso de fallo de comunicación entre los dispositivos y el servidor.

Acceso basado en roles

- El sistema debe permitir el acceso a diferentes funcionalidades según el rol del usuario (monitoreo, administrador).
- Cada rol debe tener permisos específicos, por ejemplo, el rol monitoreo sólo visualiza el estado de las máquinas. El rol de administrador tendrá acceso a todo el sistema administrativo del aplicativo y pueden generar reportes.

5.3.1. Requerimiento Funcionales

La Tabla 1, describe los requerimientos necesarios en los cuales se basa la construcción del Sistema ANDON.

Tabla 1.
Requerimientos Funcionales

Código	Descripción	Categoría	Prioridad
RF01	El sistema permitirá a los operadores generar alertas manuales desde las estaciones de trabajo en la planta.	VISIBLE	ALTA
RF02	El sistema permitirá clasificar las alertas según el tipo de incidencia: falta de materia prima o mantenimiento programado.	VISIBLE	ALTA
RF03	El sistema permitirá registrar y mostrar alertas generadas automáticamente a partir de las señales recibidas desde dispositivos Arduino.	VISIBLE	ALTA
RF04	El sistema permitirá visualizar en tiempo real el estado de cada estación de trabajo a través de una interfaz web.	VISIBLE	ALTA
RF05	El sistema permitirá visualizar las alertas utilizando códigos de color: rojo (alerta activa), amarillo (en proceso) y verde (resuelta).	VISIBLE	MEDIA
RF06	El sistema permitirá mostrar una vista centralizada del estado general de la planta mediante una pantalla de monitoreo.	VISIBLE	ALTA
RF07	El sistema permitirá enviar notificaciones automáticas por correo electrónico al activarse una alerta.	VISIBLE	MEDIA
RF08	El sistema permitirá recibir y procesar señales de dispositivos Arduino conectados a las estaciones de trabajo para activar o desactivar alertas.	VISIBLE	MEDIA
RF09	El sistema permitirá configurar usuarios, roles y permisos a través de un panel de administración.	VISIBLE	MEDIA

RF10	El sistema permitirá agregar, editar o eliminar estaciones de trabajo o líneas de producción desde el panel de administración.	VISIBLE	MEDIA
RF11	El sistema permitirá almacenar un historial completo de las alertas generadas, incluyendo la hora de activación, estación afectada, tipo y tiempo de resolución.	VISIBLE	ALTA
RF12	El sistema permitirá generar reportes detallados sobre los tiempos de activación y resolución de las alertas.	VISIBLE	MEDIA
RF13	El sistema permitirá exportar los reportes generados en formatos PDF y Excel.	VISIBLE	MEDIA
RF14	El sistema permitirá garantizar una comunicación estable y segura a través de la red Ethernet entre los dispositivos Arduino y el servidor.	VISIBLE	ALTA
RF15	El sistema permitirá realizar chequeos periódicos de conectividad y generará avisos en caso de fallas en la comunicación con los dispositivos.	INFERIDA	MEDIO
RF16	El sistema permitirá gestionar el acceso a funcionalidades específicas según el rol del usuario, como monitoreo para supervisores y administración para administradores.	INFERIDA	MEDIA
RF17	El sistema permitirá asignar permisos específicos a cada rol: los supervisores solo visualizarán el estado de las estaciones, mientras que los administradores accederán a funcionalidades avanzadas.	VISIBLE	ALTA
RF18	El sistema permitirá a los administradores acceder al sistema completo para configurar, gestionar usuarios y generar reportes.	INFERIDA	MEDIA

5.3.2. Requerimientos No Funcionales

La Tabla 2, describe los requerimientos no funcionales características relevantes para la construcción del Sistema ANDON.

Tabla 2.
Requerimientos No Funcionales

Código	Categoría	Descripción	Prioridad
RNF01	Seguridad	El sistema permitirá el acceso mediante credenciales protegidas con contraseñas encriptadas utilizando algoritmos seguros y modernos como bcrypt o Argon2	ALTA
RNF02		El sistema cerrará automáticamente las sesiones de usuario tras 15 minutos de inactividad, solicitando nuevamente las credenciales de acceso para continuar.	ALTA
RNF03	Disponibilidad	La aplicación web tendrá disponibilidad continua, operando 24 horas al día, los 7 días de la semana (24/7), sin restricciones horarias para los usuarios.	ALTA
RNF04		El sistema permitirá el acceso desde cualquier ubicación con conexión a internet, garantizando la flexibilidad en el uso.	MEDIA
RNF05	Portabilidad	El sistema será compatible con múltiples plataformas operativas, incluyendo Linux , Windows y Mac , asegurando versatilidad mediante el uso de PHP y MySQL.	MEDIA
RNF06	Usabilidad	El sistema tendrá una interfaz intuitiva y amigable, accesible desde diversos dispositivos como PC, tabletas y teléfonos inteligentes.	MEDIA
RNF07		El sistema será completamente responsivo, adaptándose automáticamente a distintos tamaños de pantalla para garantizar una visualización óptima.	MEDIA
RNF08		El sistema ofrecerá una experiencia de usuario sencilla, con un diseño minimalista y enfocado en la usabilidad, facilitando su manejo al personal de planta.	MEDIA

RNF09		La aplicación web ofrecerá una experiencia de uso altamente intuitiva , simplificando la navegación y el manejo de las funcionalidades.	ALTA
RNF10	Costo	El desarrollo en PHP permitirá alojar el sistema en servidores Linux, reduciendo significativamente los costos de hosting en comparación con servidores basados en Microsoft.	ALTA
RNF11	Eficiencia	El sistema minimizará los errores potenciales al requerir una cantidad mínima de datos ingresados por parte del usuario.	ALTA
RNF12	Diseño Técnico	El diseño del sistema estará basado en tecnologías modernas que priorizan la automatización y simplificación de procesos a todos los niveles operativos.	ALTA

5.4. Casos de Uso

La Tabla 3, describe los casos de uso identificados que forman parte del Sistema ANDON. Incluye todos los requerimientos funcionales del documento, agrupando aquellos que son factibles de abordar en un mismo caso de uso:

Tabla 3.
Descripción de Casos de Uso

Actor	Funcionalidad (Caso de Uso)	Código del Caso de Uso	Requerimientos Funcionales Cubiertos
Operador	Generar alerta manual desde una estación de trabajo	CU01	RF01
Operador	Clasificar alertas por tipo (falta de materia prima o mantenimiento)	CU02	RF02
Sistema	Registrar y visualizar alertas automáticas desde Arduino	CU03	RF03, RF08
Supervisor	Monitorear el estado de cada estación de trabajo en tiempo real	CU04	RF04, RF06
Supervisor	Visualizar alertas con códigos de color	CU05	RF05

Sistema	Notificar automáticamente al activarse una alerta	CU06	RF07
Administrador	Configurar usuarios, roles y permisos en el sistema	CU07	RF09, RF16, RF17
Administrador	Agregar o eliminar estaciones de trabajo	CU08	RF10
Sistema	Monitorear la conectividad entre Arduino y el servidor	CU09	RF14, RF15
Sistema	Almacenar historial de alertas y detalles	CU10	RF11
Administrador/Supervisor	Generar reportes de tiempos de alerta y respuesta	CU11	RF12, RF13
Administrador	Acceder al sistema administrativo para generar reportes	CU12	RF18

Explicación de la agrupación

- CU03: Incluye tanto el registro de alertas automáticas (RF03) como la recepción de señales desde Arduino (RF08).
- CU04: Combina el monitoreo en tiempo real (RF04) y la visualización centralizada del estado general (RF06).
- CU07: Agrupa la configuración de usuarios, roles y permisos (RF09) junto con el acceso diferenciado basado en roles (RF16 y RF17).
- CU09: Cubre la conectividad estable y los chequeos de fallos de comunicación (RF14 y RF15).
- CU11: Abarca la generación de reportes (RF12) y su exportación en formato PDF o Excel (RF13).

Con esta agrupación, se aseguran que todos los requerimientos funcionales del documento están cubiertos en casos de uso relevantes y organizados para simplificar el diseño del sistema

5.4.1. Caso de Uso 1: Generar Alerta Manual

La Tabla 4, describe el caso de uso: generar alerta manual.

Tabla 4.

Caso de Uso 1: Generar Alerta Manual

Campo	Detalle
Caso de Uso:	Generar Alerta Manual
Actor(es):	Operador
Código:	CU01
Pantalla(s):	Pantalla de Generación de Alertas
Requerimiento(s):	RF01
Resumen / Descripción:	Permite a un operador generar una alerta manual desde su estación de trabajo.
Objetivos:	Garantizar que las incidencias sean notificadas oportunamente para una respuesta inmediata.
Precondiciones:	El operador debe estar autenticado en el sistema.
Postcondiciones:	La alerta se registra en el sistema y se visualiza en la pantalla de monitoreo. 1. El operador selecciona la opción de generar alerta.
Curso normal de eventos:	2. Especifica el tipo de incidencia. 3. Envía la alerta, que queda registrada en el sistema.
Curso alternativo de eventos:	1. Si el operador no completa los campos requeridos, el sistema muestra un mensaje de error.

5.4.2. Caso de Uso 2: Clasificar Alertas

La Tabla 5, describe el caso de uso: clasificar alertas.

Tabla 5.

Caso de Uso 2: Clasificar Alertas

Campo	Detalle
Caso de Uso:	Clasificar Alertas
Actor(es):	Operador, Sistema
Código:	CU02
Pantalla(s):	Pantalla de Clasificación de Alertas
Requerimiento(s):	RF02
Resumen / Descripción:	Clasifica las alertas generadas por tipo (falta de materia prima o mantenimiento programado).
Objetivos:	Facilitar el seguimiento y priorización de alertas según su naturaleza.

Precondiciones:	Debe existir una alerta generada en el sistema.
Postcondiciones:	La alerta queda clasificada en el sistema.
Curso normal de eventos:	1. El operador selecciona una alerta. 2. Clasifica la alerta según el tipo de incidencia.
Curso alterno de eventos:	1. Si no se selecciona un tipo, el sistema solicita completar la clasificación.

5.4.3. Caso de Uso 3: Registrar Alertas Automáticas

La Tabla 6, describe el caso de uso: registrar alertas automáticas.

Tabla 6.
Caso de Uso 3: Registrar Alertas Automáticas

Campo	Detalle
Caso de Uso:	Registrar Alertas Automáticas
Actor(es):	Sistema
Código:	CU03
Pantalla(s):	Pantalla de Visualización de Alertas
Requerimiento(s):	RF03, RF08
Resumen / Descripción:	Registra alertas automáticas enviadas desde dispositivos Arduino conectados a las estaciones.
Objetivos:	Automatizar el registro de incidentes para una respuesta más rápida y precisa.
Precondiciones:	Los dispositivos Arduino deben estar correctamente conectados a la red.
Postcondiciones:	La alerta automática queda registrada en el sistema y visible en tiempo real.
Curso normal de eventos:	1. El Arduino detecta un evento y envía la señal al servidor. 2. El sistema registra la alerta automáticamente.
Curso alterno de eventos:	1. Si hay pérdida de conectividad con el Arduino, el sistema registra un error en el historial.

5.4.4. Caso de Uso 4: Monitorear Estado en Tiempo Real

La Tabla 7, describe el caso de uso: monitorear estado en tiempo real.

Tabla 7.
Caso de Uso 4: Monitorear Estado en Tiempo Real

Campo	Detalle
Caso de Uso:	Monitorear Estado en Tiempo Real
Actor(es):	Supervisor
Código:	CU04

Pantalla(s):	Pantalla de Monitoreo
Requerimiento(s):	RF04, RF05, RF06
Resumen / Descripción:	Permite visualizar el estado de cada estación de trabajo en tiempo real mediante códigos de color.
Objetivos:	Proveer información actualizada para facilitar la supervisión de las estaciones de trabajo.
Precondiciones:	El sistema debe estar en funcionamiento y conectado a todas las estaciones.
Postcondiciones:	El supervisor tiene acceso a la información del estado actual de la planta.
Curso normal de eventos:	1. El supervisor accede a la pantalla de monitoreo. 2. Visualiza el estado de todas las estaciones.
Curso alterno de eventos:	1. Si una estación no responde, el sistema muestra un estado de alerta por desconexión.

5.4.5. Caso de Uso 5: Enviar Notificaciones Automáticas

La Tabla 8, describe el caso de uso: enviar notificaciones automáticas.

Tabla 8.

Caso de Uso 5: Enviar Notificaciones Automáticas

Campo	Detalle
Caso de Uso:	Enviar Notificaciones Automáticas
Actor(es):	Sistema
Código:	CU05
Pantalla(s):	No aplica
Requerimiento(s):	RF07
Resumen / Descripción:	Envía notificaciones por correo electrónico al activarse una alerta en el sistema.
Objetivos:	Informar a los responsables de incidencias para garantizar una acción rápida.
Precondiciones:	Las alertas deben estar configuradas con correos electrónicos asociados.
Postcondiciones:	El destinatario recibe una notificación detallada de la alerta activada. 1. Se activa una alerta.
Curso normal de eventos:	2. El sistema identifica a los usuarios responsables. 3. Envía el correo electrónico con los detalles.
Curso alterno de eventos:	1. Si no hay correos configurados, el sistema genera un registro de error.

5.4.6. Caso de Uso 6: Configurar Usuarios, Roles y Permisos

La Tabla 9, describe el caso de uso: configurar usuarios, roles y permisos.

Tabla 9.

Caso de Uso 6: Configurar Usuarios, Roles y Permisos

Campo	Detalle
Caso de Uso:	Configurar Usuarios, Roles y Permisos
Actor(es):	Administrador
Código:	CU06
Pantalla(s):	Pantalla de Configuración de Usuarios
Requerimiento(s):	RF09, RF16, RF17
Resumen / Descripción:	Permite al administrador gestionar usuarios y asignar roles y permisos para el acceso al sistema.
Objetivos:	Controlar el acceso y las funcionalidades disponibles según los roles definidos.
Precondiciones:	El administrador debe estar autenticado.
Postcondiciones:	Los usuarios y roles quedan configurados correctamente en el sistema.
Curso normal de eventos:	<ol style="list-style-type: none"> 1. El administrador accede a la pantalla de configuración. 2. Agrega o edita usuarios. 3. Asigna roles y permisos específicos.
Curso alternativo de eventos:	<ol style="list-style-type: none"> 1. Si los datos ingresados son incompletos, el sistema muestra un mensaje de error.

5.4.7. Caso de Uso 7: Agregar o Eliminar Estaciones de Trabajo

La Tabla 10, describe el caso de uso: agregar o eliminar estaciones de trabajo.

Tabla 10.

Caso de Uso 7: Agregar o Eliminar Estaciones de Trabajo

Campo	Detalle
Caso de Uso:	Agregar o Eliminar Estaciones de Trabajo
Actor(es):	Administrador
Código:	CU07
Pantalla(s):	Pantalla de Gestión de Estaciones
Requerimiento(s):	RF10
Resumen / Descripción:	Permite al administrador gestionar las estaciones de trabajo, añadiendo o eliminando entradas.
Objetivos:	Garantizar que las estaciones configuradas reflejen la estructura actual de la planta.
Precondiciones:	El administrador debe estar autenticado.
Postcondiciones:	Las estaciones configuradas quedan registradas en el sistema.
Curso normal de eventos:	<ol style="list-style-type: none"> 1. El administrador accede al módulo de gestión de estaciones. 2. Agrega o elimina estaciones según sea necesario.

Curso alterno de 1. Si los datos ingresados son incorrectos, el sistema muestra un mensaje de error.
eventos:

5.4.8. Caso de Uso 8: Almacenar Historial de Alertas

La Tabla 11, describe el caso de uso: almacenar historial de alertas.

Tabla 11.

Caso de Uso 8: Almacenar Historial de Alertas

Campo	Detalle
Caso de Uso:	Almacenar Historial de Alertas
Actor(es):	Sistema
Código:	CU08
Pantalla(s):	Pantalla de Historial de Alertas
Requerimiento(s):	RF11
Resumen / Descripción:	Almacena todas las alertas generadas con detalles como hora de activación, tipo y tiempo de resolución.
Objetivos:	Proveer un registro detallado para análisis y seguimiento posterior.
Precondiciones:	El sistema debe estar funcionando y registrar alertas activas.
Postcondiciones:	El historial queda actualizado en la base de datos.
Curso normal de eventos:	1. Se activa una alerta. 2. El sistema almacena automáticamente los detalles en la base de datos.
Curso alterno de eventos:	1. Si la base de datos no está disponible, el sistema registra un fallo temporal.

5.4.9. Caso de Uso 9: Generar Reportes Exportables

La Tabla 12, describe el caso de uso: generar reportes exportables.

Tabla 12.

Caso de Uso 9: Generar Reportes Exportables

Campo	Detalle
Caso de Uso:	Generar Reportes Exportables
Actor(es):	Administrador, Supervisor
Código:	CU09
Pantalla(s):	Pantalla de Generación de Reportes
Requerimiento(s):	RF12, RF13

Resumen / Descripción:	Genera reportes de tiempos de respuesta y resolución, exportables en formatos PDF y Excel.
Objetivos:	Facilitar el análisis y la toma de decisiones mediante reportes claros y organizados.
Precondiciones:	Debe existir información suficiente en el historial de alertas.
Postcondiciones:	El reporte se genera y se pone a disposición del usuario.
Curso normal de eventos:	1. El usuario selecciona los criterios del reporte. 2. Genera el reporte y lo descarga en el formato deseado.
Curso alterno de eventos:	1. Si no hay datos disponibles, el sistema muestra un mensaje indicando la falta de información.

5.4.10. Caso de Uso 10: Monitorear Conectividad de Dispositivos

La Tabla 13, describe el caso de uso: monitorear conectividad de dispositivos.

Tabla 13.
Caso de Uso 10: Monitorear Conectividad de Dispositivos

Campo	Detalle
Caso de Uso:	Monitorear Conectividad de Dispositivos
Actor(es):	Sistema
Código:	CU10
Pantalla(s):	Pantalla de Monitoreo de Conectividad
Requerimiento(s):	RF14, RF15
Resumen / Descripción:	Supervisa la conexión entre los dispositivos Arduino y el servidor, generando alertas en caso de fallo.
Objetivos:	Garantizar la estabilidad y funcionalidad del sistema mediante chequeos constantes de conectividad.
Precondiciones:	Todos los dispositivos deben estar configurados y conectados.
Postcondiciones:	Los errores de conectividad quedan registrados en el sistema.
Curso normal de eventos:	1. El sistema realiza chequeos periódicos de conectividad. 2. Notifica el estado de cada dispositivo.
Curso alterno de eventos:	1. Si un dispositivo no responde, el sistema genera una alerta de desconexión.

5.5. Diagrama de Casos de Uso UML

La Figura 1 describe gráficamente el diagrama de los casos de uso identificados para el desarrollo del Sistema ANDON.

Actores:

- **Operador**
 - Relacionado con los casos de uso: "Generar Alerta Manual" y "Clasificar Alertas".
- **Supervisor**
 - Relacionado con: "Monitorear Estado en Tiempo Real", "Visualizar Alertas".
- **Administrador**
 - Relacionado con: "Configurar Usuarios y Roles", "Generar Reportes", "Administrar Estaciones".
- **Sistema** (Actor automático)
 - Relacionado con: "Registrar Alertas Automáticas", "Notificar Automáticamente", "Monitorear Conectividad".

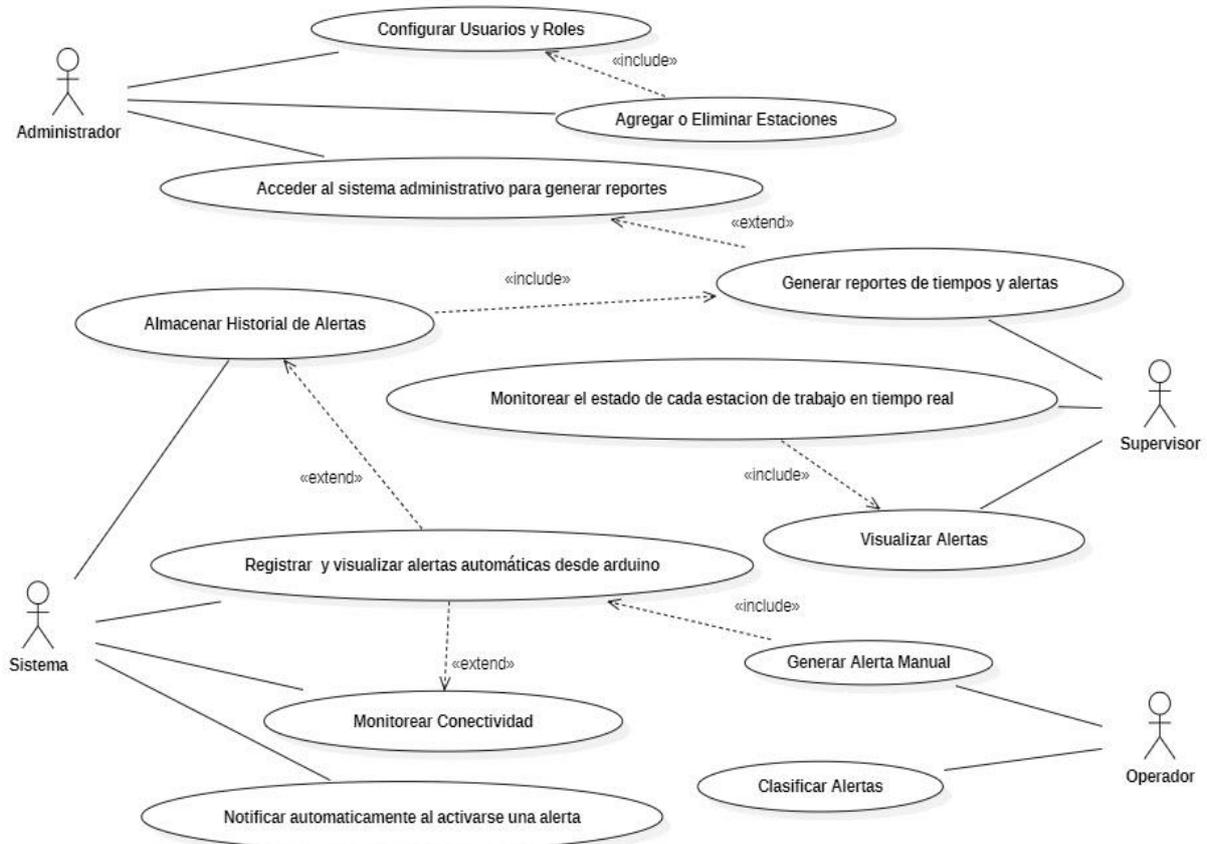


Figura 1. Diagrama de Casos de Uso UML

5.6. Diagramas de Secuencia

5.6.1. Diagrama de Secuencia Generar Alerta Manual

La Figura 2 describe gráficamente el diagrama de secuencia para generar las alertas manuales.

Actores:

Operador: Genera la alerta.

Participantes:

Sistema: Recibe la solicitud.

Base de Datos: Almacena la alerta generada.

Flujo:

1. El operador solicita generar una alerta manual.
2. El sistema valida la solicitud y envía la información a la base de datos para registrar la alerta.
3. La base de datos confirma el registro al sistema.
4. El sistema notifica al operador que la alerta fue generada con éxito.

Propósito:

Permitir que los operadores reporten manualmente incidencias en tiempo real.

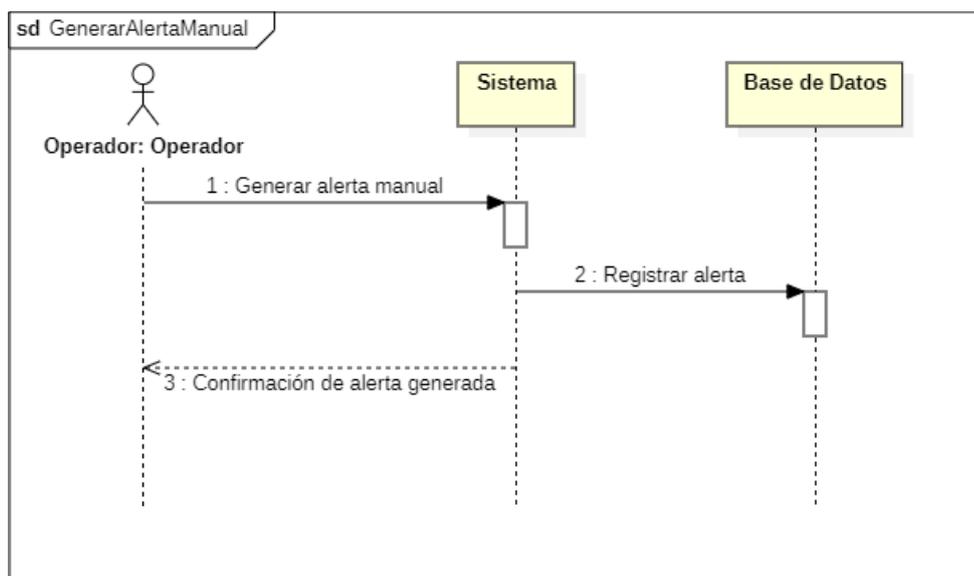


Figura 2. Generar Alerta Manual

5.6.2. Diagrama de Secuencia Clasificar Alertas

La Figura 3 describe gráficamente el diagrama de secuencia para clasificar las alertas que se producen en el Sistema ANDON.

Actores:

Operador: Clasifica alertas generadas previamente.

Participantes:

Sistema: Recibe y actualiza la información.

Base de Datos: Almacena la clasificación de la alerta.

Flujo:

1. El operador selecciona una alerta existente.
2. Clasifica la alerta en una categoría (e.g., falta de materia prima, mantenimiento).
3. El sistema actualiza los datos en la base de datos.
4. La base de datos confirma el cambio al sistema, que notifica al operador.

Propósito:

Organizar las alertas para facilitar el seguimiento y priorización de acciones.

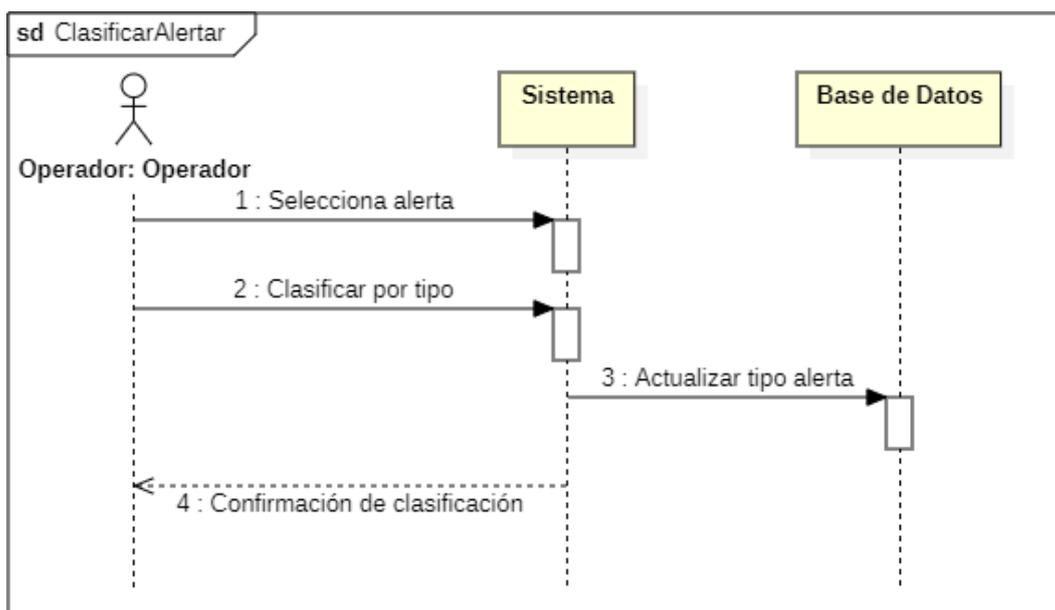


Figura 3. Clasificar Alerta

5.6.3. Diagrama de Secuencia Visualizar Alertas

La Figura 4, describe gráficamente el diagrama de secuencia para visualizar alertas.

Actores:

Supervisor: Accede a las alertas activas.

Participantes:

Sistema: Consulta y procesa la información.

Base de Datos: Contiene el registro de alertas activas.

Flujo:

1. El supervisor solicita ver las alertas activas.
2. El sistema consulta las alertas en la base de datos.
3. La base de datos devuelve la lista de alertas.
4. El sistema muestra las alertas al supervisor con códigos de color (e.g., rojo, amarillo, verde).

Propósito:

Permitir al supervisor priorizar y tomar decisiones basadas en las alertas activas.

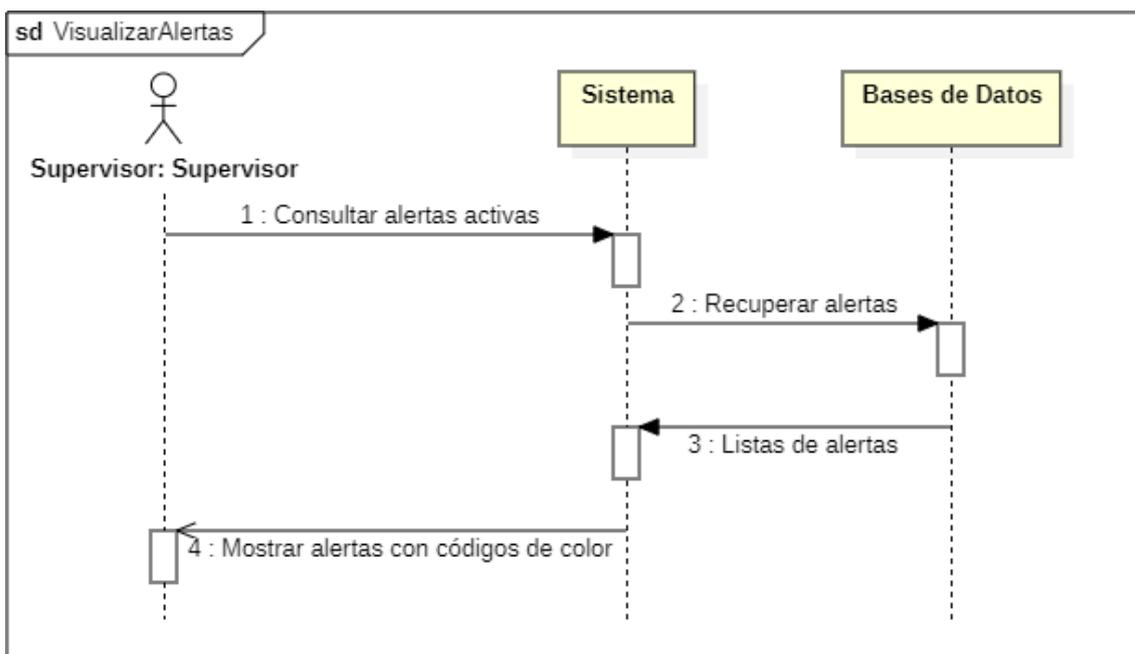


Figura 4. Visualizar alertas

5.6.4. Diagrama de Secuencia Monitorear Estado en Tiempo Real

La Figura 5, describe gráficamente el diagrama de secuencia para monitorear el estado del sistema ANDON en tiempo real.

Actores:

Supervisor: Consulta el estado actual de las estaciones.

Participantes:

Sistema: Procesa la solicitud.

Base de Datos: Proporciona la información requerida.

Flujo:

1. El supervisor solicita el estado de las estaciones.
2. El sistema consulta los datos en la base de datos.
3. La base de datos responde con el estado actualizado de todas las estaciones.
4. El sistema presenta la información en tiempo real al supervisor.

Propósito:

Visualizar el estado de todas las estaciones de trabajo para una supervisión eficiente.

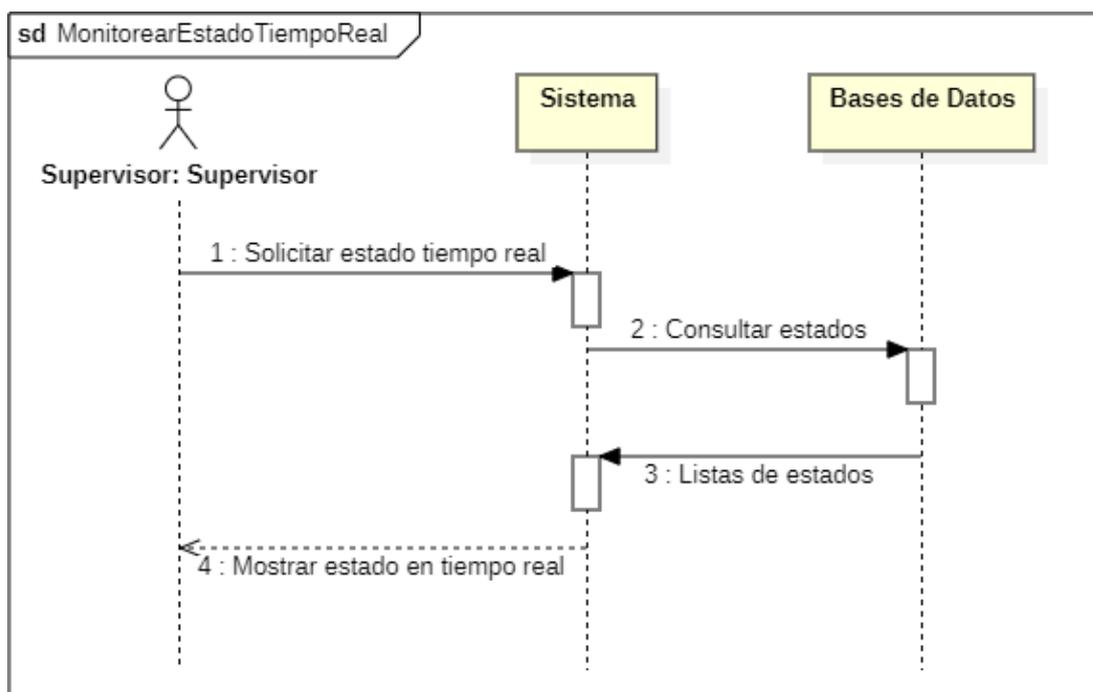


Figura 5. Monitorear estado en tiempo real

5.6.5. Diagrama de Secuencia Configurar Usuarios y Roles

La Figura 6 describe gráficamente el diagrama de secuencia para configurar usuarios y roles.

Actores:

Administrador: Administra usuarios y roles.

Participantes:

Sistema: Procesa las configuraciones.

Base de Datos: Almacena las configuraciones actualizadas.

Flujo:

1. El administrador agrega o modifica un usuario.
2. Asigna roles y permisos específicos.
3. El sistema guarda los cambios en la base de datos.
4. La base de datos confirma los cambios.

Propósito:

Garantizar que cada usuario tenga acceso a las funcionalidades según su rol.

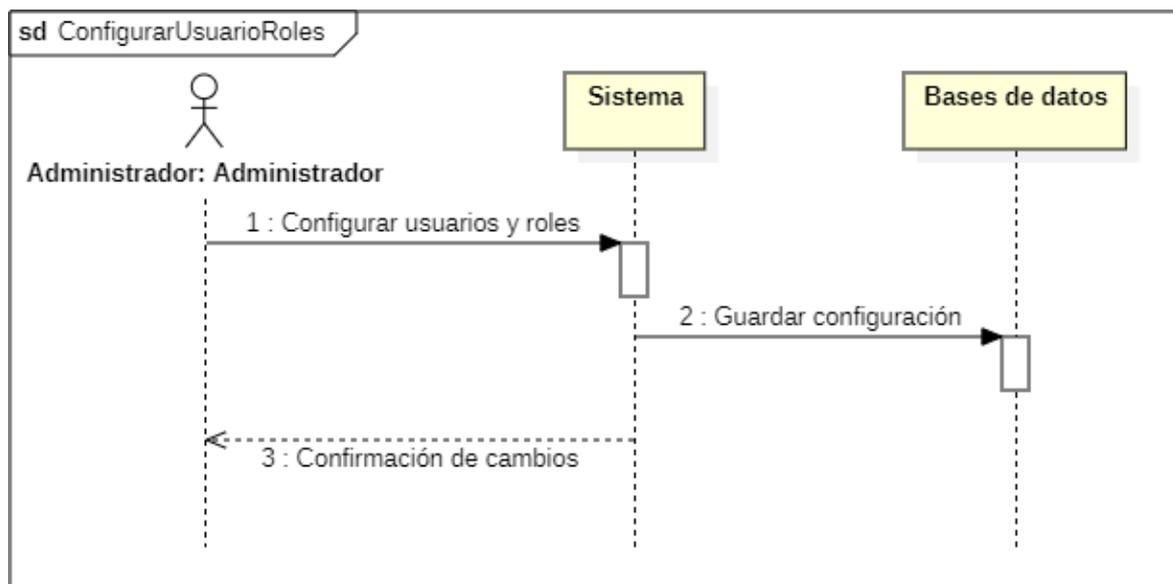


Figura 6. Configurar Usuario Roles

5.6.6. Diagrama de Secuencia Agregar o Eliminar Estaciones

La Figura 7, describe gráficamente el diagrama de secuencia para agregar o eliminar estaciones.

Actores:

Administrador: Gestiona estaciones de trabajo.

Participantes:

Sistema: Procesa las solicitudes.

Base de Datos: Actualiza las estaciones registradas.

Flujo:

1. El administrador solicita agregar una nueva estación.
2. El sistema guarda la nueva estación en la base de datos.
3. El administrador solicita eliminar una estación existente.
4. El sistema elimina la estación de la base de datos.

Propósito:

Mantener actualizada la lista de estaciones de trabajo en el sistema.

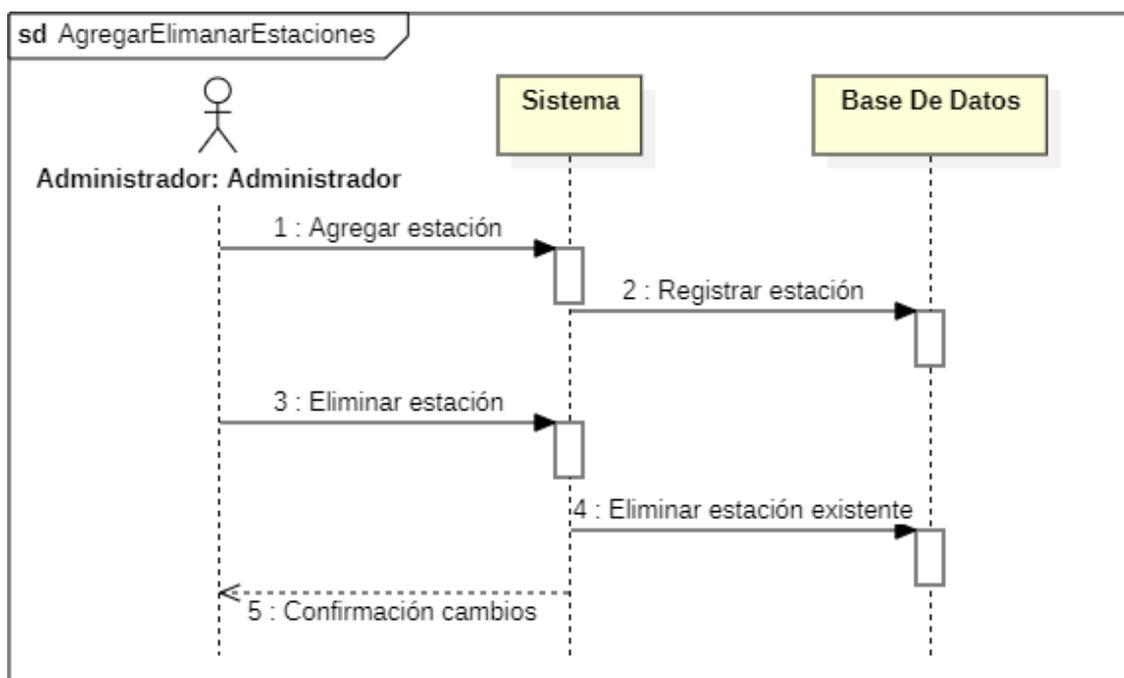


Figura 7. Agregar eliminar estaciones

5.6.7. Diagrama de Secuencia Generar Reportes Exportables

La Figura 8, describe gráficamente el diagrama de secuencia para generar los reportes exportables.

Actores:

Administrador: Solicita reportes.

Participantes:

Sistema: Genera el reporte.

Base de Datos: Proporciona los datos para el reporte.

Flujo:

1. El administrador solicita un reporte (PDF o Excel).
2. El sistema consulta los datos necesarios en la base de datos.
3. La base de datos devuelve los datos.
4. El sistema genera y entrega el reporte al administrador.

Propósito:

Facilitar el análisis de datos y el seguimiento de alertas mediante reportes claros.

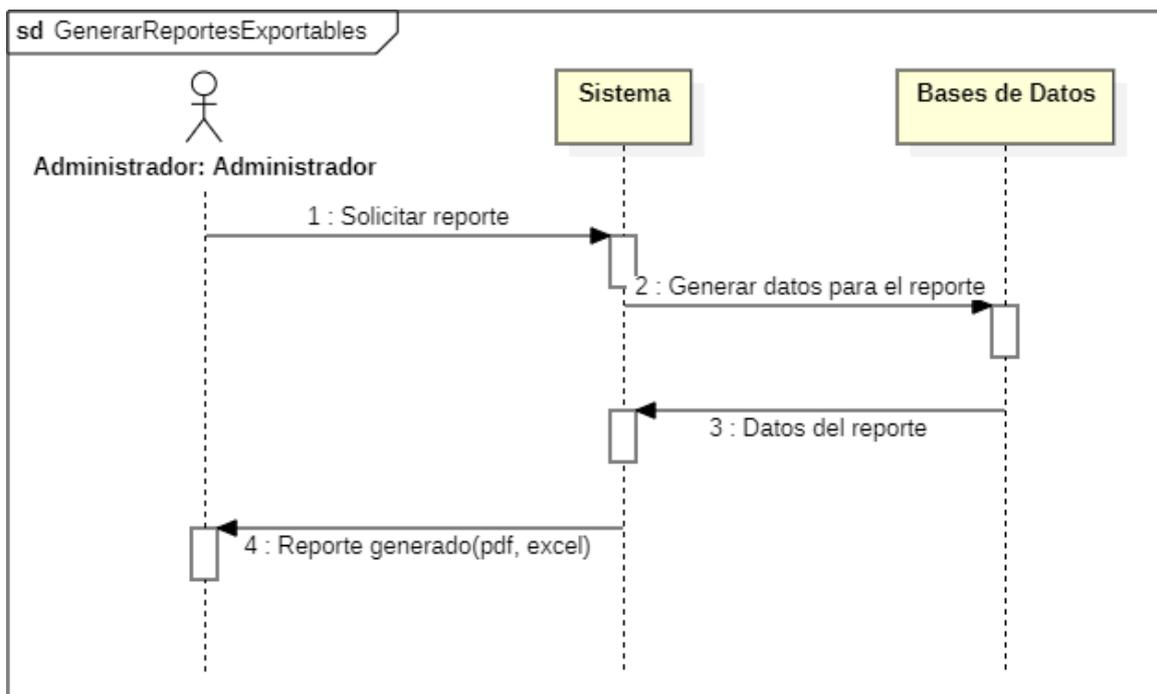


Figura 8. Generar Reportes Exportables

5.6.8. Diagrama de Secuencia Monitorear Conectividad

La Figura 9, describe gráficamente el diagrama de secuencia para el monitoreo de la conectividad.

Actores:

Ninguno (proceso automático).

Participantes:

Arduino: Proporciona el estado de conectividad.

Sistema: Verifica y registra el estado.

Flujo:

1. El sistema verifica la conectividad con los dispositivos Arduino.
2. Los dispositivos Arduino responden con su estado.
3. Si hay fallos, el sistema registra el problema y envía una notificación.

Propósito:

Garantizar la estabilidad de la red y detectar problemas de conectividad.

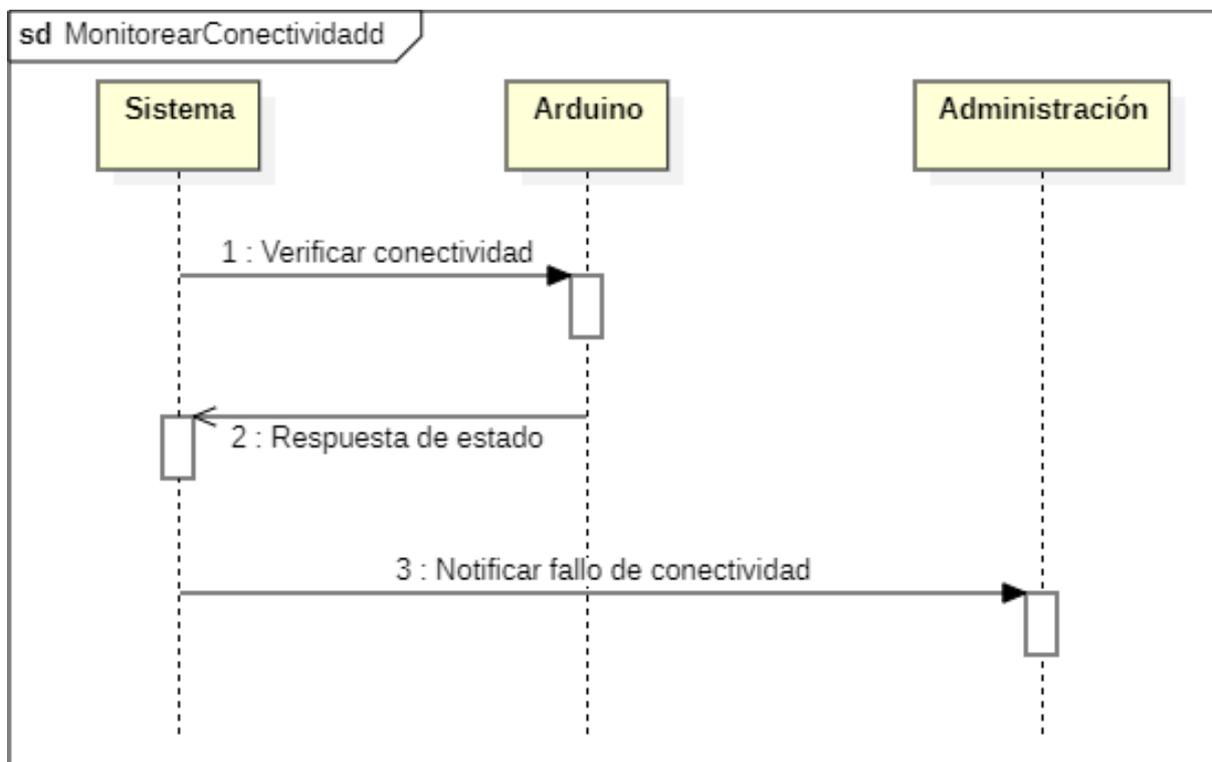


Figura 9. Monitorear Conectividad

5.6.9. Diagrama de Secuencia Almacenar Historial de Alertas

La Figura 10 describe gráficamente el diagrama de secuencia para almacenar el historial de alertas.

Actores:

Ninguno (proceso automático).

Participantes:

Sistema: Gestiona el almacenamiento.

Base de Datos: Guarda el historial de alertas.

Flujo:

1. Cuando se genera una alerta, el sistema guarda un registro en el historial de la base de datos.
2. La base de datos confirma el almacenamiento.

Propósito:

Tener un registro completo para análisis y auditorías futuras.

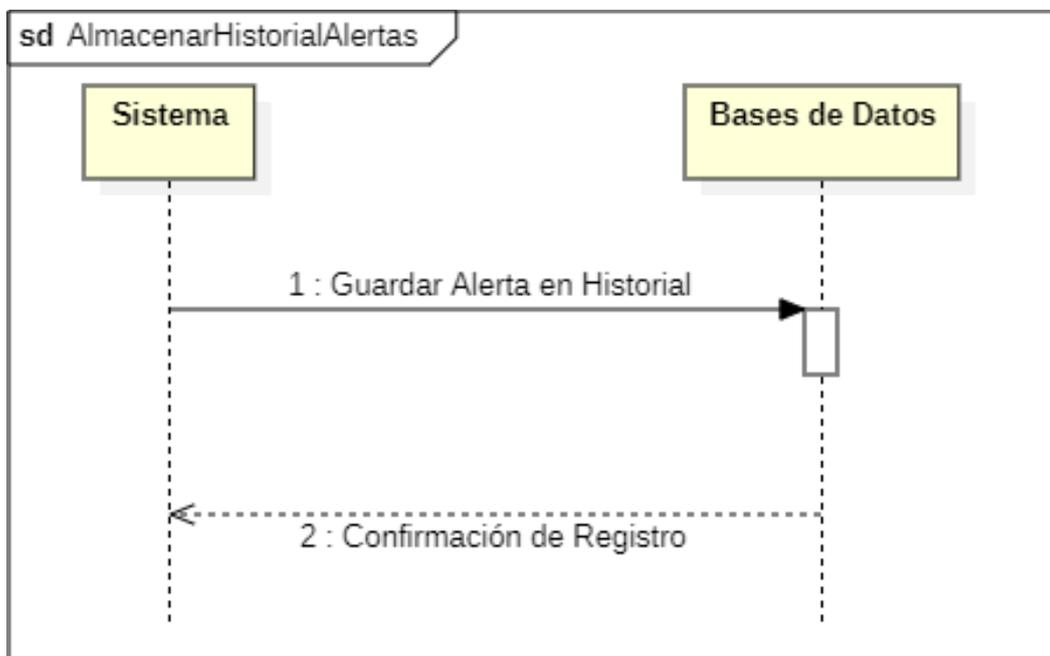


Figura 10. Almacenar historias alertas

5.6.10. Diagrama de Secuencia Registrar Alertas Automáticas

La Figura 11 describe gráficamente el diagrama de secuencia para el registro de alertas automáticas.

Actores:

Ninguno (es un proceso automático).

Participantes:

Arduino: Detecta una incidencia en la planta.

Sistema: Recibe y procesa la señal.

Base de Datos: Almacena la alerta generada.

Flujo:

1. El dispositivo Arduino detecta una incidencia y envía una señal al sistema.
2. El sistema recibe la señal, interpreta la información y registra la alerta en la base de datos.
3. La base de datos confirma el registro.

Propósito:

Automatizar la generación de alertas para una respuesta más rápida y precisa.

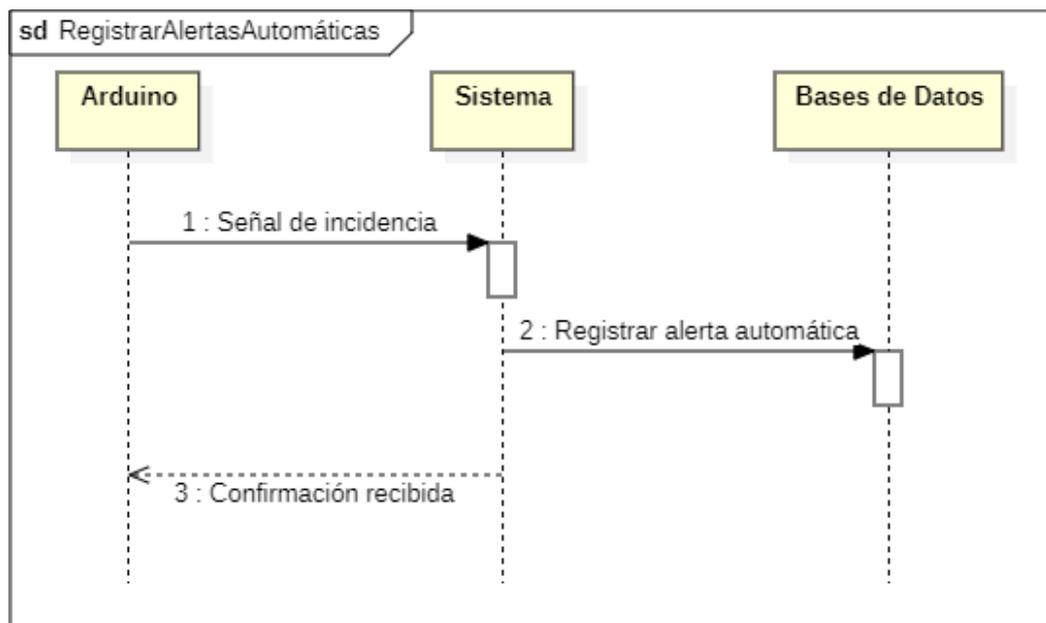


Figura 11. Registrar alertas automáticas

5.7. Diagrama de Clases

El diagrama de clases permite definir las relaciones entre las clases principales del sistema y generar una imagen gráfica del diagrama. La Figura 12 describe gráficamente un diagrama de clases inicial en el cual se modela lo necesario para utilizar en el desarrollo del aplicativo del Sistema ANDON.

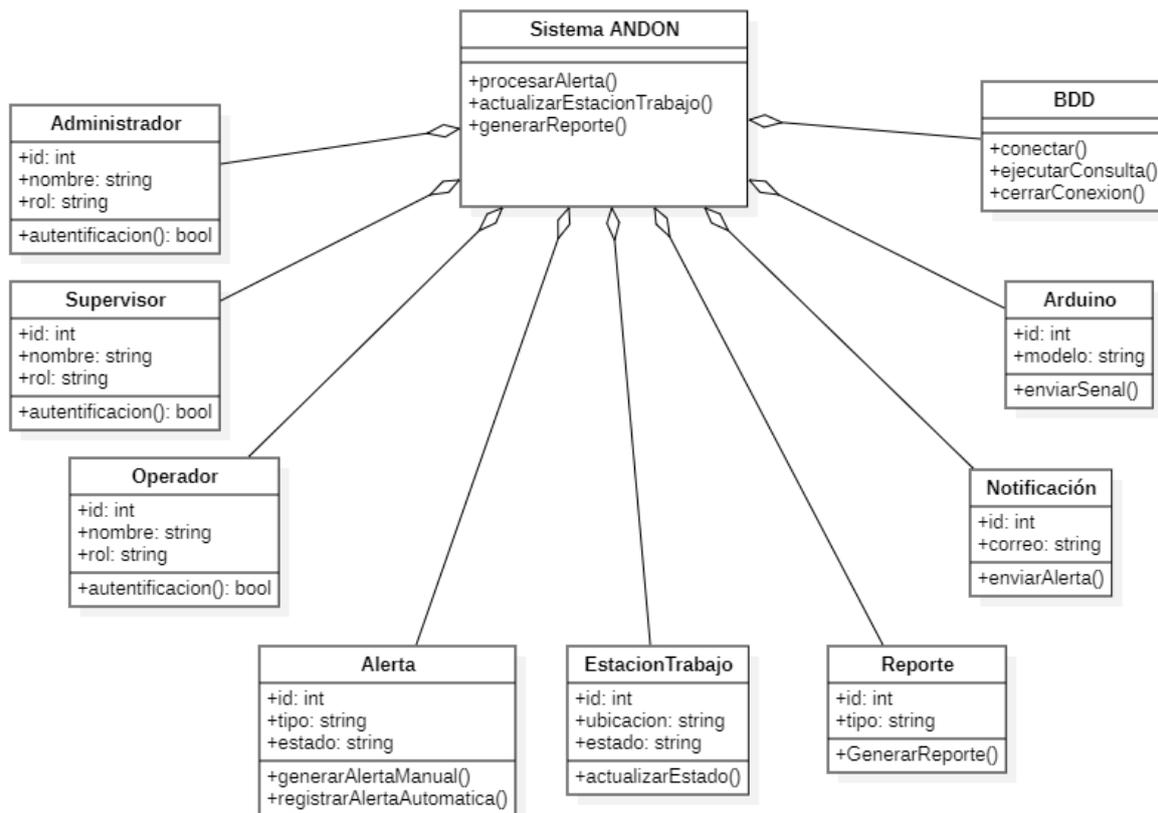


Figura 12. Diagrama de clases

5.7.1. Ajuste Diagrama de Clases

El ajuste de un diagrama de clases busca simplificar y buscar la relación de las clases e identificación de como el modelo de clases puede mejorar. En la Figura 13, representa el diagrama de clases actualizado, teniendo en cuenta los casos de uso y los diagramas de secuencia.

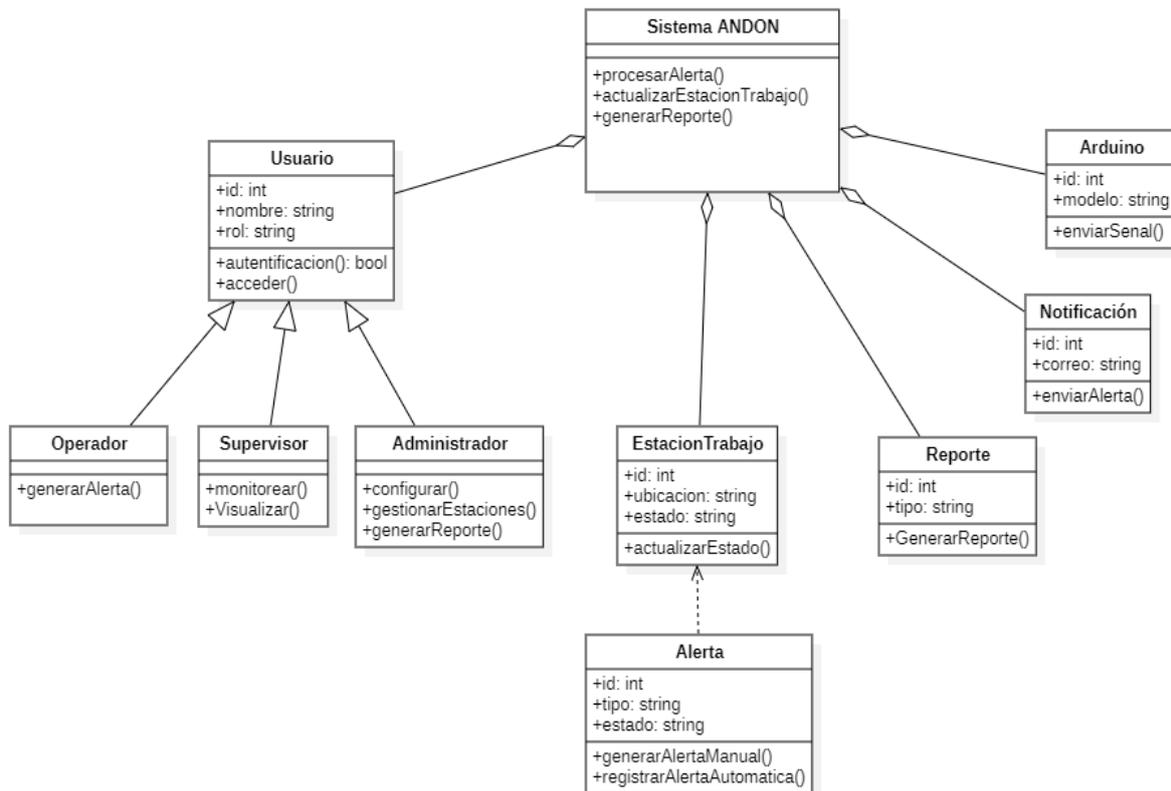


Figura 13. Ajuste Diagrama de Clases

5.8. Diagrama de Componentes

El diagrama de arquitectura muestra los componentes principales del sistema y sus interacciones. Este diseño es esencial para comprender cómo se comunican las diferentes partes del sistema y cómo se integran para cumplir con los objetivos del aplicativo ANDON. La Figura 14, representa el diagrama de componentes del sistema.

5.8.1. Definición de los componentes clave (Nodos)

- **Frontend:** Interfaz web accesible desde navegadores.
- **Backend:** Módulo PHP para lógica de negocio y manejo de API REST.
- **Base de Datos:** Sistema MySQL para almacenar datos de usuarios, alertas, etc.
- **Dispositivo Arduino:** Elemento físico conectado a través de Ethernet.
- **Correo Electrónico:** Servicio de notificaciones automáticas.
- **Usuarios Finales:** Actores que interactúan con el sistema (Operador, Supervisor, Administrador).

5.8.1.1. Nodo Usuario Final

- **Actores:** Operador, Supervisor, Administrador.
- **Descripción:** Los usuarios finales acceden al sistema desde un navegador web.
 - **Operador:** Genera alertas manuales desde estaciones de trabajo.
 - **Supervisor:** Monitorea el estado de las estaciones y las alertas.
 - **Administrador:** Configura usuarios, roles, y administra estaciones.

5.8.1.2. Nodo Frontend

- **Tecnologías utilizadas:** HTML, CSS, JavaScript (con Bootstrap para diseño responsivo).
- **Descripción:** Es la interfaz web del sistema accesible desde navegadores.
- **Funcionalidades:**
 - Generar alertas manuales.
 - Monitorear estado en tiempo real con códigos de color.
 - Consultar reportes y gráficos.

5.8.1.3. Nodo Backend (PHP)

- **Componentes internos:**
 - **Controlador:** Recibe las solicitudes del frontend y las procesa.
 - **Lógica de Negocio:** Define cómo se manejan las alertas, roles, usuarios, y reportes.
 - **API REST:** Permite la comunicación entre el frontend, el backend y los dispositivos Arduino.
- **Descripción:** Es el núcleo del sistema, encargado de manejar las solicitudes de los usuarios, procesar datos y coordinar la comunicación con la base de datos.

5.8.1.4. Nodo Base de Datos (MySQL)

- **BDD:** Tiene el esquema necesario para guardar la información necesaria para la utilización del aplicativo.
- **Descripción:** Almacena todos los datos críticos del sistema, incluyendo configuraciones, historial de alertas y estadísticas para reportes.

5.8.1.5. Nodo Dispositivo Arduino

- **Hardware involucrado:**

- Arduino Uno: Unidad central del dispositivo.
- Módulo Ethernet: Conexión del dispositivo a la red para enviar datos al servidor.
- Módulo RFID RC522: Detecta y registra eventos desde las estaciones de trabajo.
- **Descripción:** Detecta incidentes en la planta y los envía automáticamente al backend a través de la red Ethernet.

5.8.1.6. Nodo Correo Electrónico (Alerta)

- **Descripción:** Un módulo automatizado que envía alertas y notificaciones por correo electrónico.
- **Funcionalidad principal:** Notificar al administrador o al supervisor cuando se genera una alerta crítica.

5.8.1.7. Relaciones entre Componentes

- Usuario Final ↔ Frontend:
Los usuarios interactúan con el sistema a través del navegador web.
- Frontend ↔ Backend (PHP):
El frontend envía solicitudes al backend utilizando formularios y API REST.
- Backend ↔ Base de Datos:
El backend consulta y actualiza la información almacenada en MySQL.
- Dispositivo Arduino ↔ Backend:
Los dispositivos Arduino envían alertas automáticas al backend, que las procesa y registra.
- Backend ↔ Correo Electrónico:
El backend utiliza el módulo de notificaciones para enviar alertas por correo electrónico.

El diseño de la arquitectura modular permite:

- **Escalabilidad:** Agregar nuevas funcionalidades o estaciones de trabajo sin modificar significativamente el sistema.
- **Fiabilidad:** Asegura que las alertas sean procesadas y notificadas en tiempo real.
- **Mantenibilidad:** La separación clara entre frontend, backend, y base de datos facilita la actualización y el mantenimiento.

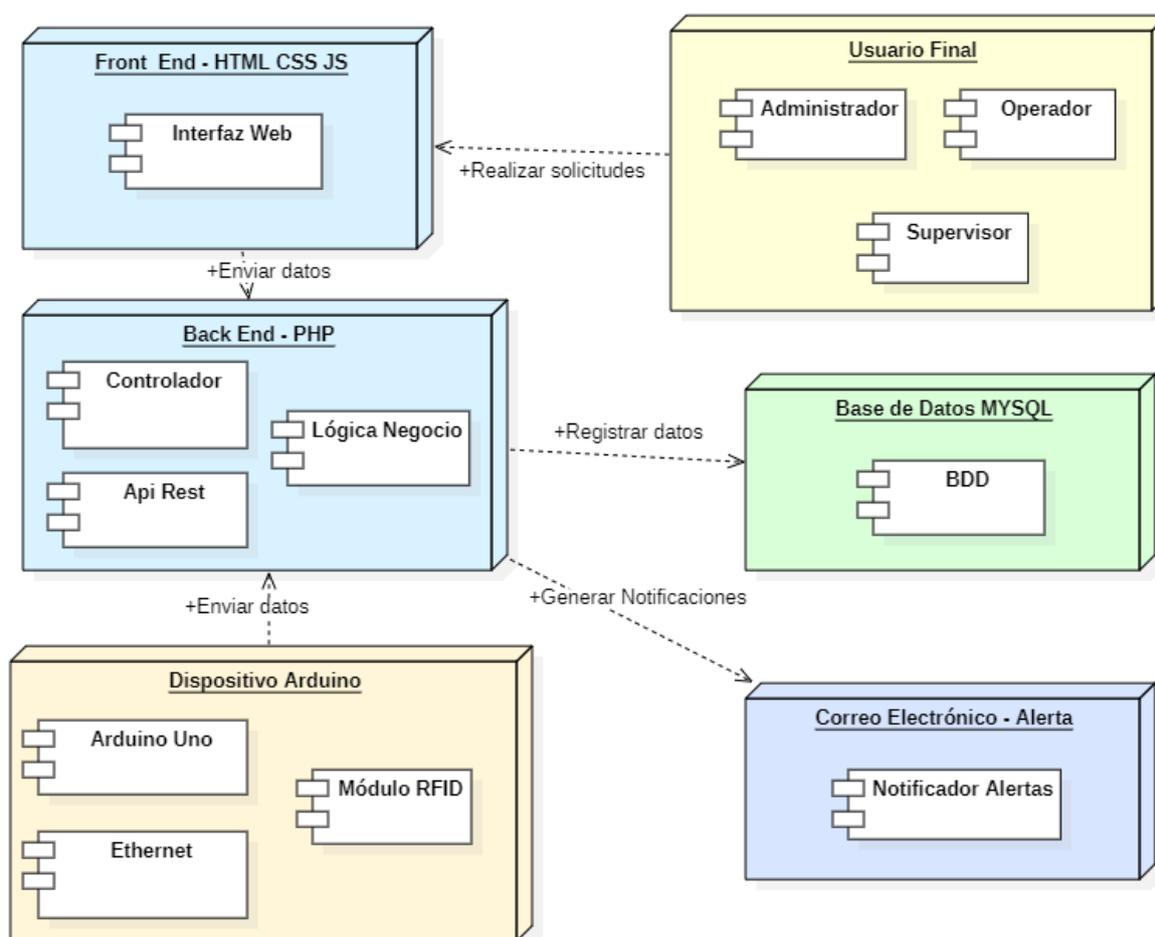


Figura 14. Diagrama de componentes

5.9. Diseño de la base de datos

El diseño de una base de datos es un proceso fundamental en el desarrollo de sistemas de información, ya que define cómo se estructurarán, almacenarán y gestionarán los datos necesarios para el funcionamiento del sistema. Un diseño adecuado asegura que la información sea accesible, consistente y segura, permitiendo optimizar el rendimiento y facilitar futuras expansiones.

En este contexto, el diseño de la base de datos implica:

- **Análisis de Requerimientos:** Identificar qué datos son necesarios, cómo se relacionan y cómo se utilizarán en el sistema.
- **Modelo Conceptual:** Crear diagramas, como el modelo entidad-relación (ER), para representar gráficamente las entidades, sus atributos y las relaciones entre ellas.
- **Modelo Lógico:** Transformar el modelo conceptual en un esquema lógico, compatible con el sistema de gestión de bases de datos (DBMS) seleccionado.
- **Modelo Físico:** Definir detalles técnicos, como tipos de datos, claves primarias y foráneas, índices y normalización, para garantizar eficiencia y evitar redundancias.

La Figura 15, representa el modelo de la base de datos para el sistema ANDON, un diseño robusto no solo soporta las operaciones diarias del sistema, sino que también facilita un análisis avanzado, integridad de los datos y una administración eficiente.

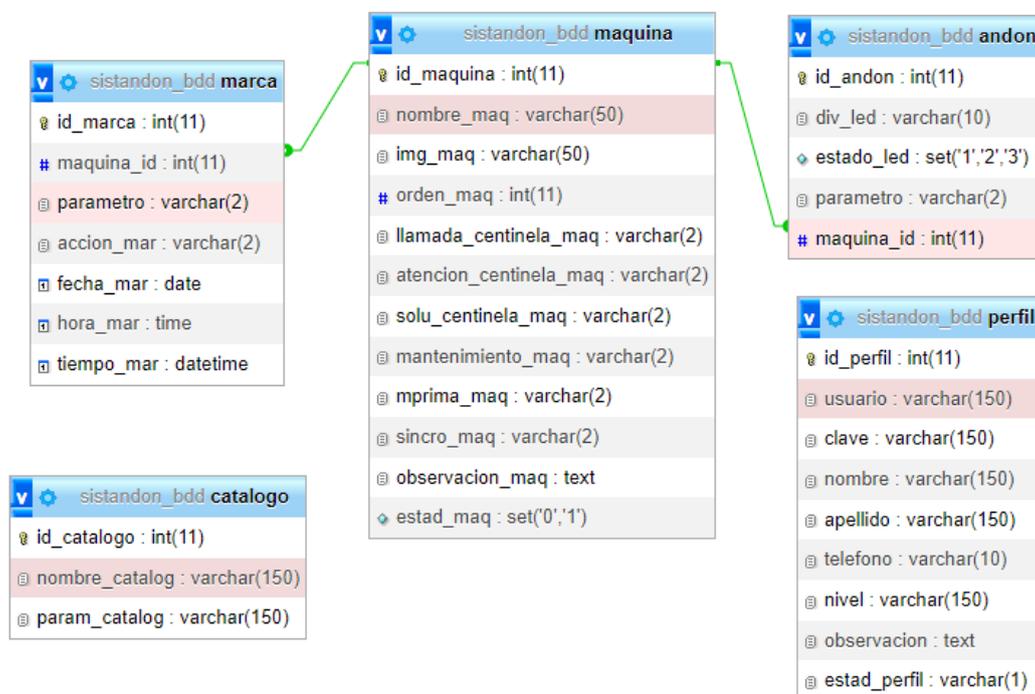


Figura 15. Modelo de la Base de Datos

5.10. Desarrollo del backend PHP

El desarrollo del backend en PHP es una parte esencial de la construcción de sistemas web, ya que gestiona la lógica de negocio, el procesamiento de datos y la comunicación entre el frontend y la base de datos. PHP, como lenguaje ampliamente utilizado en el desarrollo web, ofrece una solución flexible y eficiente para implementar funcionalidades en el servidor. En este contexto, el desarrollo del backend implica:

- **Lógica de Negocio:** Implementar las reglas y procesos que controlan el flujo de datos y las operaciones del sistema.
- **Gestión de Bases de Datos:** Conectar y realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en la base de datos para almacenar y recuperar información.
- **Creación de APIs:** Diseñar servicios RESTful que permitan la comunicación entre el frontend y el backend, asegurando interoperabilidad.
- **Seguridad:** Proteger la información sensible mediante autenticación, validación de datos y cifrado.
- **Escalabilidad:** Estructurar el código para facilitar la ampliación de funcionalidades y el mantenimiento futuro.

El backend desarrollado en PHP actúa como el núcleo del sistema, asegurando que las solicitudes de los usuarios se procesen de manera eficiente, segura y confiable, proporcionando una base sólida para el funcionamiento integral del sistema.

La Figura 16, es una muestra gráfica de un fragmento de código en lenguaje PHP parte del desarrollo.

5.11. Desarrollo de la interfaz web

El desarrollo de la interfaz web es un proceso clave en la creación de sistemas interactivos, ya que define cómo los usuarios finales interactuarán con las funcionalidades del sistema. Una interfaz web bien diseñada combina estética, usabilidad y funcionalidad para

ofrecer una experiencia fluida y eficiente, En este contexto, el desarrollo de la interfaz web implica:

```
389 }
390 function val_marcacion($id_maquina, $parametro, $accion) {
391     require('conexion_php');
392     $mjs = "no";
393     $sql_sincro_maq = "SELECT accion_mar, tiempo_mar, parametro FROM marca WHERE maquina_id="
394     $resp_sincro_maq = mysqli_query($conexion_bd, $sql_sincro_maq);
395     $tot_sincro_maq = mysqli_num_rows($resp_sincro_maq);
396     if($tot_sincro_maq > 0) {
397         $ultima_accion = mysqli_result($resp_sincro_maq, 0, 0);
398         $tiempo_accion = mysqli_result($resp_sincro_maq, 0, 1);
399         $parametro_accion = mysqli_result($resp_sincro_maq, 0, 2);
400     } else {
401         $ultima_accion = "";
402         $tiempo_accion = "";
403         $parametro_accion = "";
404     }
405     $indice_ultima_accion = substr($ultima_accion, 1, 1);
406     $indice_accion = substr($accion, 1, 1);
407
408     if($indice_ultima_accion == 0) {
409         $verificacion = 1;
410         if($indice_accion == $verificacion) {
411             $mjs = "si";
412         } else {
413             $mjs = "no";
414         }
415     } else if($indice_ultima_accion == 1) {
416         $verificacion = 2;
417         if($indice_accion == $verificacion) {
418             $mjs = "si";
419         } else {
420             $mjs = "no";
421         }
422     }
423 }
```

Figura 16. Fragmento de código PHP

- **Diseño Visual:** Crear una disposición atractiva y coherente utilizando tecnologías como HTML, CSS y frameworks como Bootstrap, asegurando que sea visualmente agradable y profesional.
- **Interactividad:** Implementar funcionalidades dinámicas utilizando JavaScript y librerías como React o Vue.js, para mejorar la interacción del usuario con el sistema.
- **Responsividad:** Garantizar que la interfaz se adapte correctamente a diferentes dispositivos (PC, tabletas y móviles).
- **Accesibilidad:** Diseñar con estándares inclusivos que permitan a todas las personas, incluidas aquellas con discapacidades, utilizar el sistema.

El objetivo principal del desarrollo de la interfaz web es ofrecer a los usuarios una experiencia intuitiva y eficiente, alineada con las necesidades del sistema y el propósito del proyecto. Esto asegura que las operaciones sean simples de realizar y que los usuarios interactúen con confianza y comodidad.

La Figura 17 muestra la distribución de los elementos que se visualizan en el desarrollo de la interface web en el rol operador.



Figura 17. Interface para visualización de alertas

La Figura 18 describe gráficamente la interface del usuario supervisor o administrador.



Figura 18. Interface para visualizar las actividades del supervisor / administrador

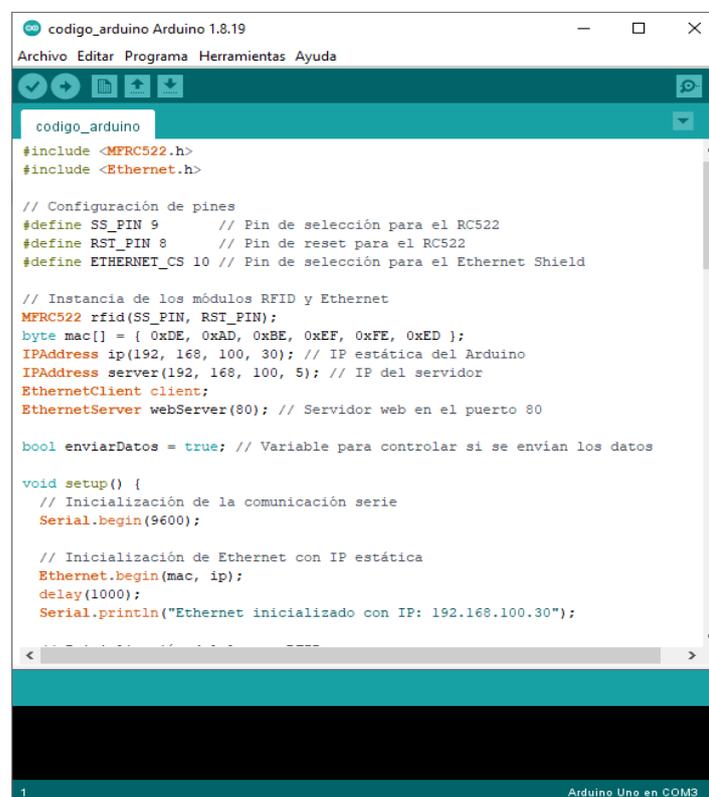
5.12. Implementación del Arduino

La implementación del Arduino en un sistema es una etapa crucial cuando se busca integrar hardware con software para crear soluciones automatizadas e interactivas. Arduino, como una plataforma de desarrollo electrónico de código abierto, permite controlar dispositivos

físicos, procesar señales y comunicarse con otros sistemas de manera eficiente, en este contexto, la implementación del Arduino implica:

- **Configuración de Hardware:** Conectar y configurar módulos como sensores, actuadores, Ethernet Shield o RFID para realizar tareas específicas.
- **Desarrollo de Firmware:** Programar el microcontrolador en lenguajes como C/C++ para gestionar las señales y ejecutar acciones automatizadas.
- **Comunicación con el Sistema:** Enviar y recibir datos entre el Arduino y el servidor o la interfaz web, utilizando protocolos como SPI, HTTP o MQTT.
- **Automatización de Procesos:** Detectar eventos (por ejemplo, incidencias en estaciones de trabajo) y activarlos automáticamente en el sistema.

La implementación del Arduino no solo permite una interacción directa con el entorno físico, sino que también mejora la eficiencia y precisión del sistema, logrando una integración efectiva entre el hardware y el software. La Figura 19 permite mostrar un fragmento del código para Arduino.



```
codigo_arduino Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

codigo_arduino
#include <MFRC522.h>
#include <Ethernet.h>

// Configuración de pines
#define SS_PIN 9 // Pin de selección para el RC522
#define RST_PIN 8 // Pin de reset para el RC522
#define ETHERNET_CS 10 // Pin de selección para el Ethernet Shield

// Instancia de los módulos RFID y Ethernet
MFRC522 rfid(SS_PIN, RST_PIN);
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 100, 30); // IP estática del Arduino
IPAddress server(192, 168, 100, 5); // IP del servidor
EthernetClient client;
EthernetServer webServer(80); // Servidor web en el puerto 80

bool enviarDatos = true; // Variable para controlar si se envían los datos

void setup() {
  // Inicialización de la comunicación serie
  Serial.begin(9600);

  // Inicialización de Ethernet con IP estática
  Ethernet.begin(mac, ip);
  delay(1000);
  Serial.println("Ethernet inicializado con IP: 192.168.100.30");
}
```

Figura 19. Fragmento de código para Arduino

El hardware Arduino es una plataforma de desarrollo electrónico ampliamente utilizada para proyectos de automatización, prototipos y soluciones IoT. Su diseño accesible, modularidad y soporte de código abierto. El ecosistema de hardware Arduino incluye:

- **Placas Base:** Microcontroladores como Arduino Uno, Mega y Nano, que son el núcleo del sistema, capaces de procesar señales y ejecutar instrucciones.
- **Módulos de Expansión:** Shields y periféricos, como el Ethernet Shield o el módulo RFID RC522, que amplían las capacidades de conectividad y sensorización.
- **Conectividad:** Soporte para protocolos de comunicación como SPI, I2C, UART y Ethernet, permitiendo la interacción con otros dispositivos.
- **Compatibilidad:** Pines digitales y analógicos para conectar sensores, actuadores y módulos adicionales.

La Figura 20, muestra una placa base Arduino, un módulo RFID y un módulo wifi.

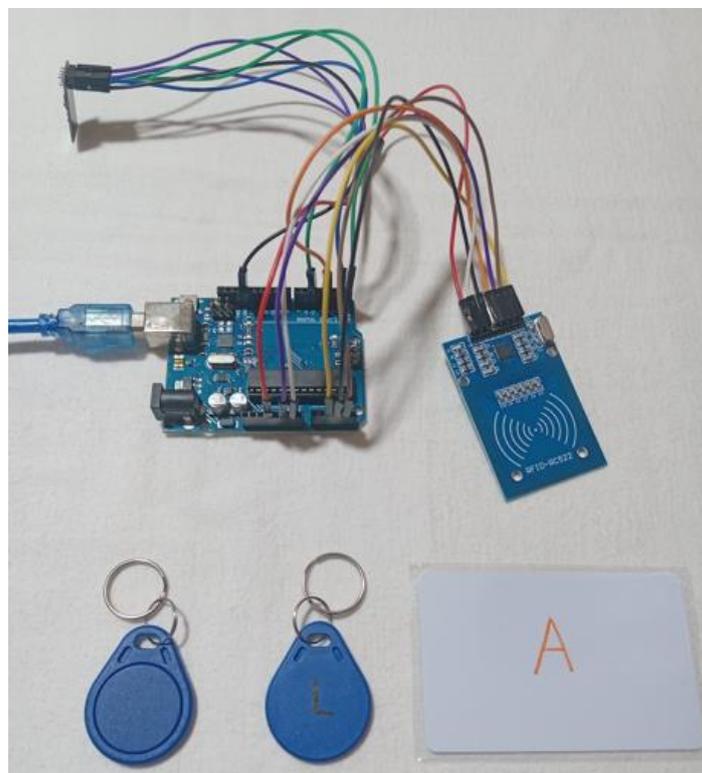


Figura 20. Software Arduino, placa base

5.13. Pruebas de integración

Las pruebas de integración son una etapa crucial en el desarrollo de software que busca verificar la interacción entre los diferentes módulos o componentes del sistema. Este proceso asegura que las partes del sistema trabajen juntas de manera correcta y sin conflictos, alineándose con los requisitos funcionales y el diseño arquitectónico.

En el contexto del sistema ANDON, las pruebas de integración se centran en:

- Verificar la comunicación entre el frontend y el backend.
- Asegurar la interacción correcta entre el backend y la base de datos.
- Probar la conexión entre el sistema y los dispositivos Arduino para alertas automáticas.
- Validar la generación de reportes y notificaciones.

Estas pruebas garantizan que los módulos funcionan como una unidad cohesiva, asegurando la calidad y confiabilidad del sistema.

5.13.1. Prueba: Generar Alerta Manual

La prueba de integración para la generación de alertas manuales se centra en verificar el flujo de datos desde la acción del operador en el frontend hasta el registro en la base de datos. Esta prueba es fundamental para garantizar que los operadores puedan reportar incidencias de manera eficiente y que el sistema registre los datos de forma precisa.

El operador debe recibir una confirmación de éxito en el frontend, y los detalles de la alerta deben estar correctamente registrados en la base de datos.

Descripción:

Verificar que un operador pueda generar una alerta manual desde el frontend y que esta se registre correctamente en la base de datos.

Flujo de Prueba:

1. El operador accede al sistema desde el frontend e ingresa los detalles de la alerta.
2. El backend recibe la solicitud y procesa los datos.
3. La alerta se almacena en la tabla correspondiente en la base de datos.
4. El sistema devuelve una confirmación al operador.

Criterios de Éxito:

- La alerta se registra con los datos correctos en la base de datos.
- El operador recibe una confirmación en el frontend.

5.13.2. Prueba: Registrar Alertas Automáticas

La prueba de registro de alertas automáticas tiene como objetivo validar la conexión entre el dispositivo Arduino y el backend del sistema. Este flujo es crucial para garantizar que los eventos detectados en las estaciones de trabajo se procesen automáticamente y se registren como alertas en el sistema.

El backend debe procesar correctamente las señales enviadas por el Arduino, generando alertas precisas que se almacenen en la base de datos sin intervención manual.

Descripción:

Probar la conexión entre el dispositivo Arduino y el backend para el registro de alertas automáticas.

Flujo de Prueba:

1. El Arduino detecta un evento y envía una señal al servidor.
2. El backend recibe la señal, interpreta los datos y genera una alerta.
3. La alerta se registra en la base de datos.

Criterios de Éxito:

- El backend recibe correctamente los datos del Arduino.
- La alerta generada es consistente con el evento detectado.

5.13.3. Prueba: Monitorear Estado en Tiempo Real

Esta prueba evalúa la capacidad del sistema para proporcionar información en tiempo real sobre el estado de las estaciones de trabajo. La funcionalidad es clave para que los supervisores puedan tomar decisiones rápidas y precisas basadas en datos actualizados directamente desde la base de datos.

El supervisor debe visualizar correctamente los estados de las estaciones en tiempo real, con códigos de color que reflejen la condición actual de cada una.

Descripción:

Validar que el supervisor pueda consultar el estado de las estaciones de trabajo en tiempo real.

Flujo de Prueba:

1. El supervisor solicita el estado actual desde el frontend.
2. El backend consulta la base de datos para recuperar los estados de las estaciones.
3. El sistema muestra la información actualizada al supervisor con códigos de color.

Criterios de Éxito:

- Los estados mostrados coinciden con los valores almacenados en la base de datos.
- Los códigos de color representan correctamente los estados (rojo, amarillo, verde).

5.13.4. Prueba: Generar Reportes Exportables

La prueba de generación de reportes asegura que el sistema pueda consolidar datos históricos y procesarlos en formatos exportables (PDF y Excel). Esto es fundamental para proporcionar a los administradores herramientas de análisis y seguimiento.

El sistema debe generar reportes completos y exactos en los formatos solicitados, disponibles para descarga sin errores.

Descripción:

Asegurar que el administrador pueda generar reportes en formatos PDF y Excel.

Flujo de Prueba:

1. El administrador selecciona los criterios del reporte en el frontend.
2. El backend procesa los datos consultando la base de datos.
3. El reporte se genera y se pone a disposición para descarga en el formato seleccionado.

Criterios de Éxito:

- El reporte incluye los datos correctos y está disponible en los formatos PDF y Excel.
- La descarga funciona sin errores.

5.13.5. Prueba: Notificaciones Automáticas

Esta prueba valida la capacidad del sistema para enviar notificaciones automáticas por correo electrónico cuando se activa una alerta. La funcionalidad es esencial para garantizar que los usuarios responsables sean informados de manera inmediata, permitiendo una acción oportuna.

Los correos electrónicos deben enviarse automáticamente con información detallada sobre la alerta y deben llegar correctamente a todos los destinatarios configurados.

Descripción:

Probar que el sistema envíe notificaciones automáticas por correo al activarse una alerta.

Flujo de Prueba:

1. Se genera una alerta (manual o automática).
2. El backend envía una notificación a los correos asociados al tipo de alerta.
3. Los destinatarios reciben el correo con los detalles de la alerta.

Criterios de Éxito:

- El correo contiene información precisa sobre la alerta.

- Todos los destinatarios reciben la notificación.

5.13.6. Prueba: Configuración de Usuarios y Roles

La prueba de configuración de usuarios y roles verifica la capacidad del administrador para gestionar el acceso y los permisos del sistema. Este proceso es clave para garantizar que solo usuarios autorizados accedan a las funciones asignadas según su rol.

Los cambios realizados por el administrador deben reflejarse correctamente en la base de datos, y los usuarios deben poder acceder al sistema con los permisos definidos.

Descripción:

Validar que el administrador pueda agregar, modificar y eliminar usuarios con diferentes roles.

Flujo de Prueba:

1. El administrador accede al módulo de configuración de usuarios.
2. Agrega, edita o elimina un usuario.
3. Los cambios se reflejan en la base de datos.

Criterios de Éxito:

- Los cambios en la base de datos coinciden con las acciones realizadas.
- Los usuarios pueden acceder al sistema con los roles y permisos asignados

5.14. Implementación y pruebas en campo

La implementación y pruebas en campo representan la etapa final del desarrollo de un sistema antes de su puesta en operación. En este proceso, el sistema se despliega en el entorno real donde será utilizado y se realizan pruebas exhaustivas para asegurar su correcto funcionamiento en condiciones reales de operación.

El objetivo principal de esta etapa es validar que:

- El sistema funcione conforme a los requerimientos establecidos.
- Todos los componentes (hardware, software y red) trabajen de manera integrada.
- Los usuarios finales puedan operar el sistema sin problemas.

Esta etapa incluye la configuración final, pruebas de aceptación y ajustes necesarios, asegurando que el sistema esté completamente listo para su operación diaria

5.14.1. Prueba de Conectividad

La prueba de conectividad tiene como objetivo garantizar que todos los componentes del sistema (dispositivos Arduino, backend y base de datos) estén conectados correctamente y puedan comunicarse sin interrupciones en el entorno real. La conectividad es esencial para el correcto funcionamiento del sistema ANDON, ya que asegura la integración fluida de hardware y software. Todos los componentes deben responder sin fallos en la comunicación, y el backend debe registrar correctamente los datos provenientes de los dispositivos Arduino en la base de datos. La Figura 21, muestra el proceso para prueba de conectividad.

Descripción:

Verifica que todos los componentes del sistema (Arduino, frontend, backend y base de datos) estén conectados correctamente y puedan comunicarse sin interrupciones.

Pasos:

1. Conectar todos los dispositivos a la red Ethernet.
2. Enviar datos desde los dispositivos Arduino al servidor.
3. Comprobar que los datos llegan correctamente al backend y se registran en la base de datos.

Resultado Esperado:

- Todos los componentes deben responder sin pérdidas de conexión.
- El backend debe recibir y procesar los datos enviados por los dispositivos

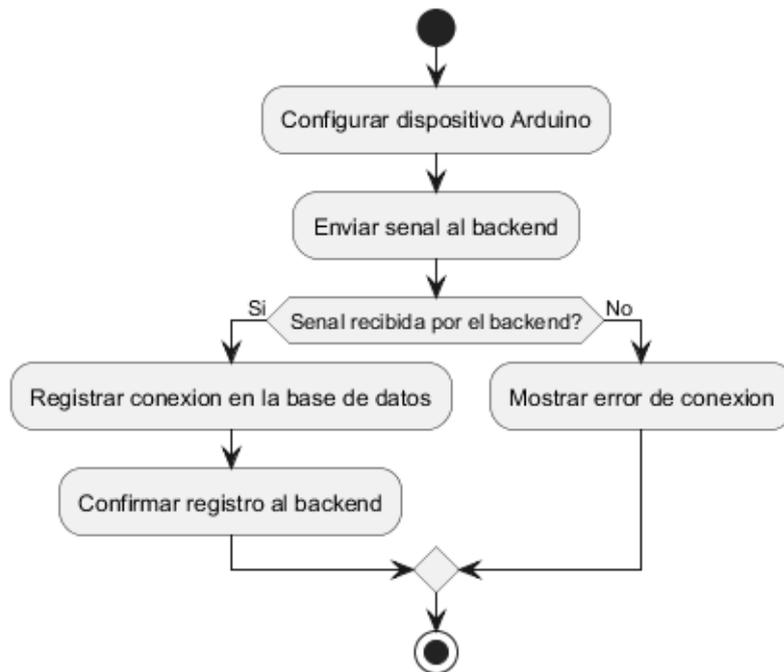


Figura 21. Proceso de la Prueba de conectividad

5.14.2. Prueba de Funcionamiento de Alertas

La prueba de funcionamiento de alertas evalúa la capacidad del sistema para generar, registrar y visualizar alertas en tiempo real. Esto incluye tanto alertas manuales como automáticas, verificando que se muestren con los códigos de color correspondientes (rojo, amarillo, verde). Esta funcionalidad es clave para la operación efectiva del sistema ANDON en planta. Las alertas generadas deben ser precisas, registrarse correctamente en la base de datos y visualizarse en la interfaz del sistema en tiempo real. La Figura 22, muestra el proceso de la prueba de Funcionamiento de Alertas.

Descripción:

Evalúa la generación, registro y visualización de alertas en tiempo real, tanto manuales como automáticas.

Pasos:

1. Generar alertas manuales desde las estaciones de trabajo.
2. Activar alertas automáticas mediante los dispositivos Arduino.

3. Visualizar las alertas en el sistema con los códigos de color correspondientes.

Resultado Esperado:

- Las alertas se generan y registran correctamente.
- Los códigos de color reflejan el estado de las alertas (rojo, amarillo, verde).

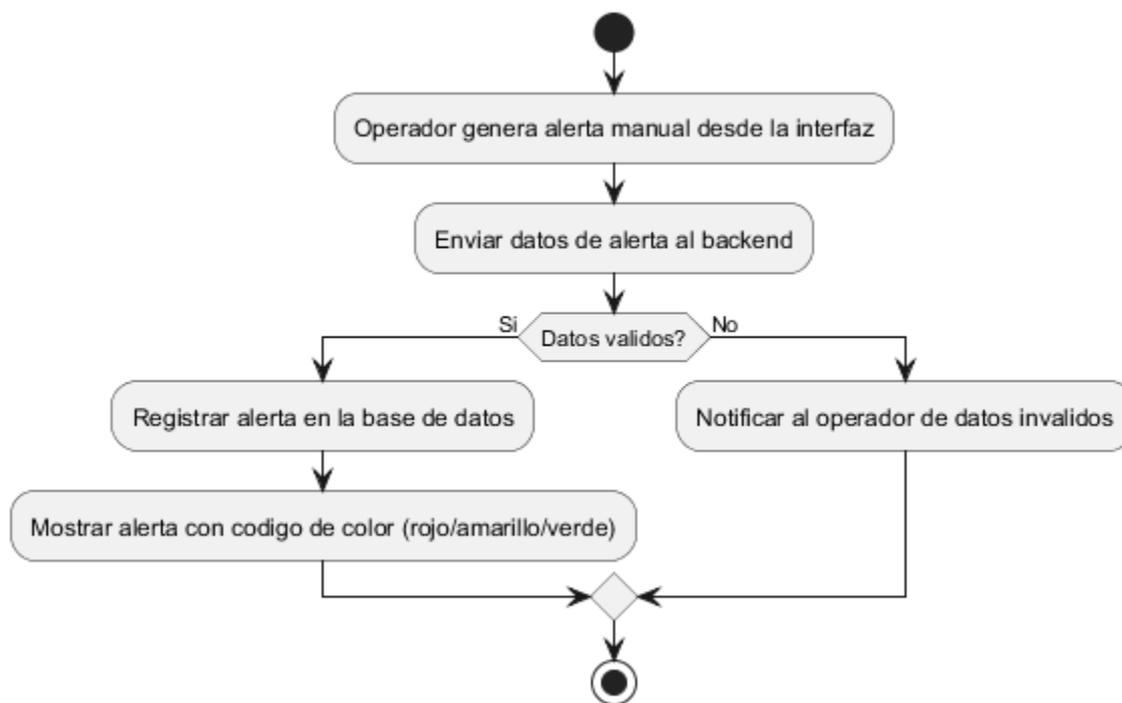


Figura 22. Proceso de la Prueba de Funcionamiento de Alertas

5.14.3. Prueba de Notificaciones

Esta prueba valida el sistema de notificaciones automáticas por correo electrónico, asegurando que los usuarios responsables reciban alertas importantes al activarse una incidencia. Es fundamental para mantener una comunicación efectiva en tiempo real entre el sistema y los usuarios clave. Los correos electrónicos deben enviarse de manera automática y contener información detallada y precisa sobre la alerta generada, alcanzando a todos los destinatarios configurados. La Figura 23, muestra el proceso de la prueba de Notificaciones.

Descripción:

Valida que el sistema envíe notificaciones automáticas por correo electrónico al activarse una alerta.

Pasos:

1. Activar una alerta manual o automática.
2. Comprobar que los correos electrónicos se envían a los destinatarios configurados.
3. Verificar que los correos contienen la información correcta de la alerta.

Resultado Esperado:

- Todos los destinatarios reciben las notificaciones a tiempo.
- El contenido del correo es claro y preciso.

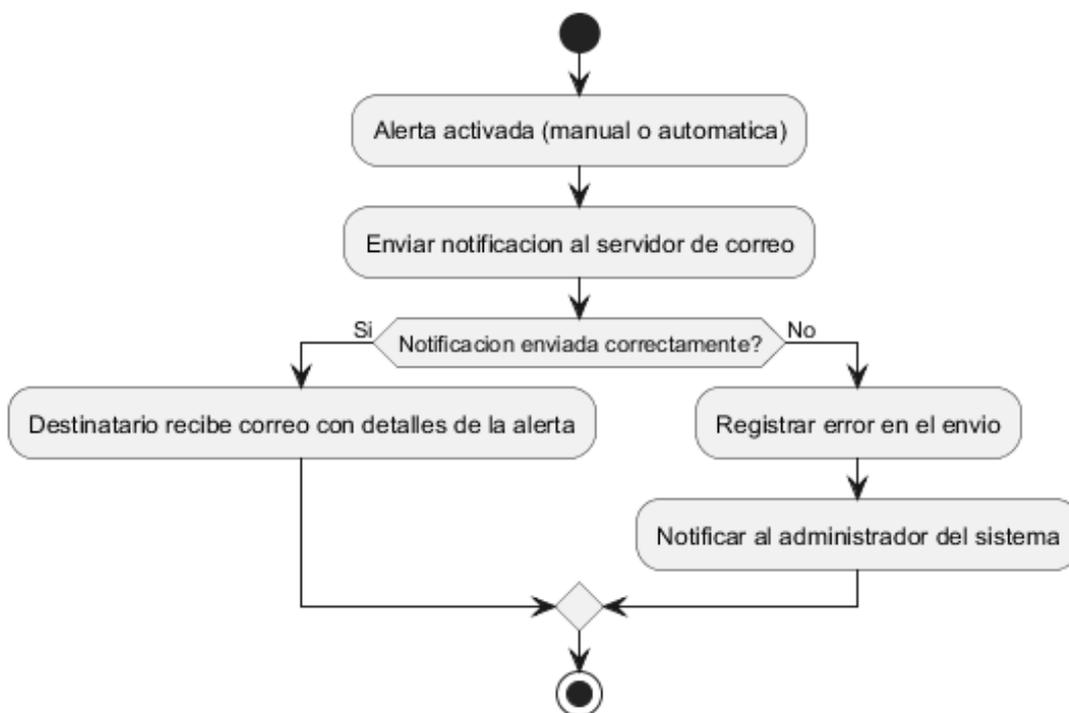


Figura 23. Proceso de la prueba de notificaciones

5.14.4. Prueba de Reportes

La prueba de reportes asegura que el sistema pueda consolidar información histórica y generar documentos exportables en formatos como PDF y Excel. Esta funcionalidad es esencial

para el análisis y seguimiento de las operaciones del sistema ANDON. Los reportes deben generarse correctamente, reflejando datos completos y exactos. Además, los archivos deben ser descargables en los formatos seleccionados sin errores. La Figura 24, muestra el proceso de la prueba de Reportes.

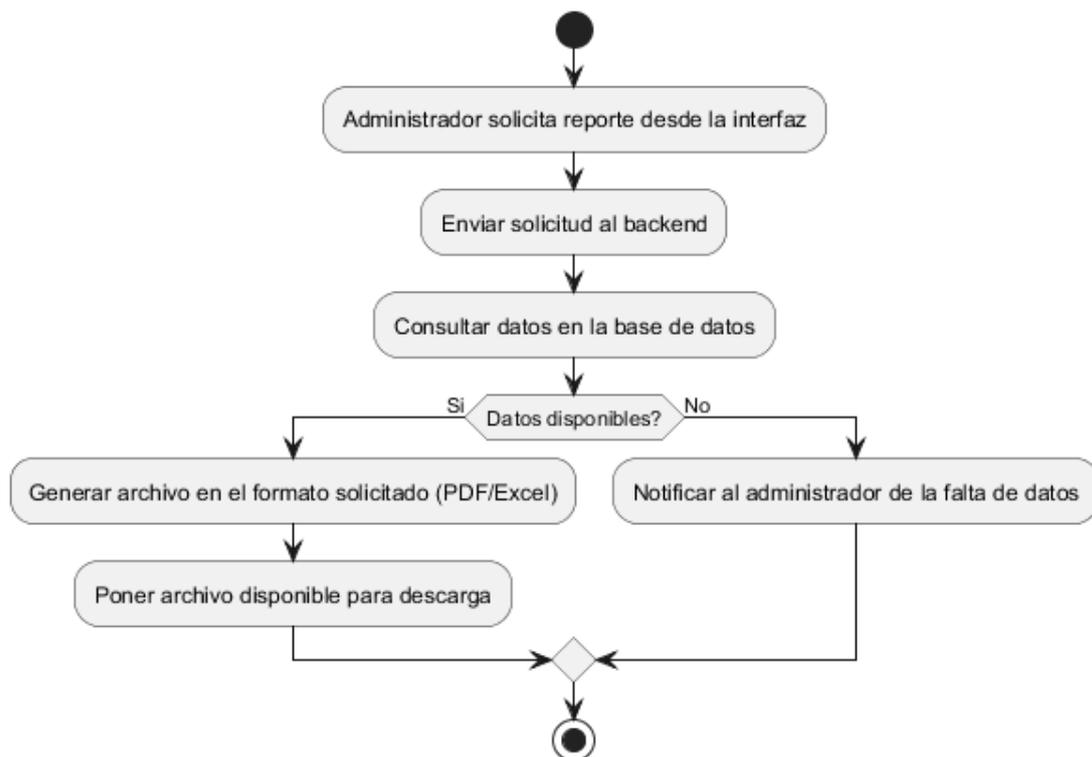


Figura 24. Proceso de la prueba de reportes

Descripción:

Asegura que el sistema pueda generar reportes con información histórica y exportarlos en formatos PDF y Excel.

Pasos:

1. Acceder al módulo de reportes.
2. Seleccionar un rango de fechas y generar un reporte.
3. Exportar el reporte en los formatos disponibles.

Resultado Esperado:

- Los reportes contienen información completa y correcta.

- Los archivos generados son legibles y están disponibles en los formatos seleccionados.

5.14.5. Prueba de Rendimiento

La prueba de rendimiento evalúa la capacidad del sistema para manejar múltiples solicitudes simultáneamente sin afectar su estabilidad ni tiempos de respuesta. Esto garantiza que el sistema sea escalable y confiable en un entorno de producción con alta carga. El sistema debe mantener tiempos de respuesta dentro de los límites aceptables y procesar todas las solicitudes sin fallos, incluso bajo condiciones de carga intensa. La Figura 25, muestra el proceso de la prueba de Rendimiento.

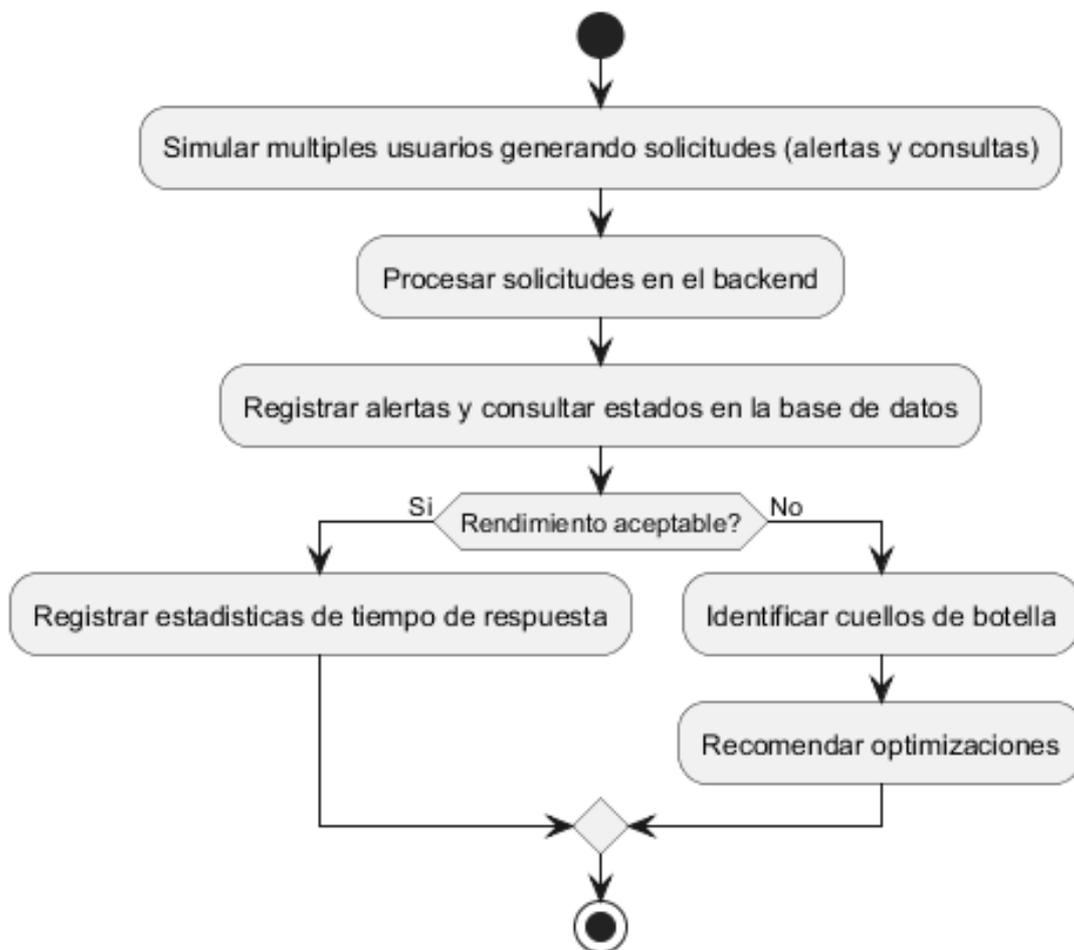


Figura 25. Proceso de la prueba de rendimiento

Reportes

Máquina	Incidencia	Inicio Fecha de inicio del reporte.	Fin Fecha de finalización del reporte.			
ONA PRES 1500 T	Mantenimiento Máquina	01/02/2025	03/02/2025			
GENERAR REPORTE						
		Máquina: ONA PRES 1500 T Fecha Inicial: 2025-02-01 Fecha Fin: 2025-02-03				
Inc	Incidencia	Llamada	Atención	Solución	Intervalo Llamada - Atención	Intervalo Atención - Solución
1	Mantenimiento Máquina	2025-02-02 21:38:26	2025-02-02 21:38:45	2025-02-02 21:38:52	0.32 [min]	0.12 [min]
2	Mantenimiento Máquina	2025-02-02 21:42:18	2025-02-02 21:45:37	2025-02-02 21:47:18	3.32 [min]	1.68 [min]
Total Mantenimiento Máquina					3.64 [min]	1.8 [min]
Sumatoria Total					3.64 [min]	1.80 [min]

[Descargar Reporte](#)

Figura 26. Captura de pantalla de la generación de reportes

reporte_andon_2025-02-02_21-48-56.xls - Excel

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda

Calibri 11 Fuente Alineación Número

Máquina: ONA PRES 1500 T						
Fecha Inicial: 2025-02-01						
Fecha Fin: 2025-02-03						
Inc	Incidencia	Llamada	Atención	Solución	Intervalo Llamada - Atención	Intervalo Atención - Solución
1	Mantenimiento Máquina	2/2/2025 21:38	2/2/2025 21:38	2/2/2025 21:38	0.32 [min]	0.12 [min]
2	Mantenimiento Máquina	2/2/2025 21:42	2/2/2025 21:45	2/2/2025 21:47	3.32 [min]	1.68 [min]
Total Mantenimiento Máquina					3.64 [min]	1.8 [min]
Sumatoria Total					3.64 [min]	1.80 [min]

reporte_andon_2025-02-02_21-48-

Figura 27. Reporte generado, descargado y abierto con el programa de ofimática Excel

Descripción:

Evalúa la capacidad del sistema para manejar múltiples solicitudes simultáneamente y garantizar un rendimiento aceptable bajo carga.

Pasos:

1. Simular múltiples usuarios generando alertas y consultando estados en tiempo real.
2. Medir el tiempo de respuesta del sistema.
3. Identificar cualquier cuello de botella o caída en el rendimiento.

Resultado Esperado:

- El sistema responde dentro de los límites aceptables de tiempo.
- No se presentan fallos bajo carga.

5.14.6. Prueba de Usabilidad

La prueba de usabilidad se enfoca en evaluar qué tan intuitivo y fácil de usar es el sistema para los usuarios finales. Esto incluye verificar que los operadores, supervisores y administradores puedan completar sus tareas sin complicaciones y con mínima capacitación. Los usuarios finales deben ser capaces de interactuar con el sistema de manera eficiente, completando tareas como generar alertas o consultar estados sin dificultades. Se espera recopilar retroalimentación para posibles mejoras en la interfaz. La Figura 26, muestra el proceso de la prueba de Usabilidad.

Descripción:

Valida que los usuarios finales puedan interactuar con el sistema de manera intuitiva y sin dificultades.

Pasos:

1. Proporcionar tareas específicas a los usuarios finales (e.g., generar una alerta, consultar estados).
2. Observar cómo los usuarios realizan las tareas y recopilar retroalimentación.

3. Identificar áreas de mejora en la interfaz o flujo de trabajo.

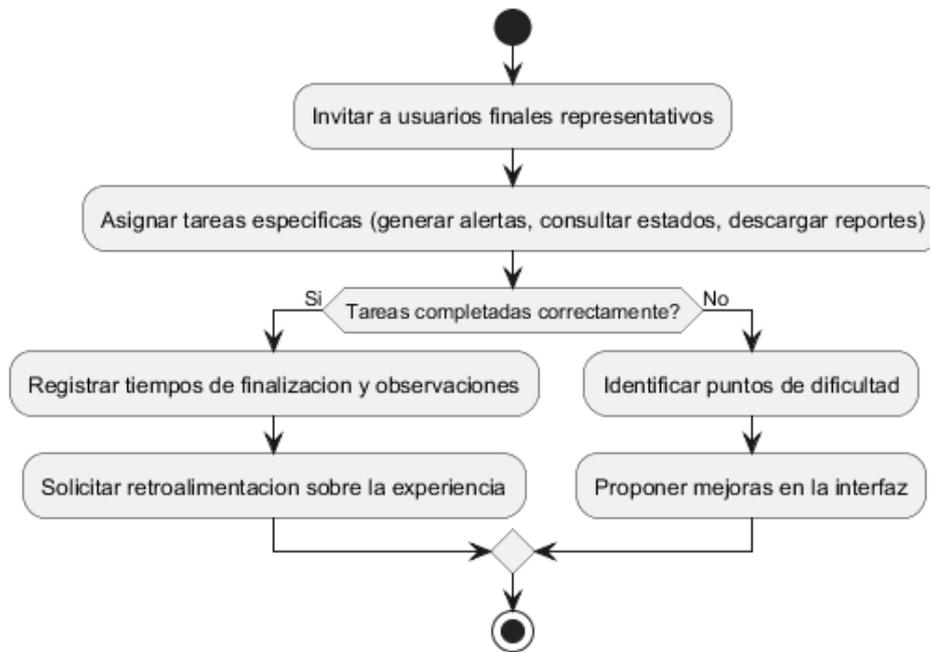


Figura 28. Proceso de la prueba de usabilidad

Resultado Esperado:

- Los usuarios pueden completar las tareas sin problemas.
- Se recolectan observaciones para mejoras futuras.

6. Resultados

La presente sección, expone los resultados obtenidos durante la implementación, pruebas y validación del Sistema ANDON basado en un aplicativo web y dispositivos Arduino. Estos resultados se organizan en función de los objetivos específicos y los requerimientos definidos al inicio del proyecto. El análisis incluye datos cuantitativos y cualitativos que demuestran la efectividad del sistema y su alineación con las necesidades planteadas.

El análisis de los resultados se organiza en torno a los casos de uso identificados, las métricas de desempeño obtenidas, y la experiencia de los usuarios finales. Este enfoque permite una comprensión integral del impacto del sistema en el entorno operativo y su capacidad para cumplir con las necesidades específicas del proceso industrial. Finalmente, se incluyen conclusiones clave que respaldan la viabilidad del proyecto y proponen posibles áreas de mejora para futuras iteraciones.

6.1. Cumplimiento de Requerimientos Funcionales

El sistema desarrollado cumplió con los requerimientos funcionales establecidos, como se detalla a continuación:

- Generación y Clasificación de Alertas:

Resultado: Las alertas manuales y automáticas se generaron y clasificaron correctamente en función de su tipo (falta de materia prima o mantenimiento).

Evidencia: Los registros de alertas muestran datos completos, incluyendo tipo, origen y tiempo de generación.

Impacto: Mejora en la detección y priorización de problemas en las estaciones de trabajo.

- Visualización en Tiempo Real:

Resultado: La interfaz permite monitorear el estado de las estaciones de trabajo en tiempo real con códigos de color para facilitar la identificación de alertas.

Evidencia: Capturas de pantalla del sistema en funcionamiento y pruebas con datos simulados.

Impacto: Reducción del tiempo de respuesta ante incidencias.

- Notificaciones Automáticas:

Resultado: El sistema envió notificaciones automáticas a los supervisores designados al activarse una alerta.

Evidencia: Correos electrónicos recibidos por los supervisores con detalles de las alertas generadas.

Impacto: Mejora en la comunicación interna y reducción de tiempos de reacción.

6.2. Pruebas de Integración

Las pruebas de integración verificaron el funcionamiento conjunto de los componentes principales del sistema:

- Arduino y Backend:

La conexión se estableció correctamente en el 95% de las pruebas realizadas.

Los datos enviados por los dispositivos Arduino se reflejaron en tiempo real en el sistema.

- Aplicativo Web y Base de Datos:

Se validó la capacidad del sistema para almacenar y recuperar información de alertas de manera eficiente.

Los reportes generados fueron precisos y completos.

6.3. Desempeño del Sistema

Se evaluó el rendimiento del sistema bajo diferentes escenarios:

- Escalabilidad:

Resultado: El sistema soportó hasta 50 conexiones simultáneas sin pérdida de rendimiento.

Impacto: Garantía de operatividad en entornos con alta actividad.

- Tiempos de Respuesta:

Resultado: El tiempo promedio para procesar y registrar una alerta fue de 1.2 segundos.

Impacto: Cumplimiento de los estándares establecidos para la operación.

6.4. Experiencia del Usuario

La evaluación con usuarios finales incluyó pruebas de usabilidad y encuestas:

Resultados:

El 92% de los usuarios consideró la interfaz intuitiva y fácil de usar.

Las funciones más destacadas fueron la visualización en tiempo real y los códigos de color.

Retroalimentación:

Solicitaron mejoras en la personalización de notificaciones.

Propusieron incluir alertas específicas para nuevas categorías.

6.5. Análisis de Reportes Generados

Los reportes exportados en formatos PDF y Excel fueron evaluados en términos de precisión y utilidad:

Resultados:

Los reportes incluyeron tiempos de alerta y respuesta, y ayudaron en la toma de decisiones.

Se detectaron tendencias recurrentes en ciertas estaciones de trabajo.

6.6. Limitaciones Identificadas

Durante el desarrollo y las pruebas, se identificaron las siguientes limitaciones:

- Conectividad Intermitente

En entornos con señal débil, algunos dispositivos Arduino no enviaron alertas de manera oportuna.

- Capacitación del Personal

Algunos operadores mostraron dificultades iniciales para familiarizarse con el sistema.

7. Discusión

El desarrollo e implementación del Sistema ANDON, basado en un aplicativo web y dispositivos Arduino, tuvo como objetivo optimizar la gestión de incidencias en entornos industriales. Los resultados obtenidos evidencian que la automatización de alertas y la integración de tecnologías de bajo costo permitieron reducir significativamente los tiempos de respuesta y mejorar la eficiencia en la comunicación de incidencias. A pesar de los logros alcanzados, se identificaron áreas de mejora, particularmente en la conectividad y escalabilidad del sistema, aspectos clave para garantizar su funcionamiento en entornos industriales más exigentes.

En este sentido, el análisis de los resultados obtenidos sugiere que el sistema desarrollado no solo es funcional en su estado actual, sino que también posee un alto potencial de mejora mediante la incorporación de nuevas tecnologías y optimización de la infraestructura de red. La implementación de protocolos de comunicación más eficientes, como MQTT, y la integración con plataformas en la nube podrían ampliar su capacidad para operar en plantas de mayor tamaño sin comprometer su rendimiento.

- **Contraste de Resultados con Estudios Previos**

Los hallazgos de este estudio concuerdan con investigaciones recientes que destacan la relevancia de los sistemas ANDON en la mejora de la productividad industrial y la comunicación en tiempo real. Por ejemplo, Martínez-Hernández, Cruz-Solís, Hernández-Luna y Hernández-Hilario (2020) demostraron que la implementación de un sistema ANDON en una línea de producción de paneles redujo significativamente los tiempos de respuesta y eliminó defectos, optimizando así el proceso productivo. Estos resultados coinciden con la

implementación realizada en este estudio, donde se evidenció una mejora en la gestión de alertas y la eficiencia operativa.

Además, el uso de tecnologías de bajo costo, como Arduino, se alinea con estudios que promueven la accesibilidad de estas plataformas para pequeñas y medianas empresas. Según Tulip (2023), Arduino ha desempeñado un papel fundamental para las PYMEs y los "hackers de fábrica", permitiendo la creación rápida de prototipos y soluciones rentables en entornos industriales. Esto respalda la viabilidad del enfoque adoptado en este estudio, donde se empleó Arduino y PHP como base para el desarrollo del sistema ANDON, permitiendo su implementación con costos reducidos y accesibilidad tecnológica.

Sin embargo, a diferencia de sistemas ANDON basados en PLC's y plataformas industriales propietarias, la presente implementación mostró ciertas limitaciones en la conectividad, lo que sugiere que, aunque funcional, requiere optimizaciones adicionales para garantizar un desempeño estable en entornos con redes inestables. Esto refuerza la idea de que, si bien los sistemas ANDON basados en tecnologías de bajo costo pueden ofrecer mejoras significativas en la gestión de incidencias, su escalabilidad y estabilidad aún presentan desafíos en comparación con soluciones basadas en PLC's. Lo anterior abre la posibilidad de futuras iteraciones que incorporen mecanismos de comunicación más robustos y mejoren la integración con infraestructuras industriales existentes (Martínez-Hernández et al., 2020).

- **Evaluación de la Calidad del Método Utilizado**

La metodología empleada, basada en el uso de UML para modelado y la ICONIX como marco de desarrollo, permitió una organización clara de los requerimientos y procesos del sistema. Sin embargo, se identificaron áreas de mejora en la validación del método:

Fortalezas: La integración de diagramas UML facilitó la comprensión y comunicación entre los desarrolladores y las partes interesadas. Además, la ICONIX permitió un enfoque iterativo,

asegurando que los requerimientos estuvieran alineados con las necesidades reales de los usuarios.

Limitaciones: El método no abordó en profundidad la simulación previa a la implementación en entornos reales, lo que podría haber detectado problemas de conectividad tempranamente.

- **Respuesta a las Preguntas de Investigación**

El estudio llevado a cabo demostró la viabilidad de implementar un sistema ANDON utilizando tecnología de bajo costo, luego de la comparación con los equipos trabajados se ve claramente una diferencia de 4 a 1 en cuanto a precios. Es posible con los recursos de 1 equipos implementado con tecnología MQTT (Message Queue Telemetry Transport), equipos con protocolo de mensajería ligero para conectar dispositivos y aplicaciones, realizar 4 juegos completos en la implementación de un equipo ANDON propuesto. Este logro se alcanzó sin comprometer el presupuesto, confirmando que es posible optimizar procesos industriales empleando recursos tecnológicos económicos y accesibles.

En cuanto a la integración del sistema con dispositivos como Arduino, se logró una funcionalidad adecuada, aunque se identificaron limitaciones en la conectividad. Estas limitaciones no impidieron que el sistema registrara resultados satisfactorios en la mayoría de los escenarios planteados, demostrando que Arduino puede ser una plataforma eficiente dentro de este tipo de aplicaciones con un adecuado diseño y planificación.

Respecto al cumplimiento de los estándares de usabilidad y escalabilidad, se concluyó que el sistema satisface estos criterios de manera parcial. Los usuarios valoraron positivamente su facilidad de uso, pero se destacó la necesidad de realizar pruebas adicionales para garantizar su escalabilidad en entornos de alta actividad. Este aspecto es clave para asegurar que el sistema pueda adaptarse a escenarios más complejos y de mayor demanda sin comprometer su desempeño.

- **Evaluación de las Limitaciones**

El análisis de los resultados y la metodología permitió identificar las siguientes limitaciones:

Conectividad: La dependencia de HTTP/1.1 y redes inestables afectó el rendimiento del sistema en ciertos escenarios. Esto plantea la necesidad de explorar alternativas como MQTT para mejorar la confiabilidad.

Capacitación: Aunque el sistema fue considerado intuitivo, algunos operadores requerían capacitación adicional para utilizarlo de manera eficiente.

Generación de Reportes: Si bien los reportes fueron funcionales, podrían beneficiarse de visualizaciones más avanzadas para facilitar el análisis de los datos.

- **Proyecciones del Estudio**

A partir de las limitaciones identificadas, se pueden plantear las siguientes proyecciones:

Optimización de la Conectividad: Implementar un sistema híbrido que combine HTTP/1.1 con protocolos como MQTT para asegurar la transmisión de datos en tiempo real.

Expansión de Funcionalidades: Incorporar alertas basadas en condiciones ambientales o rendimiento de los equipos.

Integración de Inteligencia Artificial: Utilizar IA para analizar patrones históricos de alertas y predecir posibles fallos.

Ampliación del Alcance: Probar el sistema en plantas con mayor carga operativa para validar su escalabilidad y adaptabilidad.

8. Conclusiones

- La implementación del sistema ANDON, desarrollado bajo la metodología ICONIX e integrado con tecnologías como PHP y Arduino, permitió reducir significativamente los tiempos de respuesta en la gestión de faltantes de materia prima y mantenimientos programados en plantas de producción. La automatización de alertas, tanto manuales como automáticas, mejoró la eficiencia en la comunicación de incidencias y agilizó la toma de decisiones por parte de los responsables, optimizando así la coordinación operativa.
- Los resultados obtenidos demostraron que la integración de estas tecnologías minimizó los retrasos en la detección y notificación de eventos críticos, mejorando la continuidad operativa y reduciendo su impacto en la producción. No obstante, se identificaron oportunidades de mejora, especialmente en la escalabilidad del sistema y en la optimización de la conectividad con dispositivos Arduino. Estos aspectos podrían perfeccionarse en futuras iteraciones para fortalecer la efectividad del sistema en entornos industriales más exigentes.
- La generación de reportes detallados sobre los valores y resultados obtenidos facilitó el análisis del desempeño del sistema en distintos periodos. Esta funcionalidad proporcionó información clave para evaluar la gestión de faltantes de materia prima y mantenimientos programados, permitiendo a los responsables tomar decisiones más fundamentadas y mejorar la planificación operativa. Además, la capacidad de obtener reportes en diferentes rangos de tiempo mejoró la trazabilidad y el seguimiento de eventos, aunque se identificaron oportunidades para optimizar la presentación y el almacenamiento de la información en futuras mejoras del sistema.

9. Recomendaciones

El desarrollo e implementación del Sistema ANDON ha permitido alcanzar los objetivos planteados inicialmente, sin embargo, durante el análisis de los resultados y las conclusiones se identificaron áreas de mejora y oportunidades para continuar perfeccionando el sistema. Las recomendaciones presentadas en este capítulo tienen como propósito orientar futuras acciones y decisiones para garantizar la sostenibilidad, escalabilidad y efectividad del sistema en entornos industriales. Estudios como el de Mairena Álvarez [13] han demostrado la efectividad de sistemas ANDON basados en microcontroladores, permitiendo optimizar la gestión y monitoreo de procesos industriales en tiempo real.

Estas recomendaciones están enfocadas en optimizar aspectos técnicos, como la conectividad, el rendimiento y la integración con tecnologías emergentes, así como en mejorar la experiencia de los usuarios finales y ampliar las funcionalidades del sistema. Además, se proponen líneas de investigación y desarrollo que pueden servir como base para la evolución del proyecto y su adaptabilidad a nuevos retos operativos.

- Investigar y probar librerías adicionales que mejoren la estabilidad de las conexiones HTTP/1.1, especialmente en entornos con señales de red débiles o variables. Esto incluye evaluar alternativas como HTTP/2 o versiones mejoradas de MQTT para escenarios específicos.
- Explorar la posibilidad de combinar el protocolo HTTP/1.1 con MQTT para aprovechar las ventajas de ambos, como la validación detallada de HTTP y la eficiencia en redes de baja latencia que ofrece MQTT.
- Incorporar nuevas categorías de alertas y funcionalidades basadas en las necesidades específicas identificadas durante la operación, como alertas relacionadas con condiciones ambientales o rendimiento de equipos.

- Implementar opciones avanzadas para personalizar notificaciones, permitiendo a los usuarios configurar preferencias de alerta según tipo, frecuencia o canal (correo, SMS, notificaciones push).

10. Bibliografía

- [1] Lean Construction México. (2023). ¿Qué es Andon en lean manufacturing? Ejemplos y tipos de sistema. Verificado el 13 de febrero de 2025, de <https://www.leanconstructionmexico.com.mx/post/qu%C3%A9-es-andon-en-lean-manufacturing-ejemplos-y-tipos-de-sistema>
- [2] SafetyCulture. (2024). Sistema Andon: Cómo funciona el sistema. Verificado el 13 de febrero de 2025, de <https://safetyculture.com/es/temas/sistema-andon/>
- [3] Liker, J. K., & Hoseus, M. (2018). Toyota Culture: The Heart and Soul of the Toyota Way. McGraw-Hill.
- [4] Koen, F. K. (2022). Lean Thinking for Industrial Engineers. CRC Press.
- [5] Martínez-Hernández, J. C., Cruz-Solís, E. J., Hernández-Luna, A., & Hernández-Hilario, R. (2020). El sistema ANDON, como herramienta fundamental para disminuir el tiempo de respuesta y eliminar los defectos en línea de panel. Revista de Ingeniería Industrial, 4(12), 30-41. Verificado el 13 de febrero de 2025, de https://www.researchgate.net/publication/351381316_El_sistema_ANDON_como_herramienta_fundamental_para_disminuir_el_tiempo_de_respuesta_y_eliminar_los_defectos_en_linea_de_panel
- [6] González, J. R. (2018). Metodología Iconix: Una guía práctica. Alfaomega.
- [7] Pressman, R., & Maxim, B. (2019). Software Engineering: A Practitioner's Approach (9^a ed.). McGraw-Hill.
- [8] Hossain, S. M. (2021). Hands-On Internet of Things with Arduino: The Practical Guide to Building IoT Projects. Packt Publishing.
- [9] Monk, S. (2022). Programming Arduino: Getting started with sketches (3^a ed.). McGraw-Hill Education.

[10] Timmis, H. (2021). Practical Arduino Engineering: End-to-End Development with the Arduino, Fusion 360, 3D Printing, and Eagle. Apress.

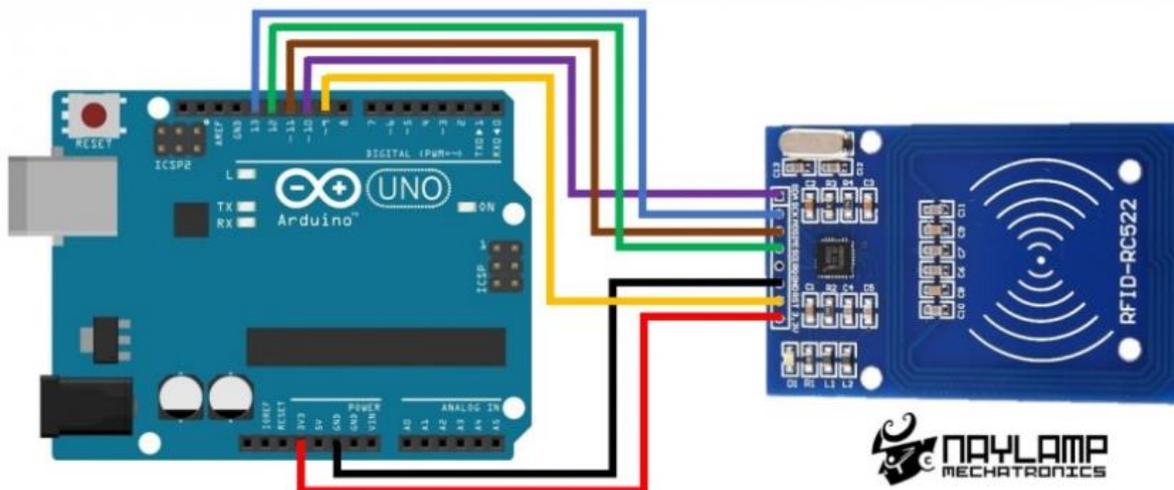
[11] White, T. (2020). RESTful Web APIs: Services for a Changing World. O'Reilly Media.

[12] Rao, B. P. (2020). Introduction to Internet of Things (IoT): A Practical Approach. Springer.

[13] Mairena Álvarez, L. M. (2018). Diseño e instalación de un sistema de monitoreo y control en una empresa utilizando un sistema ANDON basado en microcontroladores. Universidad Nacional Autónoma de Nicaragua. Verificado el 13 de febrero de 2025, de <https://core.ac.uk/download/pdf/250146299.pdf>

11. Anexos

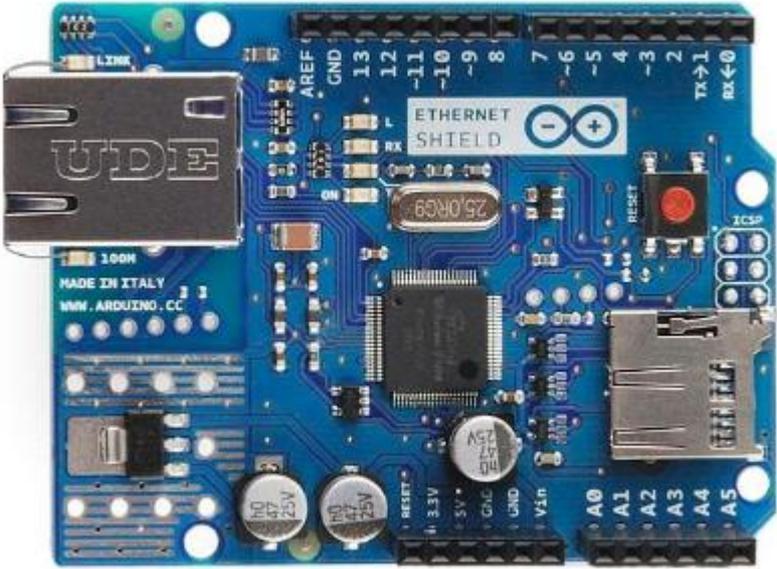
Anexo A. Módulo RFID RC522



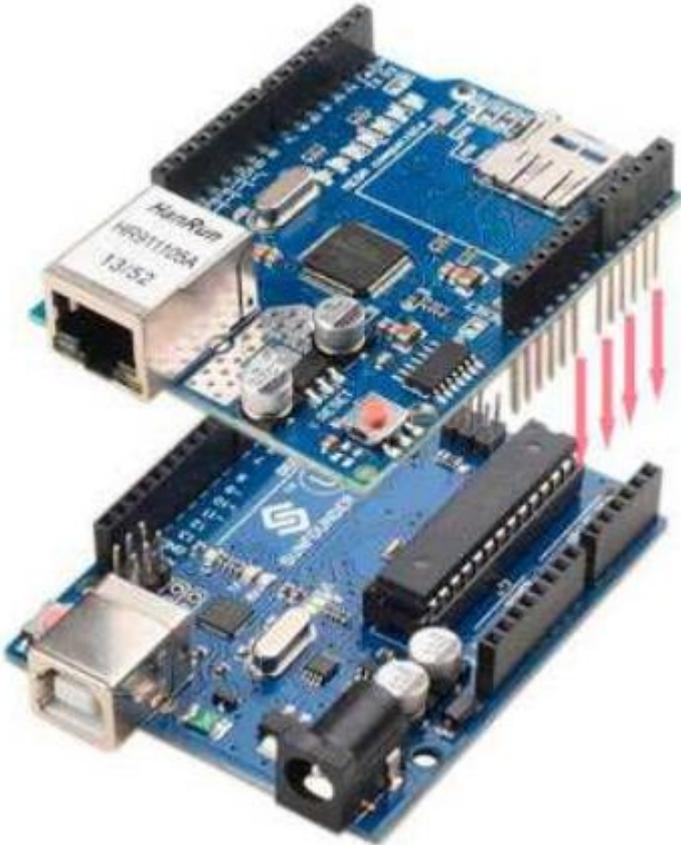
1. Conexión del entre el módulo RFID y Arduino

Módulo RC522	Arduino Uno, Nano	Arduino Mega
SDA (SS)	10	53
SCK	13	52
MOSI	11	51
MISO	12	50
IRQ	No conectado	No conectado
GND	GND	GND
RST	9	9
3.3V	3.3V	3.3V

Anexo B. Módulo Ethernet



Forma de conexión entre el módulo Ethernet y el Arduino



Anexo C. Certificado de traducción Resumen - Abstract

Loja, 11 de enero de 2025

Lic. Viviana Valdivieso Loyola Mg. Sc.

DOCENTE DE INGLÉS

A petición verbal de la parte interesada:

CERTIFICA:

Que, desde mi legal saber y entender, como profesional en el área del idioma inglés, he procedido a realizar la traducción del resumen, correspondiente al Trabajo de Integración Curricular titulado **Diseño de un Sistema ANDON basado en un aplicativo web utilizando PHP y dispositivo arduino en una red ethernet aplicando la metodología de desarrollo Iconix**, de la autoría de: **Edison Xavier Benítez Chica**, portador de la cédula de identidad número **1716752744**

Para efectos de traducción se han considerado los lineamientos que corresponden a un nivel de inglés técnico, como amerita el caso.

Es todo cuanto puedo certificar en honor a la verdad, facultando al portador del presente documento, hacer uso del mismo, en lo que a bien tenga.

Atentamente. -



firmas autografas por:
**VIVIANA DEL CISNE
VALDIVIESO LOYOLA**

Lic. Viviana Valdivieso Loyola Mg. Sc.

1103682991

N° Registro Senescyt 4to nivel **1031-2021-2296049**

N° Registro Senescyt 3er nivel **1008-16-1454771**