UNIVERSIDAD CENTRAL
VICERRECTORÍA ACADÉMICA

CARRERA DE INGENIERÍA ELECTRÓNICA

PROPUESTA DE MEJORA PARA EL MONITOREO DE LA RED INALÁMBRICA DE LA SEDE CENTRAL DEL FONAFIFO, MEDIANTE EL DISEÑO DE UN SISTEMA EMBEBIDO IMPLEMENTADO CON RASPBERRY PI, INCORPORANDO ALGUNOS PRINCIPIOS DE LA NORMA IEC 62443-4-2

MODALIDAD DE TESIS PARA OPTAR POR EL GRADO DE
LICENCIATURA EN INGENIERÍA ELECTRÓNICA

ESTUDIANTE: ERICK ESQUIVEL RUBÍ

TUTOR: ING. WILFREDO PRADO CUBILLO

SEDE METROPOLITANA
ABRIL, 2025

Contenido

Tablas

Figuras

Dedicatoria y agradecimiento

Este trabajo de investigación está dedicado a mi padre y madre, quienes desde niño me inculcaron el valor de la preparación para afrontar los retos de la vida de una mejora manera. Asimismo, está dedicado a mi esposa, quien es la que me ha acompañado en todo este duro trayecto, el cual ha significado mucho sacrificio, pero a la vez ha representado una satisfacción muy grande en el ámbito personal y profesional.

Agradezco a Dios por la salud, la fortaleza y la perseverancia que me ha brindado durante todo este tiempo para lograr cumplir una meta de vida.

Resumen

El presente trabajo de investigación tiene como tema una “Propuesta de mejora para el monitoreo de la red inalámbrica de la sede central del Fonafifo, mediante el diseño de un sistema embebido implementado con Raspberry Pi, incorporando algunos principios de la norma IEC 62443-4-2”; y como objetivo general, se ha establecido elaborar un sistema embebido con Raspberry Pi, con el que se logre una mejora en el monitoreo de la red inalámbrica en la sede central del Fonafifo, incorporando algunos principios de la norma IEC 62443-4.

Es por ello por lo que la línea teórica del estudio se desarrolló teniendo como aspecto principal la ciberseguridad de la información, planteando una solución desde el punto de vista de la Ingeniería Electrónica ante una deficiencia con que cuenta el Fonafifo con respecto al monitoreo de su red inalámbrica. Lo anterior se debe a que la ciberseguridad es aspecto fundamental hoy en día debido a la gran proliferación de ciberataques que sufren tanto las instituciones públicas como las empresas en general a cada momento.

Asimismo, se utilizó un enfoque mixto, debido a las características cualitativas y cuantitativas de las variables de estudio planteadas, todo ello bajo la metodología DMAIC, la cual tiene como actividades relevantes el definir, medir, analizar, mejorar y controlar; todo ello aplicado al diseño e implementación del sistema embebido propuesto.

Esta investigación concluyó la viabilidad de la implementación de un sistema embebido de este tipo, siempre y cuando se someta posteriormente a un proceso de revisión y mejora continua, el cual permita hacerlo más robusto, de modo que resulte aún más efectivo en cuanto a su propósito.

CAPÍTULO 1: INTRODUCCIÓN

1.1.Planteamiento del Problema

La necesidad de implementar un sistema embebido para monitorear una red inalámbrica se origina a partir de diversas causas que impactan negativamente su rendimiento y seguridad. En primer lugar, el equipo actual que gestiona la red ha superado su vida útil, lo cual limita su eficiencia y capacidad para integrarse con otras herramientas tecnológicas modernas. Dicha falta de integración dificulta visualizar de forma efectiva del tráfico de la red, sumado al hecho de que no se generan alertas de actividad sospechosa, lo cual dificulta la identificación y la atención de posibles problemas de manera oportuna; sin dejar de lado la falta de presupuesto para adquirir soluciones nuevas, situación común dentro del sector público.

Como consecuencia de lo anterior, surge la imposibilidad de establecer controles de seguridad eficientes, además de que se puede presentar lentitud en la conexión, lo que afecta la experiencia del usuario y genera insatisfacción, principalmente para aquellas personas que dependen completamente de la red inalámbrica para llevar a cabo sus funciones, lo cual resulta en una disminución en la productividad en general. Además, la exposición a ciberataques aumenta considerablemente, ya que las vulnerabilidades no son detectadas ni mitigadas adecuadamente, esto provoca una mayor probabilidad de pasar por alto situaciones de riesgo, las cuales podrían comprometer no solo la integridad de la red inalámbrica, sino también la seguridad de los datos.

Con base en lo descrito, se plantea como problema de investigación lo siguiente:

¿Cómo mejorar el monitoreo de la red inalámbrica de la sede central del Fonafifo, haciendo uso de un sistema embebido implementado con un Raspberry Pi e incorporando algunos principios de la norma IEC 62443-4-?

1.2.Objetivos

Para dar respuesta al problema de la investigación, se han definido los siguientes objetivos.

1.2.1. Objetivo General

Elaborar un sistema embebido con Raspberry Pi, con el que se logre una mejora en el monitoreo de la red inalámbrica en la sede central del Fonafifo, el cual incorpore algunos principios de la norma IEC 62443-4.

1.2.2. Objetivos Específicos

- Examinar los aspectos de seguridad de la red inalámbrica para la identificación de vulnerabilidades, utilizando la norma IEC 62443-4 como marco de referencia.
- Diseñar un sistema embebido con Raspberry Pi para el monitoreo de la red inalámbrica, el cual permita la captura y el análisis de información, así como la generación de alertas en caso de actividad sospechosa.
- Interpretar los resultados de las mediciones realizadas para la evaluación de la efectividad del sistema embebido por medio del análisis de la información recabada.
- Evaluar el retorno de inversión (ROI) del sistema embebido mediante la determinación de su impacto económico y en seguridad, y por medio de la comparación de costos y beneficios con una solución actual del mercado.

1.3.Justificación

Las redes inalámbricas son indispensables hoy en día, ya que permiten a los usuarios conectarse fácilmente a los recursos compartidos, al servicio de correo, a los sistemas de información y al servicio de internet, todo desde distintos dispositivos. Sin embargo, este acceso también ha aumentado la probabilidad de ciberataques en las organizaciones. Por ello, las conexiones inalámbricas representan un alto riesgo, pues se han convertido en uno de los principales vectores a través de los cuales los atacantes pueden infiltrarse en las redes y llevar a cabo acciones ilícitas (Kaspersky, 2024).

En Fonafifo, se cuenta con estas conexiones de red inalámbricas; sin embargo, el equipo que las administra ha llegado a su nivel de obsolescencia tecnológica desde hace algún tiempo, aunado al hecho de que no permite realizar un monitoreo efectivo del tráfico de red debido a que su interfaz de administración no cuenta con las opciones requeridas para este tipo de tareas, por lo que el riesgo de pasar por alto tráfico o actividades sospechosas es muy alto. Adicionalmente, los ya constantes recortes presupuestarios que llevan a cabo continuamente los entes gubernamentales responsables afectan no solo a los programas sociales y a la prestación de servicios por parte de las instituciones públicas a la ciudadanía, sino que además inciden claramente de manera negativa en la operatividad interna de las mismas instituciones (Solano, 2024).

Lo anterior desafortunadamente trae consecuencias en el área de tecnología, ya que en muchas ocasiones no se cuenta con los recursos necesarios para la adquisición o la actualización de herramientas y sus respectivos licenciamientos, los cuales permiten incrementar la eficiencia de las labores necesarias en el campo de la seguridad de la información, tema que en la actualidad es crítico en vista de la relevancia que esta tiene para el logro de las distintas metas que plantean las instituciones públicas; sin dejar de lado además el cumplimiento normativo al que están sujetas por parte de entes fiscalizadores como el MICITT.

Ejemplo de lo anterior es la Directriz N° 133-MP-MICITT, por medio de la cual, en resumen, se instruye a la Administración Pública Central y Descentralizada para que se realicen los procesos necesarios que garanticen la resiliencia tecnológica, la cual está directamente relacionada con los aspectos de la seguridad de la información (MICITT, 2022).

Es por ello por lo que la presente investigación pretende brindar las pautas que permitan la elaboración y la puesta en funcionamiento de un sistema embebido de bajo costo, el cual permita realizar un monitoreo de la red inalámbrica, combinando elementos de *hardware* y

software utilizando un Raspberry Pi como eje central del sistema, con el cual se pueda solventar dicha necesidad, mediante la obtención y el análisis de las señales, así como con la generación de alertas para los encargados. A su vez, dicho sistema no tendría un costo tan elevado como el que poseen las soluciones ofrecidas en el ámbito comercial por las distintas empresas dedicadas a ello.

1.4.Antecedentes

Con el fin de comprender el contexto bajo el cual se definió el problema de esta investigación, a continuación, se presentan los antecedentes relacionados con el tema. En ellos, se describe brevemente el trabajo realizado por cada autor y se presenta, en primera instancia, algunos trabajos realizados en otros países, para luego concluir con investigaciones realizadas en el ámbito nacional.

1.4.1. Internacionales

- Inicialmente, Ailaca (2011) desarrolló una investigación acerca del monitoreo y el control de redes inalámbricas para optimizar el servicio de internet de la empresa Intercompu, ubicada en Ecuador, esto debido a que, en ese momento, a pesar de que dicha empresa había tenido un incremento importante en la calidad del servicio de internet, no contaba con una gestión adecuada de sus redes inalámbricas.

  Entre sus objetivos, estaba la implementación de este sistema de monitoreo de redes inalámbricas, para lo cual se procuró primero realizar un estudio sobre la calidad del servicio de internet que le prestaba a sus usuarios. Además, se analizaron algunos sistemas de monitoreo para finalmente presentar una propuesta que permitiera subsanar las falencias en cuanto a este aspecto. Dicha investigación tuvo un enfoque cuantitativo y en ella se analizaron factores referentes a los sistemas de monitoreo de redes, todo ello ligado con la situación que se estaba presentando en la empresa.

El principal aporte de este trabajo es la realización de una propuesta detallada, en la cual se describe el paso a paso de las configuraciones necesarias hechas en los equipos que conforman el sistema de monitoreo, y en la cual se comprobó que es viable de acuerdo con sus factibilidades técnicas, operativas y económicas (Ailaca, 2011).

- Por su parte, Andrade (2022) presentó un trabajo donde se implementó un servidor LAMP con base en un Raspberry Pi y un ESP32 para el monitoreo de la temperatura, la presión y la humedad de un laboratorio de ciencias básicas en el Instituto Tecnológico Superior de la Torre de Veracruz, México, esto debido a que dicho laboratorio presentaba un problema en cuanto a la medición de dichos parámetros. Entre sus objetivos, además de la implementación del servidor LAMP (Linux, Apache, MySQL y PHP), fueron necesarios la simulación y el diseño de circuitos electrónicos que permitieran la obtención de los datos mediante diferentes sensores, así como la definición de un conjunto de pruebas para poder verificar que todo el sistema operara de manera correcta según el diseño. En cuanto a la metodología, la autora utilizó un modelo mixto, ya que se basó en un enfoque cuantitativo para la parte experimental de la investigación, mientras que usó el enfoque cualitativo para el estudio de casos.

  Su aporte radicó en que se dio una implementación exitosa del sistema, tanto para la parte de circuitería electrónica relacionada con los sensores y la obtención de la información como para su integración con el servidor LAMP en cuanto a la visualización de los resultados (Andrade, 2022).

- Navarrete (2021) realizó una propuesta de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi con el propósito de que pudiese ser implementada en

cualquier área de la ingeniería y agroindustria, pensado en ese momento principalmente para su país, Ecuador.

Sus objetivos, básicamente, se enfocaron en la implementación de dicho sistema embebido haciendo uso del Raspberry Pi junto con Python. Para ello, se investigó previamente los fundamentos necesarios para comprender el funcionamiento de los distintos sensores utilizados para la captura de datos. En cuanto al aporte, este fue la implementación con éxito del sistema embebido, el cual se logró configurar y poner en marcha bajo los parámetros planteados en su diseño; además, utilizó un recurso adicional al lograr el envío de los datos a una aplicación en la nube llamada Ubidots (Navarrete, 2021).

- Cossio (2021) presentó una prueba de concepto de una aplicación móvil para el monitoreo de redes inalámbricas del Observatorio de Seguridad de la Red Chilena (OSR), ya que se tenía una carencia con respecto a la recolección de datos de las redes locales de una manera efectiva. Los objetivos de este trabajo se enfocaron en el desarrollo de esta aplicación móvil, que pudiese recolectar datos y presentar informes relevantes y, a su vez, permitir a sus usuarios contar con una serie de pruebas para la medición de distintos parámetros sobre los cuales era de interés generar estadísticas. Su principal aporte fue que la aplicación permitió concretar algunas ideas sobre lo que se buscaba medir y acerca de cómo realizarlo, esto debido a que en general se indica que los datos no eran del todo representativos; sin embargo, como prueba de concepto dio resultados satisfactorios (Cossio, 2021).

1.4.2. Nacionales

- Pasando al ámbito nacional, Aguilar (2005) realizó una investigación acerca del

monitoreo y la administración de los equipos de la red del Banco Nacional de Costa Rica, usando SNMP y la plataforma NNM HP-Open View, esto debido a la necesidad de unificar la administración de distintas marcas de equipos de red que se tenía en ese momento. Sus principales objetivos fueron la creación de un sistema de gestión de la red que fuera eficiente y adaptado a las necesidades de ese entonces, integrando los equipos de comunicación y procurando reducir el tráfico de paquetes de administración que resultaran innecesarios.

El aporte del trabajo radicó en que se constató que el sistema propuesto era completamente factible para el banco, ya que se pudo reducir de manera considerable la cantidad de alertas que no aportaban información relevante para la toma de decisiones, de modo que se procuró que toda esta información fuera accesible vía web mediante distintas aplicaciones desarrolladas para tales efectos (Aguilar, 2005).

- Por otro lado, Berrocal (2021) aplicó los microcontroladores de bajo costo para la detección de deformaciones unitarias en estructuras, ya que se pretendía tener una herramienta que pudiese ser aplicada en diferentes contextos, por lo que su desarrollo no contó con una ubicación en específico. Su objetivo era principalmente tener claro conceptos estructurales relacionados con dichas deformaciones, ya que el trabajo estaba dirigido para ser aplicado en el campo de la Ingeniería Civil. Asimismo, la investigación de los diferentes componentes electrónicos y paquetes de *software* que permitieran la programación de los microcontroladores fue fundamental para lograr la integración total requerida para el sistema.

  La metodología utilizada fue por fases, pues se presentó la fase teórica, de implementación, de validación y de elaboración del informe; cada una de ellas con distintas subetapas que abarcaran cada parte del sistema propuesto. Como aporte

principal está el hecho de que se logró implementar un sistema de adquisición de datos (DAQ) de muy bajo costo en comparación con la inversión que se hubiese tenido que realizar por un equipo estándar del mercado. En términos porcentuales, se determinó que este costo representó el 1 % del costo de un equipo estándar (Berrocal, 2021).

- Cordero et al. (2016) propusieron un modelo de gestión de riesgos de infraestructura de tecnologías de información y comunicación con base en RISK IT de ISACA, el cual pudiera ser aplicado en una empresa de la cual los estudiantes se reservaron el nombre, esto en procura de mejorar la gestión de riesgos de TI. Su objetivo fue principalmente el diseño de dicha propuesta, para lo cual se tuvo inicialmente que contextualizar la administración de riesgos que llevaba a cabo en ese momento la Unidad de Tecnologías de Información de la empresa, así como sus condiciones y su relación con las demás áreas.

  En cuanto a la metodología utilizada, los estudiantes basaron su trabajo en el marco RISK IT de ISACA, que tiene como punto focal los riesgos de seguridad de tecnologías de información, así como en principios de la Administración del Riesgo Empresarial ERM. Su aporte, como bien se ha indicado, fue el planteamiento del modelo de gestión de riesgos de TI, que, si bien es cierto, no terminó siendo aplicado, igualmente se recomendó a la empresa que considerara su implementación dadas las ventajas que brinda el hecho de realizar una gestión de riesgos con base en estándares de alta calidad, como los definidos por ISACA (Cordero et al, 2016).

- Finalmente, Fernández (2013) realizó un trabajo acerca del control sobre los procesos de TI para garantizar la seguridad de los sistemas de una institución financiera

> denominada como CDA, debido a una necesidad específica que planteó la jefatura del Departamento de Informática de dicha empresa referente a evaluar el nivel de madurez del proceso de COBIT DS5 llamado “Garantizar la seguridad de los sistemas”. Como objetivo, se planteó la necesidad de evaluar dicho nivel de madurez, para lo cual se realizó inicialmente una verificación de las acciones llevadas a cabo por parte de los dueños de procesos con respecto a la seguridad de la información que manejaban. La metodología utilizada se basó en un enfoque cuantitativo, en vista de la operacionalidad que se les dio a las variables definidas para el estudio. Además, se aplicaron tres etapas, a saber: planificación, ejecución y comunicación de resultados. Su aporte fue que se pudo concretar con éxito la evaluación del proceso DS5, en la cual además se plantearon las recomendaciones del caso en vista de la importancia que tiene la seguridad de la información, así como las posibles consecuencias que pudo presentar la institución financiera en temas como pérdidas económicas, pérdida de imagen, enfrentarse a procesos legales e incumplimientos regulatorios (Fernández, 2013).

Como puede notarse en los distintos documentos investigados, todos cuentan con algún grado de relación con respecto a lo que se abarca en la presente investigación, pero no muestran una aplicación de la norma utilizada ni del enfoque específico para el cual se construye el sistema embebido base de este trabajo. De ahí que el aporte final se considera que será de gran ayuda para su posible implementación en otros lugares.

1.5.Proyecciones

Seguidamente, se presentan las proyecciones o la visión, las cuales incluyen los alcances, es decir, hasta dónde se espera llegar con la investigación. Asimismo, las limitaciones describen los motivos por los cuales no sería posible alcanzar los objetivos propuestos, o bien, ir más allá

de lo planteado.

1.5.1. Alcances

Con la presente investigación, se busca obtener los siguientes beneficios:

- Un sistema embebido que solvente la necesidad de contar con una herramienta para el monitoreo de la red inalámbrica del Fonafifo Sede Central, el cual integre sensores y módulos que faciliten la interpretación de los análisis de las señales provenientes de la red.
- La aplicación de la norma IEC 62443-4 en cuanto a los aspectos de seguridad del *hardware* del sistema embebido, como su protección física, el control de acceso a los puertos, el estado del *hardware* en general, entre otros.
- Una gestión simple de sistema como tal, la cual permita alertar situaciones que podrían resultar sospechosas desde el punto de vista de tráfico de red y seguridad.
- Información y alertas que puedan ser captadas mediante componentes y circuitos electrónicos.
- Una evaluación del retorno de inversión del sistema embebido, la cual permita analizar su viabilidad para ser implementado en las Oficinas Regionales de Fonafifo, o bien, en otras instituciones públicas que tengan mayores limitaciones presupuestarias.

1.5.2. Limitaciones

Adicionalmente, se cuenta con las siguientes limitaciones:

- La investigación contempla únicamente la red inalámbrica de la Sede Central de Fonafifo, debido a que no se dispone del tiempo suficiente ni de los recursos financieros para hacer visitas a las distintas Sedes Regionales con el fin de hacerlas parte del análisis.

- No se aplicará la totalidad de la norma IEC 62443-4, pues se busca adoptar solamente algunos de sus aspectos básicos para el sistema embebido.
- Los recursos monetarios empleados para adquirir los componentes necesarios para el sistema embebido corren por cuenta del investigador, lo cual representa una limitación en caso de que, a lo largo del proceso, se determine que se requiere de algún componente con costo relativamente elevado.
- El tiempo necesario para realizar las pruebas y determinar exhaustivamente la viabilidad de una posterior implementación en otros lugares podría no ser el suficiente.

Se pretende con estas proyecciones cubrir los aspectos más relevantes, delimitando de mejor manera todo el proceso investigativo, de manera tal que el producto final cumpla con las expectativas generadas.

CAPÍTULO 2: MARCO TEÓRICO

2.1. Introducción

El marco teórico es el soporte teórico contextual de los conceptos que se utilizan para el planteamiento del problema de la investigación, ya que permite justificar, demostrar y apoyar los resultados de una forma ordenada (Zita, s.f.). Seguidamente, se describen los aspectos que se consideran relevantes desde el punto de vista teórico y que ayudarán al lector a tener una comprensión de los temas que se desarrollan a lo largo del documento.

2.2. Contexto de la Organización

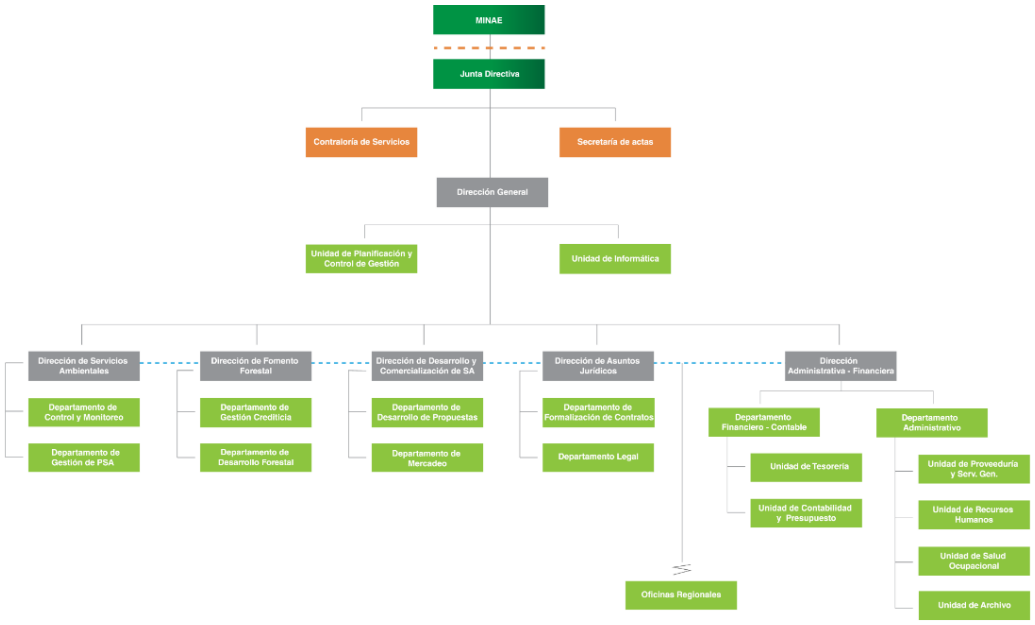
Según se indica en el sitio web oficial de Fonafifo (2018, a):

> Durante la década de 1990, Costa Rica experimentó un cambio en el sector ambiental, caracterizado por un impulso en la legislación que favorece la conservación y protección de los recursos naturales, la creación de instituciones que remozan el sector y un cambio significativo en la forma en que la sociedad percibe el manejo, la conservación y el desarrollo sostenible de los recursos naturales. (párr.1)

Es por ello que, en el año de 1996, mediante la Ley N° 7575, denominada Ley Forestal, se crea el Fondo Nacional de Financiamiento Forestal (en adelante Fonafifo), la cual es una institución pública adscrita al Ministerio de Ambiente y Energía (MINAE), que, para el beneficio de pequeños y medianos productores, se encarga de financiar los procesos de forestación, reforestación, viveros forestales, sistemas agroforestales, recuperación de áreas denudadas y los cambios tecnológicos en áreas de aprovechamiento e industrialización de los recursos forestales (Fonafifo, 2018, b).

Su estructura organizativa, como se observa en la figura 1, está compuesta por algunas direcciones, departamentos y unidades, entre las que se encuentra la Unidad de Informática, la cual se encarga de los temas relacionados con el soporte y mantenimiento de la infraestructura y la plataforma tecnológica de la institución.

Figura 1 *Organigrama de Fonafifo*



Nota. Tomado del sitio oficial del Fonafifo (Fonafifo, 2018, c).

Es importante resaltar nuevamente la problemática que se tiene en cuanto al monitoreo de las redes inalámbricas de la institución, la cual es la base de la presente investigación, dado que no se cuenta actualmente con una herramienta que permita realizar esta tarea de una manera efectiva, por lo que el riesgo de pasar por alto tráfico o actividades sospechosas en este contexto de las redes inalámbricas es muy alto.

2.3.Red Inalámbrica

Una red inalámbrica, “WiFi” por su nombre en inglés, es un entorno que permite que los dispositivos, como computadoras portátiles, tabletas, teléfonos inteligentes, entre otros, se conecten a la red de una empresa, pero sin usar cables. Además, se pueden tener *Access Points*, los cuales amplifican las señales, de forma que un dispositivo puede estar alejado de un enrutador, pero aun así permanecer conectado a la red (Cisco, 2024).

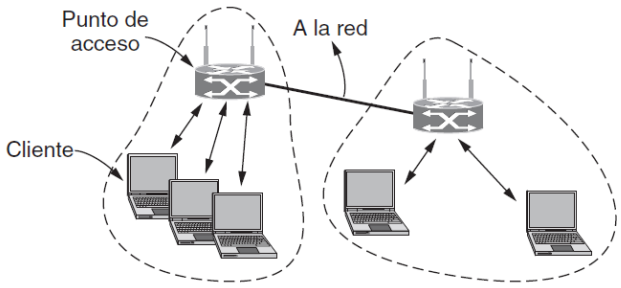
El estándar principal para este tipo de redes es el 802.11, del cual, según mencionan

Tanenbaum y Wetherall (2012), su forma más común de uso corresponde a:

> Conectar clientes, como laptops y teléfonos inteligentes, a otra red, como la intranet de una empresa o Internet. En el modo de infraestructura, cada cliente se asocia con un AP (Punto de Acceso, del inglés Access Point) que a su vez está conectado a la otra red. El cliente envía y recibe sus paquetes a través del AP. Se pueden conectar varios *Access Points* juntos, por lo general mediante una red alámbrica llamada sistema de distribución, para formar una red 802.11 extendida. En este caso, los clientes pueden enviar tramas a otros clientes a través de sus APs. (p.257)

Este tipo de conexión se visualiza en la figura 2.

Figura 2 *Aplicación del estándar 802.11*



Nota. Tomado de Tanenbaum y Wetherall (2012, p.257)

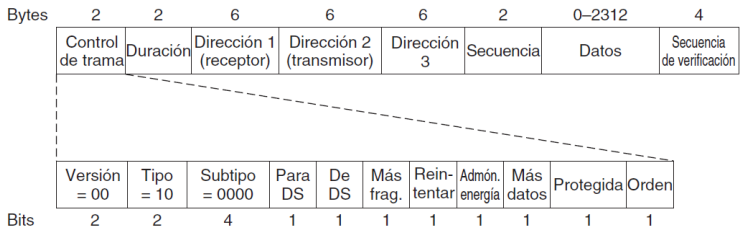
De este estándar, derivan dos técnicas de transmisión que se basan en el esquema de Multiplexación por División de Frecuencia Ortogonal (OFDM), las cuales se llaman 802.11a, que usa una banda de 5 GHz; mientras que la segunda se denomina 802.11g, con una banda de 2.4 GHz; ambas técnicas ofrecen tasas de transmisión de 54 Mbps (Tanenbaum y Wetherall, 2012, p.258).

Las técnicas 802.11 utilizan radios de corto alcance para transmitir señales en las bandas indicadas, con la ventaja de que no necesitan licencia, por lo cual pueden usarse de manera libre

para cualquier transmisor. Sin embargo, tienen la desventaja de que los fabricantes de mecanismos para apertura de puertas de cochera, teléfonos inalámbricos, hornos de microondas y otros dispositivos usan este mismo espectro, es decir, la banda de 2.4 GHz es la más saturada. Por su parte, la banda de 5 GHz resulta mejor para otras aplicaciones, a pesar de que tiene un alcance más corto, debido a que la frecuencia es más alta (Tanenbaum y Wetherall, 2012, p.259).

La información es transmitida por medio de tramas, las cuales corresponden a la unidad de datos que se transmite entre los nodos de la red (Huawei, 2023). Por su parte, Tanenbaum y Wetherall (2012) mencionan que “el estándar 802.11 define tres clases diferentes de tramas en el aire: de datos, de control y de administración. Cada una tiene un encabezado con una variedad de campos que se utilizan dentro de la subcapa MAC” (p.265). En la figura 3, se aprecia la trama correspondiente a los datos.

Figura 3 *Formato de la trama de datos 802.11*



Nota. Tomado de Tanenbaum y Wetherall (2012, p.266)

El análisis de la trama de datos, según Tanenbaum y Wetherall (2012), se describe a continuación como una serie de campos:

Campo de Control de trama, que consta de 11 subcampos. El primero es la Versión de protocolo, que se establece como 00. Está ahí para que las futuras versiones del protocolo 802.11 funcionen al mismo tiempo en la misma celda. Después están los campos de Tipo (de datos, de control o de administración) y de Subtipo (por ejemplo, RTS o CTS). Para una trama de datos regular (sin calidad de servicio), se establecen en 10 y 0000 en binario.

> Los bits Para DS y De DS se establecen para indicar que la trama va hacia o viene de la red conectada a los APS, a la cual se le conoce como sistema de distribución. El bit Más fragmentos indica que siguen más fragmentos. El bit Retransmitir marca una retransmisión de una trama que se envió antes. El bit de Administración de energía indica que el emisor va a entrar al modo de ahorro de energía. El bit Más datos indica que el emisor tiene tramas adicionales para el receptor. El bit Trama protegida indica que el cuerpo de la trama se cifró por seguridad. Por último, el bit de Orden indica al receptor que la capa superior espera que la secuencia de tramas llegue de modo riguroso en orden (p. 266).

Un aspecto que se considera importante de destacar es el cálculo de la propagación de la señal inalámbrica, ya que para efectos de esta investigación se requiere ubicar el sistema embebido en el lugar más adecuado posible, es decir, logrando que la señal llegue desde el *Access Point* (AP) hasta el sistema embebido con la potencia suficiente para evitar precisamente que las tramas de datos por analizar sufran la menor afectación posible. Para ello se puede utilizar el Modelo de Pérdida de Trayectoria de Distancia Logarítmica, el cual predice cómo se atenúa una señal conforme se propaga, para este caso en sitios internos como dentro de un edificio (Rappaport, 2002, pp.102-104). La fórmula corresponde a:

$$PL(d) = PL(d_0) + 10n \log_{10} \frac{d}{d_0} \quad (1)$$

Donde:

- $PL(d)$: Pérdida de la señal a una distancia d, la cual es entre el emisor y el receptor.
- $PL(d_0)$: Pérdida de la señal a una distancia de referencia d_0 (normalmente 1 metro).

- n: Exponente de pérdida de trayectoria, que para este caso corresponde a un valor de 4 por ser dentro de un edificio con paredes.

2.4.Seguridad en Redes Inalámbricas

Una vulnerabilidad en el ámbito informático, Santos (2024) menciona que es “cualquier fallo o error en el software o en el hardware que hace posible a un atacante o hacker comprometer la integridad y confidencialidad de los datos que procesa un sistema”. Por lo tanto, una vulnerabilidad puede deberse a fallos de diseño, errores en la configuración e inclusive, a actualizaciones de seguridad no aplicadas a los equipos en general debido a temas de obsolescencia tecnológica. Además, son una de las principales razones por las que una empresa puede sufrir un ciberataque.

Noguera (2024) señala que algunas de las vulnerabilidades más comunes en las redes inalámbricas son:

- Contraseñas débiles o fáciles de identificar: Uso de contraseñas simples como "123456" o "clave", lo que facilita el acceso no autorizado.
- Uso del protocolo WEP (*Wired Equivalent Privacy*): Protocolo de seguridad obsoleto y vulnerable a ataques de descifrado sencillos.
- Configuraciones predeterminadas en los *Access Points*: Uso de SSID y contraseñas preestablecidas sin cambios, lo que expone la red a atacantes que conocen dichas configuraciones.
- Firmware desactualizado en los *Access Points*: Falta de actualizaciones que corrigen vulnerabilidades conocidas, dejando abierta la puerta a posibles ataques.

Por otra parte, Hewlett Packard Enterprise (2024) menciona que una amenaza “se refiere a cualquier situación o caso que pueda tener consecuencias negativas para las operaciones,

funciones, marca, reputación o imagen percibida de una empresa". En lo que respecta a las amenazas más comunes de las redes inalámbricas, Rozario (2024) destaca las siguientes:

- Los atacantes pueden escuchar pasivamente las comunicaciones inalámbricas entre dispositivos y *Access Points* debido a su naturaleza abierta, obteniendo datos sensibles.
- La saturación de las redes inalámbricas mediante tráfico masivo, como solicitudes falsas, lo que provoca lentitud o interrupciones en el funcionamiento normal de la red.
- Acceso no autorizado por parte de un atacante, con lo que puede tomar control de un dispositivo conectado a la red, robar credenciales, instalar malware o modificar configuraciones.
- Dispositivos inalámbricos que son propensos a la interferencia de radiofrecuencia, por lo que se genera ruido o solapamiento de las frecuencias, lo que afecta el rendimiento de la red.
- *Access Points* que imitan redes legítimas para engañar a los usuarios y capturar datos o interceptar información transmitida.

2.5.Normativa de Seguridad

Este tipo de normativas son un conjunto de estándares o buenas prácticas para proteger los sistemas informáticos, redes y datos contra amenazas cibernéticas, estableciendo controles y medidas de seguridad para prevenir, detectar y responder a incidentes de seguridad, con el fin de asegurar la confidencialidad, integridad y disponibilidad de la información (Datos101, 2023).

Específicamente para esta investigación se toma como referencia la norma ANSI/ISA-62443-4-2-2018 forma parte de esta serie de estándares internacionales para la seguridad en

sistemas de control industrial (ICS) y redes críticas. Esta norma específica se enfoca en los requisitos de seguridad para los componentes de sistemas de control industrial, como dispositivos y equipos conectados, estableciendo las medidas de protección en áreas como la autenticación, el control de acceso, la protección contra malware, la integridad de las comunicaciones y la auditoría de eventos, con el fin de reducir las vulnerabilidades y proteger la infraestructura crítica frente a amenazas (ANSI/ISA, 2019).

2.6.Protocolos de Seguridad en Redes Inalámbricas

Son conjuntos de reglas y estándares que han sido diseñados para proteger las comunicaciones en las redes inalámbricas contra accesos no autorizados y también ataques cibernéticos. Los principales protocolos de seguridad usados en redes inalámbricas corresponden a WPA, WPA2, WPA3, y como referencia es posible mencionar el WEP, que sin embargo, ya se encuentra obsoleto (Kaspersky, 2024). Seguidamente se describe brevemente cada uno de ellos:

- WEP (*Wired Equivalent Privacy*): Creado en 1997, fue el primer protocolo de seguridad Wifi, diseñado para proporcionar una seguridad similar a las redes cableadas. Utiliza claves de cifrado estáticas y el algoritmo RC4, pero tiene vulnerabilidades graves que permiten descifrado rápido y acceso no autorizado, por lo que actualmente está obsoleto.
- WPA (Wifi Protected Access): Desarrollado en 2003 como solución provisional al WEP, usa el protocolo TKIP (*Temporal Key Integrity Protocol*), que genera claves dinámicas para cada paquete de datos.
- WPA2 (*Wifi Protected Access 2*): Data de 2004, y es una mejora respecto al WPA que usa el protocolo AES (*Advanced Encryption Standard*) para el cifrado. Es el estándar predominante y mucho más seguro, sin embargo cuenta con ciertas

vulnerabilidades que es necesario prestar atención.

- WPA3 (*Wifi Protected Access 3*): Se publicó en 2018, siendo la versión más reciente y segura. Incluye características como *Simultaneous Authentication of Equals* (SAE), que protege contra ataques de fuerza bruta y mejora la seguridad incluso en redes públicas, así como la confiabilidad de la información (Kaspersky, 2024).

2.7.Gestión de Riesgos

IBM (s.f.) describe la gestión de riesgos como "el proceso de identificar, evaluar y controlar los riesgos financieros, legales, estratégicos y de seguridad para el capital y las ganancias de una organización". Lo anterior teniendo claro que estos riesgos pueden tener origen en diferentes ámbitos, como el financiero, legal, la gestión de la empresa como tal, los desastres naturales, y por supuesto, la seguridad de la información.

Existen diferentes estándares que describen buenas prácticas acerca de la gestión de los riesgos, de los cuales Frett (2018) menciona los pasos necesarios desde el punto de vista de la norma ISO 31000 2018:

- Identificación: Reconocer los riesgos que pueden afectar los objetivos de la organización, considerando factores internos y externos.
- Análisis: Evaluar la naturaleza, probabilidad e impacto de los riesgos identificados para entender su relevancia.
- Valoración: Comparar los resultados del análisis con los criterios de riesgo definidos para priorizar qué riesgos requieren atención inmediata.
- Tratamiento: Diseñar e implementar estrategias para mitigar, evitar, transferir o aceptar los riesgos, asegurando su control dentro de los límites aceptables.

En este punto la mitigación de los riesgos tiene una importancia muy alta, ya que busca minimizar el nivel de los riesgos para que lleguen a niveles tolerables, esto con el fin de planificar las acciones por llevar a cabo ante situaciones que de alguna manera son inevitables, pero reduciendo su impacto en la continuidad del negocio (Finn y Downie, 2024).

2.8.Sistemas Embebidos

Rodríguez (s.f.) define un sistema embebido dentro del ámbito de los dispositivos electrónicos que "tiene inteligencia computacional, que está diseñado para cumplir una o varias tareas relacionadas que se determinan desde el diseño y por lo tanto, son predecibles al ejecutarse en tiempo real y que está integrado por componentes de hardware y software". Estos sistemas están inmersos en los distintos dispositivos que son de uso diario por parte de las personas, como un reloj despertador, el teléfono inteligente, una tableta, el televisor, entre otros (ver figura 4).

Figura 4 *Ejemplos de sistemas embebidos*

Nota. Tomado de López (2023)

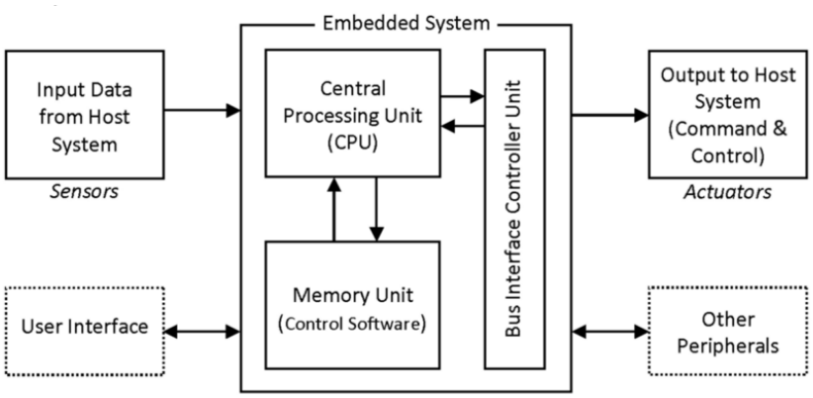
González (2024) menciona los siguientes componentes que son parte de un sistema embebido:

- Microcontrolador o Microprocesador: El cerebro del sistema, encargado de ejecutar el software embebido.

- Memoria: Incluye memoria volátil (RAM) y no volátil (ROM, Flash) para almacenar el código y los datos.
- Periféricos de entrada/salida (I/O): Interfaces para interactuar con otros dispositivos, como sensores, actuadores e interfaces de comunicación.
- Software embebido: El programa que controla el hardware y realiza las funciones específicas del sistema.

En la figura 5 se aprecia un esquema general que incluye los componentes antes mencionados.

Figura 5 *Esquema general de un sistema embebido*



Nota. Tomado de Jitendra et al (2021)

Como se indicó anteriormente, uno de los componentes principales dentro de este entorno de sistemas embebidos es el microprocesador, que para efectos de la presente investigación está representado por la plataforma Raspberry Pi. Al respecto, MCI Electronics (s.f.) menciona que este se trata de “una computadora de bajo costo y con un tamaño compacto, del porte de una tarjeta de crédito, puede ser conectada a un monitor de computador o un TV, y usarse con un mouse y teclado estándar”.

Por su parte, en la actualidad se cuenta con una serie de modelos de Raspberry Pi con

características y capacidades distintas. Entre estas se encuentran:

- La serie insignia, que se conoce con la abreviatura "Raspberry Pi", ofrece hardware de alto rendimiento, un sistema operativo Linux completo y una variedad de puertos comunes, con un tamaño similar al de una tarjeta de crédito.
- La serie *Zero* trae un sistema operativo Linux completo y puertos esenciales en un formato mínimo con bajo consumo de energía.
- La serie *Compute Module*, conocida con la abreviatura "CM", que ofrece hardware de alto rendimiento y un sistema operativo Linux completo en un formato mínimo adecuado para aplicaciones industriales e integradas. Los usuarios deben conectar los módulos que necesiten.
- También existe la serie Pico, los cuales no ejecutan Linux ni permiten almacenamiento extraíble, pero se pueden programar por medio de la instalación de un binario en el almacenamiento flash integrado (Raspberry Pi, 2024).

En la figura 6 se presenta una placa de Raspberry Pi según la serie insignia.

Figura 6 *Raspberry Pi 5*

Nota. Tomado de Raspberry Pi (2024)

Las características técnicas se presentan en la tabla 1.

Tabla 1 *Características técnicas del Raspberry Pi 5*

Característica	**Descripción**
Memoria	2 GB, 4 GB, 8GB
GPIO (*General Purpose Input Output*)	40 pines
Conectividad	2 × micro HDMI
	2 × USB 2.0; 2 × USB 3.0
	2 × Cámara CSI / Puerto de pantalla DSI
	Conector PCIe FFC
	Conector UART
	Conector para batería RTC
	Conector de cuatro pines para abanico JST-SH PWM
	Conector PoE+-capable Gigabit Ethernet (1Gb/s)
	Conexión 2.4/5GHz dual-band 802.11ac Wi-Fi 5 (300Mb/s)
	Bluetooth 5, Bluetooth Low Energy (BLE)
	Ranura para tarjeta microSD
	Conector fuente de alimentación USB-C (5V, 5A (25W) or 5V, 3A (15W) limitado a 600mA)

Nota. Elaboración propia con base en Raspberry Pi (2024)

Un componente importante dentro de estos sistemas es el *software* embebido, el cual ISO/IEC/IEEE (2010) en la norma ISO/IEC/IEEE 24765:2010 lo define como “un *software* que forma parte de un sistema más grande y realiza algunos de los requisitos de ese sistema” (p.122). Este software es posible desarrollarlo en una variedad de herramientas, de las cuales

InnovaciónDigital30 (2023) menciona algunas de ellas:

- Linux Embebido: Basado en el kernel de Linux, es utilizado en dispositivos de alta gama y ofrece ventajas como el software de código abierto y una gestión avanzada de hardware.
- Eclipse: Un IDE de código abierto que es ampliamente utilizado en el desarrollo de sistemas embebidos. Ofrece soporte para varios lenguajes de programación y arquitecturas.
- IAR Embedded Workbench: Un IDE popular para sistemas embebidos, con soporte para una amplia gama de microcontroladores.
- Keil MDK: Especializado en microcontroladores ARM, ofrece un conjunto completo de herramientas de desarrollo.
- Code Composer Studio: Desarrollado por Texas Instruments, es un IDE para la programación de microcontroladores TI MSP430 y otros dispositivos.

Asimismo, los lenguajes de programación que pueden utilizarse para este propósito también son variados, entre los que se encuentran C, C++, Python y Ensamblador, para lo cual es necesario utilizar alguna de las herramientas de desarrollo anteriormente mencionadas (González, 2024).

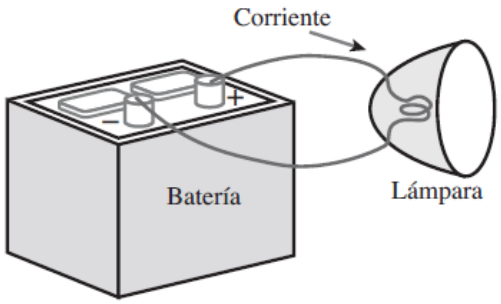
2.9.Circuitos Eléctricos - Electrónicos

Respecto a este tema, Alexander y Sadaki (2006) señalan que “en ingeniería eléctrica, a menudo interesa comunicar o transferir energía de un punto a otro. Hacerlo requiere una interconexión de dispositivos eléctricos. A tal interconexión se le conoce como circuito eléctrico, y a cada componente del circuito como elemento” (p.4).

En la figura 7 se presenta un ejemplo de circuito eléctrico básico, el cual se compone de

una fuente de alimentación, una batería y alambres para realizar las conexiones necesarias.

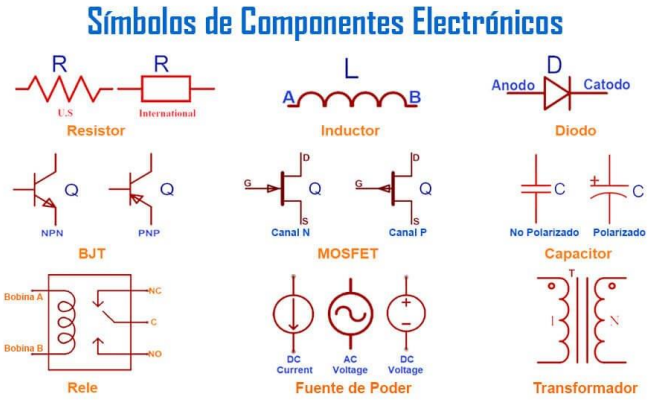
Figura 7 *Circuito eléctrico básico*



Nota. Tomado de Alexander y Sadaki (2006)

A nivel electrónico, estos circuitos pueden estar compuestos por gran variedad de componentes, cada uno de los cuales hace que el comportamiento de la corriente eléctrica varíe según los requerimientos para los cuales se esté desarrollando el circuito. En la figura 8 se presentan algunos de los componentes más utilizados en circuitos.

Figura 8 *Componentes electrónicos más utilizados en circuitos*



Nota. Tomado de ElectrónicaOnline (s.f.)

2.10. Componentes de Hardware

Para efectos de la presente investigación, el hardware corresponde a un conjunto de componentes electrónicos que conforman al sistema embebido que también sirven de soporte

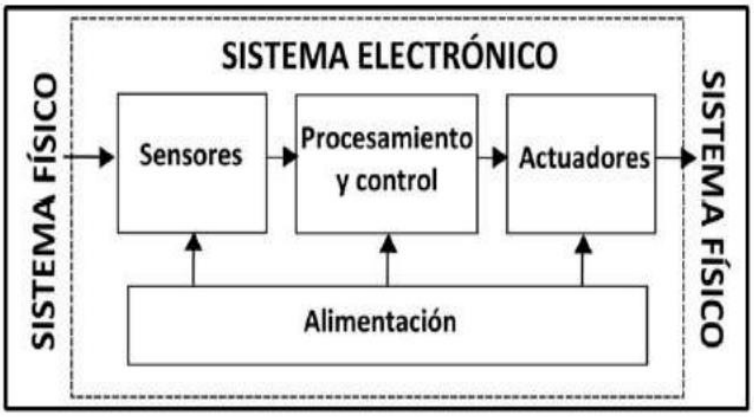
para otros componentes del sistema o permiten al usuario interactuar con él (Seoane, 2016, p.14).

Parte de estos componentes corresponden a los sensores y actuadores, de los cuales Vásquez (s.f.) menciona que:

> Son interfaces del sistema ciberfísico con el mundo real. Los sensores permiten adquirir las variaciones de las magnitudes físicas y convertirlas en señales eléctricas, y los actuadores permiten convertir señales eléctricas en acciones o variaciones de magnitudes físicas. Las magnitudes o señales físicas pueden ser de tipo mecánico, término, eléctrico, óptico y químico (p.5).

En la figura 9 se muestra un esquema del funcionamiento típico de estos componentes.

Figura 9 *Funcionamiento típico de los sensores y actuadores*



Nota. Tomado de Zamora (2019)

Para el caso de la conectividad inalámbrica del sistema embebido, el Raspberry Pi 5 cuenta con un controlador integrado, llamado Broadcom BCM43456, que es un dispositivo de un solo chip que proporciona integración para un sistema inalámbrico portátil o móvil, y que cumple con el estándar IEEE 802.11ac (MAC/banda base/radio) integrando además Bluetooth 4.2 (Broadcom Corp, s.f.). Este componente se visualiza en la figura 10.

Figura 10 *Conectividad inalámbrica del Raspberry Pi 5*

Nota. El módulo inalámbrico corresponde al elemento encerrado en color rojo

Otro elemento que forma parte del hardware del sistema embebido corresponde a la pantalla LCD, la cual está construida a base de un material cuya polarización de la luz varía en presencia de un campo eléctrico denominado cristal líquido (Zuloaga Izaguirre y Astarloa Cuéllar, 2008). Esta pantalla se compone de dos filas y quince columnas y será utilizada en el prototipo para la presentación de la información de respuesta. En la figura 11 se muestra un ejemplo de este tipo de dispositivo.

Figura 11 *Pantalla LCD*

Nota. Tomado de Söderby (2021)

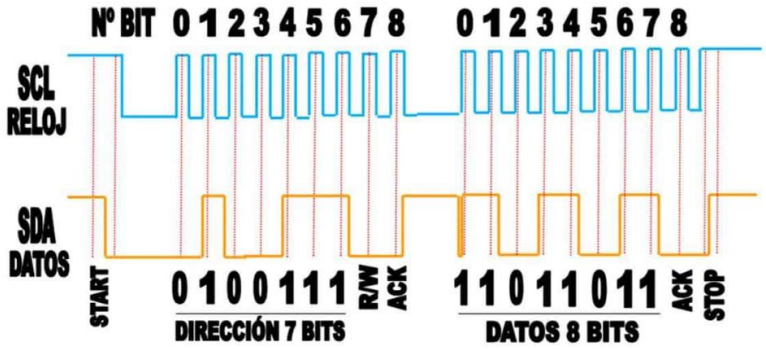
Este tipo de pantallas utilizan un protocolo de comunicación llamado I2C, el cual MCI Educación (2022) lo describe como:

Un protocolo destinado a permitir que varios circuitos integrados digitales (“chips”)

"periféricos" se comuniquen con uno o más chips "controladores". Al igual que la Interfaz Periférica Serial (SPI), sólo está pensada para comunicaciones de corta distancia dentro de un único dispositivo. Al igual que las Interfaces Seriales Asíncronas (como RS-232 o UARTs), sólo requiere dos cables de señal para intercambiar información.

Su funcionamiento se presenta en la figura 12, donde se aprecian las dos señales que se utilizan para este propósito, una llamada SCL (señal de reloj) y otra llamada SDA (señal de datos).

Figura 12 *Operación del protocolo I2C*

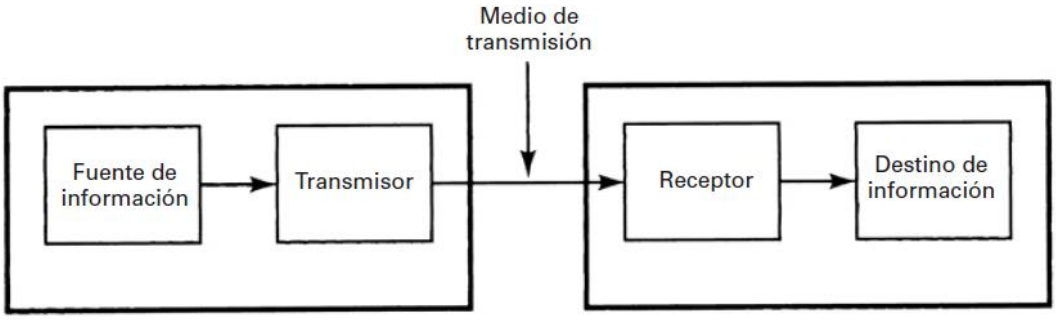


Nota. Tomado de Rivera (2020)

2.11. Señales Inalámbricas

Dado que esta investigación se enfoca en monitoreo de señales provenientes de una red inalámbrica, es importante mencionar que el aspecto que la engloba es que todo corresponde a un sistema de comunicaciones electrónicas, el cual tiene como objetivo fundamental la transferencia de información de un lugar a otro, gracias a una señal llamada "portadora", la cual se encarga de modular la señal de origen gracias a que es una señal de mayor frecuencia; es decir, la modulación se encarga de cambiar una o más propiedades de la señal portadora de acuerdo con la señal que transmite la información (Tomasi, 2003, pp.1-2). Este sistema se muestra en la figura 13.

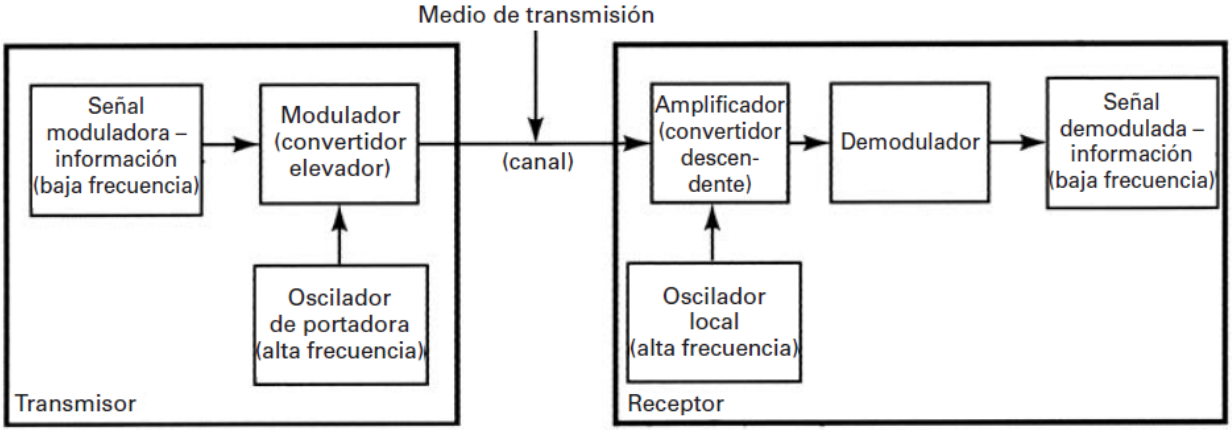
Figura 13 *Diagrama de bloques simplificado de un sistema de comunicaciones electrónicas*



Nota. Tomado de Tomasi (2003, p.2)

Por su parte, la demodulación es el proceso inverso a la modulación, es decir, toma la portadora y la convierte nuevamente en la señal de información original. Todo este proceso de modulación – demodulación es necesario ya que es difícil irradiar señales de baja frecuencia en forma de energía electromagnética con una antena, y podría darse el caso de que las señales de la información ocupen la misma banda de frecuencias, pudiendo darse una interderencia entre sí (Tomasi, 2003, p.3). La figura 14 muestra un ejemoplo completo incluyendo los procesos de modulación y demodulación de señales.

Figura 14 *Diagrama de bloques incluyendo modulación y demodulación*



Nota. Tomado de Tomasi (2003, p.4)

En el caso de la transmisión de las señales inalámbricas, se trata de propagación de ondas electromagnéticas por el espacio libre, o bien, propagación de radiofrecuencia, donde viajan a una velocidad de 300000 Km/s. El espacio libre implica al vacío, por lo que con frecuencia a la propagación por la atmósfera terrestre se le llama propagación por el espacio libre, sin embargo hay que tener claro que la atmósfera terrestre hace que las señales presenten pérdidas (Tomasi, 2003, p.347).

Para lo anterior, Tomasi (2003) menciona que se utiliza una antena, la cual "es un sistema conductor metálico capaz de radiar y capturar ondas electromagnéticas" (p.371). La antena se encarga de la irradiación y captura de energía, donde una línea de transmisión conecta la energía de un transmisor o receptor con una antena, que a su vez transmite la energía hacia la atmósfera y viceversa. En el extremo del transmisor la antena convierte la energía eléctrica de la línea de transmisión en ondas electromagnéticas que se emiten al espacio, mientras que en el receptor, la antena recibe esas ondas y las convierte de nuevo en energía eléctrica a través de la línea de transmisión (Tomasi, 2003, p.371). En la figura 15 se puede observar un de los tipos de antena que existen.

Figura 15 *Antena usada en comunicaciones inalámbricas dentro de edificios*

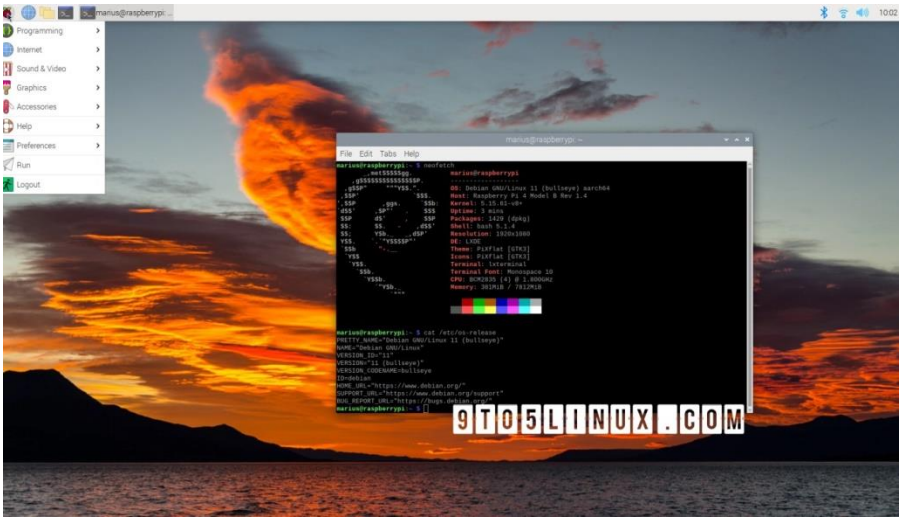
Nota. Este tipo de antena es uno de los más comunes para *routers* inalámbricos o *Access Points*

2.12. Sistema Operativo

Este corresponde, desde el punto de vista de *software*, a la base sobre la cual va a trabajar el sistema embebido (ver figura 16). Halfacree (2024) señala que:

> Un Raspberry Pi es capaz de ejecutar una amplia gama de programas de *software* y diversos sistemas operativos que son el software principal que hace que un ordenador funcione. El más popular de estos, y el oficial, es Raspberry Pi OS. Se basa el Debian Linux, ha sido adaptado especialmente para Raspberry Pi y trae una gran cantidad de programas preinstalados y listos para usar.

Figura 16 *Vista del sistema operativo Raspberry Pi OS*



Nota. Tomado de Nestor (2022)

Al ser una versión derivada de Linux Debian, prácticamente los comandos utilizados para su administración resultan ser los mismos. Todos ellos al igual que en cualquier versión de Linux, son ejecutados desde la aplicación llamada “Terminal”, ya que realizar este manejo del sistema operativo y muchas de sus configuraciones resulta ser más eficiente y rápido que hacer uso de la interfaz gráfica.

En la tabla 2 se describen brevemente los más relevantes para este propósito.

Tabla 2 *Principales comandos del Raspberry Pi OS*

Comando	Descripción
sudo apt update	Actualiza la lista de paquetes disponibles en los repositorios.
sudo apt upgrade	Actualiza los paquetes instalados a sus versiones más recientes.
sudo apt install <paquete>	Instala un paquete específico.
sudo apt remove <paquete>	Elimina un paquete instalado.
sudo reboot	Reinicia la Raspberry Pi.
sudo shutdown	Apaga la Raspberry Pi de manera segura.
ls	Muestra el contenido del directorio actual.
cd <directorio>	Cambia al directorio especificado.
pwd	Muestra la ruta del directorio actual.
mkdir <nombre>	Crea un nuevo directorio.
rm <archivo>	Elimina un archivo.
rmdir <directorio>	Elimina un directorio vacío.
cp <origen> <destino>	Copia archivos o directorios.
mv <origen> <destino>	Mueve o renombra archivos o directorios.
nano <archivo>	Abre el editor de texto Nano para editar archivos.
top	Muestra los procesos en ejecución y su consumo de recursos.
ps aux	Muestra todos los procesos en ejecución.

Comando	Descripción
df -h	Muestra el espacio libre y usado en los discos.
free -h	Muestra la memoria RAM disponible y usada.
ifconfig	Muestra la configuración de las interfaces de red.
ping <dirección>	Envía paquetes ICMP a una dirección para comprobar la conectividad de red.
hostname	Muestra el nombre del host de la Raspberry Pi.
sudo raspi-config	Accede a la herramienta de configuración del sistema Raspberry Pi.
sudo systemctl start <servicio>	Inicia un servicio del sistema (por ejemplo, un servidor web).
sudo systemctl stop <servicio>	Detiene un servicio en ejecución.
sudo systemctl restart <servicio>	Reinicia un servicio.

Nota. Elaboración propia con base en Fromaget (s.f.)

Un aspecto relevante a este punto es que bajo el sistema operativo Raspberry Pi OS se pueden realizar tareas de análisis de tráfico de una red. Fortinet (2024) indica que “el tráfico de red se refiere a la cantidad de datos que se mueven a través de una red informática en cualquier momento dado”. Por su parte, el análisis de tráfico de red es una técnica que se usa para examinar la actividad de la red, permitiendo gestionar la disponibilidad, solucionar problemas de ancho de banda, y además detectar y corregir actividades que pudieran resultar sospechosas y que atenten contra la seguridad de la información (Fortinet, 2024).

Una de las herramientas utilizadas para el monitoreo y análisis de tráfico de redes en general es *Wireshark*, la cual permite efectuar una inspección de protocolos, sean de capa física,

de enlace de datos, red, transporte y además de aplicación. La captura de información la realiza en tiempo real, ya sea desde una interfaz de red cableada o inalámbrica, para luego analizar toda esa información sin necesidad de estar en línea (De Luz, 2024).

Otra herramienta utilizada para este propósito corresponde a NMAP, de la cual Nmap.org (s.f.):

> Nmap (*Network Mapper*) es una herramienta de código abierto para la exploración de redes y la auditoría de seguridad. Fue diseñada para escanear rápidamente redes grandes, aunque funciona bien contra hosts individuales. Nmap utiliza paquetes IP sin procesar de formas novedosas para determinar qué hosts están disponibles en la red, qué servicios (nombre y versión de la aplicación) ofrecen esos hosts, qué sistemas operativos (y versiones del SO) están ejecutando, qué tipo de filtros de paquetes/cortafuegos están en uso y docenas de otras características. Si bien Nmap se usa comúnmente para auditorías de seguridad, muchos administradores de sistemas y redes lo encuentran útil para tareas rutinarias como inventario de red, administración de cronogramas de actualización de servicios y monitoreo del tiempo de actividad de host o servicio.

En este punto es importante aclarar que ambas herramientas son de uso libre, es decir, desde la perspectiva legal no hay problema en utilizarlas; sin embargo, a lo que sí se debe prestar atención es al uso que se le dé a la información que es posible obtener por medio de ambas, ya que al ser medios para recolectar información variada de una red, quedan expuestas muchas de las vulnerabilidades a las que está expuesta la red de datos de la organización en donde se estén ejecutando.

Es por ello que esta información debe ser utilizada únicamente con el propósito de realizar las mejoras necesarias que desde el punto de vista de seguridad de la información requiere la organización.

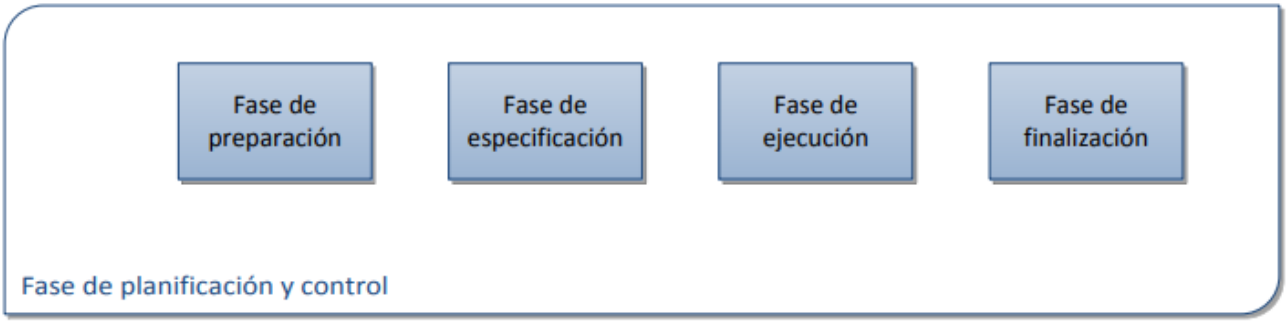
2.13. Evaluación del Sistema Embebido

Una evaluación de diferentes aspectos permite determinar si el sistema embebido logra cumplir con las expectativas para las cuales está diseñado. Algunos de los aspectos relevantes que se deben tomar en cuenta a la hora de realizar esta evaluación son:

- Desempeño: Medir la rapidez de procesamiento, tiempo de respuesta y eficiencia en el uso de recursos (CPU, memoria, etc.).
- Precisión: Evaluar la exactitud de las mediciones o resultados que genera el sistema, especialmente en aplicaciones de control y sensado de datos.
- Facilidad de uso: Valorar la interfaz de usuario, la facilidad de configuración, y la accesibilidad para el usuario final.
- Fiabilidad: Analizar la estabilidad y la capacidad del sistema para funcionar correctamente en condiciones prolongadas o adversas.
- Consumo de energía: Evaluar el uso de la energía, especialmente en sistemas portátiles o de bajo consumo (Mosquera, 2019, pp.19-24).

Parte de esta evaluación corresponde además a una etapa de pruebas, donde en la figura 17 se presenta una breve descripción gráfica del ciclo de vida de las pruebas en general.

Figura 17 *Ciclo de vida de las pruebas*



Nota. Tomado de Montero (2014, p.19)

Adicionalmente, Montero (2014) menciona estos tipos de pruebas:

- Pruebas unitarias: Comprueban el funcionamiento del sistema por separado.
- Pruebas de integración: Comprueban el funcionamiento de las interfaces entre los componentes o partes del sistema.
- Pruebas de sistema: Comprueban el comportamiento de todo el sistema.
- Pruebas funcionales: Comprueban la funcionalidad del sistema, es decir, en lo que hace este sistema.
- Pruebas de regresión: Comprueban cambios en el software provocados por modificaciones.
- Pruebas de rendimiento: Comprueban la rapidez de un sistema en realizar ciertas tareas en unas condiciones particulares de trabajo (p.23).

Para el caso de la efectividad en cuanto a la detección de situaciones sospechosas, Vectra (2025) menciona que "la tasa de detección suele calcularse como una relación entre el número de detecciones positivas verdaderas (amenazas reales correctamente identificadas) y el número total de amenazas reales". Para estos efectos, se utiliza la siguiente fórmula:

$$Tasa\ de\ detección = \frac{Cantidad\ de\ verdaderos\ positivos}{Total\ de\ alertas\ generadas}\ x\ 100 \qquad (2)$$

Donde:

- $Tasa\ de\ detección$: Eficacia del sistema embebido para detectar amenazas.
- $Cantidad\ de\ verdaderos\ positivos$: Cantidad de situaciones sospechosas que el sistema embebido detectó y eran reales, ya que fueron comparadas con otra herramienta de detección paralelo al sistema embebido.
- $Total\ de\ alertas\ generadas$: Cantidad de situaciones sospechosas que fueron detectadas por el sistema embebido.

2.14. Análisis Financiero

Para efectos del análisis financiero del sistema embebido, es importante iniciar con lo que se conoce como ROI, del cual hace mención el Grupo BBVA (2024):

> El Retorno de la Inversión (ROI) es una métrica financiera que evalúa el rendimiento económico de una inversión en relación con su coste. Esta medida es crucial para las empresas y los inversores, ya que ayuda a evaluar la rentabilidad y eficacia de las inversiones realizadas.

Para calcularlo se utiliza la siguiente fórmula:

$$ROI = \frac{Beneficio\ Obtenido\ Neto}{Coste\ de\ la\ Inversión}\ x\ 100 \qquad (3)$$

Donde:

- $Beneficio\ Obtenido\ Neto$: Ingresos o ganancias generadas por la inversión - Coste de la inversión.
- $Coste\ de\ la\ inversión$: Monto total invertido en el proyecto o sistema.

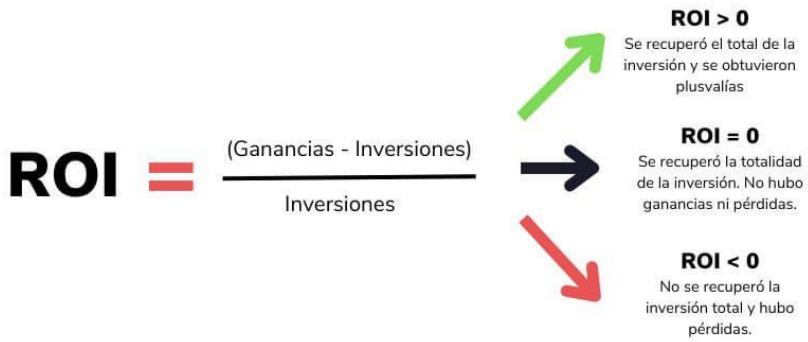
Una vez hecho el cálculo, si el resultado es positivo quiere decir que la inversión ha sido rentable, y entre más alto, mayor rentabilidad se habrá obtenido (Grupo BBVA, 2024). Adicionalmente, existen estrategias que ayudan a que este indicador mejore, como las siguientes:

- Optimización de costes: Es decir, buscando formas de reducir los costos operativos, ya sea mediante la optimización de la cadena de suministro, negociación de precios más bajos con proveedores, implementación de tecnologías más eficientes.
- Eficiencia operativa: Una forma indirecta por medio de la identificación y eliminación de procesos ineficientes o de la inversión en el desarrollo de habilidades y conocimientos de los empleados.

- Mejorar los ingresos: De forma tal que mejore el balance entre lo invertido y lo ganado, ya que además de reducir los costes, también aumentan los ingresos (Grupo BBVA, 2024).

En la figura 18 se presenta una breve descripción del concepto de ROI y su interpretación.

Figura 18 *ROI y su interpretación*



Nota. Tomado de Montezzana (2023)

Un aspecto por considerar es que el coste de la inversión mencionado en la fórmula del ROI incluye los costos que en los que se incurre para implementar soluciones innovadoras dentro de una organización, como por ejemplo, la capacitación, adopción de tecnología, cambios de procesos y actualizaciones de infraestructura, siendo estos los costos de implementación (Faster Capital, 2024).

2.15. Análisis Costo - Beneficio

Deloitte (2024) indica que este tipo de análisis corresponde a:

Una herramienta de toma de decisiones utilizada para analizar los beneficios económicos netos de un proyecto, política o inversión o para elegir entre distintas iniciativas. Análisis y clasificación de proyectos basados en valores económicos, sociales y ambientales, así como análisis de eficacia y eficiencia".

De ahí que, para efectos de verificar si el sistema embebido propuesto es viable, es necesario realizar un análisis de este tipo que permita determinar si los beneficios que brindará cumplen con las expectativas para el cual fue diseñado, claro está que también se debe valorar ese componente económico que representa uno de los factores determinantes dentro de cualquier proyecto.

CAPÍTULO 3: MARCO METODOLÓGICO

3.1.Enfoque de la Investigación

El enfoque utilizado es el denominado mixto, del cual Salas (2019) menciona que es “un proceso que recolecta, analiza y vierte datos cuantitativos y cualitativos, en un mismo estudio”. Esto por cuanto es el que mejor se adapta a los objetivos planteados. Al examinar los aspectos de seguridad de la red inalámbrica y la identificación de vulnerabilidades utilizando la norma IEC 62443-4, se utiliza el análisis cualitativo que permite una comprensión más profunda de las vulnerabilidades identificadas y las mejores prácticas para mitigarlas, proporcionando un contexto para la interpretación de los resultados.

Asimismo, en el diseño del sistema embebido con Raspberry Pi para el monitoreo de la red se obtienen datos cuantitativos, como la cantidad de alertas generadas y el rendimiento del sistema; donde al mismo tiempo con un análisis cualitativo que considere las características técnicas y operativas del sistema, permite una mejor evaluación de su efectividad. Finalmente, al evaluar el retorno de inversión (ROI), se combinan datos económicos cuantitativos con un análisis cualitativo que refleje el impacto del sistema en cuanto a su seguridad y eficiencia.

3.2.Método de la Investigación

Se utiliza un método llamado DMAIC, el cual es un acrónimo donde cada letra significa una etapa dentro de un proceso, es decir “D”: definir, “M”: medir, “A”: analizar, “I” mejorar (del inglés *improve*) y “C”: controlar. Este método busca la mejora a partir de datos, procurando la detección y eliminación de las ineficiencias que pueden resultar en defectos (Trout, 2021).

Cada una de estas etapas dentro de la investigación representa lo siguiente:

- **Definir:** Se define la problemática que se desea abordar, la cual ya fue debidamente planteada y corresponde a la necesidad de una mejora en el monitoreo de la red inalámbrica del Fonafifo.

- **Medir:** Se recopilan datos relevantes sobre el estado actual de la red inalámbrica, así como vulnerabilidades detectadas, todo ello utilizando algunos principios de la norma IEC 62443-4.
- **Analizar:** Se realiza un análisis de los datos recopilados para identificar las causas raíz de las vulnerabilidades y problemas de seguridad. Aplicando análisis de datos, se interpretan los resultados de las mediciones, permitiendo entender mejor las deficiencias del comportamiento actual de la red inalámbrica.
- **Mejorar:** Se diseñará e implementará el sistema embebido con Raspberry Pi, con base en los resultados del análisis. Esto incluirá la captura y análisis de información en tiempo real, así como la generación de alertas ante actividades sospechosas.
- **Controlar:** Se monitorea y evalúa la efectividad del sistema embebido una vez que se haya implementado. Esto incluirá la evaluación del retorno de inversión (ROI) y la comparación de costos y beneficios con soluciones actuales del mercado.

3.3.Fuentes de Información

Maranto y González (2015) mencionan que una fuente de información “es todo aquello que nos proporciona datos para reconstruir hechos y las bases del conocimiento. Las fuentes de información son un instrumento para el conocimiento, la búsqueda y el acceso de a la formación” (p.1).

Para efectos de esta investigación se utilizan fuentes de información como la documentación en general relacionada con temas de seguridad de la información, como la norma IEC 62443-4, arquitectura y configuración del Raspberry Pi, herramientas de monitoreo de la red inalámbrica así como la información que estas generan para su análisis, además del personal de TI del Fonafifo.

3.4.Instrumentos y Técnicas

Los instrumentos y técnicas que se utilizan se resumen en los siguientes párrafos, donde se procura hacer un barrido por cada uno de los objetivos de la investigación. Inicialmente se examinan los aspectos de seguridad de la red inalámbrica e identifican las vulnerabilidades, comenzando con la declaración del problema, además de realizar un análisis de causa raíz usando la técnica de los cinco porqués, para identificar vulnerabilidades, y una matriz de probabilidad e impacto asociada a los riesgos presentes, finalizando con una matriz de priorización para evaluar y clasificar dichas vulnerabilidades.

En la fase de diseño del sistema embebido, se elabora un plan de implementación que detalle cómo se llevará a cabo el desarrollo del sistema. Se desarrollan prototipos y pruebas para validar el diseño, y se lleva a cabo una lluvia de ideas para generar posibles soluciones potenciales que aborden las necesidades de seguridad. Para interpretar los resultados de las mediciones, se emplea la herramienta de análisis de Pareto, el cual permite identificar las principales causas de problemas mediante la evaluación de datos, y el análisis de correlación para determinar relaciones entre variables de rendimiento.

En la fase de evaluación del retorno de inversión (ROI) del sistema embebido, se generan informes de control que resumen el desempeño del sistema y análisis del ROI. Además, se utilizan herramientas de monitoreo continuo para evaluar el rendimiento del sistema en tiempo real y gráficos de control para visualizar el rendimiento y detectar variaciones. Finalmente, en la fase de control, se establece un cronograma de revisiones periódicas para llevar a cabo auditorías y evaluaciones de la efectividad continua del sistema, que permiten determinar si su rendimiento es el adecuado.

3.5.Variables de Análisis

Seguidamente se presenta en la tabla 3 las variables de análisis que son utilizadas para la

investigación. Estas corresponden a todo aquello que se puede medir, es decir, la información que es recabada con el fin de dar respuesta a la problemática planteada (Villasís y Miranda, 2016).

Tabla 3 *Esquema de variables de análisis del marco metodológico*

Objetivos específicos	**Variable**	**Etapas y Actividades**	**Técnicas a utilizar**	**Sujetos y Fuentes de información**
1. Examinar los aspectos de seguridad de la red inalámbrica para la identificación de vulnerabilidades, utilizando la norma IEC 62443-4 como marco de referencia.	Vulnerabilidades identificadas.	**Etapa 1:** Definición de Vulnerabilidades. **Actividades:** • Análisis de riesgos. • Identificación de vulnerabilidades. • Análisis de causa raíz. • Priorización de áreas críticas.	• Análisis documental. • Cinco porqués. • Matriz de priorización.	• Personal de TI de Fonafifo. • Documentación de seguridad de redes inalámbricas. • Documentación oficial de la norma IEC 62443-4.
2. Diseñar un sistema embebido con Raspberry Pi para el monitoreo de la red inalámbrica, que permita la captura y análisis de información, y la generación de alertas en caso de actividad sospechosa.	Desempeño del sistema embebido.	**Etapa 2:** Implementación y Medición del Sistema Embebido. **Actividades:** • Diseño del sistema embebido. • Desarrollo de circuitos para interfaces. • Creación del prototipo. • Instalación y configuración de Sistema Operativo en el Raspberry Pi.	• Lluvia de ideas. • Simulaciones de circuitos. • Prototipo. • Pruebas de integración de hardware y software. • Herramientas de monitoreo.	• Documentación oficial de Raspberry Pi. • Foros de Electrónica y Raspberry Pi. • Tutoriales en línea sobre Raspberry Pi y Linux. • Literatura sobre comunicaciones electrónicas.
3. Interpretar los	Efectividad del	**Etapa 3:** Análisis de	• Análisis de	• Registros del

Objetivos específicos	Variable	Etapas y Actividades	Técnicas a utilizar	Sujetos y Fuentes de información
resultados de las mediciones realizadas para la evaluación de la efectividad del sistema embebido, por medio del análisis de la información recabada.	sistema embebido.	Resultados. **Actividades:** • Análisis de datos generados por las alertas. • Realización de pruebas del sistema.	Pareto. • Análisis de tendencias de los datos recopilados.	sistema embebido. • Herramientas de monitoreo de red. • Páginas web de análisis de datos y estadísticas.
4. Evaluar el retorno de inversión (ROI) del sistema embebido, determinando su impacto económico y en seguridad, y comparando los costos y beneficios con una solución actual del mercado.	Retorno de inversión.	**Etapa 4:** Evaluación del ROI. **Actividades:** • Análisis de costos. • Comparación con soluciones existentes. • Establecimiento de un cronograma de revisiones periódicas.	• Análisis de costo-beneficio. • Informes de control. • Gráficos de control. • Auditorías periódicas.	• Personal de TI de Fonafifo. • Artículos sobre ROI en tecnología. • Páginas web sobre análisis de ROI.

Nota. Elaboración propia (2024)

Como se observa en la tabla 3, cada objetivo específico está ligado a una etapa que se ha determinado con base en el método DMAIC, y a su vez, está acompañado de una serie de actividades, técnicas a utilizar y sujetos y fuentes de información. Estos últimos son componentes clave que permiten obtener la información necesaria para ser procesada mediante las distintas técnicas, asegurando que cada etapa logre cumplir con su rol dentro de la investigación.

CAPÍTULO 4: ANÁLISIS DE RESULTADOS

4.1.Análisis de Vulnerabilidades y Causa Raíz

Actualmente la institución cuenta con un concentrador Cisco WC 2500 que se encuentra fuera de soporte técnico por parte del fabricante. Adicionalmente, existen cuatro *Access Points* distribuidos dos en cada piso, con el fin de brindar cobertura a los usuarios. En la tabla 4 se describen de manera resumida las vulnerabilidades detectadas para esta red inalámbrica.

Tabla 4 *Vulnerabilidades detectadas en la red inalámbrica de Fonafifo Sede Central*

Vulnerabilidad	**Descripción**	**Apartado de la norma IEC 62443-4 asociado**
Equipos fuera de soporte.	El concentrador Cisco WC 2500 ya no recibe soporte ni actualizaciones de seguridad por parte del fabricante.	Sección 13.5 EDR 3.10 – Soporte para actualizaciones.
Tráfico sin filtrar.	La falta de filtrado en el tráfico de la red aumenta el riesgo de ataques, como por ejemplo el *Man in then middle.*	Sección 15.6 NDR 3.2 – Protección contra código malicioso.
Distribución de *Access Points* (APs) sin control de cobertura.	Mala distribución de los APs puede generar áreas de baja cobertura y puntos débiles en la red.	Sección 9.3 CR 5.1 – Segmentación de la Red.
Intercepción de señales.	La comunicación inalámbrica es susceptible a ataques de *sniffing* o *jamming*.	Sección 7.3 CR 3.1 – Integridad en las Comunicaciones.
Ausencia de monitoreo en tiempo real.	La red carece de monitoreo continuo, lo que dificulta la detección temprana de incidentes.	Sección 10.4 CR 6.2 - Monitoreo continuo.

Nota. Elaboración propia con base en ANSI/ISA (2019)

Para la definición de estas vulnerabilidades se utilizó la técnica de los cinco porqués, en donde se describe mediante una serie de preguntas y respuestas la justificación de la elección de cada una de ellas. La quinta respuesta se considera que es la causa raíz de la vulnerabilidad

planteada.

1. **Equipos fuera de soporte.**

¿Por qué los equipos están fuera de soporte?

El concentrador Cisco WC 2500 ya está obsoleto y Cisco ya no ofrece soporte.

¿Por qué no se ha reemplazado el equipo?

El costo de actualizar los equipos se ha considerado demasiado alto.

Adicionalmente se han experimentado limitaciones presupuestarias a nivel institucional.

¿Por qué no se ha considerado el riesgo de no tener soporte?

No se ha evaluado adecuadamente el impacto de no tener actualizaciones de seguridad.

¿Por qué no se ha realizado un análisis de riesgo sobre este aspecto?

Por falta de un enfoque proactivo en la gestión del ciclo de vida de la red inalámbrica.

¿Por qué no se gestiona el ciclo de vida correctamente?

Para las adquisiciones en la administración pública se toma en cuanta con un mayor peso el tema presupuestario, por lo que este tipo de limitaciones constantemente se traen abajo proyectos relacionados con tecnología.

2. **Tráfico sin filtrar.**

¿Por qué no se filtra el tráfico de red inalámbrica?

No se han implementado políticas de filtrado de tráfico en la red.

¿Por qué no se ha implementado el filtrado?

Se asumió que este tráfico era seguro y no había necesidad de filtrarlo.

¿Por qué se asumió eso?

Falta de conciencia de los riesgos asociados al tráfico no filtrado.

¿Por qué no hay conciencia de esos riesgos?

La ciberseguridad implementada se ha enfocado más en la red cableada.

¿Por qué no hay una estandarización en la implementación de aspectos de ciberseguridad en toda la red por igual?

La red fue diseñada inicialmente sin tener en cuenta los estándares de seguridad adecuados para la red inalámbrica.

3. **Distribución de *Access Points* (APs) sin control de cobertura.**

¿Por qué la distribución de APs no tiene control?

Los APs fueron instalados sin realizar un estudio de cobertura.

¿Por qué no se realizó un estudio de cobertura?

Se asumió que la red inalámbrica no tendría problemas de cobertura.

¿Por qué se asumió eso?

Se consideró que no se tendrían problemas de cobertura debido a los materiales con que están hechas las divisiones de las distintas oficinas.

¿Por qué no se consideró la interferencia?

Falta de experiencia en el diseño adecuado de redes inalámbricas.

¿Por qué falta experiencia?

Los encargados de la red no tenían los conocimientos adecuados en cuanto al diseño e implementación de redes inalámbricas seguras.

4. **Intercepción de señales**

¿Por qué la red es vulnerable a la interceptación?

No se implementa cifrado en las señales inalámbricas.

¿Por qué no se implementa cifrado?

Se considera innecesario ya que la red está limitada a usuarios internos.

¿Por qué se considera innecesario?

Falta de comprensión de los riesgos asociados a la interceptación de señales.

¿Por qué no hay comprensión de los riesgos?

La seguridad de la red inalámbrica no se ha considerado como un tema crítico.

¿Por qué no se considera crítico?

No se implementaron protocolos de seguridad adecuados durante el diseño de la red inalámbrica.

5. **Ausencia de Monitoreo en Tiempo Real**

¿Por qué no hay monitoreo en tiempo real?

La red no cuenta con sistemas de monitoreo adecuados, debido a la antigüedad del equipo controlador.

¿Por qué no se implementaron sistemas de monitoreo?

Se consideró que la red no necesitaría monitoreo constante.

¿Por qué se consideró innecesario el monitoreo?

No se evaluaron los beneficios de tener un sistema de detección temprana de amenazas.

¿Por qué no se evaluaron esos beneficios?

Falta de un enfoque proactivo en la seguridad de la red.

¿Por qué falta ese enfoque?

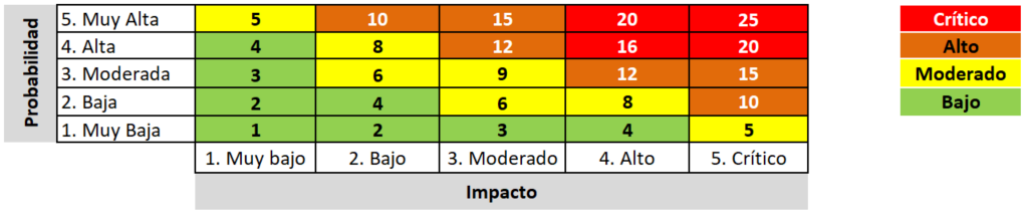
La seguridad de la red inalámbrica no ha sido gestionada adecuadamente.

4.2.Análisis de Riesgos

Seguidamente se presenta un análisis de riesgos relacionado con las vulnerabilidades detectadas. Para ello se utilizó un mapa de calor que muestra diferentes escalas de probabilidad e

impacto, y su intersección o multiplicación da como resultado un nivel de riesgo. En la figura 19 se describe dicho mapa, mientras que la tabla 5 contiene el análisis realizado.

Figura 19 *Mapa de calor utilizado para el análisis de riesgos*



Nota. Elaboración propia (2025)

Tabla 5 *Análisis de riesgos con base en las vulnerabilidades detectadas*

Vulnerabilidad	Riesgo	Probabilidad	Impacto	Riesgo (P x I)	Nivel
Equipos fuera de soporte.	Explotación de vulnerabilidades no corregidas.	4	5	20	Crítico
Tráfico sin filtrar.	Ataques *Man in the middle* y *phishing* al no filtrar el tráfico.	4	5	20	Crítico
Distribución de *Access Points* (APs) sin control de cobertura.	Pérdida de conectividad o acceso no autorizado en zonas de baja cobertura.	3	4	12	Alto
Intercepción de señales.	Intercepción y manipulación de datos por canales inalámbricos no cifrados.	3	4	12	Alto
Ausencia de monitoreo en tiempo real.	No detectar intrusiones o anomalías de tráfico a tiempo.	5	5	25	Crítico

Nota. Elaboración propia (2025)

De la tabla 5 se desprenden tres riesgos críticos, de los cuales se da una leve calificación más alta precisamente al que se considera que puede desencadenar los otros riesgos planteados, y es la falta de monitoreo en tiempo real, que como se mencionó en el capítulo 1, es una funcionalidad que no posee el equipo controlador de la red inalámbrica actualmente debido a que ya no cuenta con soporte del fabricante y por lo tanto no hay forma de actualizarlo a una versión más reciente.

4.3.Análisis de Prioridades de Atención

A continuación en la tabla 6 se muestra una matriz que sintetiza las prioridades con las que se deben atender las vulnerabilidades detectadas.

Tabla 6 *Priorización en la atención de las vulnerabilidades detectadas*

Vulnerabilidad	Riesgo total	Prioridad
Ausencia de monitoreo en tiempo real.	25	Alta
Equipos fuera de soporte.	20	Alta
Tráfico sin filtrar.	20	Alta
Distribución de *Access Points* (APs) sin control de cobertura.	12	Media
Intercepción de señales.	12	Media

Nota. Elaboración propia (2025)

Es importante mencionar que todas estas vulnerabilidades deben ser atendidas para mejorar la seguridad de la red, sin embargo, para efectos del presente trabajo solamente se abordó la que corresponde al monitoreo de la red, dado que es el propósito principal de la propuesta que se presenta más adelante.

4.4.Descripción General del Sistema Embebido Propuesto

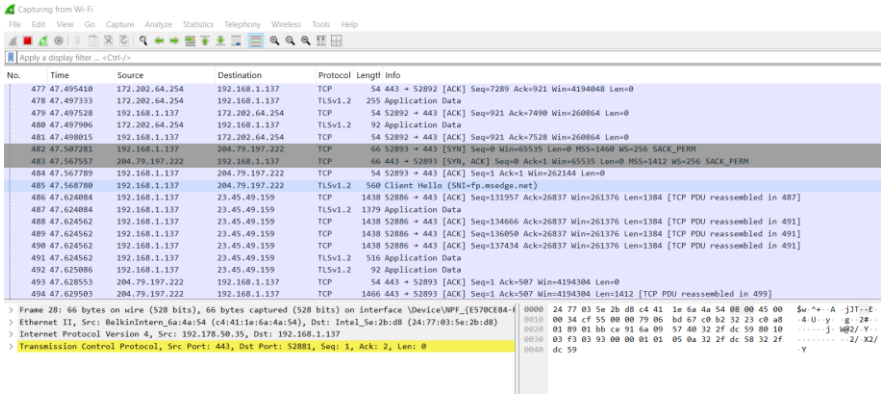
El sistema embebido se implementó haciendo uso de un Raspberry Pi 5, con el sistema operativo Raspberry Pi OS de 64 bits. Las herramientas de *software* utilizadas para el monitoreo

de la red inalámbrica son Pyshark, y Python como lenguaje de programación para la realización de los *scripts* correspondientes que combinan comandos de Pyshark, manejo de la interface I^2C y pines GPIO del Raspberry Pi, así como el almacenamiento de la información generada por las alertas en un archivo de texto plano. Además, los aspectos relacionados con las alertas visuales son gestionados por medio de una pantalla LCD y LEDs. Todo esto se describe con más detalle en el capítulo 6 de este documento.

Dos aspectos por tomar en cuenta para este sistema embebido son los referentes a la forma en cómo se realizó la recolección de información de tráfico de red y el otro acerca de qué tan efectivo es. En el caso de la información sobre las alertas, el sistema embebido fue implementado del 31 de enero de 2025 a eso de las 6am al 10 de febrero de 2025 hasta aproximadamente las 5pm. Cabe indicar que estuvo en operación durante el horario laboral solamente, es decir, desde aproximadamente las 6am hasta las 5pm (sin incluir sábado y domingo) del período indicado, para un total de 11 horas diarias.

Para la efectividad, se realizó una comparación con los datos generados haciendo uso de la herramienta WireShark instalada en una computadora portátil, con el fin de verificar qué tan acertada es la información que genera el sistema embebido.

Figura 20 *Herramienta Wireshark instalada en una computadora para pruebas*



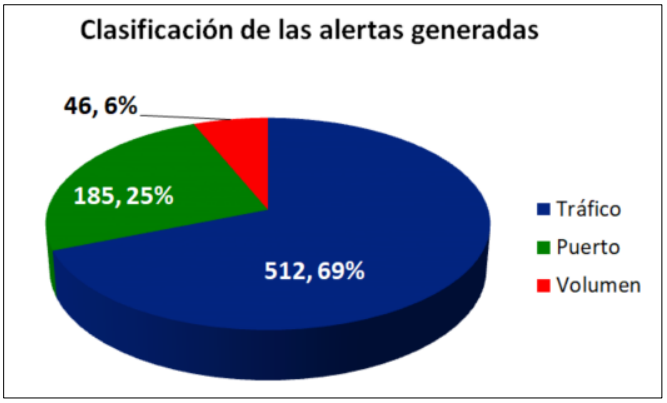
Nota. Captura tomada del equipo utilizado (2025)

Para realizar esta comparativa, la información obtenida con la herramienta Wireshark instalada en la computadora cubre el mismo rango de fechas indicado.

4.5.Análisis de Datos de las Alertas Generadas

En este apartado se toma como base la información del archivo de texto donde se fueron almacenando las alertas generadas, y se procedió a realizar un análisis de datos por medio de una hoja de Excel, con el fin de clasificarlos según las tres categorías de actividad sospechosa definidas en la tabla 9: volumen de paquetes, uso de puertos e IPs sospechosas.

Figura 21 *Clasificación de alertas generadas por el sistema embebido*



Nota. Elaboración propia (2025)

Según se observa en la figura 21, del total de alertas generadas un 69% corresponden a tráfico sospechoso (512 alertas), un 25% a intentos de acceso a puertos (185 alertas) y un 6% a volumen de paquetes (46 alertas), para un total de 743. Al analizar estos datos, el tráfico sospechoso representa la mayoría de los eventos registrados, lo que sugiere la posibilidad de intentos de acceso no autorizado, escaneo de red o tráfico malicioso. Para el caso de los puertos, se detectaron intentos de conexión en puertos que normalmente son usados para ataques, como SSH (22), Telnet (23) o RDP (3389), lo que podría estar representando intentos de explotación de vulnerabilidades en servicios expuestos, es decir, equipos con puertos habilitados que no deberían estar en esa condición. Finalmente, para el volumen de datos sospechoso por medio del protocolo

ICMP se detectaron situaciones en donde se superó el tamaño normal de paquetes, por lo que se podría tratar de intentos de ataques de denegación de servicio (DoS) o intentos de escaneo de red.

Posteriormente se analizó la información contenida en un archivo ".pcap", el cual corresponde al registro generado por el Wireshark instalado en la computadora. Este análisis se realizó mediante un *script* en Python, el cual comparó la información de ambas fuentes de datos, indicando en este caso las alertas de situaciones sospechosas generadas por el sistema embebido que fueron también capturadas en el archivo de Wireshark, obteniendo los siguientes resultados.

Tabla 7 *Comparativa Sistema Embebido vs Wireshark en computadora*

Tipo de evento	**Cantidad detectada por el Sistema Embebido**	**Cantidad detectada en Wireshark**	**Falsos positivos**
Tráfico sospechoso	512	408	104
Puertos	185	142	43
Volumen sospechoso	43	37	9
Total	**743**	**587**	**156**

Nota. Elaboración propia (2025)

La información de la tabla 7 está clasificada por tipo de evento, donde para cada uno de ellos se realizó el conteo de la cantidad de eventos detectados tanto en el sistema embebido como en el Wireshark instalado en la computadora. Los falsos positivos corresponden a situaciones que fueron detectadas por el sistema embebido pero no por Wireshark

A partir de esta información, se calcula mediante la fórmula (2) la tasa de detección del sistema embebido, que en este corresponde al porcentaje de efectividad que tiene:

$$Tasa\ de\ detección = \frac{Cantidad\ de\ verdaderos\ positivos}{Total\ de\ alertas\ generadas}\ x\ 100$$

$$Tasa\ de\ detección = \frac{587}{743}\ x\ 100 = 79{,}004\%$$

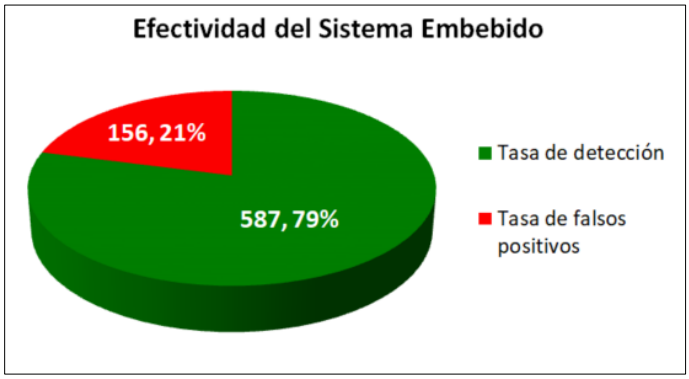
Para el caso de los falsos positivos, se puede aplicar la misma fórmula con los datos correspondientes:

$$Tasa\ de\ falsos\ positivos = \frac{Cantidad\ de\ falsos\ positivos}{Total\ de\ alertas\ generadas}\ x\ 100$$

$$Tasa\ de\ detección = \frac{156}{743}\ x\ 100 = 20{,}996\%$$

En el siguiente gráfico se muestran los resultados de la efectividad del sistema.

Figura 22 *Efectividad del sistema embebido*



Nota. Elaboración propia (2025)

De acuerdo con los resultados mostrados en la figura 22, el sistema embebido cuenta con un porcentaje alto de detecciones reales, comparadas con la información registrada por Wireshark. Sin embargo, el porcentaje de falsos positivos se considera también alto, lo cual no es del todo bueno ya que esto se refiere a situaciones que se están detectando como sospechosas pero que muy probablemente no lo son, ya que no fueron registradas por Wireshark, lo que podría hacer caer a los analistas en un trabajo adicional de análisis de incidentes que quizá no lo amerite, provocando retrasos en la atención de otras situaciones en su quehacer diario.

Es importante tener en cuenta que este sistema centra su análisis de información en *scripts* de Python, por lo que para ajustar esos porcentajes de detección es necesario revisar y depurar el código fuente de los *scripts*.

4.6.Análisis de Costo Beneficio

Una vez probado el sistema se procedió con el análisis costo beneficio, el cual tiene como propósito evaluar la viabilidad económica de implementar este sistema embebido teniendo en cuenta las limitaciones presupuestarias que tiene la institución para adquirir alguna solución ya consolidada del mercado, lo cual, para efectos de este análisis se interpreta como un ahorro ya que no se está llevando a cabo la adquisición del producto como tal. Claro está que lo ideal sería contar con una herramienta robusta y consolidada para el monitoreo de la red inalámbrica, pero no es el caso debido a la limitación presupuestaria actual. En este caso se realizaron dos análisis con dos soluciones de monitoreo, las cuales se describen a continuación.

Comparativa 1: ManageEngine Netflow Analizer (ManageEngine, 2025).

- Costo del sistema embebido: $200 aproximadamente.
- Costos recurrentes: No se cuenta con ellos ya que el sistema fue desarrollado por un funcionario de la institución, por lo que ya implícitamente recibe un salario con el cual debe cubrir no solo este proyecto, sino las demás funciones que tenga asignadas como parte de ese pago mensual que recibe.
- Duración estimada del sistema embebido en producción: 3 años.
- Costos totales: Corresponden al costo del sistema embebido, es decir, $200.
- Costo ManageEngine Netflow Analizer: $8595 por año.
- Beneficios totales: En este caso se interpreta como un ahorro anual de $8595 por la no adquisición de la herramienta ManageEngine Netflow Analizer, siendo en total:

  $$Beneficios\ totales = Costo\ solución * 3 = \$8595 * 3 = \$25785$$

- $Relación\ costo\ beneficio = \frac{Beneficios\ totales}{Costos\ totales} = \frac{\$25785}{\$200} = 128{,}925$
- $Beneficios\ netos = Beneficios\ totales - Costos\ totales = \$25785 - \$200 =$

$25585

- $ROI = \frac{Beneficios\ netos}{Costos\ totales} * 100 = \frac{\$25585}{\$200} * 100 = 12792{,}5$

Según los resultados obtenidos, la relación costo beneficio de 128,925 indica que por cada dólar que se gastó en el sistema embebido se obtendrán a los largo de los 3 años un total de $128,925, siendo esto una rentabilidad bastante alta. En el caso del ROI sucede que al ser de 12792,5 significa que esa inversión inicial de los $200 es muy favorable debido al ahorro que significa no haber adquirido el producto mencionado.

Comparativa 2: PRTG Network Monitor (Paessler, 2025).

- Costo del sistema embebido: $200 aproximadamente.
- Costos recurrentes: No se cuenta con ellos ya que el sistema fue desarrollado por un funcionario de la institución, por lo que ya implícitamente recibe un salario con el cual debe cubrir no solo este proyecto, sino las demás funciones que tenga asignadas como parte de ese pago mensual que recibe.
- Duración estimada del sistema embebido en producción: 3 años.
- Costos totales: Corresponden al costo del sistema embebido, es decir, $200.
- Costo PRTG Network Monitor: $2149 por año.
- Beneficios totales: En este caso se interpreta como un ahorro anual de $2149 por la no adquisición de la herramienta PRTG Network Monitor, siendo en total:

  $Beneficios\ totales = Costo\ solución * 3 = \$2149 * 3 = \$6447$

- $Relación\ costo\ beneficio = \frac{Beneficios\ totales}{Costos\ totales} = \frac{\$6447}{\$200} = 32{,}235$
- $Beneficios\ netos = Beneficios\ totales - Costos\ totales = \$6447 - \$200 = \$6247$

- $ROI = \frac{Beneficios\ netos}{Costos\ totales} * 100 = \frac{\$6247}{\$200} * 100 = 3123{,}5$

Según los resultados obtenidos, la relación costo beneficio de 32,235 indica que por cada dólar que se gastó en el sistema embebido se obtendrán a los largo de los 3 años un total de $32,235, siendo esto una rentabilidad bastante alta. En el caso del ROI sucede que al ser de 3123,5 significa que esa inversión inicial de los $200 es muy favorable debido al ahorro que significa no haber adquirido el producto mencionado.

Estos son indicativos de que el sistema embebido desde el punto de vista de presupuesto es bastante favorable, pero hay que tener en cuenta que las herramientas contra las que se comparó son mucho más robustas dado todas las funcionalidades que incluyen.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1.Conclusiones

De acuerdo con los objetivos planteados y los resultados obtenidos en la presente investigación, se concluye lo siguiente.

1. Desde el punto de vista técnico es factible implementar un sistema embebido como el que se propone en esta investigación, ya que, a nivel tanto de *hardware, software*, interfaces y componentes electrónicos utilizados, se cuenta con todos los recursos disponibles, y que permiten dar la funcionalidad buscada con este sistema, por tanto, dar respuesta a una problemática de seguridad de la información que no solo afecta al Fonafifo, sino también a cualquier otra institución pública o empresa privada.
2. La implementación de aspectos de la norma IEC 62443-4-2 en este sistema embebido fortalece su seguridad al establecer un marco estructurado para la protección contra ciberataques. Si bien es cierto, por motivo de tiempo disponible no se puede aplicar en la totalidad los aspectos previstos en un inicio, sí se tomaron en cuenta aspectos básicos para el tema de mapeo de los riesgos y lo implementado para el monitoreo de la red, y además de ello, se tomó como referencia para la propuesta de mejoras a futuro que pretenden fortalecer este sistema.
3. Si bien el sistema embebido implementado tiene un nivel muy básico de protección, es un buen punto de partida para ir realizando mejoras, principalmente por el tema de limitaciones presupuestarias que están presentes en la realidad de muchas instituciones públicas y que no permiten la inversión en herramientas consolidadas. De ahí la importancia que tiene este proyecto no solo desde el punto de vista académico, sino además por el beneficio que puede traer dada la importancia que reviste la seguridad de la información en la actualidad.
4. La efectividad del sistema embebido para detecciones de actividad sospechosa ronda

el 79%, mientras que para el aspecto de los falsos positivos está en un 21%. Desde el punto de vista del aporte que este sistema puede brindar al Fonafifo para subsanar una necesidad real de monitoreo de la red inalámbrica se considera muy bueno, ya que actualmente este monitoreo es inexistente prácticamente; sin embargo, para efectos de la efectividad que una herramienta de este tipo debe tener, sí se considera que la tasa de falsos positivos es muy alta, lo cual debe ser mejorado.

5. El retorno de inversión evaluado con base en soluciones actuales de mercado es bastante bueno, puesto que este sistema embebido tuvo una inversión inicial muy baja en comparación con los costos de una herramienta del mercado. Ante esta afirmación es importante tener claro que se habla de un ahorro que tuvo la institución pero por no haber adquirido aún una herramienta del mercado debido a las limitaciones de presupuesto que se han venido mencionando.

5.2.Recomendaciones

Como recomendaciones generales se plantean las siguientes:

1. Continuar con el desarrollo del sistema embebido, agregando nuevas funcionalidades para la detección de amenazas y reforzando la seguridad en *hardware* y *software*, según el plan de mejoras planteado en el capítulo 6.
2. Optimizar la detección de amenazas mediante el ajuste de los parámetros con base en el análisis de datos históricos, integrando en la medida de lo posible algoritmos de aprendizaje automático para mejorar la identificación de situaciones sospechosos y reducir los falsos positivos.
3. Implementar el sistema en las Oficinas Regionales del Fonafifo, lo que permite extender los niveles de monitoreo a otras infraestructuras en las que tampoco se cuenta con un monitoreo efectivo de la red inalámbrica.

CAPÍTULO 6: PROPUESTA

6.1.Introducción

De acuerdo con la infraestructura que posee la institución respecto a su red inalámbrica, queda claro que esta no es eficiente desde el punto de vista de monitoreo, puesto que el equipo controlador está obsoleto y desde el punto de vista de ciberseguridad no puede obtener actualizaciones de firmware, situación que no permite estar al tanto como es debido de cualquier situación anómala que se presente en cuanto al tráfico de la red.

Es importante destacar que el sistema embebido propuesto abarca únicamente esta etapa de monitoreo, puesto que las demás vulnerabilidades detectadas deben ser corregidas aplicando otro tipo de configuraciones en otros equipos que conforman la infraestructura de la red en general y que están fuera del alcance de esta propuesta.

Como en todo proyecto tecnológico que en un inicio busca ser una alternativa a productos que ofrece el mercado, este sistema embebido no cuenta con otras funcionalidades que lo hagan más robusto, pero se espera que sirva como punto de partida para lograr un producto más amplio, el cual gradualmente vaya incrementando la cantidad de funcionalidades que ofrezca, para hacerlo más atractivo y poder extenderlo a otras instituciones públicas que cuentan con limitaciones presupuestarias similares a las de Fonafifo.

6.2.Plan de Trabajo

En la tabla 8 se describe el plan de trabajo para la implementación del sistema embebido propuesto.

Tabla 8 *Plan de trabajo para la implementación del sistema embebido*

Actividad	Descripción	Pasos
1. Preparación del Raspberry Pi 5.	Configurar el sistema operativo y herramientas básicas necesarias.	• Instalar Raspberry Pi OS. • Configurar la red, SSH y habilitar I^2C. • Actualizar paquetes del sistema.

Actividad	Descripción	Pasos
2. Configuración del hardware.	Conectar los dispositivos (LCD, LEDs y otros componentes) al Raspberry Pi 5.	• Conectar la pantalla LCD usando I^2C. • Configurar LEDs con resistencias en los pines GPIO. • Verificar las conexiones físicas.
3. Instalación de las librerías y herramientas de software.	Preparar el entorno instalando librerías para hardware y monitoreo de red.	• Instalar bibliotecas como RPi.GPIO y RPLCD. • Instalar las herramientas de captura de tráfico como tcpdump o pyshark.
4. Desarrollo del sistema de monitoreo de tráfico de red.	Programar un sistema para analizar tráfico de red e identificar anomalías.	• Programar un script en lenguaje Python para capturar el tráfico. • Definir los parámetros para clasificar el tráfico sospechoso.
5. Implementación de alertas visuales.	Crear las alertas visuales con base en los eventos detectados.	• Programar LEDs para los estados de alerta. • Configurar la pantalla LCD para mostrar mensajes descriptivos.
6. Integración y pruebas del sistema.	Integrar las funcionalidades y realizar pruebas completas para determinar su efectividad.	• Integrar el monitoreo con las alertas visuales. • Simular tráfico normal y sospechoso. • Capturar tráfico real de la red. • Ajustar los umbrales y mensajes según los resultados. • Comparar la información de las alertas con la obtenida por medio de otra herramienta.

Nota. Elaboración propia (2025)

A continuación se describe cada una de las actividades, de manera tal que el lector pueda dar un recorrido completo por la solución desde su inicio hasta la implementación para la captura y análisis de la información.

6.2.1. Preparación del Raspberry Pi 5

El elemento principal del sistema embebido corresponde al Raspberry Pi 5, que para estos efectos cuenta con un chasis con ventilador que ayuda a la disipación de calor.

Figura 23 *Raspberry Pi 5 utilizado en el sistema embebido*

Nota. Elaboración propia (2025)

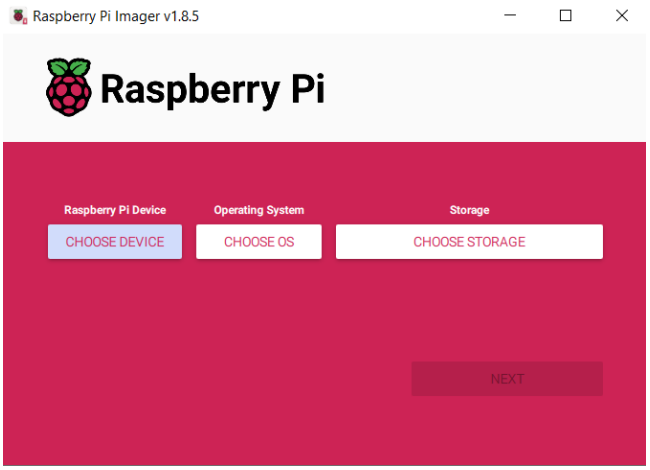
La instalación del sistema operativo Raspberry Pi OS se realiza desde una computadora que tenga un puerto para tarjeta SD ya que este es el medio de almacenamiento que utiliza el dispositivo.

Figura 24 *Medio de almacenamiento del Raspberry Pi 5*

Nota. Elaboración propia (2025)

Para ello es necesario descargar una herramienta llamada “Paspberry Pi Imager”, la cual contiene una imagen del sistema operativo. Esto se realiza desde el enlace https://www.raspberrypi.com/software/.

Figura 25 *Utilitario necesario para la instalación del sistema operativo*



Nota. Captura tomada de la herramienta Raspberry Pi Imager (2025)

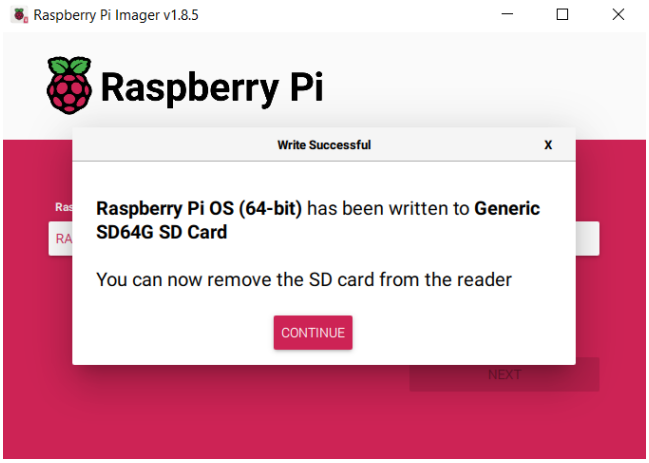
Seguidamente se seleccionan los valores necesarios en la configuración, como el modelo del Raspberry Pi 5, el sistema operativo Raspberry Pi OS (64 bits) y la tarjeta SD donde será almacenado el *software*. Una vez que inicia la instalación puede irse verificando el progreso.

Figura 26 *Valores configurados para la instalación del sistema operativo*

Nota. Captura tomada de la herramienta Raspberry Pi Imager (2025)

Una vez que finaliza la instalación del sistema operativo en la tarjeta SD, es necesario removerla y conectarla al Raspberry Pi 5, para iniciar con su uso.

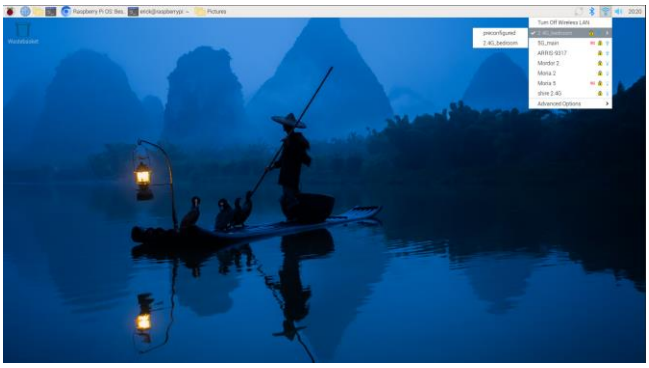
Figura 27 *Instalación del sistema operativo finalizada*



Nota. Captura tomada de la herramienta Raspberry Pi Imager (2025)

Al reiniciar el sistema operativo, se procede a conectar a la red disponible, mediante el ícono ubicado en la barra de tareas en la esquina superior derecha.

Figura 28 *Configuración de la red inalámbrica*



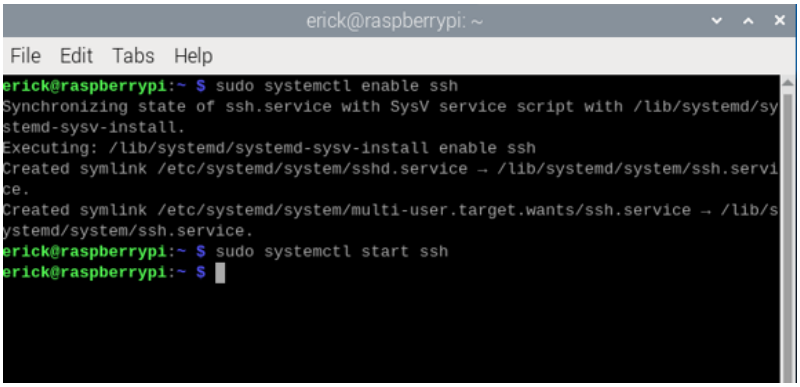
Nota. Captura tomada del Raspberry Pi OS (2025)

El SSH es el protocolo que permite realizar conexiones seguras mediante el puerto 22, y se activa mediante los comandos:

- sudo systemctl enable ssh

- sudo systemctl start ssh

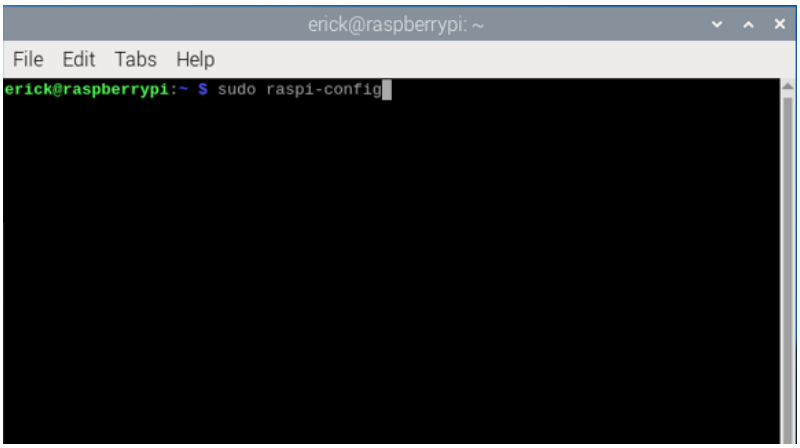
Figura 29 *Activación del protocolo SSH*



Nota. Captura tomada del Raspberry Pi OS (2025)

Luego, para configurar la interface I^2C, se debe ingresar al utilitario raspi-config, el cual entre otros, permite habilitar esta interface para posteriormente poder hacer uso de una pantalla LCD.

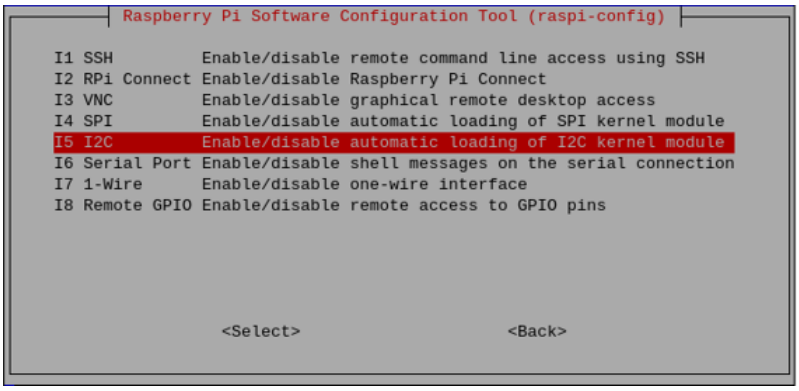
Figura 30 *Comando para ingresar a la configuración del I^2C*



Nota. Captura tomada del Raspberry Pi OS (2025)

En el submenú *Interface options* se puede encontrar la opción para habilitar el puerto I2C, el cual debe estar activo para poder hacer la conexión que se indicó de la pantalla LCD.

Figura 31 *Habilitar la interface I2C*

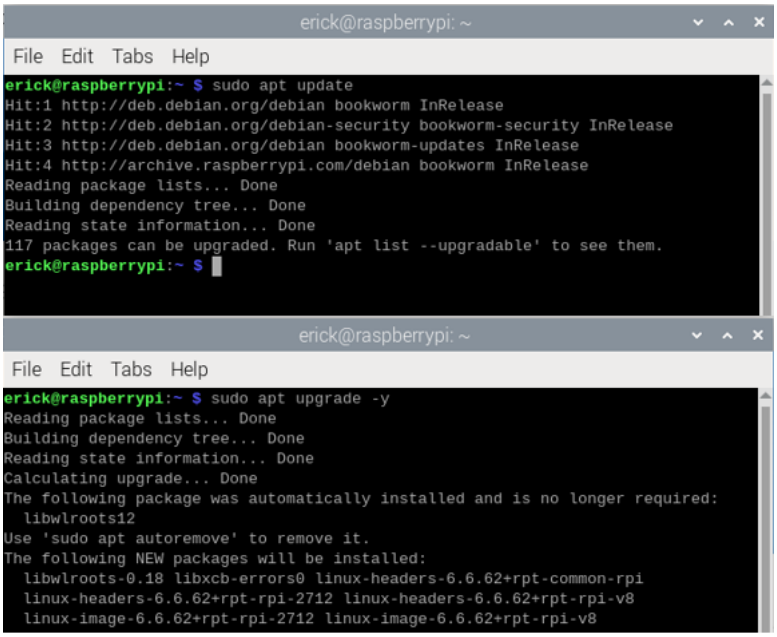


Nota. Captura tomada del Raspberry Pi OS (2025)

Seguidamente se debe realizar una actualización general del sistema operativo, esto debido a que es importante contar con las últimas actualizaciones que el fabricante haya puesto a disposición, principalmente por temas de seguridad. Se utilizan los comandos:

- sudo apt update
- sudo apt upgrade –y

Figura 32 *Actualizar el sistema operativo*



Nota. Captura tomada del Raspberry Pi OS (2025)

6.2.2. Configuración del Hardware

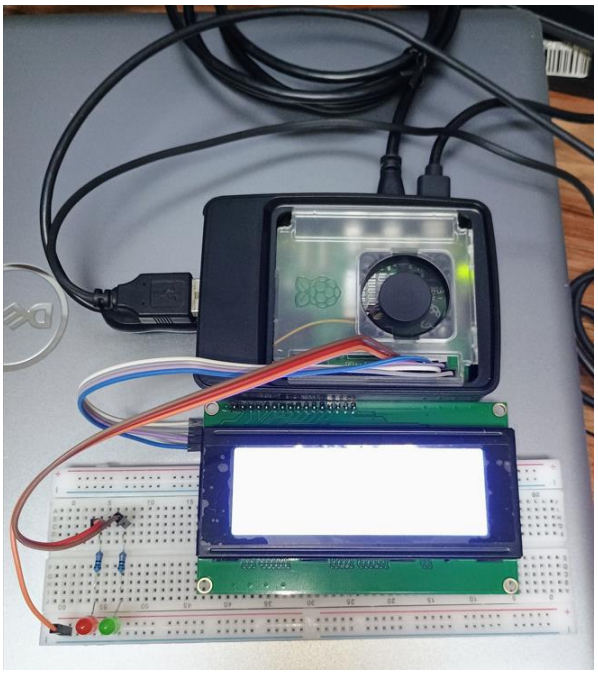
El *hardware* es relativamente sencillo gracias a que la interface I^2C solo tiene cuatro conexiones para la pantalla LCD. De igual forma los LEDs solo requieren de tres conexiones, una de las cuales corresponde a tierra. En la tabla 9 se presentan estas conexiones.

Tabla 9 *Conexiones del hardware con el Raspberry Pi 5*

Componente	**Nombre del pin (número de pin físico)**
Led rojo.	GPIO 17 (11)
Led verde.	GPIO 27 (13)
Conexión a tierra.	GND (9(
Pantalla LCD VCC	5 V (2)
Pantalla LCD tierra	GND (6)
Pantalla LCD SDA	GPIO 2 (3)
Pantalla LCD SCL	GPIO 3 (5)

Nota. Elaboración propia (2025)

Figura 33 *Hardware del sistema embebido*



Nota. Elaboración propia (2025)

6.2.3. Instalación de Librerías y Herramientas de Software

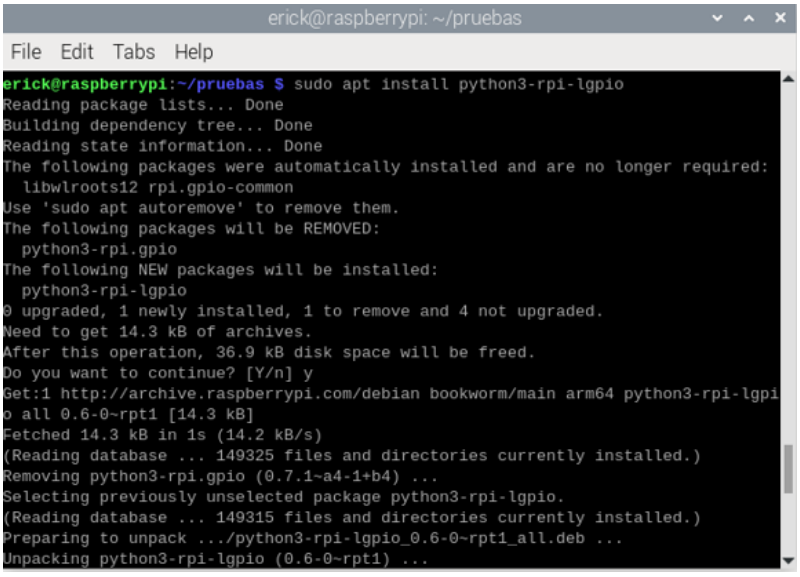
Se requiere de elementos de software para poder hacer que el Raspberry Pi se comunique con el mundo exterior mediante el hardware que se necesita agregar, en este caso la biblioteca LGPIO permite interactuar con los pines GPIO del Raspberry Pi, para controlar dispositivos externos como LEDs, sensores y motores, o bien leer entradas como botones y switches.

Por su parte, RPLCD es otra biblioteca para controlar pantallas LCD, por ejemplo las del tipo 16x2 o 20x4 basadas en el controlador HD44780, ya sea mediante conexiones paralelas o I^2C, y se utiliza para mostrar información o datos en tiempo real.

Para instalar la LGPIO se requiere de los siguientes comandos:

- sudo apt-get update
- sudo apt-get install python3-rpi-lgpio

Figura 34 *Instalar LGPIO*

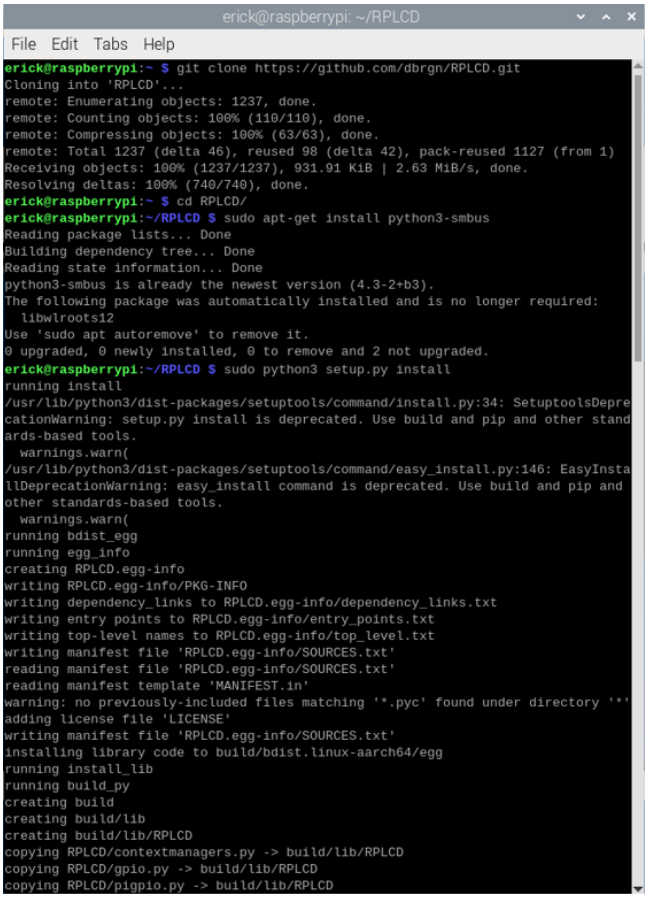


Nota. Captura tomada del Raspberry Pi OS (2025)

Para el caso de la biblioteca RPLCD es necesario clonar el repositorio desde Github y proceder con la instalación, todo ello con estos comandos:

- git clone https://github.com/dbrgn/RPLCD.git
- cd RPLCD
- sudo python3 setup.py install

Figura 35 *Instalar RPLCD*

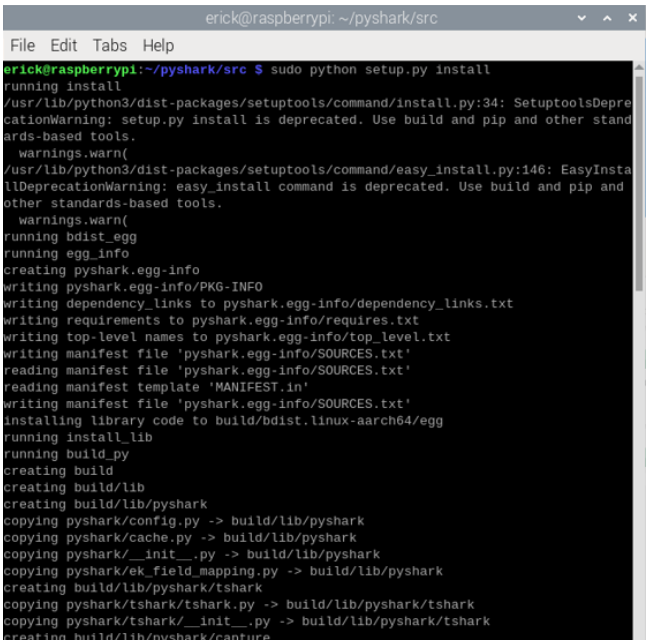


Nota. Captura tomada del Raspberry Pi OS (2025)

A continuación se instala la herramienta que permite hacer el monitoreo del tráfico de la red inalámbrica, llamada Pyshark, con los siguientes comandos:

- sudo apt-get install tshark
- git clone https://github.com/KimiNewt/pyshark.git
- cd pshark/src
- sudo python setup.py install

Figura 36 *Instalar Pyshark*



Nota. Captura tomada del Raspberry Pi OS (2025)

6.2.4. Desarrollo del Sistema de Monitoreo de Tráfico de Red

Una vez que el Raspberry Pi cuenta con las bibliotecas necesarias para poder utilizar tanto los componentes de hardware como la captura del tráfico de la red inalámbrica, es necesario programar un *script* que permita correlacionar estos elementos. El mismo está diseñado en lenguaje Python. Sin embargo, de previo es necesario parametrizar o definir los valores para los tipos de tráfico que podrían considerarse sospechosos. Seguidamente se presenta una serie de aspectos básicos relacionados con ataques que fueron tomados en cuenta para este sistema.

Tabla 10 *Criterios para definir tráfico de red sospechoso*

Parámetro	**Descripción**	**Condición en el Código**
Volumen de paquetes.	Tamaño de los paquetes capturados. Un tráfico elevado podría ser indicador	(packet.length) > TAMANO_PING_MINIMO (por ejemplo 1024 bytes).

Parámetro	Descripción	Condición en el Código
	de un ataque de tipo DoS.	
Uso de puertos.	Acceso a puertos sensibles como 22 (SSH) o 3389 (RDP). Puede indicar un intento de acceso no autorizado.	packet[port] in PUERTOS_SOSPECHOSOS (por ejemplo, [22, 23, 3389]).
IPs sospechosas.	Tráfico proveniente de direcciones IP conocidas por ser maliciosas o fuera de un rango permitido.	source_ip in IPS_SOSPECHOSAS (comparar contra una lista de IPs prohibidas).

Nota. Elaboración propia (2025)

Cabe destacar que las pruebas que se realizaron para la detección de actividad sospechosa fueron con comandos básicos, como ping y ssh, ejecutados desde cualquier equipo que estuviera conectado en la red. .

El *script* básico en lenguaje Python corresponde a las siguientes líneas de código fuente. Cualquier texto que se encuentre después del símbolo # corresponde a un comentario que ayuda a comprender de mejor manera el código fuente.

```
#
import pyshark  # Librería para capturar y analizar paquetes de red
from time import time  # Función para obtener el tiempo en segundos desde el inicio del programa
from datetime import datetime  # Para formatear la fecha y hora de las alertas
# Interfaz de red a monitorear
```

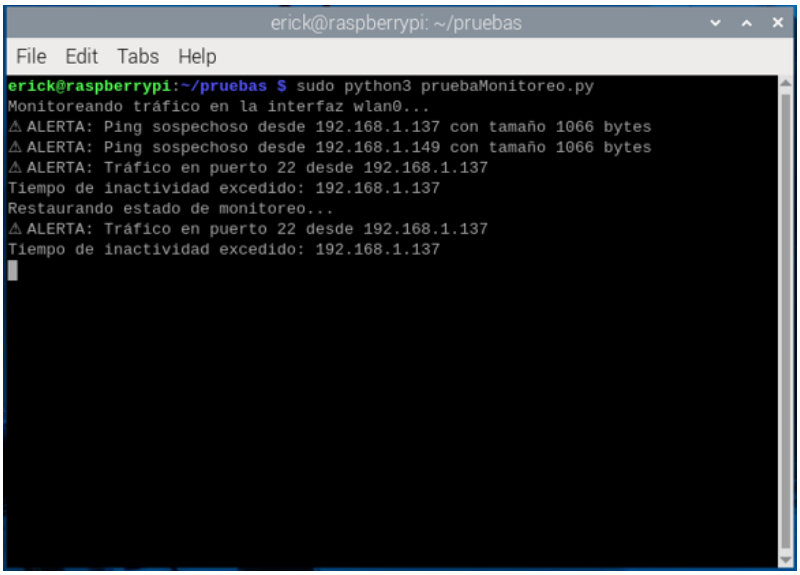
```
interface = 'wlan0'  # Interfaz inalámbrica (ajustar si es diferente)
# Lista de IPs sospechosas
IPS_SOSPECHOSAS = {"192.168.1.100", "10.0.0.50"}  # Direcciones IP que serán
monitoreadas por actividad sospechosa
# Lista de puertos por monitorear
PUERTOS_SOSPECHOSOS = {22, 23, 3389}  # Puertos comunes de servicios como SSH (22),
Telnet (23), RDP (3389)
# Diccionario para rastrear actividad y evitar spam
last_alert_time = {}  # Guarda el tiempo de la última alerta enviada para cada IP
# Tiempo entre alertas de la misma IP
ALERTA_INTERVALO = 10  # Intervalo mínimo entre alertas de una misma IP (en segundos)
# Tamaño mínimo para generar alerta de ping (en bytes)
TAMANO_PING_MINIMO = 1024  # Umbral de tamaño para generar alerta (en bytes)
# Función para registrar alertas en archivo de texto
def registrar_alerta(mensaje):
    with open("alertas.txt", "a") as file:  # Abre el archivo "alertas.txt" en modo append
        file.write(f"{mensaje}\n")  # Escribe el mensaje de alerta en el archivo
# Captura de paquetes en tiempo real
print(f"Monitoreando tráfico en la interfaz {interface}...")  # Muestra mensaje inicial sobre la
interfaz que se está monitoreando
capture = pyshark.LiveCapture(interface=interface)  # Inicia la captura en vivo de los paquetes en
la interfaz seleccionada
# Bucle para procesar continuamente los paquetes capturados
for packet in capture.sniff_continuously():  # Captura paquetes de forma continua
```

```
    try:
        if 'IP' in packet:  # Si el paquete tiene una capa IP (es un paquete de red IP)
            source_ip = packet.ip.src  # Obtiene la IP de origen del paquete
            current_time = time()  # Obtiene el tiempo actual en segundos
            # ALERTA POR IP SOSPECHOSA
            if source_ip in IPS_SOSPECHOSAS:  # Verifica si la IP de origen está en la lista de IPs sospechosas
                if source_ip not in last_alert_time or current_time - last_alert_time[source_ip] > ALERTA_INTERVALO:
                    # Si no ha sido alertada recientemente, genera una alerta
                    print(f"ALERTA: Tráfico sospechoso desde {source_ip}")
                    mensaje_alerta = f"{datetime.now()} - Alerta: Tráfico sospechoso desde {source_ip}"
                    registrar_alerta(mensaje_alerta)  # Registra la alerta en el archivo
                    last_alert_time[source_ip] = current_time  # Actualiza el tiempo de la última alerta para esa IP
            # ALERTA POR PUERTOS SOSPECHOSOS
            if 'TCP' in packet:  # Si el paquete contiene una capa TCP
                puerto_detectado = int(packet.tcp.dstport)  # Obtiene el puerto de destino
                if puerto_detectado in PUERTOS_SOSPECHOSOS:  # Verifica si el puerto está en la lista de puertos sospechosos
                    if source_ip not in last_alert_time or current_time - last_alert_time[source_ip] > ALERTA_INTERVALO:
                        # Si no ha sido alertada recientemente, genera una alerta
```

```
                print(f"ALERTA: Tráfico en puerto {puerto_detectado} desde {source_ip}")
                mensaje_alerta = f"{datetime.now()} - Alerta: Puerto {puerto_detectado} desde
{source_ip}"
                registrar_alerta(mensaje_alerta)  # Registra la alerta en el archivo
                last_alert_time[source_ip] = current_time  # Actualiza el tiempo de la última
alerta para esa IP
        # ALERTA POR PING CON TAMAÑO MAYOR AL UMBRAL
        if 'ICMP' in packet:  # Si el paquete contiene una capa ICMP (ping)
            if int(packet.length) > TAMANO_PING_MINIMO:  # Si el tamaño del paquete es
mayor que el umbral
                if source_ip not in last_alert_time or current_time - last_alert_time[source_ip] >
ALERTA_INTERVALO:
                # Si no ha sido alertada recientemente, genera una alerta
                print(f"ALERTA: Ping sospechoso desde {source_ip} con tamaño {packet.length}
bytes")
                mensaje_alerta = f"{datetime.now()} - Alerta: Ping sospechoso desde {source_ip}
con tamaño {packet.length} bytes"
                registrar_alerta(mensaje_alerta)  # Registra la alerta en el archivo
                last_alert_time[source_ip] = current_time  # Actualiza el tiempo de la última
alerta para esa IP
    except AttributeError:
        pass  # Ignorar paquetes que no tengan los atributos esperados (por ejemplo, paquetes sin
dirección IP)
#
```

Para probar su funcionamiento, se guarda el archivo con el nombre “pruebaMonitoreo.py” y se ejecuta con este comando: python3 pruebaMonitoreo.py.

Figura 37 *Salida del script para prueba de captura de datos*



Nota. Captura tomada del Raspberry Pi OS (2025)

6.2.5. Implementación de Alertas Visuales

Estas se relacionan directamente con los mensajes y alertas por medio de los LEDs que fueron implementados como parte del *hardware* electrónico del sistema embebido. Para ello se definió el siguiente *script* en Python para el manejo de estos componentes, mediante la interface I^2C y y pines GPIO del Raspberry Pi. En esta parte del proceso lo que se está planteando es el script que permita probar la funcionalidad de los componentes como LEDs y pantalla LCD, ya que más adelante se abarcará el tema de la integración con todo el sistema.

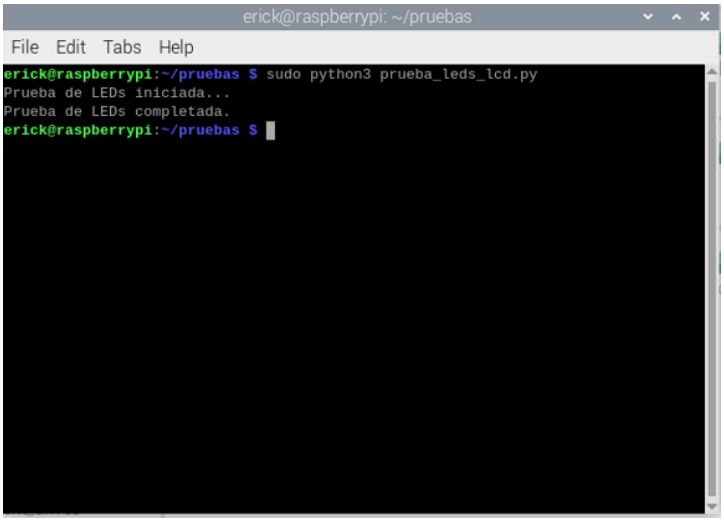
```
#
import lgpio  # Se usa la librería rpi-lgpio para el control de pines GPIO
import time  # Se importa la librería para manejar tiempos
from smbus import SMBus  # Se importa la librería para manejar la comunicación I2C
```

```
from RPLCD.i2c import CharLCD  # Librería para controlar una pantalla LCD I2C
# Abre el chip GPIO (usualmente el chip principal es 0)
chip = lgpio.gpiochip_open(0)
# Definir pines de los LEDs
LED_ROJO = 17  # Pin GPIO 17 para el LED rojo
LED_VERDE = 27  # Pin GPIO 27 para el LED verde
# Configura los pines como salidas
lgpio.gpio_claim_output(chip, LED_ROJO)  # Configura el pin 17 como salida
lgpio.gpio_claim_output(chip, LED_VERDE)  # Configura el pin 27 como salida
# Configuración de la pantalla LCD I2C
I2C_ADDRESS = 0x27  # Dirección I2C de la pantalla LCD, ajusta según la dirección detectada
con i2cdetect
lcd = CharLCD(i2c_expander="PCF8574", address=I2C_ADDRESS, port=1, cols=20, rows=4)
# Inicializa la pantalla LCD con los parámetros especificados
# Prueba de LEDs
print("Prueba de LEDs iniciada...")  # Muestra un mensaje en la consola indicando que se inicia
la prueba
for led in [LED_ROJO, LED_VERDE]:  # Itera sobre los dos LEDs
    lgpio.gpio_write(chip, led, 1)  # Enciende el LED
    time.sleep(1)  # Espera 1 segundo
    lgpio.gpio_write(chip, led, 0)  # Apaga el LED
    time.sleep(1)  # Espera 1 segundo
print("Prueba de LEDs completada.")  # Muestra un mensaje en la consola indicando que la
prueba ha finalizado
```

```
# Prueba de la pantalla LCD
lcd.clear()  # Limpia la pantalla LCD
lcd.write_string("Prueba de LCD:\nEncendiendo LEDs")  # Muestra un mensaje en la pantalla LCD
time.sleep(3)  # Espera 3 segundos
lcd.clear()  # Limpia la pantalla LCD
lcd.write_string("Raspberry OK!")  # Muestra un mensaje en la pantalla LCD
# Limpieza de los pines GPIO
lgpio.gpiochip_close(chip)  # Cierra el acceso al chip GPIO al finalizar
#
```

Para probar su funcionamiento, se guarda el archivo con el nombre “prueba_leds_lcd.py” y se ejecuta con este comando desde la terminal: python3 prueba_leds_lcd.py.

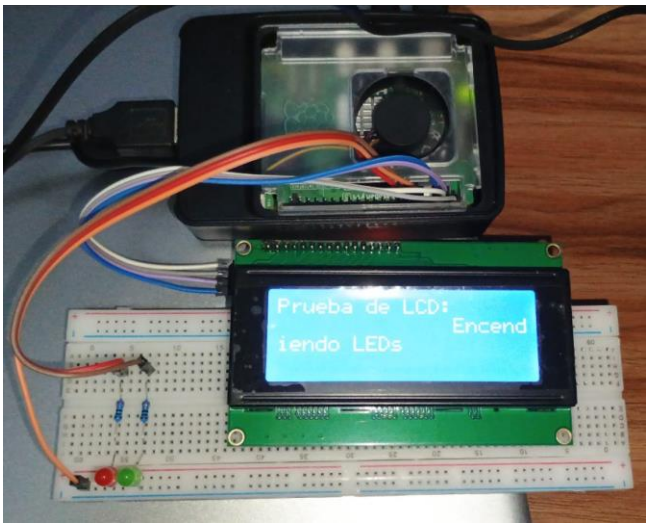
Figura 38 *Salida del script para prueba de la pantalla LCD y los LEDs*



Nota. Captura tomada del Raspberry Pi OS (2025)

El sistema presenta la siguiente salida al ejecutar la prueba de hardware para las alertas visuales.

Figura 39 *Salida del script para prueba de la pantalla LCD y los LEDs*



Nota. Foto tomada del Sistema Embebido (2025)

6.2.6. Integración y Pruebas del Sistema

Luego de haber probado las funcionalidades por separado, se procede a integrar todo en un solo *script*, con el fin de que el sistema embebido monitoree la red inalámbrica y emita alertas cuando considere necesario de acuerdo con los parámetros definidos previamente.

```
#

import pyshark  # Importa pyshark para captura de paquetes en tiempo real

from time import sleep, time  # Importa funciones de tiempo para manejar temporizadores

import lgpio  # Importa la librería lgpio para controlar pines GPIO

from RPLCD.i2c import CharLCD  # Importa la librería para controlar una pantalla LCD I2C

from datetime import datetime  # Importa la librería para manejar fechas y horas

# Configuración de los pines GPIO para los LEDs

LED_VERDE_PIN = 17  # Pin del LED verde

LED_ROJO_PIN = 27   # Pin del LED rojo

chip = lgpio.gpiochip_open(0)  # Abre el primer chip GPIO
```

```
lgpio.gpio_claim_output(chip, LED_VERDE_PIN)  # Configura el pin del LED verde como salida
lgpio.gpio_claim_output(chip, LED_ROJO_PIN)  # Configura el pin del LED rojo como salida
# Configuración de la pantalla LCD I2C
I2C_ADDRESS = 0x27  # Dirección hexadecimal de la LCD
lcd = CharLCD(i2c_expander="PCF8574", address=I2C_ADDRESS, port=1, cols=20, rows=4)
# Inicializa la pantalla LCD
# Interfaz de red a monitorear
interface = 'wlan0'  # Define la interfaz de red a monitorear (en este caso, la interfaz inalámbrica)
# Lista de IPs sospechosas (estas son las IPs que serán monitoreadas)
IPS_SOSPECHOSAS = {"5.228.183.178", "37.148.204.40", "45.195.162.189", ...}  # Las IPs que serán consideradas sospechosas
# Lista de puertos a monitorear (puertos comunes usados en ataques)
PUERTOS_SOSPECHOSOS = {22, 23, 123, 1433, 3389}  # Puertos SSH, Telnet, NTP, MSSQL, RDP
# Diccionario para rastrear actividad y evitar spam de alertas
active_ips = {}  # Diccionario para almacenar las IPs activas
last_alert_time = {}  # Diccionario para almacenar el tiempo de la última alerta por IP
prev_message = ""  # Último mensaje mostrado en la pantalla LCD
TIMEOUT = 5  # Tiempo en segundos antes de considerar una IP como inactiva
ALERTA_INTERVALO = 10  # Intervalo mínimo entre alertas de la misma IP
TIEMPO_RECUPERACION = 10  # Tiempo en segundos para restaurar el estado inicial después de una alerta
alerta_activa = False  # Bandera para saber si hay alerta activa
```

```
last_alert_reset = time()  # Registra el último reinicio de la alerta
TAMANO_PING_MINIMO = 1024  # Tamaño mínimo de ping para generar una alerta
# Función para registrar alertas en archivo de texto
def registrar_alerta(mensaje):
    with open("alertas.txt", "a") as file:  # Abre el archivo para agregar una alerta
        file.write(f"{mensaje}\n")  # Escribe la alerta en el archivo
# Función para mostrar mensajes en la pantalla LCD, evitando actualizaciones innecesarias
def mostrar_lcd(line1, line2=""):
    global prev_message  # Usa la variable global para comparar el mensaje actual
    message = f"{line1.ljust(20)}\n{line2.ljust(20)}"  # Formatea el mensaje a 20 caracteres por
línea
    if message != prev_message:  # Si el mensaje es diferente al anterior, lo actualiza
        lcd.clear()  # Limpia la pantalla LCD
        lcd.cursor_pos = (0, 0)  # Establece la posición del cursor en la primera línea
        lcd.write_string(line1.ljust(20))  # Muestra el primer mensaje
        if line2:  # Si hay un segundo mensaje, lo muestra en la segunda línea
            lcd.cursor_pos = (1, 0)
            lcd.write_string(line2.ljust(20))
        prev_message = message  # Actualiza el mensaje previo con el mensaje actual
# Mensaje inicial en la pantalla LCD
mostrar_lcd("Monitoreando red", "Esperando...")
# Enciende el LED verde al inicio y apaga el LED rojo
lgpio.gpio_write(chip, LED_VERDE_PIN, 1)
lgpio.gpio_write(chip, LED_ROJO_PIN, 0)
```

```
# Captura de paquetes en tiempo real
print(f"Monitoreando tráfico en la interfaz {interface}...")
capture = pyshark.LiveCapture(interface=interface)  # Inicia la captura de paquetes
# Loop principal de monitoreo
for packet in capture.sniff_continuously():  # Continúa capturando paquetes
    try:
        if 'IP' in packet:  # Si el paquete tiene una capa IP
            source_ip = packet.ip.src  # Obtiene la IP de origen del paquete
            current_time = time()  # Obtiene la hora actual
            alerta_generada = False  # Variable para saber si se generó una alerta
            lcd_mensaje1 = ""  # Primer mensaje para LCD
            lcd_mensaje2 = ""  # Segundo mensaje para LCD
            # ALERTA POR IP SOSPECHOSA
            if source_ip in IPS_SOSPECHOSAS:  # Si la IP es sospechosa
                if source_ip not in last_alert_time or current_time - last_alert_time[source_ip] >
ALERTA_INTERVALO:
                    print(f"ALERTA: Tráfico sospechoso desde {source_ip}")  # Muestra el mensaje en
consola
                    lcd_mensaje1 = "Alerta: Tráfico"  # Mensaje para la LCD
                    lcd_mensaje2 = f"IP: {source_ip}"
                    alerta_generada = True  # Marca que se generó una alerta
                    alerta_activa = True  # Establece que hay una alerta activa
                    last_alert_time[source_ip] = current_time  # Guarda el tiempo de la última alerta
                    mensaje_alerta = f"{datetime.now()} - Alerta: Tráfico sospechoso desde
```

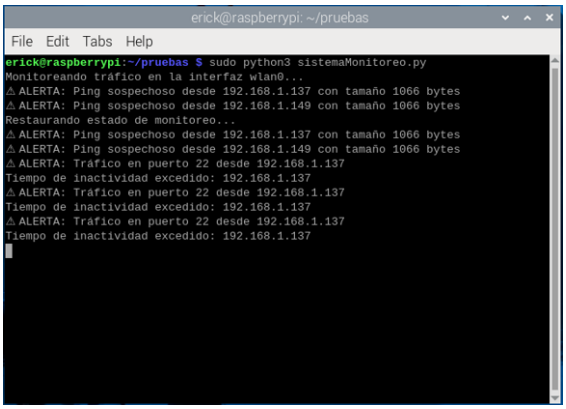
```
{source_ip}"  # Registra la alerta con timestamp
                registrar_alerta(mensaje_alerta)  # Llama a la función para registrar la alerta
            # ALERTA POR PUERTOS SOSPECHOSOS
            if 'TCP' in packet:  # Si el paquete tiene una capa TCP
                puerto_detectado = int(packet.tcp.dstport)  # Obtiene el puerto de destino del paquete
                if puerto_detectado in PUERTOS_SOSPECHOSOS:  # Si el puerto es sospechoso
                    if source_ip not in last_alert_time or current_time - last_alert_time[source_ip] >
ALERTA_INTERVALO:
                        active_ips[source_ip] = current_time  # Marca la IP como activa
                        print(f"□ ALERTA: Tráfico en puerto {puerto_detectado} desde {source_ip}")  #
Muestra el mensaje en consola
                        if not lcd_mensaje1:  # Si no hay mensaje anterior, usa este
                            lcd_mensaje1 = f"Alerta: Puerto {puerto_detectado}"
                            lcd_mensaje2 = f"IP: {source_ip}"
                        alerta_generada = True  # Marca que se generó una alerta
                        alerta_activa = True  # Establece que hay una alerta activa
                        last_alert_time[source_ip] = current_time  # Guarda el tiempo de la última alerta
                        mensaje_alerta = f"{datetime.now()} - Alerta: Puerto {puerto_detectado} desde
{source_ip}"  # Registra la alerta
                        registrar_alerta(mensaje_alerta)  # Llama a la función para registrar la alerta
            # ALERTA POR PING CON TAMAÑO MAYOR AL UMBRAL
            if 'ICMP' in packet:  # Si el paquete es un paquete ICMP (ping)
                if int(packet.length) > TAMANO_PING_MINIMO:  # Si el tamaño del ping es mayor
que el umbral
```

```
                print(f"ALERTA: Ping sospechoso desde {source_ip} con tamaño {packet.length}
bytes")  # Muestra el mensaje en consola
                lcd_mensaje1 = "Alerta: Ping"
                lcd_mensaje2 = f"IP: {source_ip} - {packet.length}B"
                alerta_generada = True  # Marca que se generó una alerta
                alerta_activa = True  # Establece que hay una alerta activa
                last_alert_time[source_ip] = current_time  # Guarda el tiempo de la última alerta
                mensaje_alerta = f"{datetime.now()} - Alerta: Ping sospechoso desde {source_ip}
con tamaño {packet.length} bytes"  # Registra la alerta
                registrar_alerta(mensaje_alerta)  # Llama a la función para registrar la alerta
        # ACTUALIZAR LCD SI HAY ALERTA
        if alerta_generada:  # Si se generó una alerta
            mostrar_lcd(lcd_mensaje1, lcd_mensaje2)  # Muestra el mensaje de la alerta en la LCD
            lgpio.gpio_write(chip, LED_ROJO_PIN, 1)  # Enciende el LED rojo
            lgpio.gpio_write(chip, LED_VERDE_PIN, 0)  # Apaga el LED verde
            last_alert_reset = current_time  # Reinicia el temporizador de recuperación
        # COMPROBAR SI HAY QUE RESTAURAR AL ESTADO DE MONITOREO
        if alerta_activa and current_time - last_alert_reset >= TIEMPO_RECUPERACION:  # Si
ya pasó el tiempo de recuperación
            print("Restaurando estado de monitoreo...")  # Muestra el mensaje en consola
            mostrar_lcd("Monitoreando red", "Esperando...")  # Muestra el mensaje inicial en la
LCD
            lgpio.gpio_write(chip, LED_ROJO_PIN, 0)  # Apaga el LED rojo
            lgpio.gpio_write(chip, LED_VERDE_PIN, 1)  # Enciende el LED verde
```

```
            alerta_activa = False  # Restablece el estado de alerta
        # ELIMINA IPs INACTIVAS
        inactive_ips = [ip for ip, last_seen in active_ips.items() if time() - last_seen > TIMEOUT]  # Revisa las IPs inactivas
        for ip in inactive_ips:  # Elimina las IPs inactivas
            del active_ips[ip]  # Elimina la IP del diccionario de IPs activas
            print(f"Tiempo de inactividad excedido: {ip}")  # Muestra que la IP ha excedido el tiempo de inactividad
    except AttributeError:  # Si el paquete no tiene los atributos esperados
        pass  # Simplemente ignora ese paquete
#
```

Para probar su funcionamiento, guardar el archivo con el nombre “sistemaMonitoreo.py” y se ejecuta con este comando desde la terminal: sudo python3 sistemaMonitoreo.py.

Figura 40 *Salida del script para prueba del script completo*

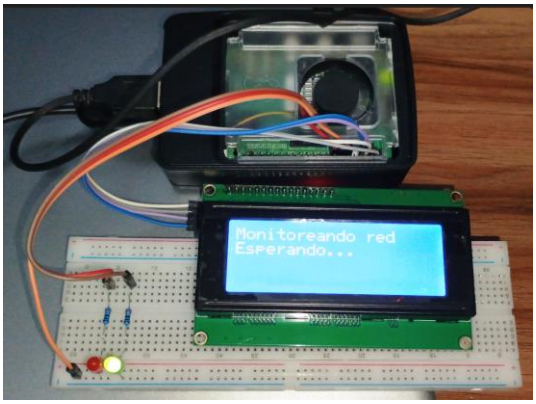


Nota. Captura tomada del Raspberry Pi OS (2025)

Para la lista de direcciones IP sospechosas se agregaron algunas de una lista negra obtenida en internet, así como algunas IP internas de los dispositivos que normalmente hacen uso de la red inalámbrica. Como parte de las alertas visuales, el comportamiento del *hardware*

sistema se muestra en las siguientes figuras. En su estado de reposo, el sistema está a la espera de cualquier actividad que considere sospechosa de acuerdo con toda la programación de parámetros que se realizó. En este caso, la pantalla muestra el mensaje “Monitoreando red Esperando…” mientras que el *led* verde se encuentra encendido y el *led* rojo apagado.

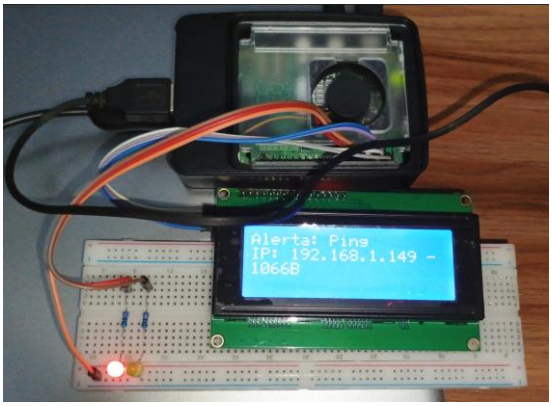
Figura 41 *Sistema en estado de espera*



Nota. Foto tomada del Sistema Embebido (2025)

Este comportamiento de los *leds* se debe a que si no hay detección de actividad sospechosa, el color verde es el que permanece encendido, sin embargo, al momento de realizar alguna detección, el color rojo es el que se va a encender, y la pantalla indicará la alerta respectiva, como se parecía en la siguiente figura.

Figura 42 *Sistema detectando una situación sospechosa*



Nota. Foto tomada del Sistema Embebido (2025)

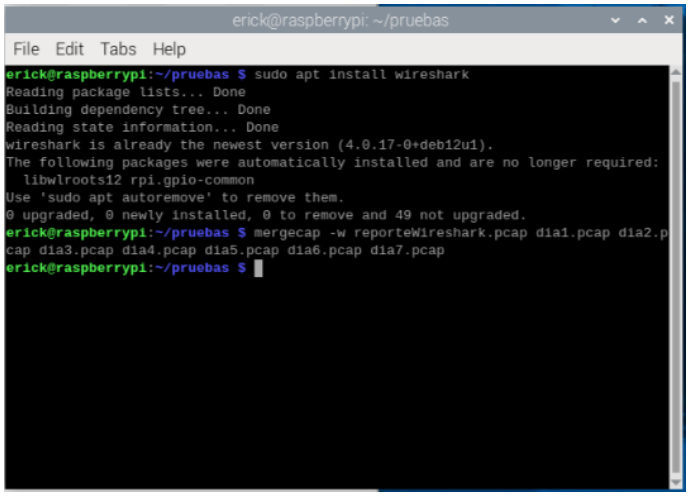
6.2.7. Efectividad del Sistema Embebido

Para efectos de tener un método que permita determinar la efectividad del sistema embebido en cuanto a la detección de situaciones sospechosas reales, se instala la herramienta Wireshark en una computadora portátil para mantenerla capturando información de la red inalámbrica paralelamente con el sistema embebido.

En vista de que son varios días de captura de datos en horarios no continuos, se genera un archivo ".pcap" para cada día, los cuales posteriormente es necesario combinar en un único archivo para proceder con el análisis de la información.

Los comandos utilizados se muestran en la siguiente figura. Es importante aclarar que es necesario que previamente esté instalada la herramienta Wireshark en el Raspberry Pi OS, en vista de que es más simple la unificación de los archivos respecto a realizarla en la computadora portátil de pruebas, ya que esta posee sistema operativo Windows.

Figura 43 *Consolidación de archivos pcap*



Nota. Captura tomada del Raspberry Pi OS (2025)

El archivo consolidado se utiliza para ser comparado con el archivo de las alertas en formato TXT que genera el sistema embebido, esto mediante el siguiente *script* en Python.

```
#
from scapy.all import rdpcap, IP, TCP, UDP, ICMP
import datetime
# Archivos de entrada
ALERTAS_FILE = "alertasFinal.txt"  # Archivo de alertas a procesar
PCAP_FILE = "reporteWireshark.pcap"  # Archivo PCAP que contiene el tráfico capturado
# Cargar tráfico del PCAP
print(f"□ Cargando PCAP: {PCAP_FILE}...")
paquetes = rdpcap(PCAP_FILE)  # Carga los paquetes desde el archivo PCAP
# Leer alertas y extraer datos
alertas = []  # Lista para almacenar las alertas procesadas
with open(ALERTAS_FILE, "r") as f:
    for linea in f:
        partes = linea.strip().split(" - Alerta: ")  # Divide la línea en timestamp y mensaje de alerta
        if len(partes) == 2:
            timestamp_str, mensaje = partes
            timestamp = datetime.datetime.strptime(timestamp_str, "%Y-%m-%d %H:%M:%S.%f")
# Convierte el timestamp a formato datetime
            # Extraer IPs de la alerta
            palabras = mensaje.split()  # Divide el mensaje en palabras
            ips = [p for p in palabras if p.count(".") == 3]  # Filtra las palabras que son direcciones IP
            if len(ips) >= 1:
                ip_src = ips[0]  # Asigna la primera IP como origen
```

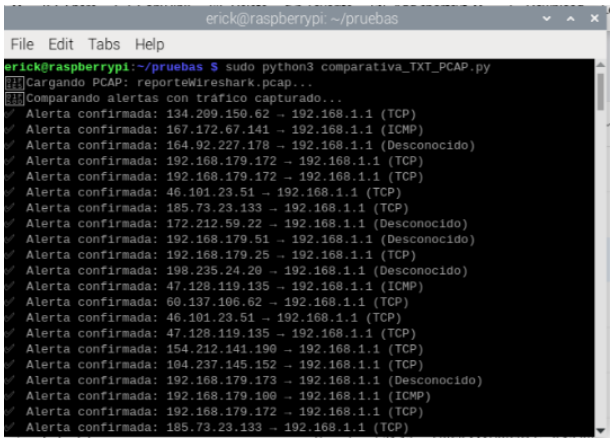
```
            ip_dst = ips[1] if len(ips) > 1 else "192.168.1.1"  # Asigna la segunda IP como destino
(o valor por defecto)
            tipo_evento = "Desconocido"  # Inicializa tipo de evento
            # Clasifica el tipo de evento según el mensaje
            if "Tráfico sospechoso" in mensaje:
                tipo_evento = "TCP"
            elif "Puerto sospechoso" in mensaje:
                tipo_evento = "UDP"
            elif "Ping sospechoso" in mensaje:
                tipo_evento = "ICMP"
            # Agrega la alerta procesada a la lista
            alertas.append((timestamp, ip_src, ip_dst, tipo_evento))
# Comparar alertas con el tráfico del PCAP
print("□ Comparando alertas con tráfico capturado...")
alertas_confirmadas = 0  # Contador de alertas confirmadas
falsos_positivos = 0  # Contador de falsos positivos
ventana_tiempo = datetime.timedelta(seconds=2)  # Margen de tiempo para la comparación (2
segundos)
# Recorre cada alerta y verifica si corresponde con un paquete en el PCAP
for alerta in alertas:
    timestamp, ip_src, ip_dst, tipo_evento = alerta  # Extrae los datos de la alerta
    encontrada = False  # Bandera para verificar si la alerta se encuentra en el tráfico
    # Recorre todos los paquetes del PCAP
    for pkt in paquetes:
```

```
        if IP in pkt:  # Solo procesa paquetes con capa IP
            pkt_time = datetime.datetime.fromtimestamp(float(pkt.time))  # Obtiene la marca de
tiempo del paquete
            if abs(pkt_time - timestamp) > ventana_tiempo:  # Si el paquete no está dentro del margen
de tiempo, lo salta
                continue
            # Compara la IP de origen y destino
            if pkt[IP].src == ip_src and pkt[IP].dst == ip_dst:
                # Verifica si el tipo de evento coincide con el protocolo del paquete
                if (tipo_evento == "TCP" and TCP in pkt) or \
                   (tipo_evento == "UDP" and UDP in pkt) or \
                   (tipo_evento == "ICMP" and ICMP in pkt) or \
                   (tipo_evento == "Desconocido"):  # Caso de otros protocolos (sin especificar)
                    encontrada = True  # Marca la alerta como encontrada
                    break
    # Clasificación de la alerta según si fue confirmada o es un falso positivo
    if encontrada:
        print(f"Alerta confirmada: {ip_src} → {ip_dst} ({tipo_evento})")
        alertas_confirmadas += 1
    else:
        print(f"Falso positivo: {ip_src} → {ip_dst} ({tipo_evento})")
        falsos_positivos += 1
# Resumen final
print("\n Resumen de Comparación:")
```

```
print(f"Alertas Confirmadas: {alertas_confirmadas}")  # Muestra el total de alertas confirmadas
print(f"Falsos Positivos: {falsos_positivos}")  # Muestra el total de falsos positivos
#
```

Para probar su funcionamiento, guardar el archivo con el nombre "comparativa_TXT_PCAP.py" y se ejecuta con este comando desde la terminal: sudo python3 comparativa_TXT_PCAP.py.

Figura 44 *Salida de la comparación de información*



Nota. Captura tomada del Raspberry Pi OS (2025)

6.3.Propuesta para Mejoras Prioritarias

Seguidamente se plantea una propuesta en la que se mencionan las principales acciones por seguir que requiere el sistema embebido, esto con el fin de mejorar tanto su funcionalidad como su efectividad para la detección y alerta.

Tabla 11 *Propuesta para mejoras*

Mes	Mejoras en monitoreo de actividad sospechosa	Mejoras funcionales y de operación	Mejoras en hardware
1	Ajuste de parámetros para detección de tráfico, puertos y volumen sospechoso.	Implementación de autenticación de usuarios y control de acceso.	Revisión de temperatura y conexiones de la Raspberry Pi.

Mes	Mejoras en monitoreo de actividad sospechosa	Mejoras funcionales y de operación	Mejoras en hardware
2	Mejora en la visualización de alertas con pantalla LCD y LEDs.	Cifrado de *logs* y comunicaciones usando TLS.	Instalación de disipadores y ventilación activa.
3	Implementación de listas negras de IPs basadas en actividad previa.	Integración de *firewall* y filtrado de paquetes.	Implementación de modo de bajo consumo cuando no hay actividad sospechosa.
4	Análisis de patrones avanzados en tráfico con *machine learning* básico.	Uso de firmas digitales en los archivos de configuración.	Uso de un *Watchdog Timer* (WDT) para reiniciar el sistema en caso de fallos.
5	Monitoreo de tiempos de conexión y patrones anómalos en puertos abiertos.	Implementación de detección de intrusiones (IDS) con Suricata.	Uso de botones físicos para reinicio manual o cambios de configuración.
6	Detección de dispositivos desconocidos en la red.	Validación de integridad del sistema con *hashes* SHA256.	Integración de batería de respaldo en caso de cortes de energía.

Nota. Elaboración propia (2025)

Cabe resaltar que cualquiera de estas acciones requiere de una etapa de pruebas, con el fin de determinar que el sistema embebido realmente esté mejorando no sólo en cuanto a la detección real y disminución de falsos positivos, sino que además, el hardware como tal soporte el procesamiento que conllevan todas estas funcionalidades.

Referencias

330 Ohms. (30 de julio de 2024). *Cómo habilitar la comunicación I2C en Raspberry Pi - 330ohms*. https://www.330ohms.com/blogs/blog/como-habilitar-la-comunicacion-i2c-en-raspberry-pi

Aguilar, C. (mayo de 2005). *Monitoreo y administración de los equipos de la red del Banco Nacional de Costa Rica por medio de SNMP basado en la plataforma NNM de HP-Open View* . https://repositorio.sibdi.ucr.ac.cr/server/api/core/bitstreams/5b2484a2-9571-43b6-9eba-d7e60adc6b22/content

Ailaca, C. (noviembre de 2011). *Sistema de monitoreo y control de redes inalámbricas para optimización del servicio de internet en la empresa Intercompu.* https://repositorio.uta.edu.ec/bitstream/123456789/442/1/Tesis_t655ec.pdf

Alexander, C., y Sadaki, M. (2006). *Fundamentos de circuitos eléctricos.* México: McGraw Hill.

Andrade, Y. (julio de 2022). *Implementación de un servidor LAMP con una Raspberry Pi y ESP32 para monitorear la temperatura, presión y humedad de un laboratorio de ciencias básicas del Instituto Tecnológico Superior de Martínez de la Torre.* https://rinacional.tecnm.mx/jspui/bitstream/TecNM/4612/1/IMPLEMENTACI%C3%93N%20DE%20UN%20SERVIDOR%20LAMP%20CON%20UNA%20RASPBERRY%20PI%20Y%20ESP32%20PARA%20MONITORER%20LA%20TEMP%2C%20PRESI%C3%93N%20Y%20HUM%20DE%20UN%20LAB%20DE%20C.B%C3%81SICAS%20DEL%20ITSM

ANSI/ISA. (28 de agosto de 2019). *Security for industrial automation and control systems, Part 4-2: Technical security requirements for IACS components.* https://s3.us-east-1.amazonaws.com/fonteva-customer-media/00D1I000002JB6nUAG/OvIGvqBd_S_62443_4_2_pdf

Berrocal, J. (diciembre de 2021). *Implementación de micro controladores electrónicos de bajo costo en la adquisición de deformaciones unitarias es estructuras.* https://repositorio.sibdi.ucr.ac.cr/server/api/core/bitstreams/3121d532-f444-4f02-a350-b813dcc7c85f/content

Broadcom Corp. (s.f.). *BCM43455 / BCM43456 / BCM43454 / BCM43458*. https://device.report/broadcom/bcm43456

Cisco. (2024). *¿Qué es Wi-Fi (red inalámbrica) frente a una red cableada?* https://www.cisco.com/c/es_mx/solutions/small-business/resource-center/networking/wireless-network.html

Cordero, R., Gómez, Y., Salazar, S., y Vargas, E. (julio de 2016). *Propuesta de un modelo de gestión de riesgo de infraestructura de tecnologías de información y comunicaciónen la empresa XYZ, basado en RISK IT de ISACA.* https://repositorio.sibdi.ucr.ac.cr/server/api/core/bitstreams/bff1430d-e327-4e56-96e3-288a741cd82b/content

Cossio, K. (2021). *Aplicación móvil para el monitoreo de la seguridad en redes inalámbricas e internet.* https://repositorio.uchile.cl/bitstream/handle/2250/181120/Aplicacion-movil-para-el-monitoreo-de-la-seguridad-en-redes-inalambricas-en-Internet.pdf?sequence=1&isAllowed=y

Cuadrado, L. (23 de noviembre de 2023). *Importancia de la Pérdida de Trayectoria en Espacio Libre en el diseño y planificación de radioenlaces*. https://abcxperts.com/importancia-de-la-perdida-de-trayectoria-en-espacio-libre-en-el-diseno-y-planificacion-de-radioenlaces/

Datos101. (05 de octubre de 2023). *La importancia del cumplimiento normativo en ciberseguridad y cómo lograrlo*. https://www.datos101.com/blog/normativa-ciberseguridad-como-lograrlo/

De Luz, S. (03 de octubre de 2024). *Cómo usar Wireshark para capturar y analizar el tráfico de red*. https://www.redeszone.net/tutoriales/redes-cable/wireshark-capturar-analizar-trafico-red/#303790-que-es-wireshark

Ediciones ENI. (s.f.). *Raspberry Pi y GPIO*. https://www.ediciones-eni.com/libro/python-y-raspberry-pi-aprenda-a-desarrollar-en-su-nano-ordenador-9782409014284/raspberry-pi-y-gpio

ElectrónicaOnline. (s.f.). *Símbolos de Componentes Electrónicos: Leerlos y Comprenderlos*. https://electronicaonline.net/componentes-electronicos/simbolos-de-componentes-electronicos-leerlos-y-comprenderlos/

Faster Capital. (2024). *Costos Implementación*. https://fastercapital.com/es/palabra-clave/costos-implementaci%C3%B3n.html#:~:text=Costos%20de%20Implementaci%C3%B3n%3A%20Son%20los,procesos%20y%20actualizaciones%20de%20infraestructura.

Fernández, M. (2013). *Control sobre el proceso de TI: Garantizar la seguridad de los sistemas.* https://www.kerwa.ucr.ac.cr/server/api/core/bitstreams/b4eb0f4c-dca8-4302-b5d9-33a4ab1664e1/content

Finn, T., y Downie, A. (07 de mayo de 2024). *¿Qué es la mitigación de riesgos?* https://www.ibm.com/mx-es/topics/risk-mitigation

Fonafifo. (2018). *Historia*. https://www.fonafifo.go.cr/es/conozcanos/historia/

Fonafifo. (2018). *Objetivos*. https://www.fonafifo.go.cr/es/conozcanos/objetivos/

Fonafifo. (2018). *Organigrama*. https://www.fonafifo.go.cr/es/conozcanos/organigrama/

Fortinet. (2024). *¿Qué es el tráfico de red?* https://www.fortinet.com/lat/resources/cyberglossary/network-traffic

Frett, N. (12 de junio de 2018). *Cuatro pasos para la evaluación de riesgos según Norma ISO 31000 2018*. https://www.auditool.org/blog/control-interno/cuatro-pasos-para-la-evaluacion-de-riesgos-segun-norma-iso-31000-2018

Fromaget, P. (2025). *Actualizar Raspberry Pi OS a la Ultima Versión*. https://raspberrytips.es/actualizar-raspberry-pi-ultima-version/

Fromaget, P. (s.f.). *57 Comandos De Raspberry Pi Que Todo El Mundo Debería Conocer*. https://raspberrytips.es/comandos-raspberry-pi/

Github. (2023). *RPLCD*. https://github.com/dbrgn/RPLCD

Gómez, C., y Pedraza, L. (junio de 2018). *Ubicación de dispositivos móviles en ambientes interiores por medio de análisis de radiación de redes WiFi y deformaciones de campo magnético*. https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052018000200203

González, J. (03 de julio de 2024). *Bases de la programación en sistemas embebidos*. https://www.tecneu.com/blogs/noticias/bases-de-la-programacion-en-sistemas-embebidos

Grupo BBVA. (15 de abril de 2024). *ROI: qué es el retorno de la inversión y cuál es su fórmula*. https://www.bbva.com/es/salud-financiera/roi-que-es-el-retorno-de-la-inversion-y-cual-es-su-formula/

Halfacree, G. (2024). *La guía oficial de Raspberry Pi para principiantes.* Reino Unido: Raspberry Pi.

Hewlett Packard Enterprise. (2024). *¿Qué es una amenaza de ciberseguridad?* https://www.hpe.com/lamerica/es/what-is/cybersecurity-threats.html

Huawei. (21 de diciembre de 2023). *Descripción general de las tramas Ethernet*. https://forum.huawei.com/enterprise/es/descripci%C3%B3n-general-de-las-tramas-ethernet/thread/737876946756452352-667212883219591168

IBM. (s.f.). *¿Qué es gestión de riesgos?* https://www.ibm.com/mx-es/topics/risk-management

InnovaciónDigital360. (22 de setiembre de 2023). *Sistemas embebidos: qué son y para qué se utilizan*. https://www.innovaciondigital360.com/iot/sistemas-embebidos-que-son-y-para-que-se-utilizan/

ISO/IEC/IEEE. (15 de diciembre de 2010). *Systems and software engineering — Vocabulary* [Archivo PDF]. https://www.cse.msu.edu/~cse435/Handouts/Standards/IEEE24765.pdf

Jitendra, P., Samal, A., Pani, S., y Charaborty, C. (2021). *Elementary framework for an IoT based diverse ambient air quality monitoring system*. https://www.researchgate.net/figure/Block-diagram-of-an-embedded-system-showing-basic-functional-components_fig1_354329894

Kaspersky. (2024). *¿Qué es WEP, WPA, WPA2 y WPA3 y cuáles son sus diferencias?* https://latam.kaspersky.com/resource-center/definitions/wep-vs-wpa

Kaspersky. (2024). *Protección de redes inalámbricas*. https://latam.kaspersky.com/resource-center/preemptive-safety/protecting-wireless-networks

López, M. (20 de setiembre de 2023). *Sistema Embebido*. https://chsos202335806.wordpress.com/2023/09/20/sistema-embebido/

Lozano, R. (19 de agosto de 2020). *Como usar pantalla lcd con i2c con raspberry*. https://www.taloselectronics.com/blogs/tutoriales/como-usar-pantalla-lcd-con-i2c-con-raspberry

ManageEngine. (2025). *Comparación de versiones de NetFlow Analyzer*. https://www.manageengine.com/latam/netflow/comparacion-de-versiones.html

Maranto, M., y González, M. (febrero de 2015). *Fuentes de información.* https://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/16700/LECT132.pdf

MCI Educación. (23 de agosto de 2022). *Protocolo I2C (Inter Integrated Circuit).* https://cursos.mcielectronics.cl/2022/08/23/i2c/

MCI Electronics. (s.f.). *¿Que es Raspberry Pi?* https://raspberrypi.cl/que-es-raspberry/

MICITT. (21 de abril de 2022). *DIRECTRIZ N° 133-MP-MICITT*. https://micitt.go.cr/sites/default/files/2023-06/DIRECTRIZ-N%C2%B0-133-marca-de-hora.pdf

Montero, J. (27 de marzo de 2014). *Técnicas de diseño de pruebas en sistemas embebidos.* https://oa.upm.es/34071/1/PFC_julian_montero_somavilla.pdf

Montezzana, M. (28 de junio de 2023). *ROI Capacitación: Mide el Retorno de la Inversión en Programas de Desarrollo Corporativo*. https://voxy.com/es/blog/roi-capacitacion/

Mosquera, N. (2019). *Evaluación de sistemas embebidos para un sistema avanzado de asistencia al conductor en ROS* [Archivo PDF]. https://repositorio.escuelaing.edu.co/bitstream/handle/001/1161/Mosquera-Seligmann%2c%20Nicol%c3%a1s-2020.pdf?sequence=1&isAllowed=y

Navarrete, J. (marzo de 2021). *Implementación de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi.* http://repositorio.ucsg.edu.ec/bitstream/3317/16249/1/T-UCSG-PRE-TEC-ITEL-401.pdf

Nestor, M. (07 de setiembre de 2022). *New Raspberry Pi OS Update Brings Desktop Enhancements, NetworkManager Support*. https://9to5linux.com/new-raspberry-pi-os-update-brings-desktop-enhancements-networkmanager-support

Noguera, D. (2024). *¿Son seguras las redes Wi-Fi? Descubre los riesgos y vulnerabilidades.* https://flashstart.com/es/son-seguras-las-redes-wifi-descubre-los-riesgos-y-vulnerabilidades/

Paessler. (2025). *Precios de PRTG Network Monitor*. https://www.paessler.com/es/pricing

Rappaport, T. (2002). *Wireless Communications - Principles and Practice.* Prentice Hall.

Raspberry Pi. (2024). *Raspberry Pi hardware*. https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

Rivera, J. (09 de julio de 2020). *Pantalla LCD 1602 mediante I2C*. https://pasionelectronica.com/pantalla-lcd-1602-mediante-i2c/

Rodríguez, F. (s.f.). *Sistemas embebidos Smart: La era del Internet de las Cosas*. https://www.infotec.mx/sistemas-embebidos-smart-la-era-del-internet-de-las-cosas

Rozario, R. (10 de abril de 2024). *5 Wi-Fi vulnerabilities that you should be aware of*. https://www.linkedin.com/pulse/5-wi-fi-vulnerabilities-you-should-aware-ryan-rozario-0f0qc

Salas, D. (04 de junio de 2019). *El enfoque mixto de investigación: algunas características*. https://investigaliacr.com/investigacion/el-enfoque-mixto-de-investigacion/

Santos, J. (22 de agosto de 2024). *¿Qué es una vulnerabilidad informática y cómo protegerse?* https://www.deltaprotect.com/blog/vulnerabilidad-informatica

Seoane, E. (2016). *Sistema operativo, búsqueda de la información: Internet, intranet y correo electrónico.* España: Ideaspropias.

Soderby, K. (18 de noviembre de 2021). *Liquid Crystal Displays (LCD) with Arduino*. https://docs.arduino.cc/learn/electronics/lcd-displays

Solano, G. (07 de agosto de 2024). *Gobierno incumpliría meta de la regla fiscal en el presupuesto 2026*. https://www.unacomunica.una.ac.cr/index.php/agosto-2024/5513-gobierno-incumpliria-meta-de-la-regla-fiscal-en-el-presupuesto-2026

Solectro. (2025). *5. Pines GPIO y su programación*. https://solectroshop.com/es/content/60-5-pines-gpio-y-su-programacion

Tanenbaum, A., & Wetherall, D. (2012). *Redes de computadoras.* México: Pearson Educación.

Tomasi, W. (2003). *Sistemas de comunicaciones electrónicas.* México: Pearson Educación.

Trout, J. (21 de julio de 2021). *DMAIC: Una guía completa*. https://cmc-latam.com/2021/07/21/dmaic-una-guia-completa/

Vásquez, F. (s.f.). *Sensores y actuadores.* https://openaccess.uoc.edu/bitstream/10609/141046/12/PLA3_Sensores%20y%20actuadores.pdf

Vectra. (2025). *Métricas de ciberseguridad*. https://es.vectra.ai/topics/cybersecurity-metrics

Villasís, M., y Miranda, M. (30 de mayo de 2016). *El protocolo de investigación IV: las variables de estudio*. https://revistaalergia.mx/ojs/index.php/ram/article/view/199/350#:~:text=Las%20variables%20en%20un%20estudio,de%20los%20protocolo%20de%20investigaci%C3%B3n.

Zamora, M. (2019). *Prototipando soluciones tecnológicas.* https://www.uaeh.edu.mx/docencia/P_Presentaciones/prepa3/2019/prototipando-soluciones-tecnologicas.pdf

Zita, A. (s.f.). *Marco Teórico*. https://www.significados.com/marco-teorico/

Zuloaga Izaguirre, A., y Astarloa Cuéllar, A. (2008). *Sistemas de procesamiento digital.* Madrid, España: Delta, Publicaciones Universitarias.