REPÚBLICA DEL ECUADOR

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE POSGRADO

MAESTRÍA EN COMPUTACIÓN MENCIÓN SEGURIDAD INFORMÁTICA

IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE DE SEGURIDAD DOMÉSTICA BASADO EN IOT Y APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN DE INTRUSOS

Trabajo de titulación previo a la obtención del Título de Magíster en Computación mención en Seguridad Informática

AUTOR:

Pablo David Toledo Iñiguez

TUTOR:

PhD. Cathy Pamela Guevara Vega

ASESOR:

PhD. Geovany Raura Ruiz

IBARRA – ECUADOR

AÑO – 2025

REPÚBLICA DEL ECUADOR

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1104570237		
APELLIDOS Y NOMBRES:	Pablo David Toledo Iñiguez		
DIRECCIÓN:	Loja, Calle Rodríguez Soto y Juan Larrea		
EMAIL:	pablodavids11mail.com		
TELÉFONO FIJO:		**TELÉFONO MÓVIL:**	0986349152

DATOS DE LA OBRA	
TÍTULO:	IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE DE SEGURIDAD DOMÉSTICA BASADO EN IOT Y APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN DE INTRUSOS
AUTOR (ES):	Pablo David Toledo Iñiguez
FECHA:	25-09-2025
PROGRAMA:	[ ] **PREGRADO** [X] POSGRADO
TITULO POR EL QUE OPTA:	Magíster en Computación con mención en seguridad informática
ASESOR /TUTOR:	PhD. Geovany Raura Ruiz / PhD. Cathy Pamela Guevara Vega

CONSTANCIA

El autor Pablo David Toledo Iñiguez, manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 18 días del mes de septiembre del 2025

EL AUTOR:

Nombre: Pablo Toledo

APROBACIÓN DEL TUTOR

Yo Ph.D. Guevara Vega Cathy Pamela, en calidad de director de la tesis titulada: "IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE DE SEGURIDAD DOMÉSTICA BASADO EN IOT Y APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN DE INTRUSOS" de auditoría del Ing. Pablo David Toledo Iñiguez, para optar por el grado de Magister en Computación con Mención en Seguridad Informática, doy fe de que dicho trabajo reúne los requisitos y méritos suficientes para ser sometidos a presentación privada y evaluación por parte del jurado examinador que se designe.

En la ciudad de Ibarra, a los 18 días del mes de septiembre de 2025

Lo certifico

1002334835 CATHY PAMELA GUEVARA VEGA

Firmado digitalmente por 1002334835 CATHY PAMELA GUEVARA VEGA
Fecha: 2025.09.24 10:29:22 -05'00'

Ph.D. Guevara Vega Cathy Pamela

TUTOR DE TESIS

DEDICATORIA

A mi amada hija Samantha, cuya luz y alegría iluminan cada día de mi vida y me motivan a crecer y ser mejor persona, este logro es el reflejo de mi compromiso de ofrecerte un futuro lleno de oportunidades, y de que encuentres en tu propio camino la fuerza para perseguir y alcanzar tus sueños. Cada esfuerzo que realizo está inspirado en ti y en la esperanza de que sirva como ejemplo y guía en tu vida, este trabajo es para ti y por ti, con todo mi amor.

Pablo Toledo

AGRADECIMIENTO

Quiero expresar mi sincero agradecimiento a mi director de tesis por su paciencia, sabiduría y orientación, que fueron esenciales para completar este proyecto, de igual manera a los profesores de la maestría por brindarme los conocimientos y herramientas necesarios para mi formación, gracias a mis compañeros por el apoyo y el intercambio de ideas durante todo este proceso. Mi mayor gratitud es para mi familia, por su comprensión y respaldo constante, que me dieron fuerza y motivación. Finalmente, agradezco a todas las personas que de alguna forma contribuyeron a que este trabajo fuera posible.

Pablo Toledo

INDICE DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE POSGRADO MAESTRÍA EN COMPUTACIÓN MENCIÓN SEGURIDAD INFORMÁTICA - EN LÍNEA

IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE DE SEGURIDAD DOMÉSTICA BASADO EN IOT Y APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN DE INTRUSOS

AUTOR: Pablo David Toledo Iñiguez

TUTOR: PhD. Cathy Pamela Guevara Vega

Año: 2025

RESUMEN

En la presente investigación se desarrolla e implementa un sistema inteligente de seguridad doméstica basado en tecnologías de Internet de las Cosas (IoT) y aprendizaje automático (Machine Learning o ML) para la detección de intrusos, debido a la creciente inseguridad en entornos residenciales del país. El estudio se llevó a cabo en una vivienda unifamiliar ubicada en la ciudad de Loja Ecuador, donde se instaló una infraestructura compuesta por sensores Aqara bajo protocolo ZigBee, un coordinador ConBee II, un Gateway Raspberry Pi 5 y un sistema de respaldo energético solar. La metodología utilizada integró Design Science Research (DSR) para el diseño e implementación del sistema y CRISP-DM para el desarrollo del modelo de ML, por otra parte, los datos se recolectaron durante aproximadamente dos meses y medio (marzo–mayo 2025), registrando 28 variables de los sensores. El procesamiento se realizó mediante una arquitectura de microservicios desplegada en Docker, con backend en FastAPI, frontend en React, y bases de datos InfluxDB y PostgreSQL. En la fase experimental se evaluaron los algoritmos Isolation Forest, One-Class SVM y Elliptic Envelope, resultando en que este último alcanzó el mejor desempeño en validación cruzada con un accuracy promedio de 0.85. Sin embargo, al validar el sistema con patrones de intrusión simulados se evidenciaron limitaciones críticas: un recall bajo (20%), lo que implica que solo se detecta una de cada cinco intrusiones, y un alto índice de falsos positivos (86.7%), clasificando gran parte de la actividad normal como anómala. Estos resultados muestran que la integración de IoT y machine learning constituye una base sólida

para sistemas de seguridad doméstica inteligentes, aunque el modelo actual presenta restricciones que limitan su confiabilidad en escenarios de seguridad crítica sin ajustes y optimizaciones significativas, como un mayor tiempo en la toma de muestras.

Palabras Clave: Seguridad doméstica inteligente, Internet de las Cosas (IoT), Aprendizaje automático (ML), Detección de intrusos, ZigBee, Elliptic Envelope.

NORTHERN TECHNICAL UNIVERSITY FACULTY OF POSTGRADUATE STUDIES MASTER'S IN COMPUTER SCIENCE WITH A SPECIALIZATION IN COMPUTER SECURITY - ONLINE

IMPLEMENTATION OF AN INTELLIGENT HOME SECURITY SYSTEM BASED ON IoT AND MACHINE LEARNING FOR INTRUDER DETECTION

AUTHOR: Pablo David Toledo Iñiguez

ADVISOR: PhD. Cathy Pamela Guevara Vega

Year: 2025

ABSTRACT

In the present research, an intelligent home security system based on Internet of Things (IoT) technologies and Machine Learning or ML is developed and implemented for intruder detection, in response to the growing insecurity in residential environments in the country. The study was conducted in a single-family home located in the city of Loja, Ecuador, where an infrastructure composed of Aqara sensors under the ZigBee protocol, a ConBee II coordinator, a Raspberry Pi 5 gateway, and a solar energy backup system was installed. The methodology used integrated Design Science Research (DSR) for the system's design and implementation and CRISP-DM for the development of the ML model. Data was collected over approximately two and a half months (March–May 2025), recording 28 variables from the sensors. The processing was carried out using a microservices architecture deployed in Docker, with a backend in FastAPI, a frontend in React, and InfluxDB and PostgreSQL databases. In the experimental phase, the Isolation Forest, One-Class SVM, and Elliptic Envelope algorithms were evaluated, with the latter achieving the best performance in cross-validation with an average accuracy of 0.85. However, when validating the system with simulated intrusion patterns, critical limitations were observed: a low recall (20%), meaning only one out of five intrusions was detected, and a high rate of false positives (86.7%), classifying a significant portion of normal activity as anomalous. These results demonstrate that the integration of IoT and ML provides a solid foundation for intelligent home security systems, although the current model presents constraints that limit its

reliability in critical security scenarios without significant adjustments and optimizations, such as extending the data collection period.

Keywords: Smart home security, Internet of Things (IoT), Machine learning (ML), Intrusion detection, ZigBee, Elliptic Envelope.

1. CAPÍTULO I

EL PROBLEMA

1.1. Problema de investigación

1.1.1. Contexto temático

En la era digital actual la seguridad doméstica se ha convertido en una gran preocupación para los propietarios de viviendas en todo el mundo, con el aumento de las tecnologías de la información y la comunicación los sistemas de seguridad tradicionales están dando paso a soluciones más avanzadas y eficientes. El IoT y el ML están revolucionando la forma en que protegemos nuestros hogares y ofreciendo nuevas posibilidades para la detección de intrusos (Long et al., 2021).

El IoT permite la interconexión de dispositivos físicos a través de internet facilitando la recopilación y el intercambio de datos en tiempo real, que combinado con el ML permite a los sistemas aprender y mejorar automáticamente a partir de la experiencia sin ser programados explícitamente, lo que está abriendo nuevas fronteras en la seguridad doméstica inteligente (Albulayhi & Sheldon, 2021).

Los sistemas de seguridad basados en IoT y ML ofrecen ventajas significativas sobre los métodos convencionales, incluyendo una mayor precisión en la detección de intrusos, la capacidad de aprender y adaptarse a los patrones de comportamiento de los residentes, así como la posibilidad de enviar alertas en tiempo real a través de múltiples canales de comunicación.

En el contexto ecuatoriano, específicamente en la provincia de Loja la implementación de estos sistemas avanzados de seguridad doméstica contiene mucha relevancia debido al aumento de la delincuencia en la región, lo cual sugiere la necesidad de soluciones más efectivas y autónomas para proteger los hogares, esto se ha vuelto esencial considerando que muchos robos ocurren durante la noche o los fines de semana, cuando las casas están desocupadas (INEC, 2023).

1.1.2. Problematización

El incremento de la inseguridad en los hogares del Ecuador y particularmente en la provincia de Loja plantea un gran desafío para la comunidad y las autoridades, ya que los sistemas de seguridad tradicionales han demostrado ser insuficientes frente a esta creciente amenaza y presentan varias limitaciones que comprometen su eficacia (INEC, 2024). Una de las causas de la poca eficiencia de los sistemas tradicionales es su poca o nula flexibilidad y autonomía, ya que por lo general requieren intervención humana constante para su operación y monitoreo, lo que los hace vulnerables a errores y retrasos en la respuesta. Además, su capacidad para detectar intrusos es limitada, ya normalmente son basados en sensores simples que pueden activarse por falsos positivos como el movimiento de mascotas o cambios en las condiciones ambientales en general (Manivannan, 2023).

Otra problemática es la falta de conectividad y comunicación en tiempo real de estos sistemas, ya que en muchos casos las alertas no llegan a los usuarios con la rapidez necesaria, especialmente cuando los hogares están vacíos durante largos períodos como noches o fines de semana, esta demora en la notificación puede ser perjudicial para los propietarios, dando a los intrusos más tiempo para actuar sin ser detectados.

Esta incapacidad de los sistemas tradicionales para aprender y adaptarse a los patrones de comportamiento de los residentes representa un problema importante, resultando en una alta tasa de falsas alarmas que no solo pueden ser molestas para los propietarios, sino que también pueden llevar a una desensibilización ante las alertas, reduciendo la eficacia general del sistema de seguridad.

Las consecuencias de estas limitaciones son múltiples, como la sensación de inseguridad entre los residentes que aumenta y afecta su calidad de vida y bienestar psicológico. Además, la ineficacia de los sistemas de seguridad puede resultar en pérdidas materiales y daños a la propiedad cuando ocurren intrusiones exitosas. adicional a esto, a nivel comunitario, la percepción de inseguridad puede afectar negativamente la convivencia social y el desarrollo económico de la región.

Frente a esta problemática surge la necesidad de implementar soluciones más avanzadas y eficientes como los sistemas de seguridad doméstica inteligentes basados en IoT y ML, que se presentan como una alternativa prometedora para abordar estas limitaciones. Estos sistemas tienen el potencial de ofrecer una detección de intrusos más precisa, alertas

en tiempo real a través de múltiples canales de comunicación, capacidad de aprender y adaptarse a los patrones de comportamiento de los residentes, reduciendo así las falsas alarmas y mejorando la eficacia general de la seguridad doméstica (Alani & Awad, 2023).

1.2. Interrogantes de la investigación

¿Cuáles son las mejores prácticas y tecnologías más avanzadas para la implementación de sistemas de seguridad doméstica basados en IoT y ML?

¿Cuáles son los componentes y configuraciones necesarios para desarrollar una infraestructura IoT robusta utilizando dispositivos con tecnología ZigBee para la recolección de datos en tiempo real?

¿Qué algoritmo de ML es más efectivo para la detección de intrusos y cómo puede integrarse en un sistema de seguridad doméstica con notificaciones multicanal?

1.3. Objetivos de la investigación

1.3.1. Objetivo general

Implementar un sistema de seguridad doméstica inteligente que utilice tecnologías IoT y aprendizaje automático (Machine Learning) para la detección de intrusos dentro de una vivienda, basado en alertas en tiempo real y la conectividad mediante varios canales de comunicación.

1.3.2. Objetivos específicos

- Realizar una revisión de literatura relacionada con sistemas de seguridad doméstica basados en IoT y aprendizaje automático, identificando las mejores prácticas y tecnologías más avanzadas.
- Desarrollar una infraestructura IoT robusta utilizando dispositivos con tecnología ZigBee para la recolección de datos en tiempo real de diversos sensores en el entorno doméstico.
- Implementar un sistema de seguridad doméstica que incorpore un algoritmo de aprendizaje automático para la detección de intrusos de acuerdo a los patrones de actividad del hogar y alertas de notificación multicanal, asegurando la comunicación continua.
- Validar el sistema de seguridad doméstica en un caso de estudio, considerando diferentes patrones de comportamiento de intrusos.

1.4. Hipótesis de trabajo

La implementación de un sistema de seguridad doméstica inteligente que utilice tecnologías IoT y aprendizaje automático (Machine Learning) permitirá la detección de intrusos dentro de una vivienda, basado en alertas en tiempo real y la conectividad mediante varios canales de comunicación.

1.4.1. Hipótesis alternativa

La implementación de un sistema de seguridad doméstica inteligente que utilice tecnologías IoT y ML (Machine Learning) no permitirá la detección de intrusos dentro de una vivienda, basado en alertas en tiempo real y la conectividad mediante varios canales de comunicación.

1.5. Categorización de variables

1.5.1. Variable Independiente

Sistema de seguridad doméstica inteligente que utilice tecnologías IoT y ML.

1.5.2. Variable dependiente

Detección de intrusos dentro de una vivienda, basado en alertas en tiempo real y la conectividad mediante varios canales de comunicación.

1.6. Justificación

La seguridad en los hogares es una gran preocupación especialmente en lugares como la provincia de Loja Ecuador, donde los índices de delincuencia están aumentando y los sistemas de seguridad tradicionales como las alarmas, cámaras, etc. no siempre son suficientes, ya que suelen depender de la intervención humana y no siempre pueden detectar todas las formas de intrusión de manera efectiva.

El uso de la IoT y el ML en sistemas de seguridad doméstica proporciona una solución eficiente para mejorar la seguridad, los dispositivos IoT como sensores y cámaras inteligentes pueden monitorear el hogar las 24 horas recopilando datos en cada momento. Además, con la ayuda de algoritmos de ML estos datos se pueden analizar para detectar actividades anormales y alertar a los propietarios inmediatamente (Saba et al., 2022).

1.6.1. Beneficios importantes

Un sistema de seguridad que utiliza IoT y ML puede enviar alertas instantáneas a los propietarios cuando se detecta algo sospechoso, de esta manera ofrece una respuesta rápida, lo cual es muy importante para prevenir robos y otras amenazas, especialmente cuando la casa está vacía.

Los algoritmos de ML pueden distinguir mejor entre actividades normales y sospechosas, reduciendo las falsas alarmas y evitando que los propietarios sean molestados por alertas innecesarias y podrán confiar en que una alerta es realmente importante (Singh et al., 2021).

Los sistemas basados en IoT pueden adaptarse fácilmente a diferentes tipos de hogares y crecer según sea necesario, con la tecnología ZigBee por ejemplo, ya que permite que muchos dispositivos trabajen juntos de manera eficiente, creando una red de seguridad robusta. Además, estos sistemas pueden aprender y mejorar con el tiempo, haciéndose cada vez más efectivos.

Estos sistemas pueden enviar notificaciones a través de varias vías como aplicaciones móviles, correos electrónicos y mensajes de texto, lo que asegura que los propietarios siempre estén informados y puedan tomar acciones rápidas en caso de emergencia.

1.6.2. Impacto social y económico

Un sistema de seguridad doméstica inteligente, además de ayudar a proteger el hogar también influye en la calidad de vida de las personas alrededor, ya que, si los vecinos perciben que la zona es más segura, baja el nerviosismo y la preocupación diaria y con eso se evitan gastos innecesarios por robos o daños. Además, todo lo que implica diseñar, instalar y mantener este tipo de sistemas abre la puerta a nuevos empleos y hasta puede motivar a que aparezcan pequeñas empresas o proyectos tecnológicos en la misma comunidad.

1.6.3. Viabilidad técnica

En cuanto a la parte técnica, este proyecto es bastante realizable porque hoy en día las tecnologías IoT y los algoritmos de ML están al alcance y siguen mejorando, el protocolo ZigBee por ejemplo, es muy usado y tiene buena reputación para conectar varios sensores entre sí sin problemas, a eso se suman un montón de bibliotecas y herramientas que hacen más sencilla la parte de programación y análisis de datos. Todo junto crea una base bastante

sólida para poner en marcha un sistema de seguridad doméstica que funcione bien y que se pueda mejorar con el tiempo (Saba et al., 2022).

2. CAPÍTULO II

MARCO REFERENCIAL

2.1. Antecedentes

El desarrollo de sistemas de seguridad doméstica ha avanzado significativamente con la integración de tecnologías IoT y el uso de algoritmos de ML. Por tal motivo, este marco de antecedentes se centra en la revisión de estudios y proyectos relevantes que abordan la aplicación de estas tecnologías en la detección de intrusos, proporcionando un contexto sólido para la investigación propuesta.4

2.1.1. Evolución de la seguridad doméstica con IoT

La evolución de la seguridad doméstica ha sido impulsada por la creciente adopción de tecnologías IoT, Jacinto et al. desarrollaron un prototipo para IoT en viviendas inteligentes destacando la importancia de integrar sensores y dispositivos en un sistema integrado y coordinado que permita automatizar la seguridad del hogar, este estudio muestra la viabilidad de utilizar IoT para mejorar la seguridad doméstica, proporcionando un marco práctico que puede ser adaptado y expandido en la investigación propuesta (Jacinto et al., 2024). En Ecuador, se ha comenzado a aplicar esta tecnología en la seguridad doméstica, adaptándola a las necesidades locales.

Por ejemplo,(González G. Alvaro F., 2024) de la Universidad Nacional de Loja, estudió la seguridad en redes IoT como LoRaWAN que normalmente se usan en industrias, pero también pueden aplicarse en hogares, estos estudios son importantes porque ayudan a entender cómo usar IoT para proteger mejor las casas en la región

2.1.2. Algoritmos de ML en seguridad doméstica

El uso de algoritmos de ML para la detección de intrusos ha demostrado ser una herramienta efectiva en la mejora de la seguridad doméstica, Alani & Awad presentaron un sistema inteligente de detección de intrusos de dos capas, que combina técnicas de aprendizaje profundo con IoT para detectar amenazas en tiempo real, este enfoque es particularmente relevante para la tesis ya que ofrece un modelo de referencia sobre cómo implementar la detección de intrusos utilizando tecnologías avanzadas (Alani & Awad, 2023).

De manera similar**,** Saba et al. desarrollaron un sistema de detección de intrusos basado en anomalías utilizando un modelo de aprendizaje profundo para redes IoT, este estudio es importante para la investigación propuesta ya que proporciona evidencia empírica sobre la eficacia de los algoritmos de aprendizaje en la mejora de la precisión y la reducción de falsos positivos en sistemas de seguridad doméstica (Saba et al., 2022).

En el contexto ecuatoriano, trabajos como la **"Propuesta de mejoras de alertas de seguridad de dispositivos de IoT mediante inteligencia artificial"** resaltan cómo la implementación de técnicas avanzadas de ML puede optimizar la respuesta ante amenazas en entornos domésticos, este estudio sugiere que la aplicación de algoritmos más avanzados mejora la precisión de las alertas y también permite una respuesta más rápida y eficiente ante posibles intrusiones, lo que es muy importante en la protección del hogar (Duque Quevedo Odalys Rashel, 2024).

2.1.3. Desafíos y soluciones en la seguridad de redes IoT

La seguridad en redes IoT es fundamental para el diseño e implementación de sistemas de seguridad doméstica, garantizando la integridad, confidencialidad y disponibilidad de la información, Trujillo Borja et al. muestran un análisis comparativo de protocolos de enrutamiento aplicados en redes de sensores inalámbricos (WSN) utilizadas en ambientes industriales, aunque el enfoque es industrial las conclusiones sobre la eficiencia y seguridad de los protocolos pueden ser extrapoladas a entornos domésticos (Trujillo Borja et al., 2023).

Asimismo, Manivannan abordó la mejora de la seguridad en IoT mediante la detección de anomalías y la prevención de intrusiones impulsadas por inteligencia artificial, este estudio aporta una perspectiva integral sobre cómo combinar el ML con IoT para reforzar la seguridad en entornos domésticos, lo que es directamente aplicable a la investigación propuesta (Manivannan, 2023).

2.1.4. Estudios de caso y proyectos relevantes en el contexto Local

En el contexto ecuatoriano, la necesidad de mejorar la seguridad doméstica se refleja en las estadísticas de criminalidad proporcionadas por la Policía Nacional del Ecuador y la Dirección Nacional de Análisis de la Información, documentan que sigue siendo preocupante los delitos contra la propiedad en la subzona de Loja, estos datos justifican la urgencia de

desarrollar sistemas de seguridad doméstica avanzados y contextualizan la relevancia del proyecto propuesto (Policía Nacional Del Ecuador & Dirección Nacional De Análisis De La Información, 2024).

Figura 1.

Comparativa de delincuencia en Loja (2023-2024).

SUBZONA LOJA	ENE A DIC 2023	DEL 01 DE ENE AL 26 DE JUL 2023	DEL 01 DE ENE AL 26 DE JUL 2024	VARIACION ABSOLUTA	VARIACION PORCENTUAL	PESO DELICTUAL
ROBO A PERSONAS	275	156	154	-2	-1%	43%
ROBO DOMICILIOS	302	159	105	-54	-34%	29%
ROBO A UNIDADES ECONÓMICAS	81	45	36	-9	-20%	10%
ROBO DE BIENES, ACCESORIOS Y AUTOPARTES DE VEHÍCULOS	102	55	25	-30	-55%	7%
ROBO A MOTOS	34	20	20	0	0%	6%
ROBO A CARROS	43	22	18	-4	-18%	5%
ROBO EN EJES VIALES O CARRETERAS	3	0	4	4	400%	1%
TOTAL	840	457	362	-95	-21%	100%

Nota. Reporte de la DAI de la Policía Nacional Del Ecuador, Sobre niveles de violencia y delincuencia subzona Loja.

Como se observa en la Figura 1 durante el período de tiempo del 01 de enero al 26 de julio del 2024, registra un decremento del -21% (-95 eventos) en comparación al mismo período del año2023.

Figura 2.

Indicadores de delincuencia, subzona Loja, julio 2023-julio 2024.

SUBZONA LOJA	JULIO	DEL 01 AL 26 DE JULIO 2023	DEL 01 AL 26 DE JULIO 2024	VARIACION ABSOLUTA	VARIACION PORCENTUAL	PESO DELICTUAL
ROBO DOMICILIOS	29	25	16	-9	-36%	37%
ROBO A UNIDADES ECONÓMICAS	9	6	4	-2	-33%	9%
ROBO A MOTOS	2	2	1	-1	-50%	2%
ROBO DE BIENES, ACCESORIOS Y AUTOPARTES DE VEHÍCULOS	6	5	4	-1	-20%	9%
ROBO EN EJES VIALES O CARRETERAS	0	0	0	0	0%	0%
ROBO A PERSONAS	22	19	14	-5	-26%	33%
ROBO A CARROS	6	6	4	-2	-33%	9%
TOTAL	74	63	43	-20	-32%	100%

Nota. Reporte de la DAI de la Policía Nacional Del Ecuador, Sobre niveles de violencia y delincuencia subzona Loja.

Según la Figura 2, se muestra que del 01 al 26 de julio del 2024 registra un decremento del -32% (-20 eventos) en comparación al mismo períododelaño2023, a pesar de la reducción en los incidentes de robos a domicilios en Loja durante el 2024, con un notable decremento del 21% y 32% en los períodos analizados, la situación sigue siendo preocupante. Este

descenso no debe interpretarse como una eliminación del problema ya que los niveles de criminalidad permanecen significativamente altos, la persistencia de estos incidentes indica que, aunque se han logrado avances, las medidas de seguridad actuales podrían no ser suficientes para garantizar la protección completa de los hogares en la ciudad. Es esencial mantener y fortalecer las estrategias de seguridad para continuar reduciendo estos índices y brindar una mayor tranquilidad a los residentes de Loja.

El estudio de Saa Ayala y Soto Valle sobre un sistema de seguridad basado en IoT para viviendas urbanas ofrece un ejemplo práctico de cómo estas tecnologías pueden ser implementadas en entornos residenciales, destacando los desafíos y soluciones que surgieron durante el desarrollo del proyecto. Este estudio es relevante para la tesis ya que proporciona lecciones aprendidas y mejores prácticas que pueden ser aplicadas en la investigación propuesta (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023).

2.1.5. Integración de IoT y ML en la seguridad doméstica

La integración de IoT y ML en sistemas de seguridad doméstica representa un avance significativo en la protección de hogares, Long et al. realizaron un estudio sobre las técnicas de detección de intrusos basadas en ML para IoT, destacando cómo estas tecnologías pueden ser optimizadas para mejorar la seguridad y la eficiencia de los sistemas de seguridad doméstica. Este estudio es importante para entender las diferentes estrategias de implementación y las herramientas disponibles para desarrollar un sistema de seguridad confiable (Long et al., 2021).

Por otro lado, Taiwo et al. exploraron un sistema de control y seguridad inteligente basado en aprendizaje profundo, aplicable a hogares inteligentes, este estudio proporciona un marco conceptual para la implementación de algoritmos de aprendizaje profundo en la seguridad doméstica, demostrando cómo estas tecnologías pueden ser utilizadas para anticipar y minimizar amenazas potenciales en tiempo real (Taiwo et al., 2022).

2.2. Marco teórico

Este marco teórico se construye sobre una revisión de la literatura relevante en torno a la implementación de sistemas de seguridad doméstica inteligentes, basados en IoT y ML para la detección de intrusos, esta revisión abarca diversas fuentes académicas y científicas para asegurar un fundamento robusto para el proyecto.

2.2.1. Proceso de revisión de la literatura

El proceso de revisión de la literatura fue diseñado para identificar estudios relevantes y actuales que informen sobre las tecnologías utilizadas en la seguridad doméstica inteligente, se emplearon varias bases de datos de alta reputación, incluyendo IEEE Xplore y Google Scholar para garantizar que se cubran los avances más recientes en el campo.

2.2.2. Identificando la literatura relevante

Para identificar la literatura relevante, se formularon cadenas de búsqueda específicas y se utilizaron operadores booleanos para combinar términos clave, tanto en IEEE Xplore y Google Scholar.

2.2.3. Cadena de búsqueda

La Tabla 1 muestra la cadena de búsqueda que permitió focalizar la indagación en estudios que abordan tanto la seguridad doméstica basada en IoT y ML como la detección de intrusos, dando un total de 30 publicaciones: (2018 - 2024).

Tabla 1.

Secuencia de búsqueda utilizada en la base de datos científica

Criterio	IEEE Xplorer	Google Scholar
Cadena de búsqueda	("All Metadata":Intelligent home security system based on iot and machine learning) AND ("All Metadata":Intrusion detection).	("IoT home security" AND "machine learning" AND "ZigBee")
SUBTOTAL	**30**	**21**
TOTAL	**51**	

Nota. Obtenido de (IEEE Xplorer), (Google Scholar).

2.2.4. Búsqueda de documentos

La búsqueda de documentos se llevó a cabo en las bases de datos mencionadas con un enfoque en artículos publicados entre 2015 y la fecha actual, para esto se consideraron estudios que ofrecen una visión integral sobre el desarrollo de sistemas de seguridad basados en IoT y ML, y cómo estos sistemas abordan los desafíos de la detección de intrusos en entornos domésticos.

2.2.5. Selección de artículos

Para la selección de los artículos y documentos que sustentan esta investigación, se diseñó un proceso de tres fases con el fin de garantizar que las fuentes seleccionadas sean pertinentes, actuales y directamente aplicables al tema de estudio.

2.2.5.1. Primera fase: Aplicación de criterios de inclusión y exclusión. En la primera fase se establecieron criterios claros de inclusión y exclusión para filtrar los trabajos de investigación, aquí solo se consideraron artículos y tesis publicadas en los últimos 9 años, que abordaran temas como la seguridad doméstica, IoT y los algoritmos de ML aplicados a la detección de intrusos, también se incluyeron investigaciones sobre la implementación de infraestructuras IoT, la configuración de redes ZigBee, la redundancia en canales de comunicación y sistemas de notificaciones inteligentes. Se excluyeron aquellos trabajos que no guardaran relación directa con estos temas o que no cumplieran con altos estándares de calidad académica, como artículos con baja relevancia o publicados en revistas de menor impacto.

2.2.5.2. Segunda fase: Selección y organización de los artículos. En la segunda fase se procedió a seleccionar los artículos más relevantes utilizando las cadenas de búsqueda previamente definidas, los documentos seleccionados fueron organizados en una base de datos categorizados por título, año de publicación, resumen y palabras clave, esto permitió identificar rápidamente los estudios que ofrecían una mayor contribución al desarrollo del sistema de seguridad doméstica basado en IoT y ML, asegurando una cobertura amplia y detallada de todos los aspectos críticos del proyecto.

2.2.5.3. Tercera fase: Evaluación y análisis detallado. En la etapa final se realizó una evaluación del contenido de cada documento seleccionado, para ello se puso especial énfasis en las secciones como el resumen, la introducción, el desarrollo

metodológico y las conclusiones/recomendaciones. Los documentos que aportaban estudios empíricos, análisis de casos concretos o propuestas metodológicas directamente aplicables al proyecto de seguridad doméstica fueron priorizados, aquellos que demostraron mayor relevancia y aplicabilidad tras esta evaluación fueron los que se incluyeron finalmente en la revisión de la literatura.

Tabla 2.

Selección de documentos

Base de datos	Fase I	Fase II	Fase III
IEEE Xplorer	30	10	8
Google Scholar	21	10	2
Total	51	20	10

Nota. Obtenido de (IEEE Xplorer), (Google Scholar).

En la Tabla 3, se muestra la selección completa de todos los documentos relevantes para el presente proyecto.

Tabla 3.

Documentos seleccionados.

Código	Título	Autor	Información relevante
A1	MCIDS-Multi Classifier Intrusion Detection system for IoT Cyber Attack using Deep Learning algorithm.	(Singh et al., 2021)	Sistema de detección de intrusos basado en aprendizaje profundo, utiliza dataset UNSW-NB15, alta precisión y bajos falsos positivos en la detección de múltiples tipos de ataques,
A2	A Survey of Machine Learning-based IoT Intrusion Detection Techniques.	(Long et al., 2021)	Investigación sobre detección de intrusos en IoT utilizando técnicas de ML, análisis de amenazas y evaluación de tecnologías de detección,
A3	An Intelligent Two-Layer Intrusion Detection System for the Internet of Things.	(Alani & Awad, 2023)	Sistema de detección de intrusos en dos capas, utilizando técnicas de ML, alta precisión y bajo tiempo de procesamiento en sistemas IoT,
A4	Anomaly Detection in IoT Based PIR Occupancy Sensors to Improve	(Samani et al., 2020)	Detección de anomalías en sensores PIR para mejorar la eficiencia energética en edificios, utilizando una red neuronal profunda.

	Building Energy Efficiency.		
A5	Enhanced Intelligent Smart Home Control and Security System Based on Deep Learning Model.	(Taiwo et al., 2022)	Sistema de automatización del hogar basado en IoT con detección de intrusos utilizando un modelo de aprendizaje profundo, alta precisión en la clasificación de patrones de movimiento.
A6	A survey on Security and Privacy Challenges in Smarthome based IoT.	(Ahmed & Zeebaree, 2021)	Revisión de los desafíos de seguridad y privacidad en aplicaciones de hogares inteligentes basadas en IoT, clasificación en categorías y propuestas de mejora.
A7	An Improved Data Anomaly Detection Method Based on Isolation Forest	(Xu DWang YMeng YZhang Z, 2017)	SA-iForest: Método mejorado de Isolation Forest que selecciona árboles de aislamiento óptimos usando simulated annealing, aumentando precisión y eficiencia en la detección de anomalías en datos.
A8	Isolated forest-based ZigBee Device Identification Using Adaptive Filter Coefficients	(Chen ZPeng LFu H, 2022)	Identificación ZigBee: Método que extrae huellas RF usando coeficientes LMS y combina Random Forest e Isolation Forest para identificar dispositivos IoT, logrando alta precisión incluso con ruido.
A9	Deep Isolation Forest for Anomaly Detection	(Deep Isolation Forest for Anomaly Detection, 2023)	Deep iForest: Extensión de Isolation Forest que usa representaciones aleatorias mediante redes neuronales para particiones no lineales, mejorando la detección de anomalías complejas y manteniendo alta escalabilidad.
A10	Isolation Forest Based Anomaly Detection: A Systematic Literature Review	(Al Farizi WHidayah IRizal M, 2021)	IF Review: Revisión sistemática de mejoras a Isolation Forest, identificando debilidades por selección aleatoria de variables y soluciones pre-IF, post-IF y de mejora de método.

Nota. Obtenido de (IEEE Xplorer), (Google Scholar).

2.3. Marco conceptual

2.3.1. Introducción a la seguridad doméstica con IoT

2.3.1.1. Definición y conceptualización del IoT. El IoT se refiere a la interconexión de dispositivos físicos a través de la red de internet, permitiendo la recopilación

y el intercambio de datos. (Li et al., 2015) explican que esta tecnología permite que objetos cotidianos se comuniquen y actúen en conjunto para cumplir tareas específicas.

Relevancia del IoT en la Seguridad Doméstica: El IoT ha evolucionado considerablemente en los últimos años, proporcionando soluciones avanzadas para el monitoreo y control remoto en tiempo real, el IoT combinado con modelos de aprendizaje profundo permite crear sistemas de seguridad más inteligentes y adaptativos, estos sistemas mejoran la detección de intrusos y también optimizan la respuesta a eventos, lo que es muy importante con la actual creciente preocupación por la seguridad en entornos nos residenciales (Taiwo et al., 2022).

2.3.1.2. Impacto del IoT en la seguridad doméstica.

- **Ventajas del IoT en Seguridad:** IoT aporta varias ventajas en el ámbito de la seguridad doméstica, como la vigilancia continua en tiempo real, la automatización de respuestas ante incidentes y la interconexión inteligente de múltiples dispositivos para conformar un sistema integral. Estas capacidades permiten detectar y prevenir intrusiones de forma proactiva y optimizar la reacción ante emergencias, logrando que los sistemas de seguridad sean más eficientes, coordinados y confiables en la protección del hogar (Taiwo et al., 2022).
- **Desafíos del IoT en la Seguridad Doméstica:** A pesar de sus ventajas, el IoT en la seguridad doméstica enfrenta muchos desafíos, especialmente en lo que respecta a la seguridad de los datos y la complejidad de integrar dispositivos de diferentes fabricantes. La protección de la información es esencial ya que los dispositivos conectados pueden ser vulnerables a ciberataques, lo que podría comprometer la seguridad del sistema y la privacidad de los usuarios (Saba et al., 2022).

2.3.1.3. Evolución del IoT en la seguridad doméstica

- **Historia y Desarrollo:** IoT ha recorrido un largo camino desde sus inicios, evolucionando de sistemas básicos de automatización a redes complejas que ahora incorporan inteligencia artificial para mejorar la seguridad doméstica, la tecnología IoT ha avanzado permitiendo la integración de múltiples dispositivos y la reducción de costos y facilitando su adopción en hogares comunes. Este desarrollo ha democratizado el acceso a sistemas de seguridad más sofisticados mejorando la protección en entornos residenciales (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023).

- **Estudios de Caso Globales y Locales:** Revisión de proyectos que han implementado sistemas de seguridad basados en IoT, tanto a nivel global como en contextos locales como Ecuador (González G. Alvaro F., 2024), por ejemplo, explora la implementación de redes LoRaWAN en la seguridad doméstica demostrando cómo estas tecnologías pueden ser adaptadas para mejorar la protección en entornos urbanos.

2.3.2. Tecnologías utilizadas en IoT para seguridad doméstica

2.3.2.1. Protocolos de comunicación en IoT

- **ZigBee:** es un protocolo de comunicación de bajo consumo energético que es especialmente adecuado para redes de sensores y dispositivos en el hogar, ZigBee es preferido en sistemas de seguridad doméstica debido a su capacidad para crear redes de malla robustas y confiables, esto que garantiza una comunicación estable incluso cuando se utilizan múltiples dispositivos en diferentes ubicaciones dentro de un hogar (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023).
- **Z-Wave:** es un protocolo similar a ZigBee pero con un enfoque más comercial, es una alternativa para la conectividad de dispositivos de seguridad en el hogar debido a su amplia compatibilidad con una gran variedad de dispositivos, Z-Wave es altamente eficiente para la integración de sistemas de seguridad doméstica, ofreciendo una conectividad confiable y fácil de implementar en diversas configuraciones residenciales (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023).
- **Wi-Fi y Bluetooth:** son tecnologías ampliamente utilizadas para conectar dispositivos en el hogar, cada una con sus particularidades, aunque Wi-Fi ofrece una mayor velocidad de transmisión de datos, ZigBee y Z-Wave son más adecuados para aplicaciones de seguridad doméstica que requieren baja latencia y alta eficiencia energética, Wi-Fi es ideal para dispositivos que necesitan transmitir grandes volúmenes de datos, mientras que Bluetooth se utiliza comúnmente para conexiones de corto alcance como controles remotos o sensores de proximidad (Manivannan, 2023).

2.3.2.2. Plataformas de gestión de IoT

- **Raspberry Pi como Centro de Control:** La Raspberry Pi, especialmente en su versión más reciente es ampliamente utilizada como el cerebro de los sistemas de seguridad doméstica IoT debido a su capacidad de procesamiento mejorada, la Raspberry Pi

con su potencia de cómputo y versatilidad es una excelente opción para gestionar múltiples dispositivos IoT simultáneamente, esto la convierte en el núcleo de muchos sistemas de automatización del hogar (Taiwo et al., 2022).

- **ConBee II Zigbee USB Dongle:** Este Dongle permite que la Raspberry Pi se comunique con dispositivos ZigBee actuando como un coordinador de red, la importancia de un coordinador confiable en la red ZigBee para asegurar la estabilidad y eficiencia del sistema, facilitando una comunicación robusta entre los dispositivos IoT en el hogar (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023).
- **Debian 12:** Llamada "Bookworm" y lanzada en junio de 2023, es la versión estable más reciente de Debian, incluye más de 64,000 paquetes de software con actualizaciones que mejoran la seguridad y la estabilidad del sistema, usa el kernel Linux 6.1 LTS que ofrece mejor soporte para hardware moderno y un rendimiento más eficiente. Además, Debian 12 agrega una nueva sección en sus repositorios llamada non-free-firmware que permite incluir firmware propietario necesario para que algunos dispositivos funcionen correctamente. Esta versión ofrece un sistema libre, seguro y confiable, adecuado para diferentes tipos de proyectos y entornos. (Debian Project, 2023).

2.3.3. Sensores ZigBee en seguridad doméstica

2.3.3.1. Tipos de sensores ZigBee y su aplicación en seguridad

2.3.3.1.1. Sensores de movimiento simples. Los sensores de movimiento simples como los sensores infrarrojos pasivos (PIR) detectan cambios en la radiación infrarroja del entorno, típicamente emitida por personas o animales en movimiento, estos sensores funcionan al medir el calor y el movimiento, activando alarmas o luces cuando se detecta una presencia en su campo de visión (Samani et al., 2020).

- **Aplicación:** Generalmente se utilizan en sistemas de iluminación automática y alarmas de seguridad por su bajo costo y efectividad en la detección de movimiento general.

2.3.3.1.2. Sensores de movimiento con reconocimiento de personas. Estos sensores combinan tecnologías de detección de movimiento con algoritmos de reconocimiento de patrones, permitiendo diferenciar entre humanos y otros objetos en movimiento, estos sensores utilizan técnicas avanzadas como visión por computadora y aprendizaje profundo para identificar la forma y movimiento específicos de una persona (Khalid et al., 2019).

- **Aplicación:** Son esenciales en sistemas de seguridad avanzados donde se requiere minimizar los falsos positivos, ofreciendo una detección más precisa y enfocada en la seguridad perimetral y control de acceso.

2.3.3.1.3. Sensores de puertas y ventanas. Estos sensores monitorean la apertura y cierre de puertas y ventanas, detectando accesos no autorizados, (Adhikary et al., 2024) explican que integrados en una red ZigBee estos sensores proporcionan una cobertura completa del hogar, alertando inmediatamente en caso de una intrusión.

2.3.4. Funcionamiento y configuración de sensores ZigBee

2.3.4.1. Redes de Malla ZigBee. Las redes de malla ZigBee son fundamentales para garantizar la robustez y confiabilidad de los sistemas de seguridad doméstica, especialmente en áreas grandes o con obstáculos, estas redes permiten que cada dispositivo actúe como un nodo, retransmitiendo señales y garantizando que la comunicación continúe incluso si uno de los dispositivos falla, esto mantiene la integridad del sistema en entornos complejos. La capacidad de auto-reparación y la escalabilidad de las redes ZigBee las hacen ideales para aplicaciones en seguridad doméstica donde la estabilidad y cobertura completa son muy importantes (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023).

La topología de malla también facilita la adición de nuevos dispositivos sin comprometer el rendimiento de la red, permitiendo que los sistemas crezcan y se adapten a las necesidades cambiantes de seguridad en un hogar, su baja latencia y el consumo reducido de energía son beneficios adicionales que consolidan a ZigBee como una de las opciones más efectivas para la interconexión de dispositivos en sistemas de seguridad.

2.3.4.2. Instalación y Configuración de Sensores ZigBee. La correcta instalación y configuración de sensores ZigBee es importante para maximizar su efectividad y asegurar una cobertura completa del área a proteger, (Manivannan, 2023) proporciona una guía detallada sobre cómo optimizar la ubicación de los sensores en un entorno doméstico, para una correcta configuración se debe comenzar con la identificación de puntos críticos de acceso y áreas con mayor riesgo de intrusión, colocar los sensores estratégicamente en estos puntos maximiza la detección y minimiza los puntos ciegos.

Para garantizar una cobertura completa es recomendable que los sensores estén separados por distancias óptimas que permitan la comunicación sin interferencias y aseguren que cada sensor esté dentro del rango de al menos dos dispositivos más, reforzando así la red

de malla, verificar que todos los dispositivos estén correctamente emparejados con el coordinador ZigBee como un dongle ConBee II, y que el software de gestión como deCONZ Phoscon esté configurado adecuadamente para integrar todos los sensores en el sistema de seguridad.

El proceso de configuración también incluye la calibración de los sensores para evitar falsos positivos y garantizar que solo se activen en presencia de una amenaza real, este ajuste puede incluir la sensibilidad del sensor, el ángulo de detección y la programación de zonas de alerta específicas en la interfaz de control.

2.3.5. Algoritmos de ML para detección de Intrusos

Concepto y principios del ML. El ML es una herramienta poderosa en la detección de intrusos dentro de sistemas de seguridad doméstica, este tipo de algoritmos permiten analizar patrones de comportamiento y detectar anomalías que podrían indicar una intrusión, mejorando la efectividad de los sistemas de seguridad. El uso de algoritmos de ML en sistemas de seguridad doméstica optimiza la detección de amenazas al analizar grandes volúmenes de datos, ofreciendo respuestas más rápidas y precisas. Esta capacidad de adaptarse convierte al ML en una herramienta poderosa en la protección de hogares (Manivannan, 2023).

El ML es una rama de la inteligencia artificial que trata de la capacidad de las máquinas para aprender de datos previos y mejorar su rendimiento sin ser explícitamente programadas, en la seguridad doméstica estos algoritmos permiten a los sistemas identificar comportamientos normales y detectar desviaciones que podrían indicar una amenaza, lo que enfoque mejora significativamente la efectividad de los sistemas de seguridad permitiendo una detección proactiva de intrusiones antes de que ocurran daños (Taiwo et al., 2022).

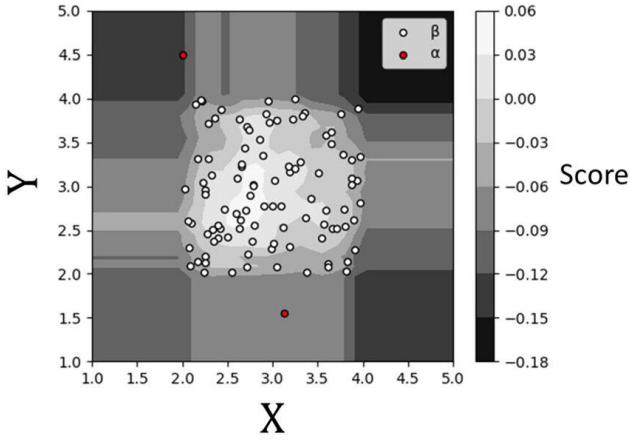
2.3.5.1. Algoritmos comunes utilizados en la detección de intrusos.

- **Isolation Forest:** Parte de la idea de que las anomalías son más fáciles de separar que los datos normales (véase Figura 4), para hacerlo construye árboles de decisión muy simples que dividen los datos de forma aleatoria. De tal manera que, si un dato es en realidad anómalo, quedará aislado en pocas divisiones mientras que los datos normales requieren más pasos para ser separados, este comportamiento se muestra en la Figura 3. El rendimiento de Isolation Forest es muy eficiente porque su complejidad es $\boldsymbol{O(n \log n)}$, al aumentar la cantidad de datos el tiempo de cómputo crece más que de manera lineal, pero no

llega a ser tan costoso como en algoritmos cuadráticos ($\mathbf{O}(\boldsymbol{n^2})$), como los algoritmos que comparan distancias entre todos los puntos, ya que, si se duplica la cantidad de registros el tiempo no se duplica exactamente pero tampoco se dispara de manera exagerada, Isolation Forest se puede usar en conjuntos de datos muy grandes y con muchas variables y lo convierte en una herramienta popular en áreas como la detección de intrusiones, fraudes financieros y la seguridad en entornos IoT (Liu et al., 2008).

Figura 3.

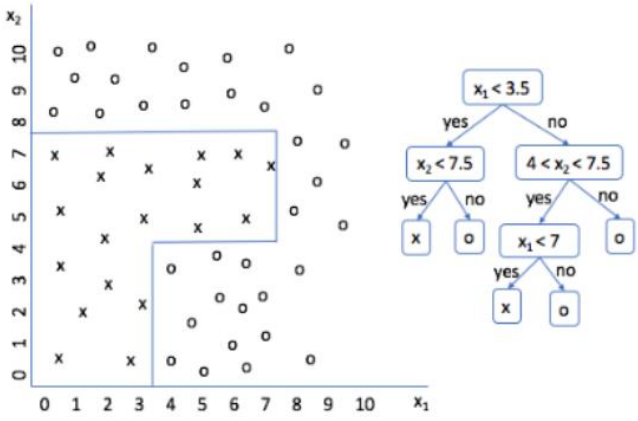
Puntuaciones de anomalías de Isolation Forest.



Nota. Obtenido de (Dougal Ferguson, 2025)

Figura 4.

Fronteras en un árbol de decisión.



Nota. Obtenido de (Iturbe-Araya & Rifà-Pous, 2025)

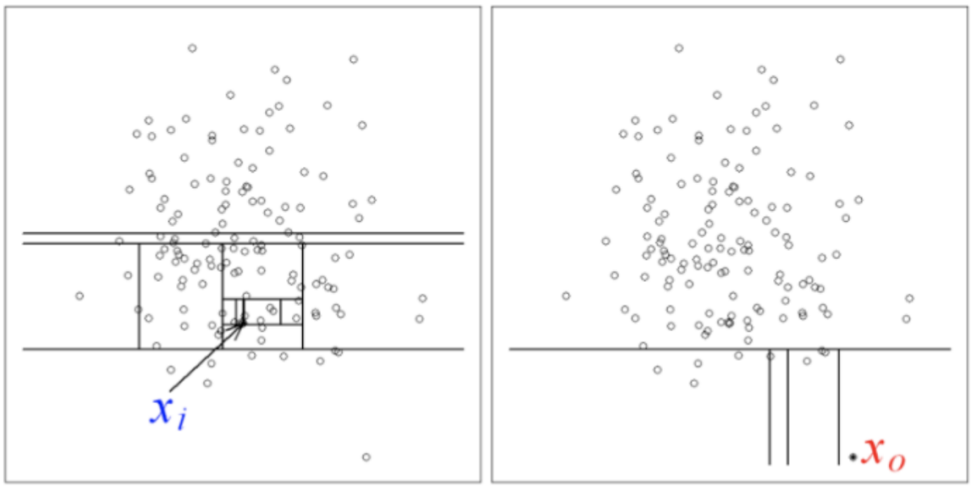
Isolation Forest detecta anomalías midiendo la profundidad que se necesita para aislar una observación de las demás, a esta medida se le llama path length.

- **Menor path length (menos profundidad):** Si una observación se aísla rápidamente (en pocas particiones) es probable que sea una anomalía, esto se debe a que las anomalías suelen ser puntos "extraños" que están alejados de la mayoría de los datos.
- **Mayor path length (más profundidad):** Si una observación requiere muchas particiones para ser aislada es probable que sea un dato normal, esto sucede porque los datos normales están agrupados y se necesita más esfuerzo para separarlos.

El valor final para cada observación es el promedio de su path length en varios árboles dando como resultado una puntuación de anomalía, cuanto menor sea la puntuación mayor será la probabilidad de que se trate de un valor atípico. En la Figura 5 se muestra que en la primera observación (dato normal) tiene un path length mayor que la segunda (anomalía), por lo que la segunda se clasifica como una anomalía con una puntuación más alta.

Figura 5.

Ejemplo de un árbol de Isolation Forest.



Nota. Obtenido de (Iturbe-Araya & Rifà-Pous, 2025)

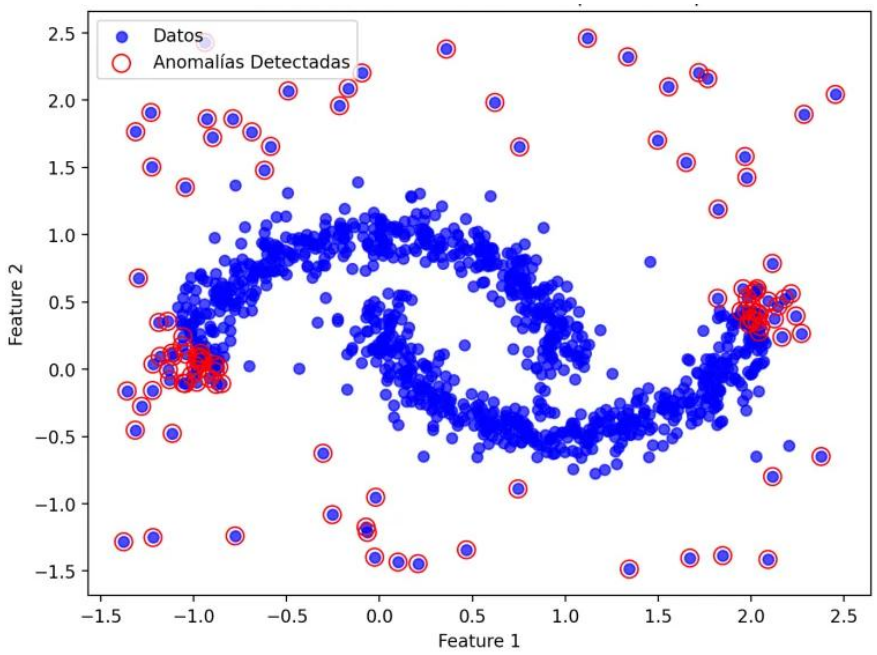
- **Support Vector Machines (SVM):** Son uno de los métodos de aprendizaje supervisado más robustos y ampliamente utilizados para la clasificación de datos. En la detección de intrusos, las SVM son particularmente efectivas porque pueden separar de manera precisa los comportamientos normales de los anómalos utilizando un hiperplano óptimo en un espacio de alta dimensionalidad. Las SVM, al ser entrenadas con datos etiquetados de comportamientos legítimos y maliciosos, pueden aprender a identificar patrones específicos asociados con intrusiones. Esto permite que el sistema de seguridad doméstica responda con alta precisión, reduciendo la incidencia de falsos positivos y

aumentando la confianza en la detección de amenazas reales. La capacidad de las SVM para manejar problemas de clasificación binaria y multicategoría las hace adecuadas para sistemas de seguridad que requieren una clasificación precisa y eficiente (Taiwo et al., 2022).

- **Redes Neuronales Convolucionales (CNN):** Son una de las arquitecturas más avanzadas de redes neuronales, especialmente diseñadas para el procesamiento de datos visuales. En sistemas de seguridad doméstica, las CNN se utilizan para analizar imágenes y videos capturados por cámaras de vigilancia. Las CNN son excepcionalmente efectivas para identificar patrones y características visuales como formas, contornos y movimientos, que son cruciales para la detección precisa de intrusos. Estas redes pueden aprender a reconocer las diferencias entre actividades cotidianas y comportamientos sospechosos, lo que reduce significativamente los falsos positivos y permite una vigilancia más efectiva y en tiempo real. La capacidad de las CNN para generalizar a partir de grandes conjuntos de datos de imágenes las hace ideales para aplicaciones en las que la seguridad visual es una prioridad (Saba et al., 2022).
- **Elliptic Envelope:** Esta técnica mostrada en la Figura 6 permite la detección de anomalías en conjuntos de datos que siguen una distribución Gaussiana, se basa en la idea de que los datos normales tienden a concentrarse en una región central y densa, mientras que los datos anómalos se sitúan en las zonas periféricas y de baja densidad, parte de la premisa de que de los datos normales pueden ser modelados por una distribución Gaussiana multivariante donde visualmente los puntos “normales” forman una nube elíptica definida por la media y la matriz de covarianza. Para detectar anomalías, se construye esta elipse de forma robusta frente a outliers usando el Minimum Covariance Determinant, y la distancia de Mahalanobis determina qué tan lejos está cada punto de la elipse, los puntos que superan un umbral basado en la proporción estimada de anomalías se clasifican como outliers. Es eficaz en contextos multivariados y resistente a outliers pero pierde precisión con distribuciones no gaussianas, datos multimodales y conjuntos de alta dimensión debido a su coste computacional (Rousseeuw & Driessen, 1999).

Figura 6.

Detección de anomalías con Elliptic Envelope.



Nota. Obtenido de *(Iturbe-Araya & Rifà-Pous, 2025)*

- **One-Class SVM:** Es un algoritmo de ML no supervisado orientado a la detección de anomalías o valores atípicos dentro de un conjunto de datos, su objetivo es identificar la distribución de una sola clase denominada clase "normal", y detectar las instancias que se desvían significativamente de esta distribución como anomalías. Esto se logra mediante la construcción de un hiperplano de separación en un espacio de características de alta dimensión que encapsula la mayoría de las instancias normales, maximizando el margen con respecto a los datos anómalos. Los datos que quedan fuera de esta frontera se clasifican como anomalías.

Entre sus parámetros más importantes están el tipo de función kernel, el parámetro nu que controla la proporción de valores atípicos permitidos y gamma que defina la influencia de un solo punto de datos en la construcción del hiperplano, One-Class SVM es útil en situaciones donde hay abundancia de datos normales y escasez o ausencia de datos anómalos para entrenamiento como en detección de fraudes, seguridad informática o control de calidad industrial (Schölkopf et al., 2001).

2.3.5.2. Algoritmos comunes utilizados en la detección de intrusos.

- **Detección de anomalías:** La detección de anomalías es fundamental en la identificación de intrusos a través del ML, ya que permite la identificación temprana de comportamientos inusuales antes de que se conviertan en amenazas, estos sistemas con algoritmos como Isolation Forest y Support Vector Machines (SVM) pueden aprender de los datos históricos recopilados por sensores, cámaras, identificando comportamientos que se desvían de lo normal. La importancia de la detección de anomalías en la seguridad destaca en que esta capacidad permite la activación proactiva de alertas que pueden prevenir daños antes de que ocurran, la efectividad de estos algoritmos radica en su habilidad para analizar grandes volúmenes de datos en tiempo real, lo que los convierte en herramientas muy importantes en la protección de los hogares (Al Farizi et al., 2021; Chen et al., 2022; Manivannan, 2023; D. Xu et al., 2017; H. Xu et al., 20re23).
- **Integración de algoritmos con sensores IoT:** La integración de algoritmos de ML con sensores IoT como ZigBee, ha demostrado ser un avance importante en el desarrollo de sistemas de seguridad doméstica que operan de manera continua y en tiempo real. Esto mejora la capacidad de los sistemas para detectar eventos anómalos, así como también permite una respuesta rápida y precisa ante posibles amenazas amenazas (D. Xu et al., 2017).

Según el estudio de (Taiwo et al., 2022), la integración de sensores IoT con algoritmos avanzados como las Redes Neuronales Convolucionales (CNN) permite a los sistemas de seguridad procesar grandes volúmenes de datos visuales capturados por cámaras de vigilancia, pueden analizar patrones complejos en las imágenes como el movimiento o la forma de los objetos, lo que les permite distinguir entre actividades normales y comportamientos que podrían indicar una intrusión. Al detectar un evento sospechoso el sistema activa de inmediato un conjunto de notificaciones inteligentes que alertan al usuario a través de diversas plataformas como aplicaciones móviles o correos electrónicos, garantizando que la información llegue al destinatario en tiempo real.

(González G. Alvaro F., 2024) señala que la combinación de estos algoritmos con sensores avanzados como ZigBee mejora la capacidad de detección y permite que el sistema aprenda y se adapte a las rutinas específicas del hogar, de esta manera se reduce la cantidad de falsos positivos y mejora la precisión de las alertas. Los sensores ZigBee están conectados

en una red de malla y proporcionan una cobertura completa del área vigilada, garantizando la transmisión de datos incluso si uno de los dispositivos falla.

Asimismo, (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023) destacan que la eficiencia energética de los sensores ZigBee es fundamental para su operación continua en sistemas de seguridad doméstica, ya que estos sensores están diseñados para funcionar de manera autónoma durante largos períodos de tiempo para asegurar que los datos necesarios del análisis y la detección estén siempre disponibles para los algoritmos de ML.

- **Entrenamiento y validación de modelos.** El entrenamiento y la validación de modelos de ML son procesos esenciales para asegurar la precisión y confiabilidad de los sistemas de detección de intrusos, especialmente en redes de sensores inalámbricos (WSN, que al estar basados en inteligencia artificial requieren de un entrenamiento íntegro utilizando datos históricos de la red para aprender a identificar patrones de comportamiento anómalos, durante este entrenamiento se alimenta al modelo con un conjunto de datos etiquetados que representan tanto comportamientos normales como anómalos dentro de la red. Esto permite que el sistema aprenda a diferenciar entre el tráfico de red legítimo y posibles ataques o intrusiones (Al Farizi et al., 2021; Velastegui Morales Jhoselyn Lizeth & Cuzme Rodríguez Fabián Geovanny, 2024).

Una vez que el modelo ha sido entrenado es importante realizar un proceso de validación para evaluar su rendimiento en un entorno real, esto se lleva a cabo utilizando un conjunto de datos separado que no ha sido visto por el modelo durante el entrenamiento, esta práctica es importante para medir la capacidad del modelo para funcionar correctamente con datos nuevos y desconocidos. La validación implica la simulación de escenarios reales de intrusión para comprobar si el modelo puede identificar correctamente las amenazas sin generar un número elevado de falsos positivos o negativos (Velastegui Morales Jhoselyn Lizeth & Cuzme Rodríguez Fabián Geovanny, 2024).

Es importante implementar ciclos de retroalimentación continua donde el modelo sea reentrenado periódicamente con nuevos datos recopilados por la red de sensores, esto permite que el sistema se actualice y que su capacidad de detección mejore con el tiempo, adaptándose a nuevos tipos de ataques o cambios en el entorno de la red (Al Farizi et al., 2021; H. Xu et al., 2023).

2.3.5.3. Desafíos y futuro del ML en la seguridad doméstica.

2.3.5.3.1. Desafíos en la implementación. La implementación de algoritmos de ML en sistemas de seguridad doméstica ofrece numerosas ventajas, pero también enfrenta desafíos significativos, como la necesidad de contar con grandes volúmenes de datos de alta calidad para entrenar modelos efectivos, estos datos deben ser representativos de una amplia gama de escenarios posibles en un entorno doméstico para garantizar que el modelo pueda reconocer tanto comportamientos normales como intrusiones (Velastegui Morales Jhoselyn Lizeth & Cuzme Rodríguez Fabián Geovanny, 2024).

Además, la complejidad computacional asociada con el uso de algoritmos avanzados como las Redes Neuronales Convolucionales (CNN) es otro desafío importante, ya que la implementación de estos algoritmos requiere un procesamiento intensivo, lo que puede ser un impedimento en sistemas que operan en hardware limitado, como los dispositivos IoT que se utilizan comúnmente en hogares inteligentes. Esto también plantea problemas en términos de eficiencia energética ya que los dispositivos deben poder funcionar continuamente sin agotar rápidamente sus recursos (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023).

Por último, la integración de estos algoritmos con sistemas IoT plantea desafíos en tema de seguridad de datos y protección de la privacidad, ya que la interconexión de múltiples dispositivos a través de redes inalámbricas puede aumentar la vulnerabilidad del sistema a ciberataques, lo que muestra la importancia de implementar medidas de seguridad para proteger la información sensible de los usuarios (González G. Alvaro F., 2024).

2.3.5.3.2. Futuro del ML en seguridad. El futuro del ML en la seguridad doméstica promete avances significativos, particularmente en la detección proactiva de intrusos y en la personalización de sistemas de seguridad basados en los hábitos específicos de los usuarios. Los avances en el aprendizaje profundo y el procesamiento de grandes volúmenes de datos permitirán a los sistemas de seguridad adaptarse mejor a las amenazas emergentes, ofreciendo una protección más robusta y personalizada para los hogares (Taiwo et al., 2022).

Uno de los desarrollos más prometedores es la capacidad de los sistemas para aprender y evolucionar continuamente a medida que recopilan más datos, esto significa que los sistemas de seguridad futuros serán capaces de reaccionar a incidentes y podrán anticiparse a ellos mediante la identificación de patrones de comportamiento que preceden a

una intrusión. La incorporación de técnicas de aprendizaje no supervisado podría permitir que los sistemas de seguridad detecten incluso aquellos tipos de intrusiones que no han sido previamente etiquetados o identificados, mejorando así la eficacia del sistema en escenarios complejos y desconocidos (Velastegui Morales Jhoselyn Lizeth & Cuzme Rodríguez Fabián Geovanny, 2024).

Además, la personalización será una característica importante en los sistemas de seguridad doméstica del futuro, (Saa Ayala Juan Fernando & Soto Valle Cerlos, 2023) sugieren que los sistemas podrán ajustarse automáticamente a los hábitos y preferencias de los usuarios, optimizando las configuraciones de seguridad para cada hogar en particular, lo que aumentará la efectividad de los sistemas y mejorará la experiencia del usuario al reducir las falsas alarmas y ofrecer soluciones de seguridad más intuitivas y adaptadas a las necesidades individuales.

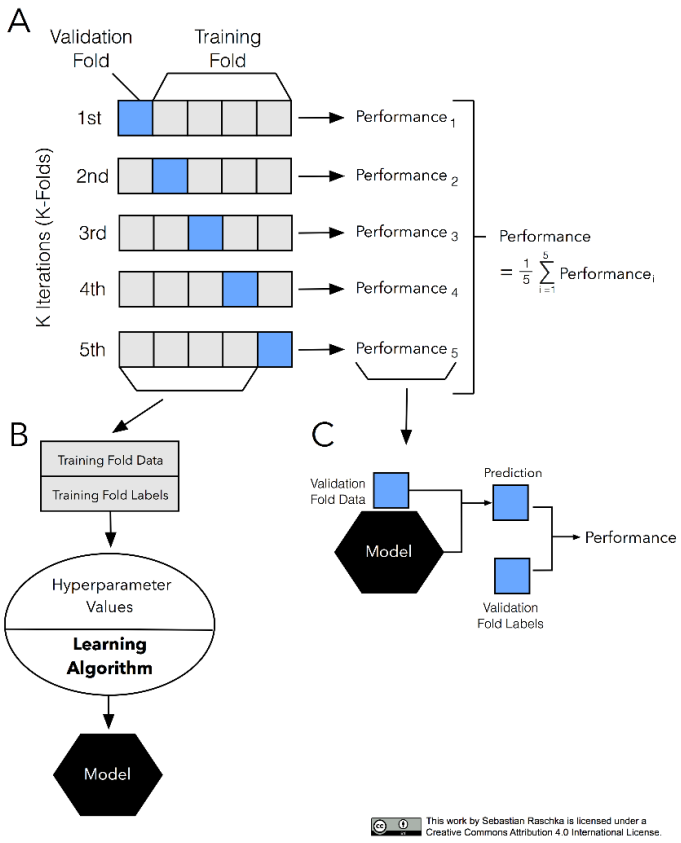
2.3.6. Selección, validación y evaluación de modelos predictivos

2.3.6.1. GridSearchCV. Es una herramienta de la librería scikit-learn diseñada para la optimización de hiperparámetros en modelos de ML, el cual consiste en una búsqueda en una "rejilla" definida por el usuario, probando todas las combinaciones posibles de hiperparámetros especificados para un modelo dado, esta búsqueda se realiza junto con la validación cruzada que divide el conjunto de datos en varios subconjuntos ("folds") para evaluar de manera robusta el desempeño de cada combinación de parámetros, evitando el sobreajuste y obteniendo una estimación precisa de la capacidad generalizadora del modelo.

GridSearchCV como se muestra en la Figura 7 resultará en la selección automática del conjunto de hiperparámetros que maximiza el desempeño del modelo según una métrica especificada, esto es útil para modelos sensibles a sus hiperparámetros como las máquinas de vectores de soporte o modelos de redes neuronales (Pedregosa, 2011).

Figura 7.

Diagrama de funcionamiento de GridSearchCV.

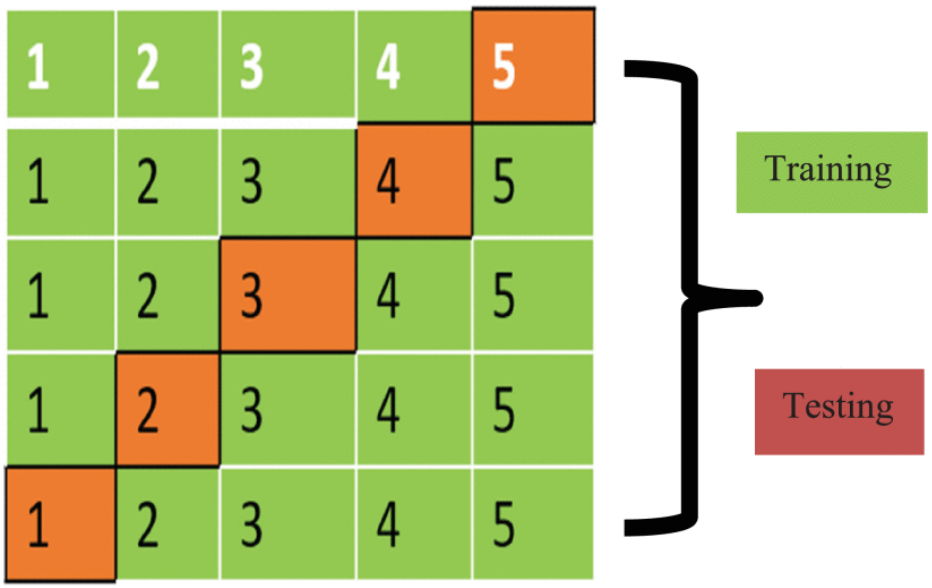


Nota. (How Is Cross Validation Performed and How GridSearchCV() Specifically?, n.d.)

2.3.6.2. Validación cruzada. La validación cruzada como se muestra en la Figura 8 es una técnica que se utiliza para evaluar el rendimiento y la generalización de modelos de ML que consiste en dividir el conjunto de datos en varios subconjuntos o "folds", este modelo se entrena con todos los folds menos uno que se usa para probarlo, este proceso se repite hasta que cada fold haya sido utilizado una vez como conjunto de prueba, este método proporciona una evaluación más estable y fiable del desempeño real del modelo y minimiza el sesgo que podría introducir la selección arbitraria de un solo conjunto de prueba, también es una medida para prevenir el sobreajuste, garantizando que el modelo mantenga buen desempeño en datos no vistos, la validación cruzada es la base con la que técnicas como GridSearchCV evalúan distintas configuraciones de hiperparámetros para seleccionar la mejor opción (Pedregosa, 2011).

Figura 8.

Representación gráfica de la validación cruzada.



Nota. Obtenido de (Iturbe-Araya & Rifà-Pous, 2025)

2.3.6.3. Matriz de confusión y métricas de desempeño. Es una herramienta para la evaluación del rendimiento en algoritmos de clasificación, muestra el desempeño de un modelo al contrastar las predicciones que realiza con los valores reales permitiendo cuantificar los aciertos y errores de manera tabular (Castillo Castro, 2024), a continuación, se muestran sus dos variantes:

- **Exactitud (Accuracy):** Mide el porcentaje total de predicciones que el modelo realizó correctamente y se calcula como la proporción de aciertos sobre el total de casos (Sokolova, 2009).
- **Precisión (Precision):** Evalúa de todas las instancias cuántas eran realmente positivas, por tal motivo esta métrica es esencial cuando el coste de un falso positivo es alto (Fawcett, 2006).
- **Recuperación (Recall):** Es una métrica que mide la capacidad del modelo para identificar correctamente todas las instancias que son realmente positivas, esto es necesario cuando no se requiere omitir ningún caso positivo (Fawcett, 2006).
- **Puntuación F1 (F1-Score):** Combina la Precisión y la Recuperación de manera equilibrada para resumir en un solo número el desempeño del modelo, esto es útil

cuando las clases están desbalanceadas ya que considera tanto la exactitud como la capacidad de capturar todos los casos positivos (Sokolova, 2009).

2.3.6.4. Ensamblador de algoritmos. Esta técnica combina las predicciones de varios modelos de ML para generar una predicción más robusta. A continuación, se muestra sus dos formas principales:

- **Votación dura (Hard Voting):** Es un método donde la predicción final se basa en la mayoría de los votos de los modelos individuales. La clase que es predicha con mayor frecuencia por la mayoría de los modelos es la elegida como predicción final (Ganaie, 2022).
- **Votación suave (Soft Voting):** Se basa en el promedio de las probabilidades predichas por cada modelo para cada clase, la clase con la probabilidad promedio más alta es seleccionada como el resultado final (Xu et al., 2022).

2.3.6.5. Técnicas para el tratamiento de datos desbalanceados (Sobremuestreo). En situaciones donde existe un desequilibrio significativo entre las clases, las técnicas de sobremuestreo (oversampling) son empleadas para balancear el conjunto de datos con el objetivo principal de generar instancias sintéticas de la clase minoritaria para mejorar el rendimiento del modelo y evitar que este se incline a favor de la clase mayoritaria. Las técnicas abordadas se muestran a continuación:

- **SMOTE (Synthetic Minority Over-sampling Technique):** Es el método más común, el cual crea nuevas instancias sintéticas de la clase minoritaria interpolando entre varios ejemplos vecinos existentes de esa misma clase (Valdovinos Rosas, 2006).
- **ADASYN (Adaptive Synthetic Sampling):** Es una versión adaptativa de SMOTE que genera más datos sintéticos para aquellos ejemplos de la clase minoritaria que son más difíciles de aprender, es decir los que se encuentran cerca de la frontera de decisión (He, 2008).
- **SMOTE-ENN:** Esta técnica híbrida combina el sobremuestreo de SMOTE con el submuestreo de ENN (Edited Nearest Neighbors), primero genera datos sintéticos y luego elimina las instancias de ambas clases que son consideradas ruido o se encuentran en la frontera entre clases (Sasada et al., 2020).
- **SMOTE-Tomek:** Es otro enfoque híbrido que primero aplica SMOTE para generar nuevas muestras minoritarias y luego utiliza los "Tomek Links" para eliminar pares

de instancias de clases opuestas que son vecinas más cercanas, limpiando así el espacio entre las clases (Cai et al., 2023; Chatterjee & Dethlefs, 2021).

2.3.7. Notificaciones inteligentes en tiempo real

2.3.7.1. Importancia de las notificaciones en sistemas de seguridad. En los sistemas de seguridad doméstica, la capacidad de recibir notificaciones en tiempo real es fundamental para garantizar una respuesta rápida ante posibles intrusiones o incidentes., etas notificaciones inteligentes permiten que los usuarios reciban alertas inmediatas sobre actividades sospechosas independientemente de su ubicación, lo que mejora significativamente la eficacia del sistema de seguridad.

Las notificaciones en tiempo real informan a los propietarios sobre eventos que ocurren en su hogar y también les permiten tomar decisiones rápidas como contactar a la policía o activar una alarma adicional, esto es importante en situaciones donde cada segunda cuenta para prevenir o minimizar el daño causado por una intrusión (Gupta et al., 2018).

2.3.7.2. Protocolo MQTT en la comunicación IoT. El protocolo MQTT (Message Queuing Telemetry Transport) es uno de los más utilizados en la entrega de notificaciones en sistemas IoT debido a su ligereza y eficiencia en la transmisión de datos, es un protocolo de publicación/suscripción que es ideal para entornos donde la latencia y el consumo de ancho de banda deben minimizarse. En un sistema de seguridad doméstica permite que los dispositivos IoT, como sensores y cámaras envíen alertas instantáneas a un servidor central o directamente al usuario (Hillar, 2017).

2.3.7.3. Desafíos en la implementación de notificaciones inteligentes. A pesar de los beneficios, la implementación de notificaciones inteligentes en sistemas de seguridad doméstica enfrenta varios desafíos como la latencia en la entrega de mensajes, que puede ser causada por la congestión de la red o por la configuración incorrecta del sistema. La elección de un protocolo de comunicación eficiente como MQTT, y la optimización de la red de IoT son muy importantes para reducir la latencia y asegurar que las notificaciones lleguen de manera oportuna (Yalçnkaya et al., 2020).

Otro desafío es la fiabilidad del sistema de notificaciones, especialmente en situaciones donde la conectividad a internet puede ser intermitente, por tal motivo, la implementación de canales de comunicación redundantes como el uso de redes 4G/LTE,

puede ayudar a mitigar este problema, asegurando que las notificaciones se entreguen incluso si la red Wi-Fi principal falla.

2.3.7.4. Mejores prácticas para la configuración de notificaciones. Para aumentar la efectividad de las notificaciones en un sistema de seguridad doméstica es importante seguir ciertas mejores prácticas, asegurando que las alertas sean precisas, oportunas y útiles para los usuarios. Uno de los aspectos importantes es la configuración de reglas de notificación claras y específicas, lo que evita el envío de alertas innecesarias que podrían desensibilizar al usuario ante eventos críticos, un fenómeno conocido como "fatiga de alertas", que puede llevar a que los usuarios ignoren notificaciones importantes, disminuyendo la eficacia del sistema de seguridad.

En el estudio realizado por (Yalçnkaya et al., 2020) se destaca la importancia de un diseño adecuado en la configuración del sistema de notificaciones para asegurar la transmisión eficiente de datos y la entrega confiable de alertas en tiempo real. Se observa que la precisión en la detección de eventos y la rapidez en la entrega de notificaciones son esenciales para el funcionamiento óptimo del sistema, así como el uso del protocolo MQTT en el sistema de hogar inteligente muestra ser efectivo en garantizar que las notificaciones lleguen a los usuarios de manera oportuna, incluso en redes con limitaciones de ancho de banda.

Las pruebas regulares del sistema de notificaciones son esenciales para garantizar su correcto funcionamiento, estas deben incluir la verificación de la precisión en la detección de eventos, la rapidez en la entrega de las notificaciones y la fiabilidad del sistema bajo diferentes condiciones de red. Ajustes periódicos basados en los resultados de estas pruebas son necesarios para mejorar tanto la precisión como la velocidad de las alertas, asegurando que el sistema sea capaz de responder adecuadamente a situaciones de emergencia.

2.3.8. Redundancia en canales de comunicación

2.3.8.1. Importancia de la redundancia en sistemas de seguridad. La redundancia en los sistemas de comunicación es esencial para garantizar la operatividad continua de un sistema de seguridad incluso en casos de fallos en la red principal, que en sistemas de IoT aplicados a la seguridad doméstica mantener la comunicación y el control del sistema es crítico, especialmente en situaciones de emergencia. La implementación de protocolos de comunicación como MQTT en sistemas IoT asegura la continuidad del servicio

mediante mecanismos de redundancia que minimizan el riesgo de interrupciones (Yalçnkaya et al., 2020).

Implementación práctica y beneficios. La implementación práctica de redundancia en los sistemas de seguridad puede realizarse mediante la configuración de redes secundarias utilizando tecnologías como 4G/LTE, en combinación con la red Wi-Fi principal, esto asegura que en una caída de la red Wi-Fi el sistema de seguridad continúe operando, utilizando la red móvil. La configuración de un módulo 4G/LTE en una Raspberry Pi, proporciona un canal de comunicación secundario que se activa automáticamente cuando la red principal falla, asegurando así la integridad del sistema de seguridad.

2.3.9. Herramientas de desarrollo de software y tecnologías

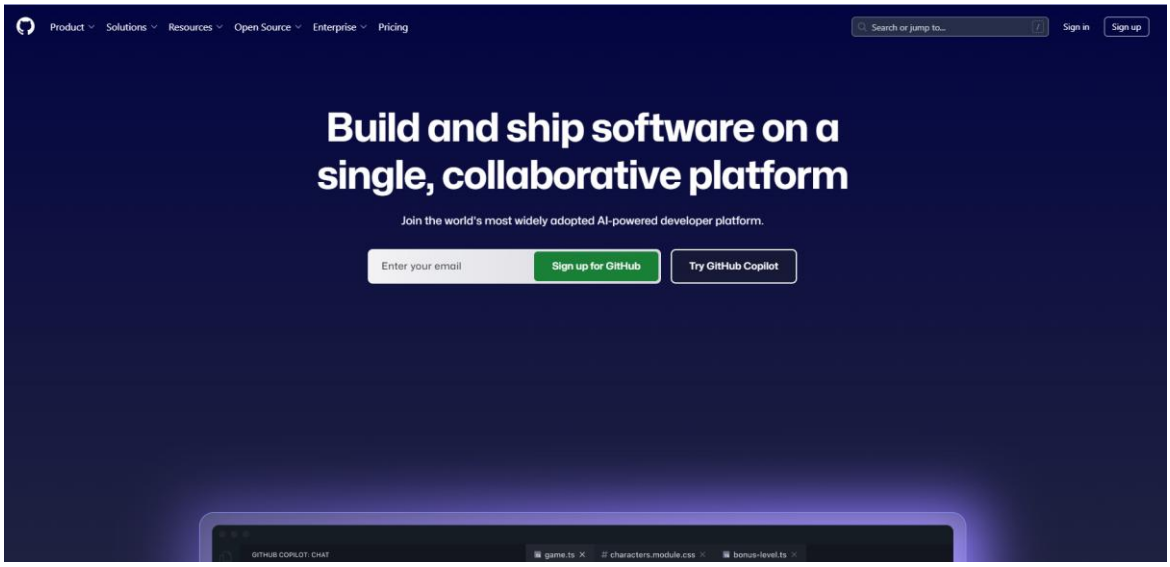
2.3.9.1. Git y Github. Git es un sistema de control de versiones que gestiona los cambios en archivos de código fuente a lo largo del tiempo, guardando el historial de cambios de un proyecto, es ideal para proyectos grandes y colaborativos, permite crear versiones paralelas del proyecto para implementar nuevas funcionalidades sin afectar el código principal evitando conflictos y posibles errores.

Github es considerada como una red social para desarrolladores basada en la web para alojar repositorios que usan Git, permite colaborar, revisar código, reportar errores y más, sus características principales incluyen la integración continua / despliegue automático (CI/CD), control de acceso y permisos para equipos, alojamiento para repositorios Git en la nube, etc.

En la Figura 9 se muestra como luce la interfaz de GitHub.

Figura 9.

Página principal de GitHub.



Nota. GitHub (2024). https://github.com/

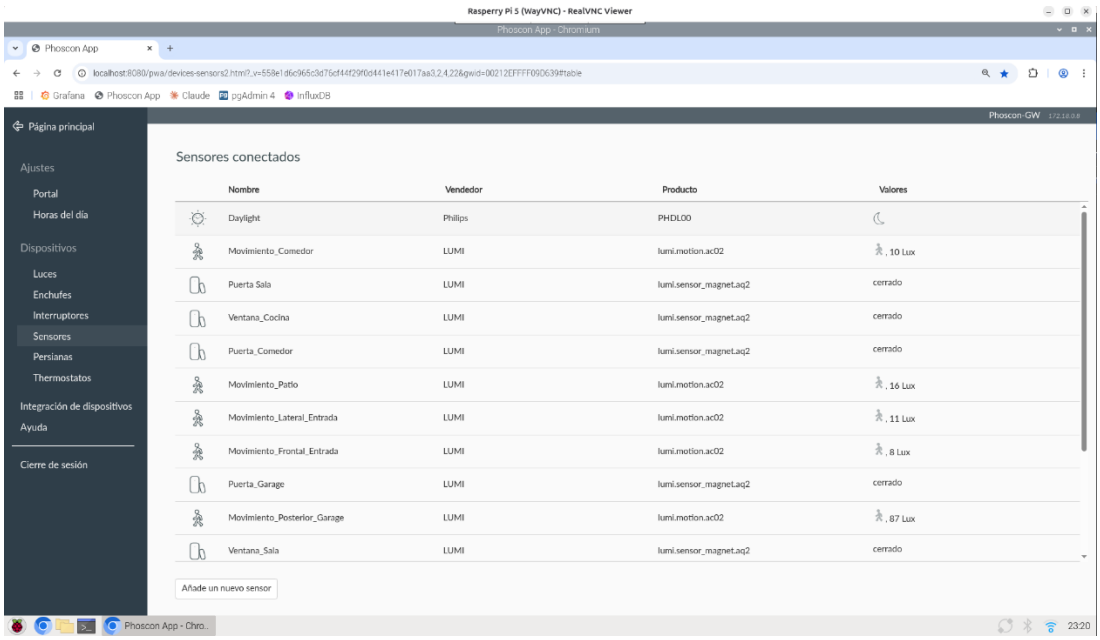
2.3.9.2. deCONZ y Phoscon. El software deCONZ permite gestionar la comunicación y controlar los dispositivos ZigBee, permitir el control y la supervisión y exponer estos dispositivos para su consumo como una API REST.

Phoscon funciona como una plataforma web que permite la configuración y el control de los dispositivos ZigBee de manera amigable, el sistema permite administrar la amplia la gama de dispositivos según sus funcionalidades dentro del sistema, incluyendo la vinculación de nuevos dispositivos como sensores de movimiento, de vibración e impacto y de establecer configuraciones de comunicación del software deCONZ.

En la Figura 10 se presenta la apariencia de la plataforma:

Figura 10.

Interfaz de Phoscon.



2.3.9.3. Python. Python es un lenguaje de programación con sintaxis sencilla y clara, especializado en múltiples campos como ML, desarrollo web, ciencia de datos y entre otras categorías, ofrece un amplio número de módulos y bibliotecas gratuitas con actualizaciones constantes para gran variedad de ámbitos mantenidos por la comunidad de desarrolladores o por empresas que emplean estos sistemas.

Este lenguaje se considera practico para el procesamiento de datos, automatización y creación de Apis por contar con bibliotecas especializadas, por lo cual su conexión con dispositivos IoT como ZigBee se puede comprender de manera dinámica y lograr la implementación de nuevas funcionalidades.

2.3.9.4. FastAPI. FastAPI es un framework que se especializa dentro de Python como una forma para construir APIs ágilmente, esencialmente una API (Interfaz de programación de aplicaciones) es un intermediario que permite la comunicación entre de dos entidades, un solicitante (cliente) y otra que responde (servidor), de esta forma se pueden crear formas de consumo según los requerimientos de uso, como rutas para conocer información o generar consultas de procesamiento que se responden al cliente en cuanto obtiene una respuesta dentro del servidor.

Dentro de sus ventajas es considerado por la comunidad como un marco robusto y escalable y su documentación recalca permitir la integración de seguridad como OAuth2 y

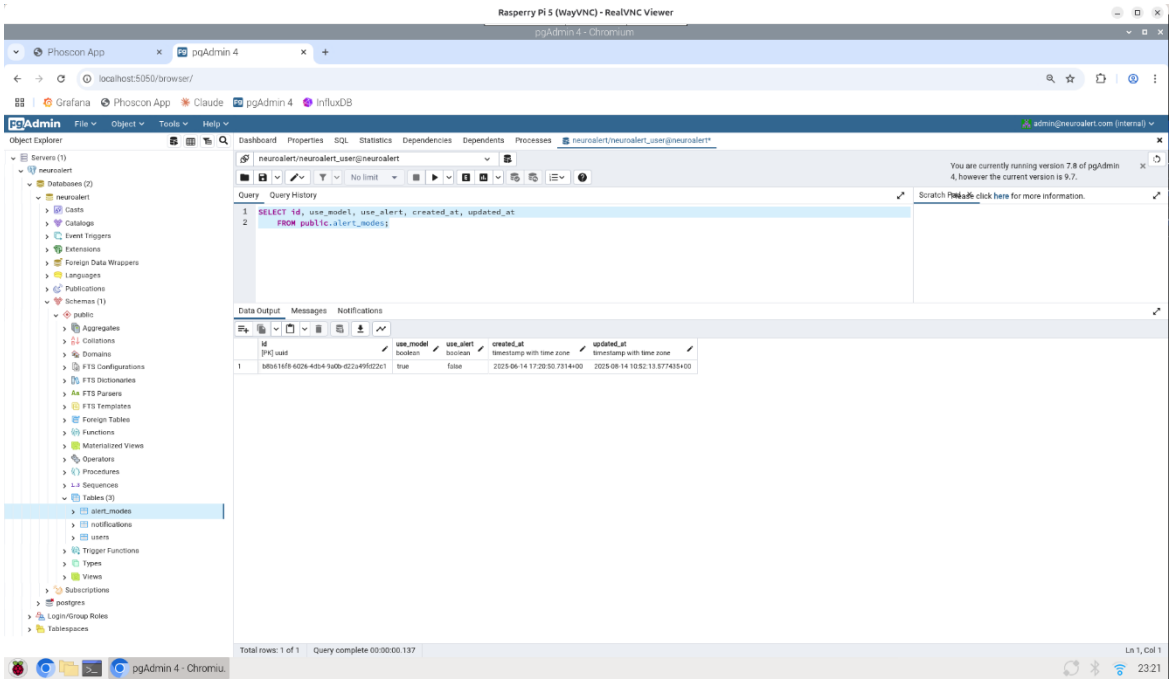
JWT, procesamiento asincrónico, mantener una velocidad de ejecución competente en comparación con otros marcos como NodeJS y Go y entre más características.

2.3.9.5. Postgres y PgAdmin. Postgres es una base de datos (BD) relacional basada en código SQL (Lenguaje de consulta estructurado) el cual es un lenguaje de programación orientado a BDs, flexible con gran poder de configuración y operaciones para el almacenamiento de información, como ejemplificación sus principales funciones permiten insertar, consultar, modificar y eliminar datos, así mismo permite administrar la estructura completa de una BD.

Ahora bien, postgres a pesar de mantener grandes funcionalidades dentro del campo de BDs no emplea un entorno gráfico, por lo que surge PgAdmin como un entorno con capacidad de conectarse a la BD y obtener la información de los registros fácilmente, entre algunas de sus funcionalidades implementa herramientas de desarrollo como un editor visual para crear, modificar funciones, triggers y procedimientos hasta un dashboard con métricas del servidor en tiempo real, la Figura 11 muestra la interfaz de PgAdmin.

Figura 11.

Interfaz de PgAdmin.

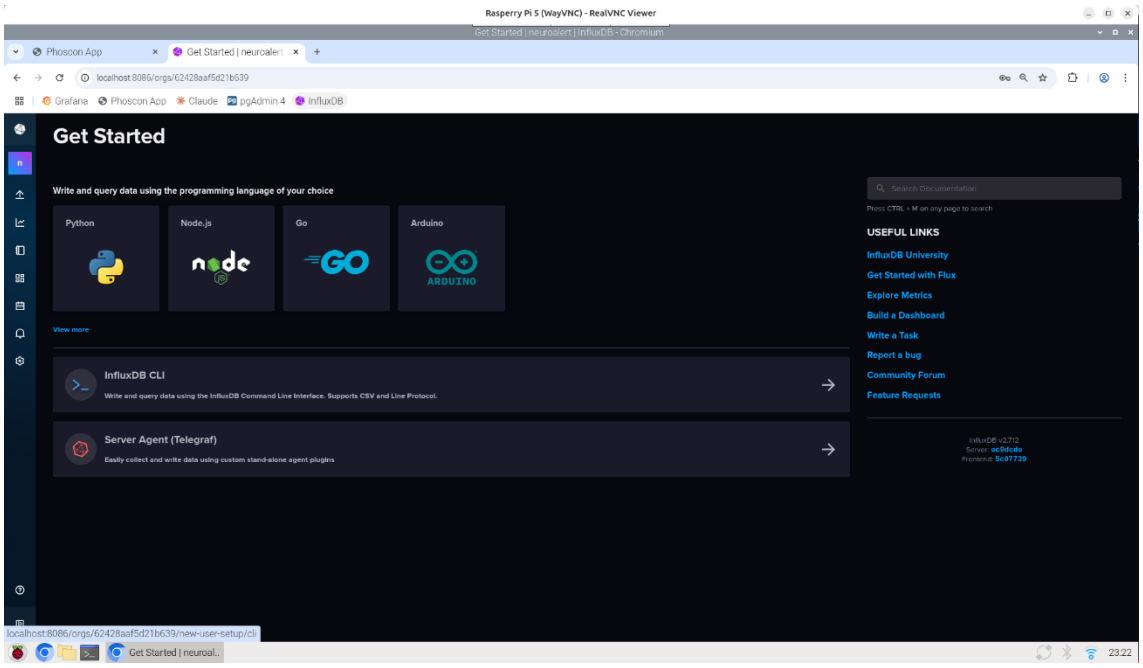


2.3.9.6. InfluxDB y Grafana. InfluxDB es una base de datos dedicada al almacenamiento de datos de series temporales especialmente útil para dispositivos IoT, misma que se encuentra optimizada para escribir y consultar grandes volúmenes de

información, su funcionamiento incluye el InfluxQL (Similar a SQL) para consultas básicas y Flux un lenguaje más avanzado para análisis complejos, transformaciones de datos y operaciones matemáticas avanzadas. En la **Figura 12** se muestra la apariencia de InfluxDB.

Figura 12.

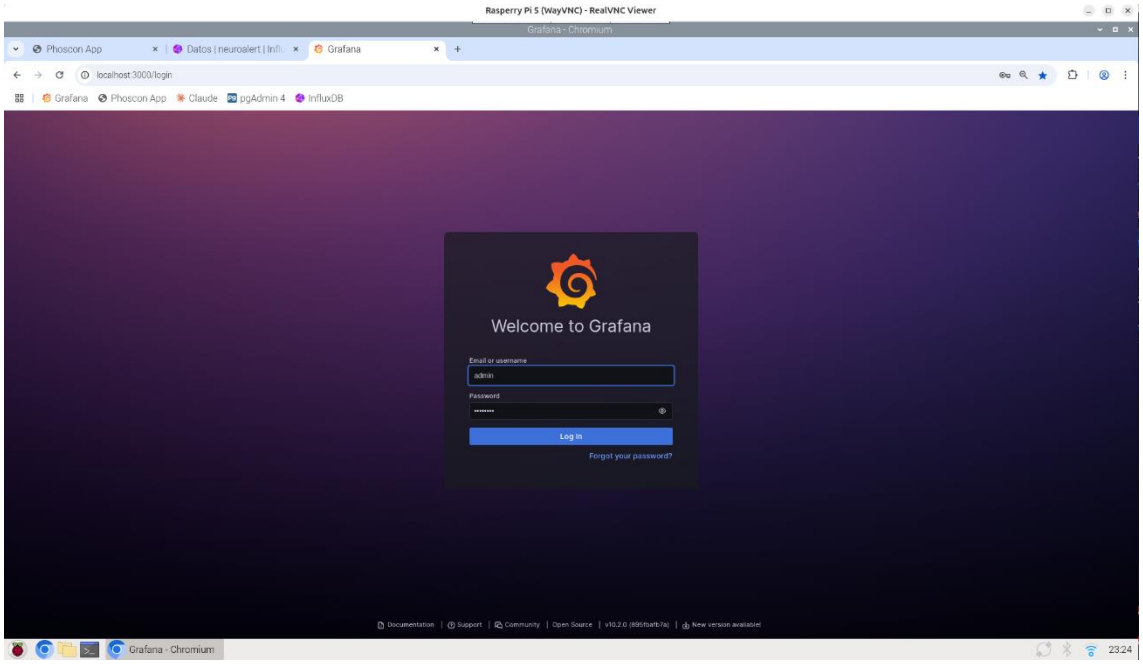
Interfaz de InfluxDB.



Ahora si bien, InfluxDB actúa como el cerebro de almacenamiento, Grafana actúa como los ojos del sistema transformando la información de la base de datos en visualizaciones comprensibles y accionables, con funcionalidades de crear dashboards en tiempo real con alertas automáticas y permitir visualizar patrones y tendencias, su apariencia se puede observar Figura 13.

Figura 13.

Interfaz de Grafana.



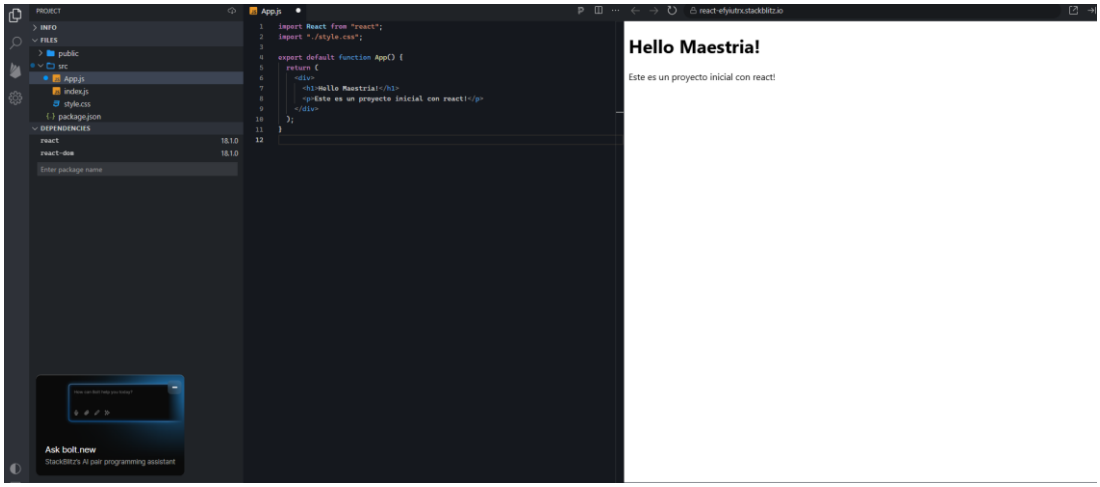
2.3.9.7. JavaScript y React. JavaScript es un lenguaje de programación de alto nivel orientado al desarrollo para ecosistemas web modernos y es considerado versátil puesto que su uso se puede encontrar tanto del lado del Backend como NodeJS y Frontend como React, además tras la incorporación de tipado surge TypeScript siendo adoptando en tecnologías como Nest, Angular, Next, Nuxt, Remix y más opciones.

React por su parte es una biblioteca de JavaScript considerada como un Framework a la par que Angular, enfocada en construir interfaces de usuario con el uso de componentes reutilizables que pueden manejar su propio estado, mantiene un ecosistema amplio con herramientas complementarias que permiten crear interfaces y funcionalidades de forma simple como, por ejemplo: React Router, Redux, Material-UI, Next y React Native, esta última siendo enfocada para dispositivos móviles multiplataforma.

En la Figura 14 se representa parte del código inicial de un proyecto con React, el cual emplea JavaScript como lenguaje principal.

Figura 14.

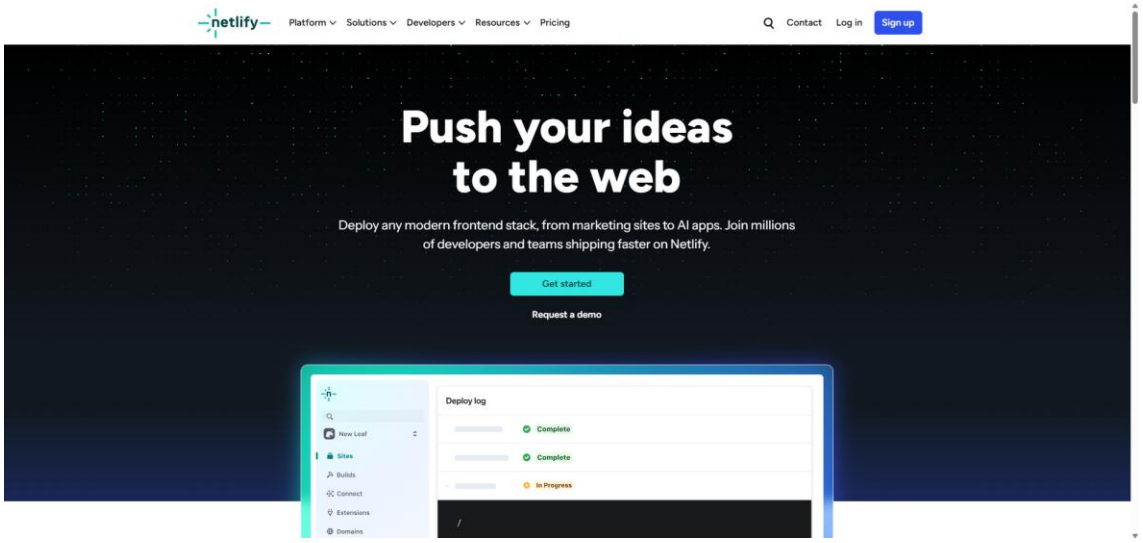
Proyecto inicial con React.



2.3.9.8. Netlify. Es una plataforma de alojamiento de sitios web conocida debido a su simplicidad y velocidad para desplegar proyectos con capacidad para realizar despliegues continuos con conexión a GitHub, GitLab o BitBucket o emplear carpetas con el proyecto de despliegue, por otro lado, emplea características de seguridad proporcionando certificados SSL, cifrado con HTTPS y control de acceso con autenticación y autorizaciones. En la **Figura 15** se presenta la interfaz de la plataforma Netlify.

Figura 15.

Página principal de Netlify.



Nota. Netlify (2024). https://www.netlify.com/

2.4. Marco legal

El marco legal de esta tesis se fundamenta en las leyes y normativas ecuatorianas que regulan el uso de tecnologías de la información, la protección de datos personales y la seguridad ciudadana. A continuación, se detallan las disposiciones legales relevantes que guían la implementación de este tipo de tecnologías.

2.4.1. Constitución de la república del Ecuador

La Constitución de la República del Ecuador, aprobada en 2008 establece derechos fundamentales que son esenciales para el desarrollo de sistemas de seguridad doméstica, incluyendo aquellos basados en tecnologías de IoT y ML.

- **Art. 66.19 Protección de datos personales:** Reconoce el derecho de toda persona a acceder, decidir y controlar el uso de sus datos personales, impidiendo su tratamiento sin autorización.
- **Art. 66.20 Inviolabilidad del domicilio:** Prohíbe el ingreso o registro del interior de un domicilio sin consentimiento del titular o sin orden legal.
- **Art. 66.21 Inviolabilidad de la correspondencia:** Garantiza que cartas, correos electrónicos u otras comunicaciones escritas privadas no puedan ser abiertas ni revisadas sin autorización.
- **Art. 66.22 Inviolabilidad de las comunicaciones:** Protege las llamadas, mensajes y cualquier comunicación en tiempo real contra vigilancia o interceptación no autorizada (Constitución de la República del Ecuador, 2008).

2.4.2. Ley de comercio electrónico y firmas digitales

Esta ley creada en 2002 es fundamental para regular el uso de tecnologías de la información y comunicación en Ecuador.

Mensajes de Datos: Regula los mensajes de datos garantizando que las notificaciones en tiempo real enviadas por el sistema de seguridad doméstica sean reconocidas legalmente, siempre que se utilicen métodos de transmisión seguros y confiables (Congreso Nacional, 2021).

2.4.3. Ley orgánica de protección de datos personales

Modificada en 2021, establece el marco regulatorio para la protección de datos personales en Ecuador, siendo relevante para cualquier sistema que gestione información sensible.

- **Art. 8 Consentimiento informado:** Exige autorización libre, específica, informada e inequívoca antes de recolectar o tratar datos personales, incluidas imágenes, audio o información biométrica.
- **Art. 10(j) Principio de seguridad:** Obliga a implementar medidas técnicas y organizativas para prevenir el acceso no autorizado, pérdida o destrucción de datos.
- **Art. 39 Protección de datos desde el diseño:** Los sistemas deben integrar la seguridad y privacidad desde su desarrollo inicial.
- **Art. 40 Evaluación de riesgos:** Requiere identificar y analizar posibles amenazas que comprometan la privacidad y seguridad de los datos.
- **Art. 43 Notificación de vulneraciones:** Obliga a informar a la autoridad y a los titulares en caso de filtraciones o incidentes que afecten datos personales (Asamblea Nacional, 2021).

2.4.4. Código orgánico integral penal (COIP)

El COIP regula diversos aspectos relacionados con la seguridad, tanto física como digital y es necesario para prevenir y sancionar delitos que puedan afectar la integridad de los sistemas de seguridad.

- **Art. 230 Interceptación ilegal de datos:** Penaliza captar o interceptar datos, mensajes o comunicaciones privadas sin autorización.
- **Art. 232 Ataque a la integridad de sistemas informáticos:** Sanciona acciones que alteren, dañen o eliminen datos o programas.
- **Art. 234 Acceso no consentido:** Castiga ingresar sin permiso a sistemas informáticos o redes.
- **Art. 181 Violación de propiedad privada:** Prohíbe ingresar sin autorización a un lugar privado.
- **Art. 189 Robo:** Penaliza apropiarse de bienes ajenos mediante violencia o amenaza. (COIP, 2021).

3. CAPÍTULO III

MARCO METODOLOGICO

3.1. Descripción del área de estudio

El área de estudio fue una vivienda unifamiliar ubicada en la ciudad de Loja, Ecuador, seleccionada como entorno de prueba para el desarrollo e implementación de un sistema inteligente de seguridad doméstica. Dicha vivienda sirvió como espacio controlado para instalar y configurar los dispositivos IoT, y donde se realizaron las pruebas para evaluar el rendimiento del sistema de seguridad basado en tecnologías de IoT y algoritmos de ML. La ubicación de la vivienda, así como las características del entorno, resultaron representativas para estudiar los desafíos de seguridad en el ámbito doméstico, especialmente en una ciudad con problemas de seguridad como Loja. La investigación se centró en la detección de intrusiones a través de dispositivos conectados, garantizando la redundancia de comunicación en caso de fallos en la red principal.

3.2. Enfoque y tipo de investigación

El enfoque de la investigación fue de tipo mixto, combinando tanto métodos cuantitativos como cualitativos, como se describe a continuación:

Para la revisión de literatura se utilizó un enfoque cualitativo, ya que se revisaron y analizaron estudios, artículos, libros y tesis relacionados con seguridad doméstica, IoT y ML. El análisis se enfocó en la comprensión de teorías, conceptos y marcos existentes, los cuales fundamentaron las bases teóricas de la investigación. Durante esta etapa, se identificaron las principales tendencias en la implementación de sistemas de seguridad inteligentes, así como las mejores prácticas en el uso de IoT y algoritmos de ML para la detección de intrusos.

Para el desarrollo de Infraestructura IoT se empleó un enfoque mixto, ya que se combinó la implementación técnica de los dispositivos IoT con un análisis cualitativo de su funcionalidad y efectividad. Se seleccionaron y adquirieron dispositivos basados en la tecnología ZigBee y se configuró la red IoT en puntos estratégicos dentro del hogar. Durante este proceso, se recopilaron datos cuantitativos relacionados con la instalación y monitoreo de los dispositivos, mientras que, paralelamente, se realizó un análisis cualitativo del

rendimiento y la operatividad de la infraestructura. Este análisis permitió ajustar y optimizar la configuración según las necesidades específicas del sistema de seguridad.

Para la implementación del Sistema de Seguridad se adoptó un enfoque predominantemente cuantitativo, ya que se centró en la implementación de los algoritmos de ML para la detección de intrusos. Se evaluaron y seleccionaron varios modelos de ML, y el modelo final fue entrenado con datos recolectados previamente. Se realizaron ajustes en los parámetros del modelo basados en métricas de rendimiento como la precisión y la tasa de detección de intrusos. Los resultados obtenidos permitieron medir la efectividad del sistema, y se generaron informes detallados sobre el desempeño del modelo bajo diferentes escenarios.

Finalmente, para la validación del Sistema se adoptó un enfoque mixto. Se llevaron a cabo simulaciones de intrusiones para recolectar datos cuantitativos sobre la tasa de falsos positivos y negativos. Además, se documentaron las experiencias obtenidas durante las pruebas para el ajuste final del sistema. Este análisis cualitativo complementó los datos numéricos, brindando una visión completa del comportamiento del sistema en situaciones reales. El enfoque mixto permitió evaluar de manera integral la eficacia del sistema desde múltiples perspectivas, lo que contribuyó a la mejora continua de su rendimiento.

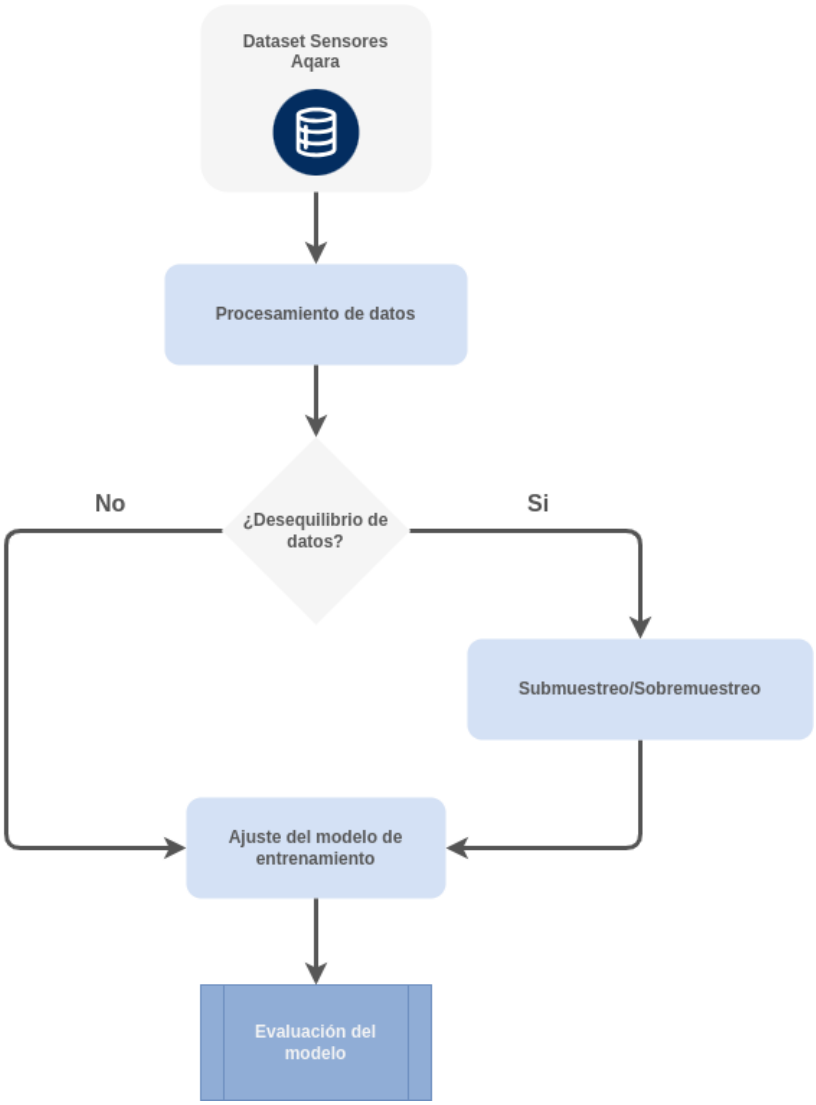
3.3. Técnicas e instrumentos

3.3.1. Metodología CRISP-DM para algoritmos ML

Se emplea una adaptación de la metodología CRISP-DM para guiar el proceso de desarrollo de un modelo de ML para la detección de anomalías tras la recolección de datos mediante los sensores aqara, su explicación se define en la Figura 16.

Figura 16.

Flujograma metodológico para modelos de ML.



El flujograma define los elementos más importantes considerados para la elaboración del algoritmo útil, en primer lugar, se realiza la importación del dataset, subsecuentemente se realiza el procesamiento con enfoque a la distribución de los datos y la distinción entre si aplicar submuestreo o sobremuestreo de acuerdo con el desequilibrio de datos, posteriormente se realiza el ajuste del modelo de acuerdo con sus características y finalmente a la evaluación de este.

3.3.2. Datos registrados de los sensores

Los datos registrados y empleados provienen de la recolección de los sensores instalados en la vivienda donde se registraron 28 variables por aproximadamente dos meses

y medio entre marzo y mayo del año 2025, la Tabla 4 explica cada una de las variables almacenadas.

Tabla 4.

Variables recolectadas de sensores.

N°	Nombre de la variable	Descripción	Unidad
1	fecha_hora	Fecha y hora del registro del evento	DateTime (YYYY-MM-DD HH:MM:SS)
2	id_sensor	Identificador único del sensor	Numérico entero
3	nombre	Nombre descriptivo del sensor	Texto
4	tipo	Tipo de sensor (ej: ZHAPresence)	Texto
5	modelo	Modelo del sensor	Texto
6	fabricante	Fabricante del sensor	Texto
7	es_evento_cambio	Indica si el registro representa un cambio de estado	Booleano (True/False)
8	cambio_estado_real	Indica si hubo un cambio real en el estado del sensor	Booleano (True/False)
9	estado_lastupdated	Timestamp de la última actualización del estado	DateTime ISO 8601
10	estado_presence	Estado de presencia detectada por el sensor	Booleano (True/False)
11	caract_hora	Hora extraída del timestamp (0-23)	Numérico entero
12	caract_minuto	Minuto extraído del timestamp (0-59)	Numérico entero
13	caract_dia_semana	Día de la semana (0-6, donde 0=lunes)	Numérico entero
14	caract_periodo_dia	Período del día (ej: noche, mañana, tarde)	Texto
15	caract_es_fin_semana	Indica si es fin de semana	Numérico binario (0/1)
16	caract_tiempo_desde_ultimo_evento	Tiempo transcurrido desde el último evento	Segundos (decimal)
17	caract_conteo_actividad_reciente	Contador de actividad reciente	Numérico entero
18	caract_cambio_valor	Indica si hubo cambio en el valor	Numérico binario (0/1)
19	caract_cambio_valor_porcentaje	Porcentaje de cambio en el valor	Porcentaje (0.0-100.0)
20	caract_es_hora_inusual	Indica si la hora es considerada inusual	Decimal (0.0-1.0)
21	estado_open	Estado de apertura (para sensores de puerta/ventana)	Booleano o vacío
22	estado_dark	Estado de oscuridad	Booleano o vacío
23	estado_daylight	Estado de luz diurna	Booleano o vacío
24	estado_status	Estado general del sensor	Texto o vacío
25	estado_sunrise	Hora del amanecer	DateTime o vacío
26	estado_sunset	Hora del atardecer	DateTime o vacío
27	estado_lightlevel	Nivel de luz detectado	Numérico o vacío
28	estado_lux	Medida de iluminancia en lux	Numérico (lux) o vacío

Nota. Número (N°).

3.3.3. Metodología de desarrollo Desing Science Research (DSR)

La metodología Desing Science Research (DSR) o en español Investigación en ciencias del diseño enfoca en crear soluciones para problemas reales combinando el rigor

científico con la relevancia práctica, es idealmente pensada como metodología principal cuando se actúa como investigador y desarrollador simultáneamente.

En la Figura 17 se puede observar el ciclo de la metodología DSR.

Figura 17.

Metodología empleada para el desarrollo de software.

Esta metodología opera mediante tres ciclos interconectados, el primer ciclo denominado relevancia conecta los problemas reales, el segundo construye y evalúa interactivamente y el tercer ciclo denominado rigor fundamenta teorías existentes, esta metodología permite investigar las soluciones implementadas estructurando el proceso en pasos: identificar el problema, definir objetivos, diseñar, desarrollar, demostrar, evaluar y comunicar.

3.3.3.1. Definición de elementos de la metodología DSR. La Tabla 5 brinda una explicación breve en base los componentes de la Figura 17.

Tabla 5.

Componentes de la metodología DSR.

Nombre	Explicación
Entorno	
Personas	Usuarios finales y stakeholders.
Sistemas organizacionales	El contexto donde se aplicará.
Sistemas técnicos	La infraestructura existente.
Problemas y oportunidades	Desafíos que motivan la investigación.
Investigación en ciencias del diseño	
Construir	Crear artefactos de diseño y procesos

Evaluar	Validar la efectividad y utilidad de los artefactos
Base de conocimiento	
Teoría científicas y métodos	Fundamentos teóricos
Experiencia	Conocimiento practico acumulado

En la siguiente Tabla 6 se definen los ciclos de DSR:

Tabla 6.

Ciclos 3 de la metodología DSR

Nombre	Explicación
Ciclo de relevancia	• Identifica los problemas reales del entorno • Define requisitos para la solución • Realiza pruebas de campo para validar la aplicabilidad • Asegura que la investigación sea útil
Ciclo de diseño	• Proceso iterativo de construcción y evaluación • Refinamiento continuo de los artefactos • Mejoras continuas
Ciclo de rigor	• Usa teorías existentes como base • Contribuye con nuevos conocimientos • Hay que asegurar que la investigación tenga solidez científica

3.4. Procedimiento de investigación

El procedimiento de investigación constó de varias fases estructuradas, que aseguraron una implementación ordenada del proyecto.:

3.4.1. Fase 1: Revisión de la literatura

Se realizó la revisión de la literatura en bases de datos como IEEE Xplore y Google Scholar, donde se analizaron estudios previos relacionados con la seguridad doméstica, IoT y ML, esta revisión proporcionó un marco teórico robusto que fundamentó la investigación.

3.4.2. Fase 2: Desarrollo de la infraestructura IoT

Se adquirieron y configuraron los dispositivos IoT basados en la tecnología ZigBee, incluyendo sensores de movimiento, de puertas/ventanas, estos dispositivos se instalaron en puntos estratégicos de la vivienda para garantizar una cobertura integral.

3.4.3. Fase 3: Implementación del sistema de seguridad

Se tomaron como opciones diferentes algoritmos de ML y se seleccionó el mejor resultado del algoritmo con mejor desempeño en la detección de patrones anómalos en la actividad doméstica. El modelo entrenado fue integrado en la infraestructura IoT para la detección en tiempo real de intrusos. Además, se configuraron notificaciones inteligentes multicanal para alertar a los usuarios.

3.4.4. Fase 4: Validación del sistema

Se definieron escenarios de prueba que incluyeron simulaciones de intrusiones, evaluando la eficacia del sistema en términos de detección de falsos positivos y negativos. Los resultados fueron analizados cuantitativamente, y se realizaron ajustes finales al sistema con base en los datos obtenidos. Finalmente, se documentaron los resultados obtenidos y se generó un informe detallado para su evaluación.

3.5. Consideraciones bioéticas

Durante el desarrollo del proyecto, se tomaron en cuenta consideraciones bioéticas importantes. Se garantizó la privacidad y protección de los datos personales durante la recolección de información a través de los sensores IoT instalados en la vivienda. Aunque no se trataba de un estudio que involucrara la interacción directa con seres humanos, se aplicaron medidas para proteger la integridad de los datos recogidos, asegurando que los mismos no fueran compartidos ni utilizados para fines no autorizados.

Además, se respetaron todos los principios éticos de la investigación científica, asegurando que los procedimientos implementados no causaran daño a terceros, respetando las normativas locales de protección de datos y cumpliendo con los estándares de seguridad y privacidad.

4. CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados obtenidos a lo largo de las diferentes fases del proyecto, analizando los resultados de la implementación del sistema de seguridad doméstica basado en IoT y ML para la detección de intrusos, las tecnologías seleccionadas y los algoritmos implementados fueron escogidos basándose en una revisión de literatura y en las necesidades específicas del entorno doméstico. A continuación, se detallan los hallazgos más importantes de cada fase del proyecto.

4.1. Evaluación inicial del sistema de seguridad doméstica

Durante la revisión de la literatura se identificaron varias tecnologías y enfoques relevantes para la implementación de un sistema de seguridad doméstica eficiente, particularmente en el contexto de la utilización de dispositivos IoT, redes ZigBee y algoritmos de ML para la detección de intrusos.

Se seleccionaron estudios y proyectos que destacaron las ventajas de combinar estas tecnologías para mejorar la seguridad y eficiencia en hogares inteligentes. Por ejemplo, la implementación de sistemas de monitoreo de intrusos con redes de sensores inalámbricos (WSN) utilizando inteligencia artificial mostró ser una solución altamente eficaz para detectar accesos no autorizados, como lo demuestra el trabajo de (Velastegui Morales Jhoselyn Lizeth & Cuzme Rodríguez Fabián Geovanny, 2024), quienes desarrollaron un sistema similar para la detección de intrusos en redes WSN.

Asimismo, se revisaron estudios que presentaban desafíos en la implementación de IoT, tales como la privacidad y seguridad de los datos, los cuales fueron abordados en la investigación de (Ahmed & Zeebaree, 2021), quienes realizaron una encuesta sobre los retos de seguridad y privacidad en entornos domésticos inteligentes.

La selección de dispositivos IoT y el protocolo de comunicación también fue fundamentada en estudios recientes que subrayan la importancia del uso de tecnologías como ZigBee y MQTT para garantizar una transmisión eficiente y de bajo consumo en sistemas de seguridad (Yalçnkaya et al., 2020). Estas tecnologías permiten integrar múltiples dispositivos en una red segura y escalable, proporcionando una base sólida para el desarrollo de un sistema robusto de detección de intrusos.

A continuación, se presentan las respuestas a las interrogantes de la investigación en base al análisis de la literatura y la metodología descrita en su tesis.

4.1.1. Revisión de sistemas IoT y ML: Mejores prácticas

Con la revisión de literatura se identifica un conjunto de tecnologías y protocolos más adecuados y eficientes para sistemas de seguridad doméstica inteligente, en la Tabla 7 se muestra las tecnologías destacadas en el marco teórico del proyecto que constituyen las prácticas recomendadas.

Tabla 7.

Tecnologías y prácticas recomendadas en la literatura.

Tecnología/Práctica	Aplicación principal	Total	Referencias importantes
Protocolo ZigBee	Comunicación de bajo consumo entre sensores y dispositivos que forman redes de malla robustas.	5	(Albornoz & Soto, 2018; Chen et al., 2022a; Dresden elektronik ingenieurtechnik GmbH, 2025; InfluxData, 2025a; Vera Romero et al., 2017)
Protocolo MQTT	Utilizado para la transmisión de datos y notificaciones en tiempo real.	5	(Calatayud Sánchez, 2021; elektronik ingenieurtechnik GmbH, 2025; Hillar, 2017; Rosas Cruz, 2021; Yalçnkaya et al., 2020)
Raspberry Pi	Unidad principal del sistema para gestión de dispositivos y ejecución de algoritmos.	1	(elektronik ingenieurtechnik GmbH, 2025)
Conectividad	Multicanal que asegura la operatividad continua del sistema con los usuarios.	4	(Calatayud Sánchez, 2021; Duque Quevedo Odalys Rashel, 2024; Khalid et al., 2019; Taiwo et al., 2022)
Bases de Datos	Almacenamiento de datos de sensores en el tiempo.	4	(Calatayud Sánchez, 2021; Group, 2025; InfluxData, 2025b; pgAdmin Development Team, 2025)

Nota. Fuente: (IEEE Xplorer), (Google Scholar).

En la Tabla 7 se muestra que ZigBee es el protocolo preferido para la comunicación entre sensores por su bajo consumo y capacidad de formar redes de malla, MQTT para transmitir datos y alertas en tiempo real, mientras que la conexión 4G/LTE ofrece redundancia para asegurar la continuidad del sistema ante fallos de la red principal. El uso combinado de estas tecnologías es una práctica recomendada para sistemas de seguridad doméstica inteligentes.

4.1.2. Análisis de infraestructura IoT con ZigBee

Para determinar los elementos de hardware y software necesarios para desarrollar una infraestructura IoT robusta y eficiente basada en ZigBee, requiere la integración de componentes de hardware especializados y configuraciones de software que permitan la recolección de datos en tiempo real, la gestión de la red de sensores y la garantía de continuidad operativa, en la Tabla 8 se resume los principales componentes identificados.

Tabla 8.

Componentes de hardware y software para IoT con ZigBee.

Categoría	Componente / Tecnología	Aplicación Principal	Referencias importantes	Total
Hardware	Raspberry Pi 5.	Unidad central de procesamiento y control del sistema.	(Adhikary et al., 2024; Chen et al., 2022b; Jacinto et al., 2024; Rosas Cruz, 2021; Samani et al., 2020)	5
	ZigBee USB Dongle.	Coordinador de red para gestionar la comunicación ZigBee.		
	Sensores ZigBee de movimiento y apertura.	Dispositivos finales para la recolección de datos en tiempo real.		
	Módulo 4G/LTE.	Respaldo de conectividad ante fallas de la red principal.		
	Fuente de alimentación ininterrumpida (UPS).	Mantener continuidad operativa durante cortes de energía.		
Software	Debian 12	Sistema operativo estable y seguro que sirve como base para ejecutar todos los servicios del proyecto.	(elektronik ingenieurtechnik GmbH, 2025; Group, 2025; InfluxData, 2025b; pgAdmin Development Team, 2025; Project, 2023; Ramírez, 2025)	6
	deCONZ + Phoscon	Software para gestionar la red ZigBee y controlar sensores de movimiento y apertura mediante el dongle ConBee II.		
	FastAPI	Framework para crear la API backend que recibe datos de los sensores para procesar la información, además permite la comunicación con otros servicios.		
	PostgreSQL	Base de datos relacional para almacenar registros de eventos, usuarios, credenciales y logs del sistema.		
	InfluxDB	Base de datos para almacenar los datos de los sensores y métricas del sistema en tiempo real.		

Nota. (IEEE Xplorer), (Google Scholar).

Como se muestra en la Tabla 8, el sistema combina hardware y software para asegurar la recopilación de datos en tiempo real, gestionar la red ZigBee de forma adecuada y tener

redundancia en la conectividad para garantizar la robustez, eficiencia y seguridad de una infraestructura IoT basada en ZigBee para monitoreo doméstico.

4.1.3. Evaluación de algoritmos de ML para detección de intrusos

Para determinar qué algoritmo es el más adecuado para la detección de intrusos y definir su integración en sistemas IoT domésticos, se realizó un análisis comparativo de la literatura de los algoritmos no supervisados eficaces en la detección de anomalías para estos entornos. La Tabla 9 resume los algoritmos más frecuentes en la literatura, no obstante se considera relevante los algoritmos señalados en la literatura como Elliptic Envelope o SVM.

Tabla 9.

Frecuencia de algoritmos de ML en la literatura seleccionada.

Algoritmo / Modelo	Referencias en la tesis	Total
Isolation Forest	(Chen et al., 2022b; Farizi et al., 2021; D. Xu et al., 2017; H. Xu et al., 2023)	4
Deep Learning / Redes Neuronales	(Saba et al., 2022; Singh et al., 2021; Taiwo et al., 2022)	3
Random Forest	(Meidan et al., 2017; V et al., 2021)	2

Nota.(IEEE Xplorer), (Google Scholar).

La Tabla 9 muestra que Isolation Forest destaca por mantener mayor cantidad de documentos encontrados, Deep Learning está a la par con tres conteos y Random Forest con dos.

4.2. Desarrollo e implementación de la infraestructura IoT

4.2.1. Arquitectura de sistema IoT basado en ZigBee

4.2.1.1. Requisitos de hardware. Los dispositivos físicos empleados para la construcción de una infraestructura iot se explican en los siguientes apartados:

La Tabla 10 define los elementos físicos que permiten una detección de actividad dentro del domicilio, posteriormente validación de eventos y notificación al usuario final.

Tabla 10.

Requisitos de hardware para detección de actividad.

Componente	Características	Cantidad
Rasperry Pi 5	CanaKit Raspberry Pi 5 Starter Kit PRO - Turbina Negro 8GB RAM	1
Memoria MicroSD con Debian 12;	Sistema operativo principal para el Rasperry Pi 5.	1
ConBee II Zigbee USB Dongle	Dogle USB, funciona como Gateway para comunicación de dispositivos Zigbee.	1
Aqara Motion Sensor P1 (Exteriores)	Modelo RTCGQ14LM Alcance 7 metros Detección PIR con ángulo de 150°.	4
Aqara Motion Sensor P1 (Interiores)	Modelo RTCGQ14LM Alcance 7 metros Detección PIR con ángulo de 150°	3
Aqara Door and Window Sensor	Modelo MCCGQ11LM Detección magnética de apertura y cierre.	6

La Tabla 11 define los componentes que permiten mantener una red energética robusta frente a fallas, asegurando que los sensores y la comunicación se mantenga continua mientras se mantenga el respaldo de almacenamiento y el funcionamiento de los paneles solares.

Tabla 11.

Requisitos de hardware para redundancia de energética.

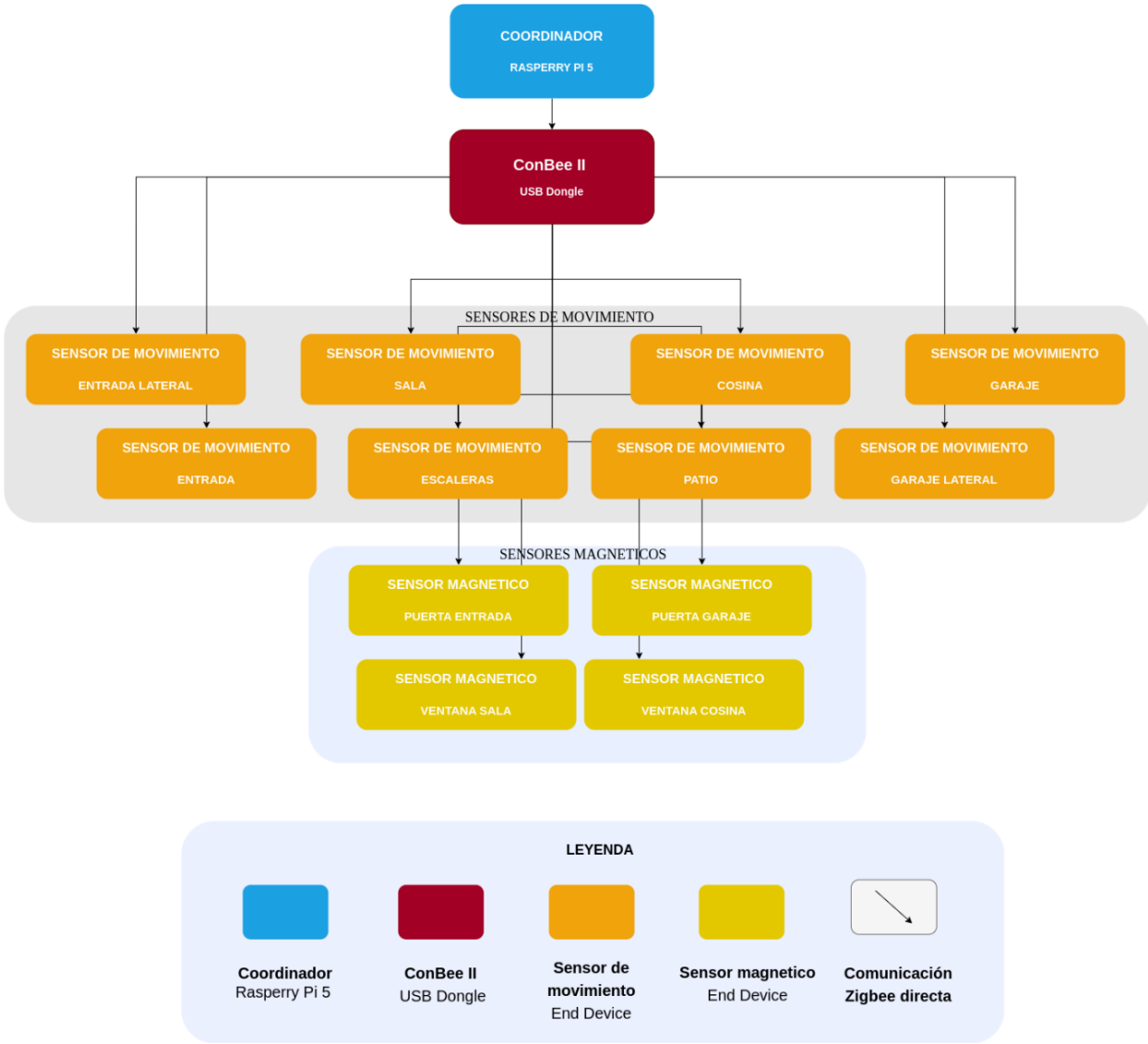
Componente	Características	Cantidad
Panel Solar	Potencia total: 1100W Tecnología: Monocristalino Voltaje nominal: ~24V Corriente máxima: ~23A por panel	2
Controlador MPPT	Máxima corriente de carga: 40A Voltaje del sistema: 12V/24V auto Protecciones: sobrecarga, cortocircuito	1
Inversor Onda Senoidal Pura	Potencia continua: 2000W Potencia pico: ~4000W (2-3 segundos) Voltaje entrada: 12V DC Voltaje salida: 120V AC / 60Hz	1
Batería de Gel	Capacidad: 150 Amperios-hora Tecnología: Gel (libre mantenimiento) Ciclos de vida: 1200-1500 ciclos Temperatura operación: -15°C a +50° Voltaje: 12V	1

4.2.1.2. Estructura lógica de conectividad ZigBee. La representación esquemática de esta conectividad se ilustra en la Figura 18 , la cual está pensada para

proporcionar una solución escalable para la monitorización y control de dispositivos en un entorno doméstico.

Figura 18.

Topología lógica de la red ZigBee.

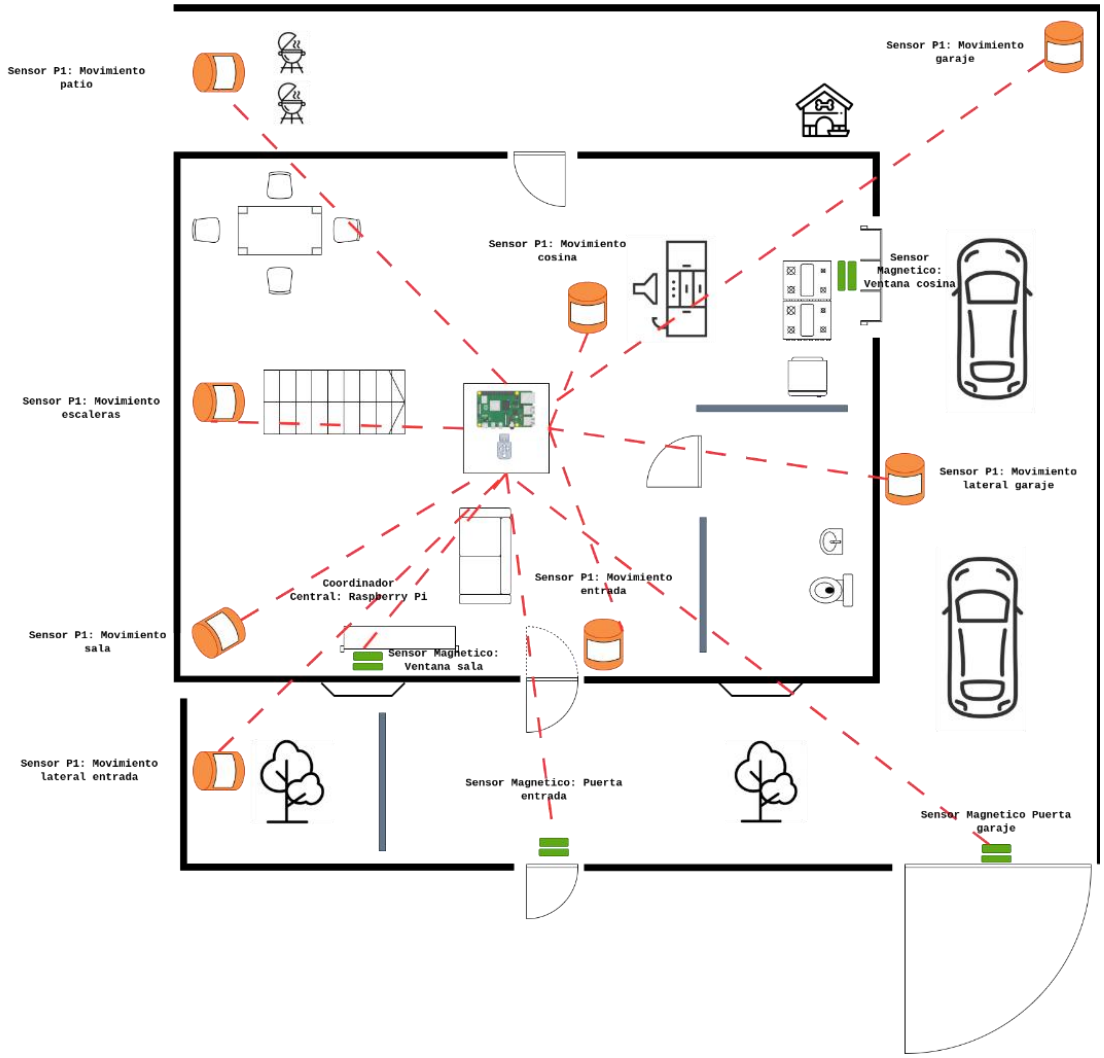


Doce dispositivos finales conectados divididos en sensores magnéticos y de movimiento se comunican directamente con el coordinador sin dispositivos intermediarios, esta es considerada una topología óptima para un entorno doméstico por su simplicidad de configuración, mantenimiento, máxima duración de batería en sensores y una cobertura suficiente para el área de 120m².

4.2.1.3. Distribución física de la red de sensores. La distribución empleada se basa en una topología en estrella que consta de ocho sensores de movimiento y cuatro sensores magnéticos interconectados hasta el coordinador, su representación dentro del domicilio se puede observar en la Figura 19.

Figura 19.

Topología física de la red ZigBee.



La topología aplicada mantiene rutas de comunicación directa hasta el coordinador (líneas interconectadas), estos sensores están ubicados para cubrir la mayor parte de los puntos de acceso en casa: puertas de acceso, ventanas, sala, cocina y garaje, esto permite la recolección de información del comportamiento habitual en casa para el entrenamiento del modelo.

4.2.2. Implementación y despliegue operativo de la red de sensores

4.2.2.1. Sistema de captura de información en tiempo real. Los sensores de movimiento y magnéticos envían los datos recopilados hasta el coordinador central (Rasperry Pi 5 y ConBee II Zigbee USB Dongle), donde se procesan los registros obtenidos, validando los eventos reales, eliminando los eventos duplicados y filtrando estos registros para almacenar series temporales limpias, evitando el registro de ruido y finalmente estos se almacenan dentro de la base de datos InfluxDB donde su interfaz se puede observar en la Figura 20, por otra parte, Grafana funciona como un sistema que tiene la capacidad de transformar la información almacenada en gráficos comprensibles y accionables, esta se puede observar en la Figura 21.

Figura 20.

Interfaz de InfluxDB.

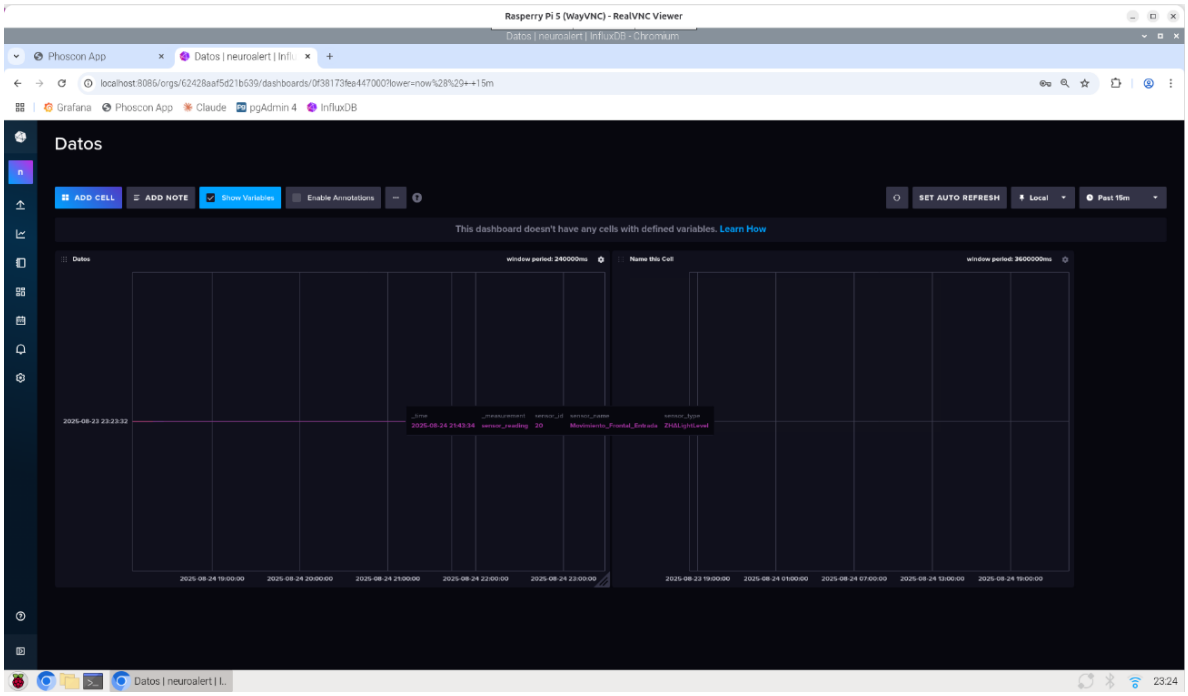
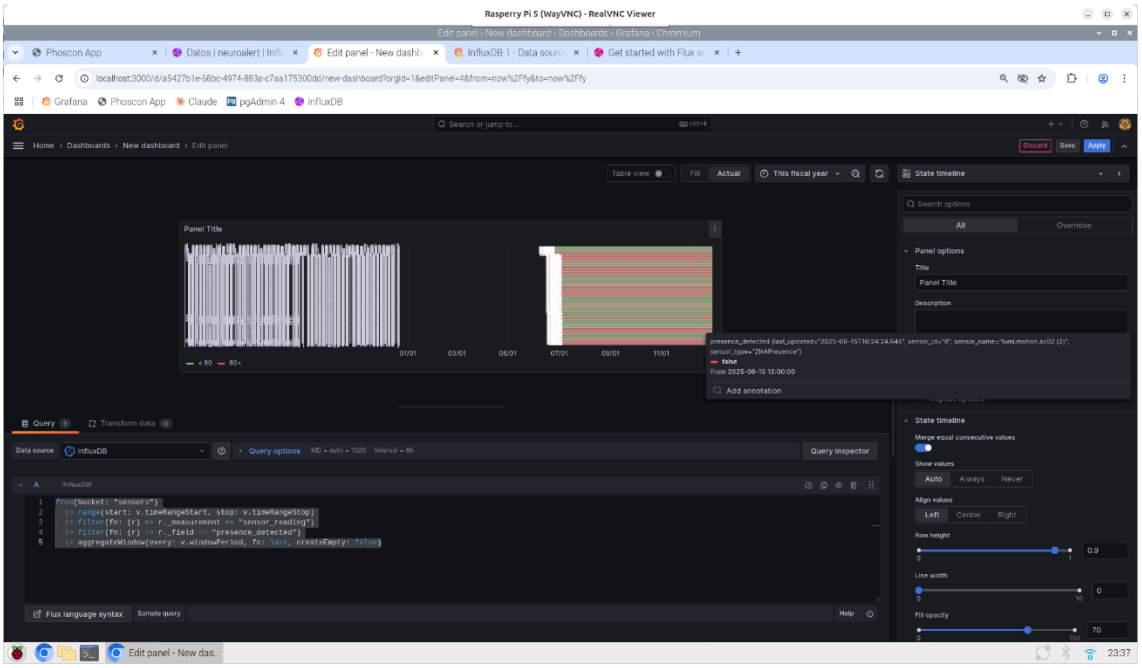


Figura 21.

Interfaz de Grafana.

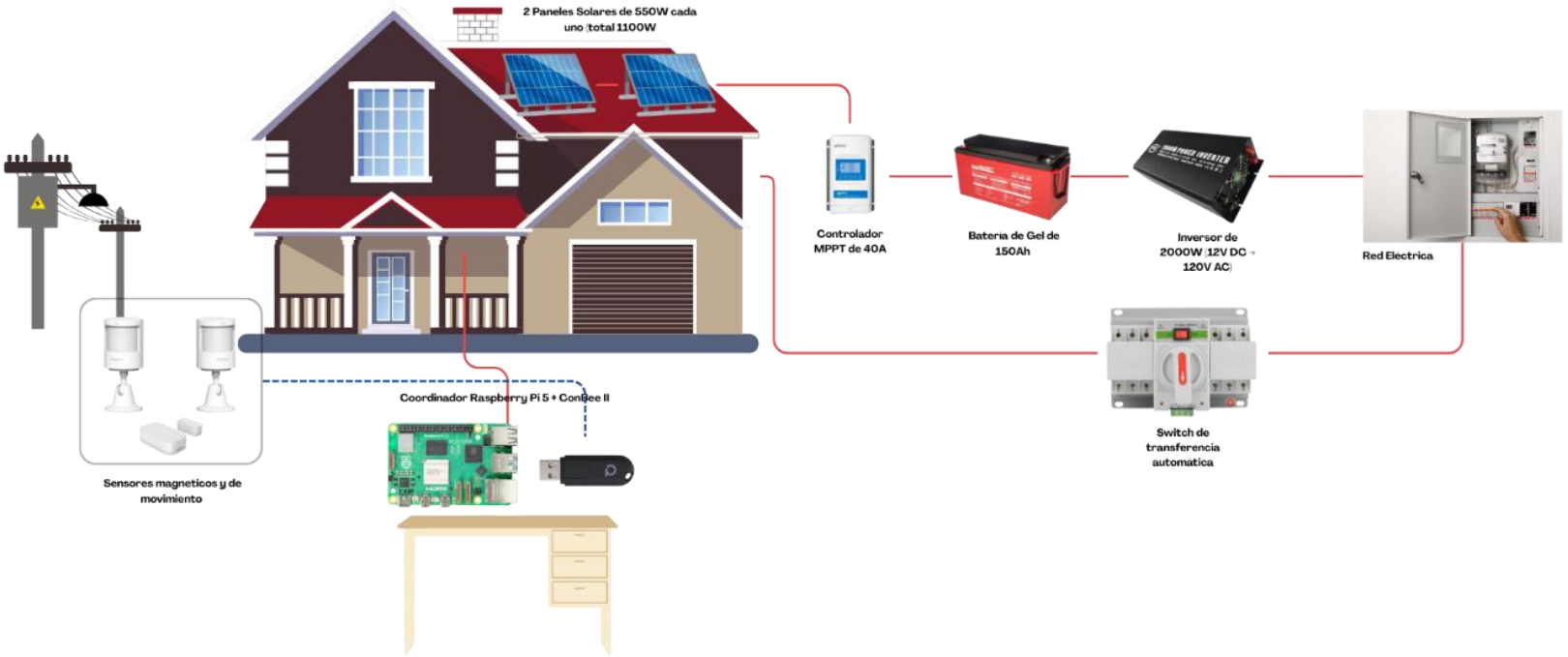


4.2.2.2. Documentación visual de sensores implementados. Se presenta los sensores instalados en casa, su evidencia se puede observar dentro del **Anexo 4. Sensores instalados en vivienda.**

4.2.2.3. Continuidad operativa ante interrupciones del suministro energético. Frente a posibles interrupciones de energía se emplea un sistema de alimentación autónoma mediante paneles solares que permite tener un respaldo de energía, en la Figura 22 se representa el funcionamiento del sistema integrado.

Figura 22.

Sistema de energía solar como respaldo energético.



El funcionamiento del sistema es el siguiente:

1. Los paneles solares capturan la energía solar durante todo el día con una capacidad de 1100 Watts
2. El controlador MPPT optimiza la conversión y regula la carga:
3. Carga las baterías cuando hay exceso de energía
4. Protege contra sobrecargas y cortocircuitos
5. Ajusta automáticamente el voltaje (12V/24V)
6. La batería de gel de 12V 150A se mantiene cargada y lista como respaldo
7. El inversor convierte 12V DC a 120V AC
8. EL switch de transferencia mantiene la alimentación desde el sistema solar.

Cuando no hay suficiente energía solar, la batería automáticamente suministra energía al inversor, el sistema puede funcionar aproximadamente ocho a doce horas sin sol y en modo de emergencia cuando la batería se agota el switch de transferencia automáticamente cambia a la red eléctrica, lo que garantiza una operación continua al sistema de energía en casa, de esta forma el coordinador Conbee II en conjunto con Rasperry Pi 5 puede mantener conexión ininterrumpida con los sensores.

4.2.2.4. Documentación fotográfica del sistema de respaldo. Se presenta el sistema de energía de respaldo, cuya evidencia puede observarse en el **Anexo 5. Sistema de respaldo energético.**

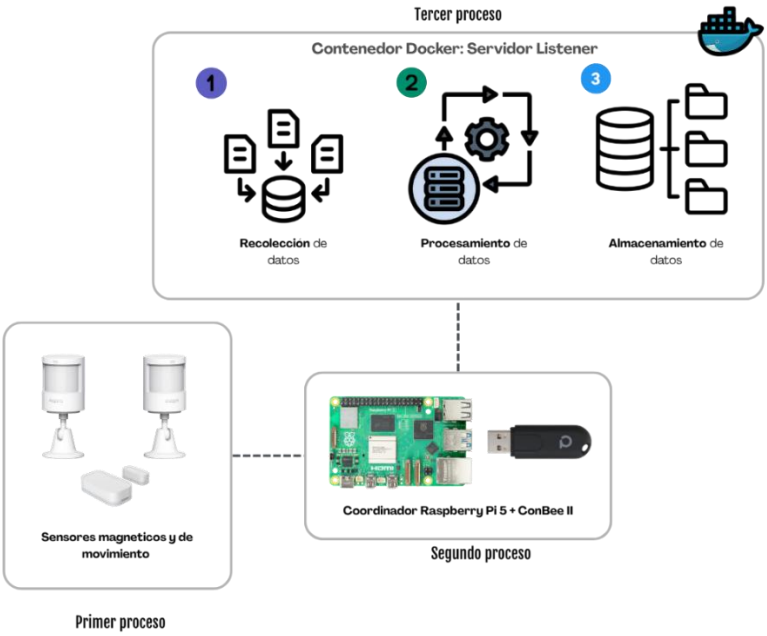
4.3. Implementación del sistema de detección de intrusos

4.3.1. Modelo de ML para detección de intrusos

4.3.1.1. Recolección de información. La Figura 23 ilustra el proceso para la recolección de datos dentro del domicilio.

Figura 23.

Proceso de recolección de datos.

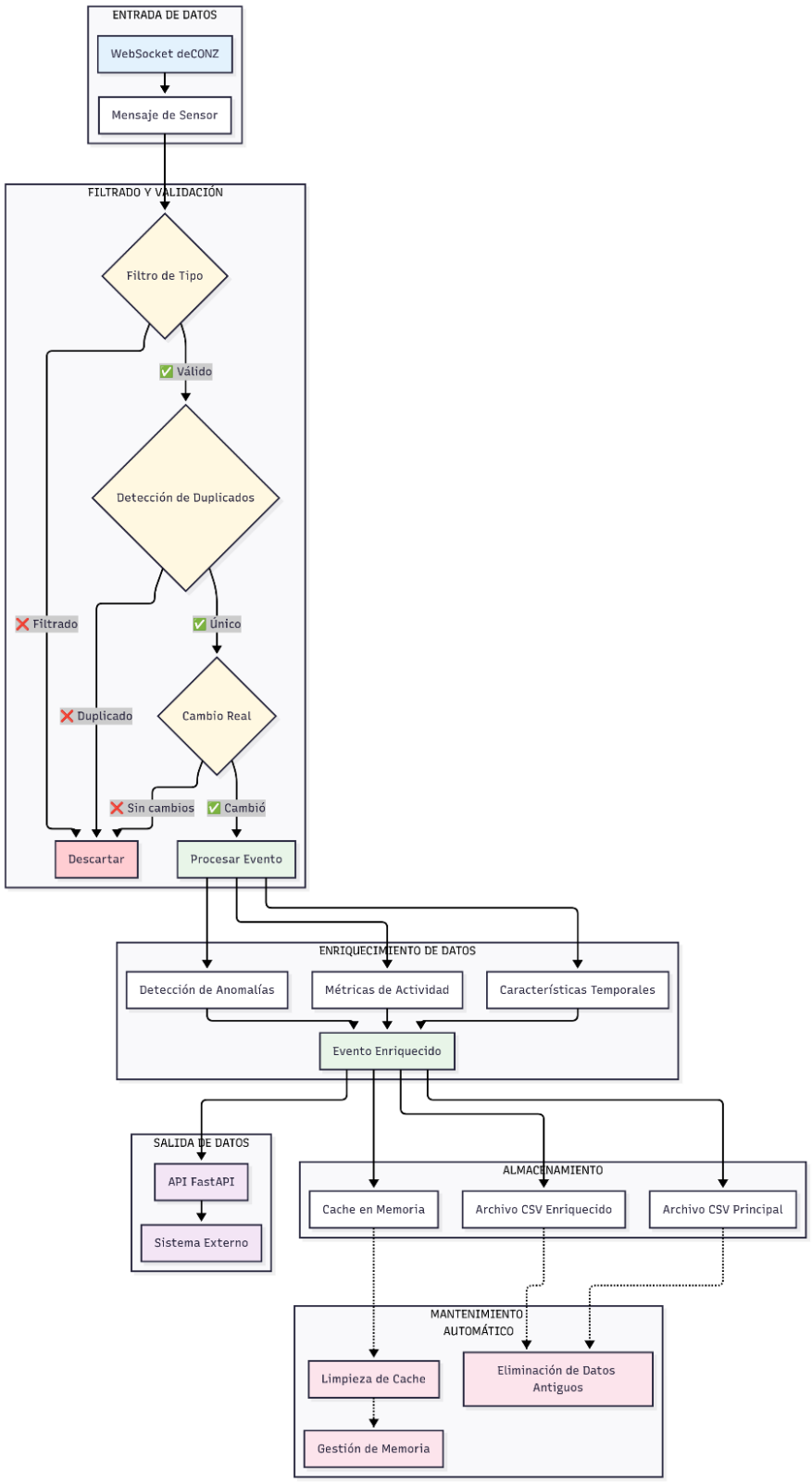


La recolección de información consta de tres pasos esenciales, el primero sucede cuando los sensores Aqara se encargan de transmitir las señales capturadas al coordinador Conbee II que se encuentra conectado al Rasperry Pi 5 donde a su vez esta información es obtenida dentro de un contenedor Docker, que se encarga de recolectar, procesar y almacenar la información.

4.3.1.2. Procesamiento de información. La Figura 24 muestra el tratamiento de los datos en un flujograma luego de que estos registros son transmitidos de los sensores Aqara hasta el coordinador.

Figura 24.

Flujograma de procesamiento de datos.



En primera instancia los sensores magnéticos y de movimiento transmiten los datos al servidor mediante una conexión WebSocket y tras lo cual inicia la etapa de procesamiento de los datos.

Como primer paso se determina si el tipo de sensor debe ser procesado y si el evento recibido es duplicado siendo calculado en comparación al tiempo del evento anterior registrado, dentro del enriquecimiento de datos se amplía la información extraída del evento original, este proceso se ejecuta de forma paralela en tres dimensiones distintas, la detención de anomalías evalúa si el evento ocurre en condiciones inusuales particularmente en horarios nocturnos, las métricas de actividad calculan la frecuencia de activación del sensor, incluyendo el tiempo transcurrido desde el ultimo evento y el conteo de eventos ocurridos en los últimos treinta minutos, finalmente las características temporales analiza el contexto temporal del evento, para identificar patrones de tiempo en el comportamiento de los sensores.

Los eventos que superan los filtros se envían a un servidor principal donde se evalúa según la configuración del sistema para determinar si debe generarse una notificación, así mismo también se realiza el almacenamiento de los eventos en dos tipos de archivos csv, uno principal con la información original y otro enriquecido con la información ampliada, además se guarda la actividad reciente dentro de la cache de memoria.

Como último paso el proceso de mantenimiento realiza una gestión de memoria, limpieza en cache, eliminación de datos antiguos dentro del csv principal y enriquecidos cada tres meses.

4.3.1.3. Análisis exploratorio de datos (correlaciones Pearson y Spearman). Se presenta la matriz de correlación empleando las variables recolectas con el método de Pearson para datos lineales en la Figura 25 y Spearman para los datos no lineales en la Figura 26.

Figura 25.

Matriz de correlación - Pearson (Lineal).

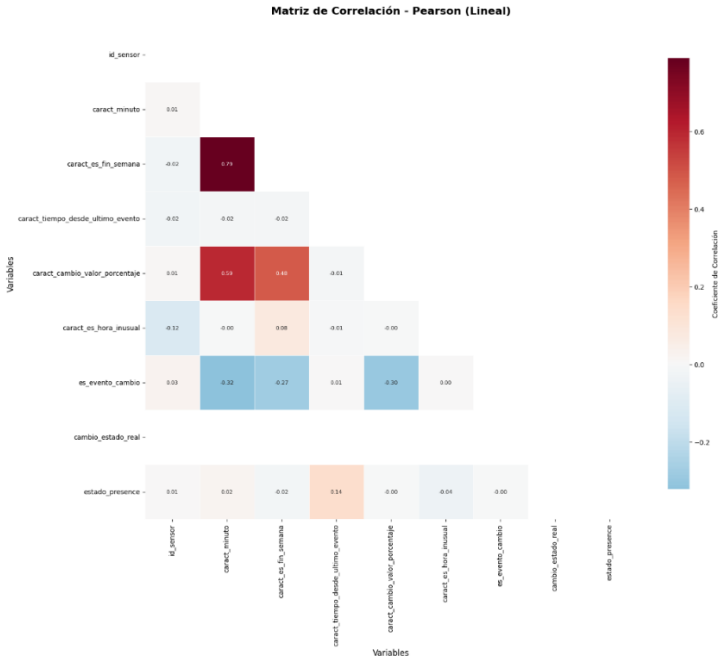
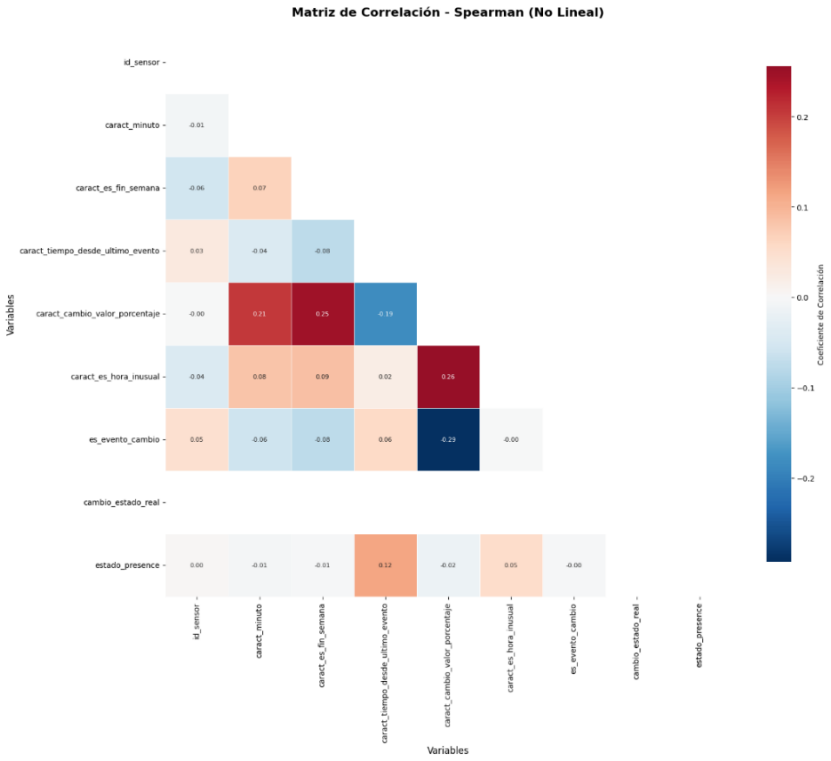


Figura 26.

Matriz de correlación - Spearman (No lineal).



- **Pearson (Lineal):** Muestra correlaciones más fuertes entre caract_minuto y caract_es_fin_semana con 0.79
- **Spearman (No lineal):** Las correlaciones son más débiles con valores máximos alrededor de 0.25 y 0.29

La gran diferencia entre las correlaciones de Pearson (hasta 0.79) y Spearman (máximo ~0.29) revela que las relaciones lineales simples no capturan adecuadamente la naturaleza del dataset.

El análisis revela tres características importantes en el comportamiento del sistema, en primer lugar, se observar una estructura temporal compleja donde existe una fuerte relación entre las variables temporales y el comportamiento del sensor, pero esta relación se muestra de manera no lineal.

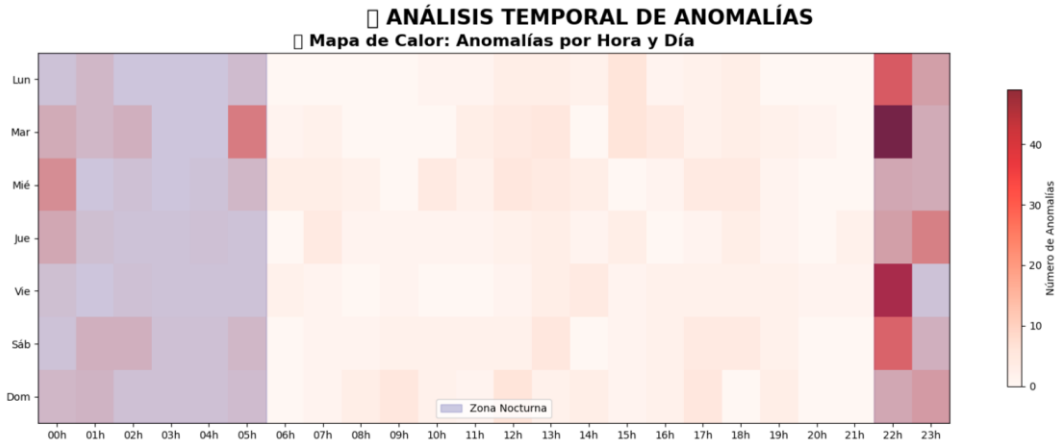
En segundo lugar, los eventos de cambios de estado no siguen patrones regulares lo que siguiere una naturaleza impredecible en las transiciones del sistema y finalmente se observa una interdependencia entre variables donde las características del sistema influyen de formas no obvias en análisis lineales tradicionales.

Como conclusión la naturaleza no lineal de las relaciones identificas sugiere el empleo de modelos de ML no lineales como, por ejemplo: Random Forest o máquinas de vectores de soporte con kernels no lineales porque resultarían más apropiados para capturar la complejidad de los sensores.

4.3.1.4. Análisis temporal de anomalías encontradas. La Figura 27 presenta un mapa de calor que muestra la distribución temporal de anomalías a lo largo de la semana.

Figura 27.

Mapa de calor - Análisis temporal de anomalías.



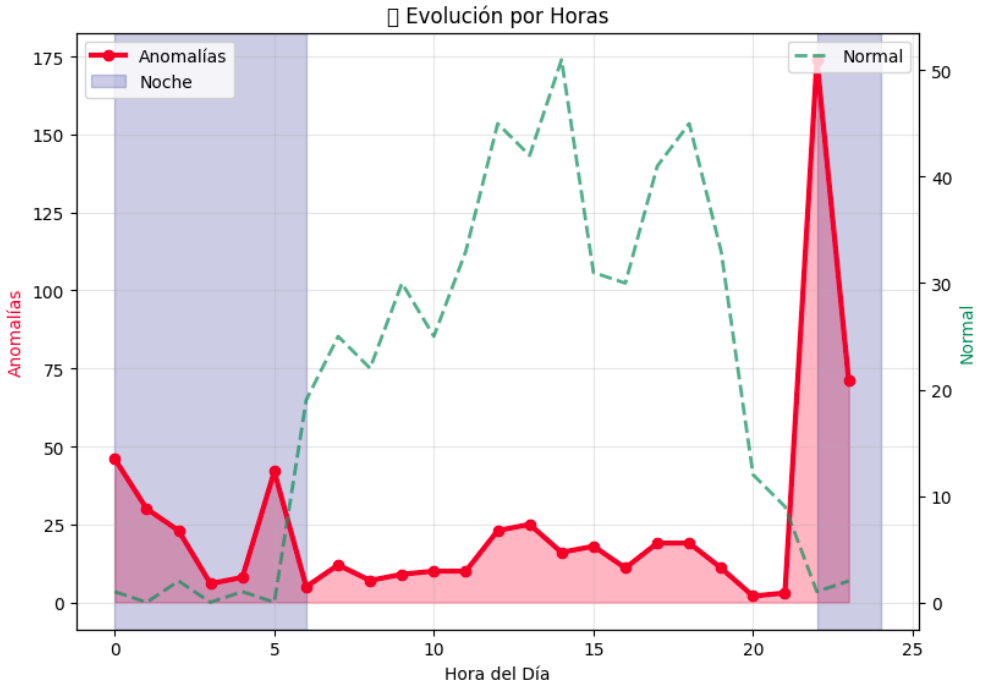
En el eje vertical se presentan los días de la semana y en el eje horizontal las horas del día, la intensidad del color indica el número de anomalías detectadas con una escala desde el 0 hasta 40 de menor a mayor intensidad de color, esta figura se puede observar una mayor concentración de anomalías durante las horas nocturnas representada por la zona sombreada en púrpura claro.

Los valores más altos de anomalías se concentran principalmente en lunes en las primeras horas (00h-03h) con valores cercanos a 40 anomalías, martes, miércoles (22h-23h) y viernes en las horas nocturnas, por otra parte, durante el periodo diurno se observa activada anómala menor con rango predominante de 0-10 anomalías.

La Figura 28 ilustra la evolución temporal de las anomalías en comparación con el comportamiento normal esperado a lo largo del periodo de 24 horas.

Figura 28.

Evolución temporal de las anomalías.



El comportamiento normal esperado representado por una línea punteada verde sigue un ciclo diario con valores bajos en la madrugada, aumento progresivo durante el periodo diurno y descenso durante la noche a niveles bajos. Mientras que las anomalías presentan un comportamiento diferenciado con un pico inicial significativo en la hora 0 con aproximadamente 45 anomalías, un descenso progresivo llegando a valores mínimos entre las 2h y 4h, un pico segundario alrededor de la hora 5 con una actividad baja y estable durante

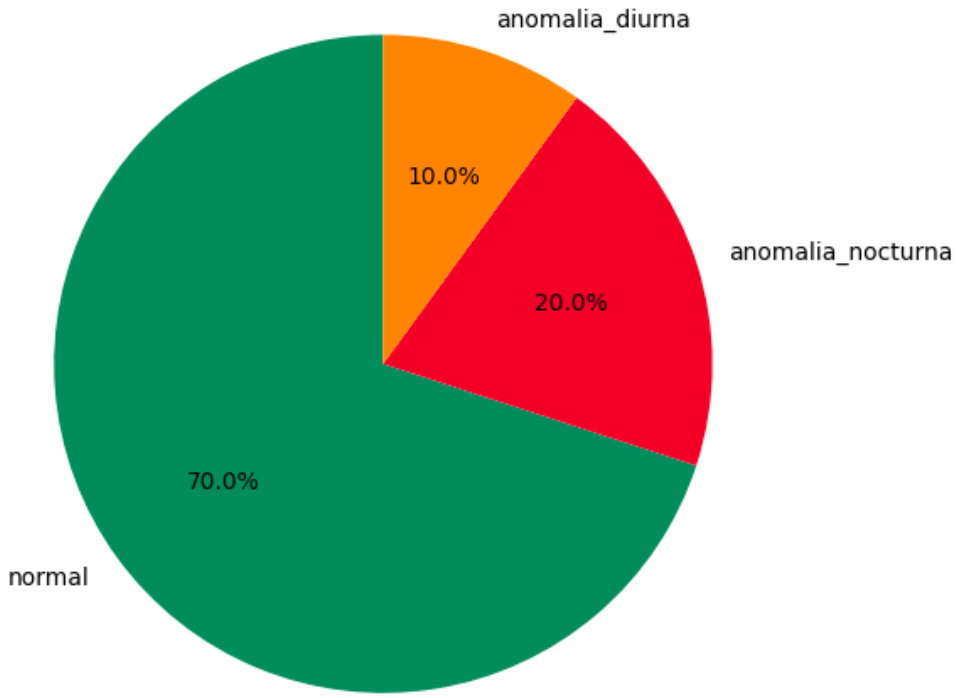
el periodo diurno seguidamente con un incremento al final del día alcanzando picos máximos de aproximadamente 170 anomalías en la hora 22.

El análisis revela un comportamiento anómalo que se opone al patrón normal esperado, mientras que la actividad normal presenta sus valores máximos durante las horas centrales del día las anomalías se concentran predominantemente en las horas nocturnas, particularmente en las transiciones día-noche (horas 5am y 22pm).

La Figura 29 presenta la distribución porcentual de los datos clasificados según las tres categorías principales.

Figura 29.

Distribución porcentual de los datos.



Se puede observar que el 70% de los datos constituyen a registros de comportamiento esperado, las anomalías nocturnas con 20% y las anomalías diurnas con 10% constituyen la menor proporción de anomalías detectadas, estas relaciones indican que los datos son consistentes con las Figura 27 y Figura 28 presentadas anteriormente.

4.3.1.5. Selección y de algoritmos de detección. La Tabla 12 describe las características principales, ventajas y justificación para su selección en el contexto específico de este estudio.

Tabla 12.

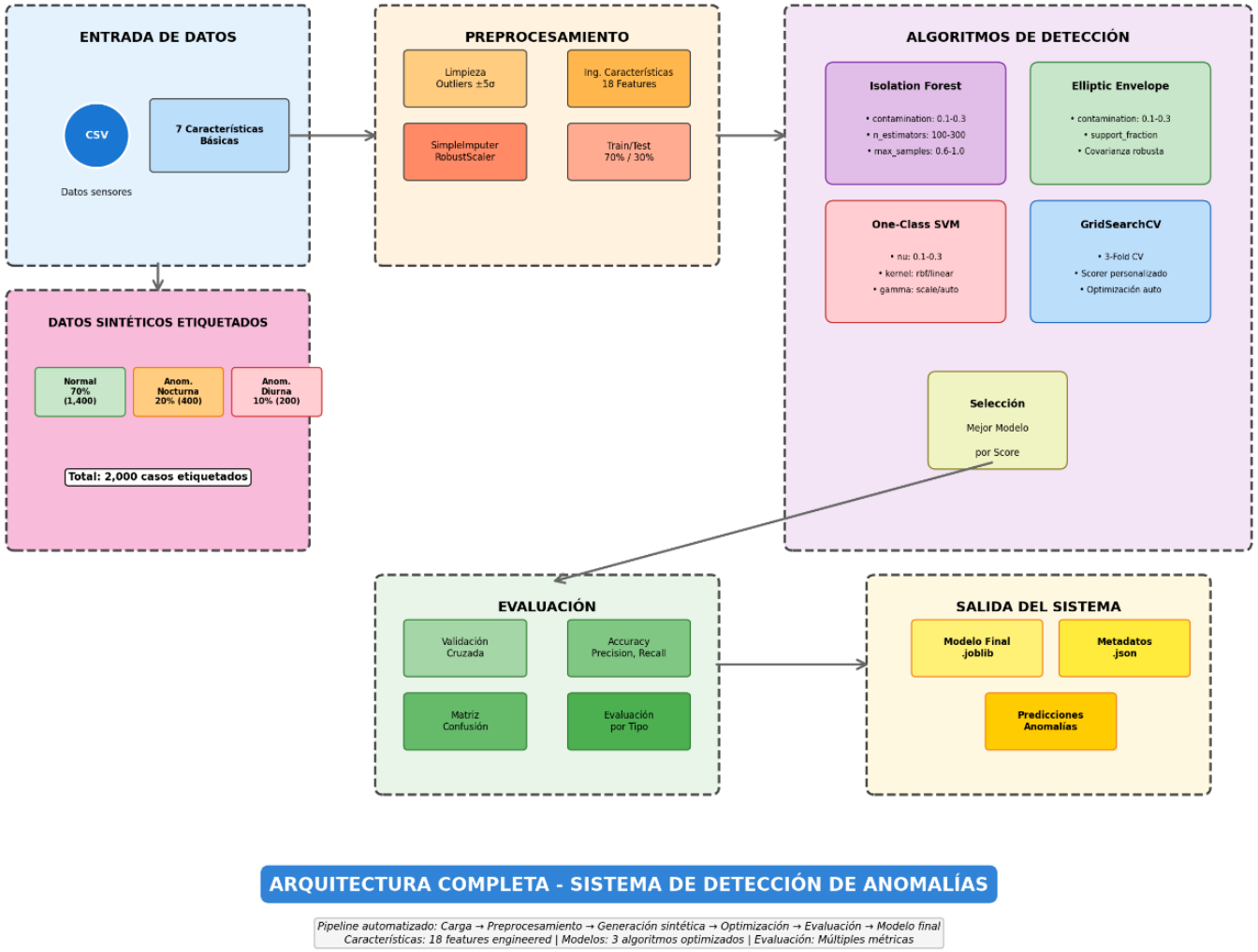
Características de algoritmos seleccionados.

Algoritmo	Principio	Ventajas clave	Uso en sensores
Isolation Forest	Aísla anomalías con árboles aleatorios; requieren menos particiones.	Escalable, eficiente, no asume distribuciones, robusto a ruido.	Ideal para grandes volúmenes de datos IoT y sensores de movimiento.
Elliptic Envelope	Modela datos normales como distribución gaussiana multivariada.	Interpretable, eficaz con datos gaussianos, maneja correlaciones.	Adecuado para sensores calibrados con comportamiento gaussiano.
One-Class SVM	Separa normales de anomalías en espacio de alta dimensión con kernels.	Flexible, no asume distribución, robusto a outliers, útil en alta dimensión.	Útil para patrones no lineales y series temporales de sensores.

4.3.1.6. Arquitectura de modelo de detección de anomalías. En la Figura 30 se presenta la arquitectura establecida para la detección de anomalías.

Figura 30.

Arquitectura de modelo de detección de anomalías.



El modelo implementa un pipeline automatizado con el ingreso de los datos de los sensores, aplica limpieza e ingeniería de datos, genera un dataset balanceado con 2000 casos y evalúa tres algoritmos (Isolation Forest, Elliptic Envelope, One-Class SVM) mediante validación cruzada y métrica de desempeño selecciona el mejor modelo, entregando un artefacto con el modelo final, metadatos y las predicciones.

La Tabla 13 presenta un resumen de los componentes de la arquitectura presentada en la Figura 30.

Tabla 13.

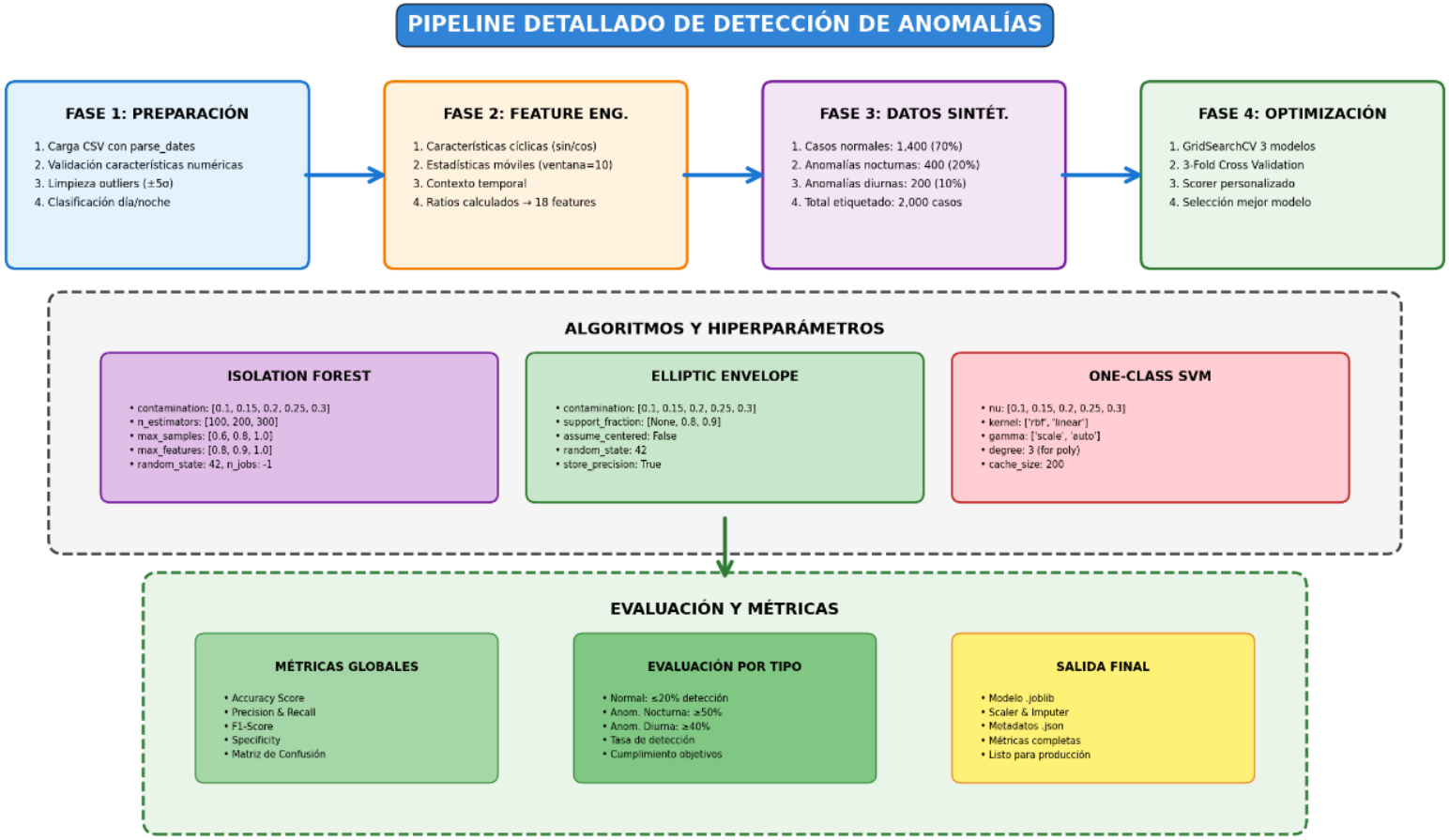
Resumen de componentes de la arquitectura de ML.

Fase del Pipeline	Componentes/Configuración	Detalles Técnicos
Entrada de datos	• Formato: CSV • Características iniciales • Datos etiquetados: 2,000 casos	• Normales: 1,400 • Anomalías: 600 (20% sintéticas)
Preprocesamiento	• Limpieza • Manejo de outliers (±3σ) • Ingeniería de características • Escalado	• Features finales: 18 • RobustScaler • División train/test: 70%/30%
Algoritmos	• Isolation Forest • Elliptic Envelope • One-Class SVM	• Contaminación ajustada (0.1–0.3) • Kernel: RBF/Linear • Optimización automática
Optimización	• GridSearchCV • Validación cruzada (3-fold) • Métrica personalizada	• YaHiperparámetros ajustados: • max_samples, support_fraction, • gamma
Evaluación	• Métricas: Precisión, Recall, AUC-ROC • Validación cruzada • Análisis por tipo de anomalía	• Resultados desglosados en JSON
Salida	• Modelo final empaquetado • Reporte de métricas globales y específicas	• Exportación en JSON

4.3.1.7. Especificaciones técnicas del pipeline de detección. La Figura 31 presenta el pipeline detallado para la detección de anomalías, el cual sigue un enfoque sistemático de cuatro fases secuenciales: preparación de datos, ingeniería de características, generación de datos sintéticos y optimización del modelo.

Figura 31.

Pipeline detallado de detección de anomalías.



La Tabla 14 apoyada de Figura 31 desglosa los elementos técnicos más importantes organizados por etapas del pipeline, incluyendo sus parámetros específicos, criterios de evaluación y artefactos de salida.

Tabla 14.

Elementos del pipeline para detección de anomalías.

Fase / Sección	Descripción / Actividades
Fase 1: Preparación	• **Carga de datos:** Importación de CSV con parser de fechas • **Validación numérica:** Verificación de formatos • **Limpieza de outliers:** Regla ±3σ • **Clasificación día/noche:** Categorización para patrones temporales
Fase 2: Feature engineering	• **Características cíclicas:** Transformaciones con senos y cosenos • **Estadísticas móviles:** Ventanas deslizantes de 10 períodos • **Contexto temporal**: Variables históricas • **Ratios calculadas:** 18 nuevas características derivadas
Fase 3: Datos sintéticos	• **Casos normales:** 1,400 (70%) • **Anomalías nocturnas:** 400 (20%) • **Anomalías diurnas:** 200 (10%) • Total: 2,000 casos etiquetados
Fase 4: Optimización	• **GridSearchCV:** Prueba de 3 modelos • **Validación cruzada:** 3-fold cross validation • **Scoring personalizado:** Métrica específica • **Selección:** Mejor algoritmo según rendimiento
Isolation Forest - Hiperparámetros	• **contamination:** [0.1, 0.15, 0.2, 0.25, 0.3] • **n_estimators:** [100, 200, 300] • **max_samples:** [0.6, 0.8, 1.0] • **max_features:** [0.8, 0.9, 1.0] • **random_state:** 42, n_jobs: -1
Elliptic envelope - Hiperparámetros	• **contamination:** [0.1, 0.15, 0.2, 0.25, 0.3] • **support_fraction:** [None, 0.8, 0.9] • **assume_centered:** False • **random_state:** 42 • **store_precision:** True
One-Class Svm - Hiperparámetros	• **nu:** [0.1, 0.15, 0.2, 0.25, 0.3] • **kernel:** ['rbf', 'linear'] • **gamma:** ['scale', 'auto'] • **degree:** 3 (for poly) • **cache_size:** 200
Métricas globales	• **Accuracy Score:** Precisión general del modelo • **Precision & Recall**: Métricas de clasificación • **F1-Score:** Media armónica precision/recall • **Specificity:** Tasa de verdaderos negativos • **Matriz de Confusión:** Análisis detallado de errores
Evaluación por tipo	• **Normal**: ≥20% detección correcta • **Anomalías Nocturnas:** ≥50% detección

Salida final	• **Anomalías Diurnas**: ≥40% detección • **Tasa de detección:** Porcentaje por categoría • **Cumplimientos objetivos:** Validación de metas • **Modelo joblib:** Algoritmo seleccionado serializado • **Scaler & Imputer:** Objetos de preprocesamiento • **Metadatos json:** Configuración y parámetros • **Métricas completas:** Reporte de rendimiento
Consideraciones técnicas	• **Balanceamento de clases:** 70/20/10 sintético estratificado • **Validación temporal:** Respeta secuencia cronológica • **Feature scaling:** Normalización para SVM y Elliptic • **Memory efficiency:** Optimización para datasets grandes • **Reproducibilidad:** Seeds fijos (random_state=42)

4.3.2. Sistema de seguridad con notificaciones multicanal

4.3.2.1. Fase 1: Planificación

- **Requerimientos funcionales del sistema**: En la Tabla 15 se muestran los requerimientos funcionales del sistema.

Tabla 15.

Requerimientos funcionales del sistema.

Número	Nombre del requerimiento	Descripción del requerimiento
RF001	Ingreso al sistema	El administrador debe poder acceder al sistema mediante un usuario y contraseña definidos.
RF002	Actualizar clave	El sistema debe permitir el cambio de clave para el acceso al sistema
RF003	Actualización de modo de uso	El sistema debe permitir alternar entre los modos de uso (Modo alerta y uso del modelo de ML)
RF004	Modificación de canales de comunicación	El sistema debe permitir escoger entre los canales de comunicación para él envió de alertas.

- **Requerimientos no funcionales del sistema:** En la Tabla 16 se muestran los requerimientos no funcionales del sistema.

Tabla 16.

Requerimientos no funcionales del sistema.

Número	Nombre del requerimiento	Descripción del requerimiento
RNF001	Usabilidad	El sistema debe tener interfaces intuitivas y amigables.
RNF002	Seguridad	El acceso se otorgar a administrador mediante sus respectivas credenciales.
RNF003	Desarrollo	El sistema sera desarrollado bajo tecnologías como

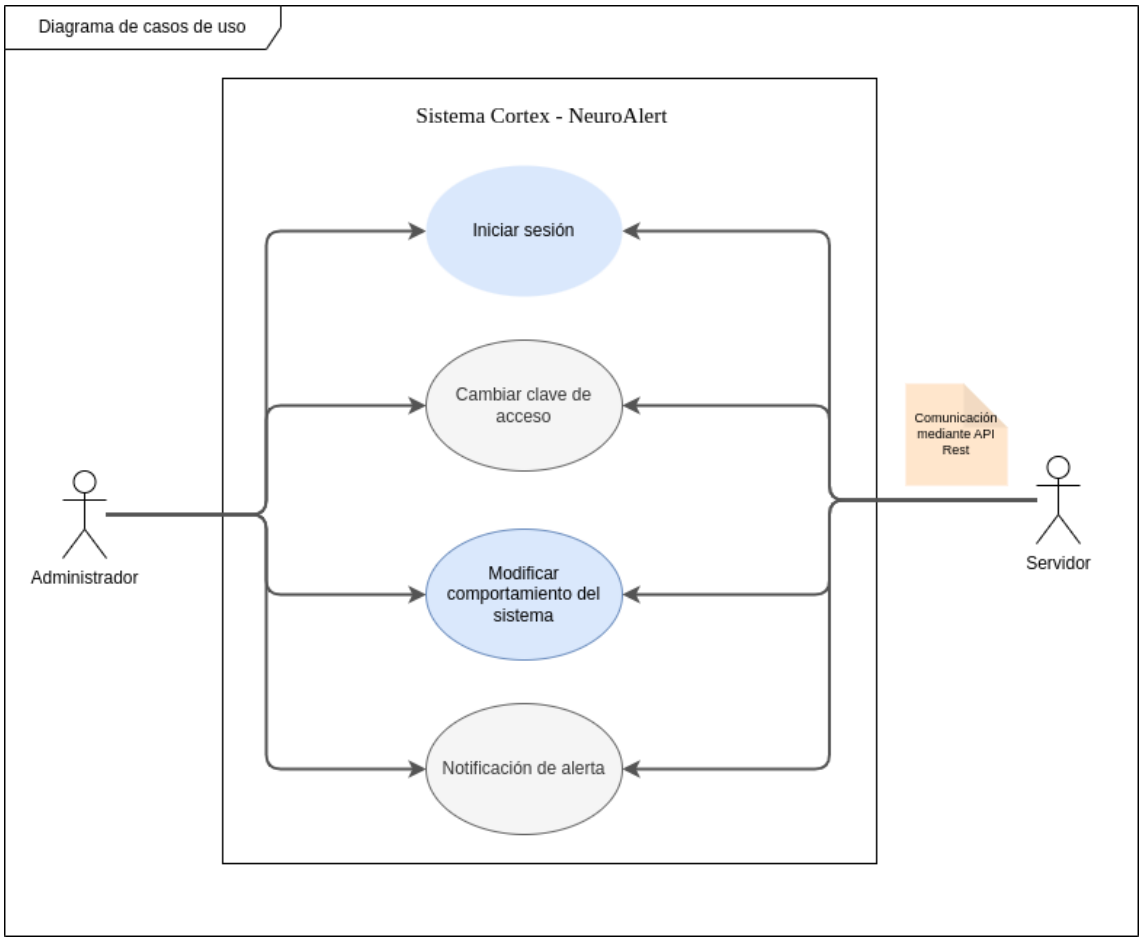
		Docker, React, FastApi, InfluxDB y Visual Estudio Code como IDE de desarrollo.
RNF004	Éticos	El sistema se desarrolla bajo principio de respeto e igualdad, sin discriminar a ninguna persona, comunidad u organización.

4.3.2.2. Fase 2: Diseño

4.3.2.2.1. Casos de uso. En la Figura 32 se representa mediante el diagrama de casos de uso los escenarios del sistema y los actores que intervienen.

Figura 32.

Casos de uso.

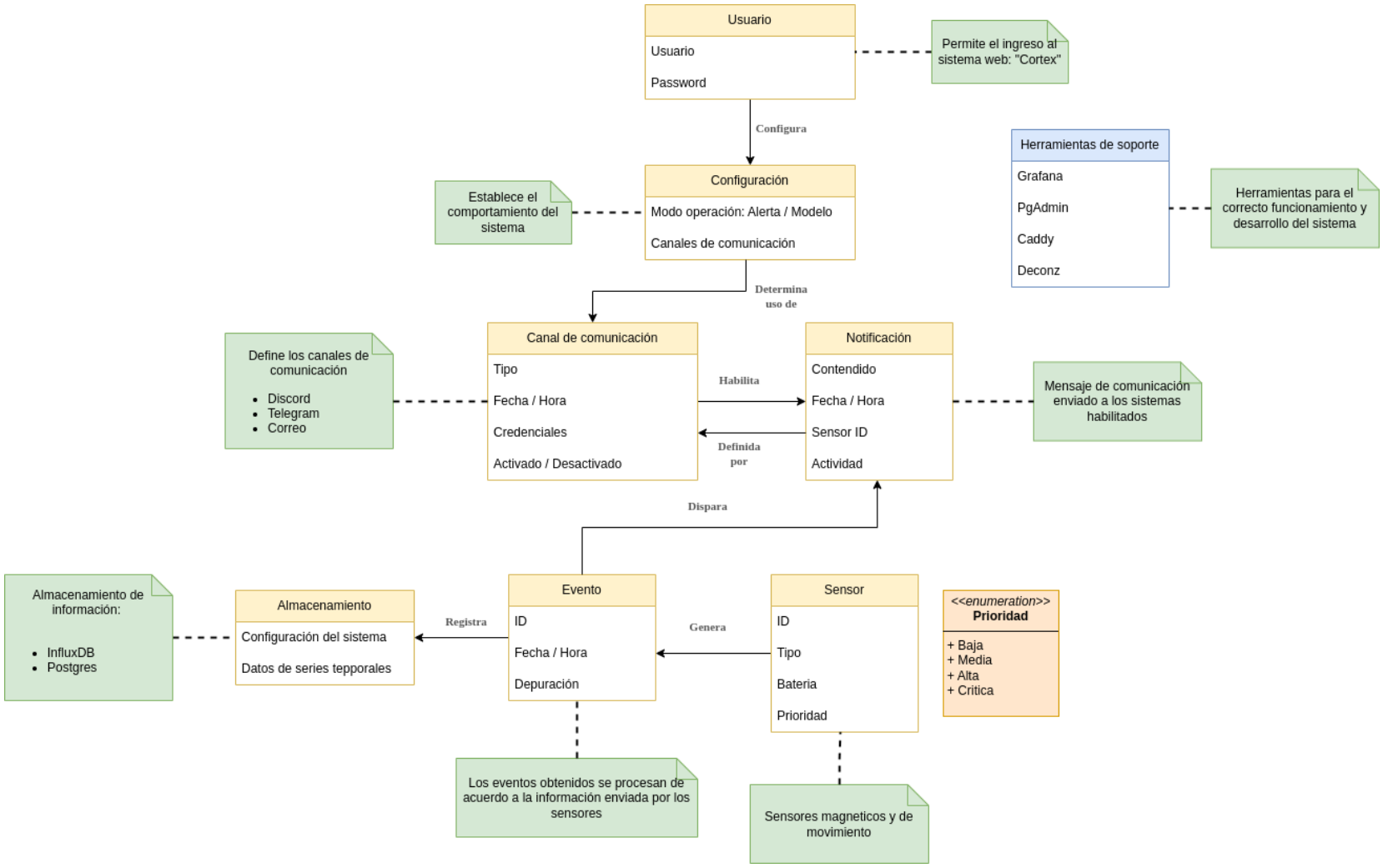


El sistema permite que el administrador dentro del sistema inicie sesión, cambie su clave de acceso y modifique el comportamiento del sistema para así recibir notificaciones de acuerdo con sus preferencias, estas funciones las gestiona el servidor principal mediante una comunicación API REST con los sistemas integrados.

4.3.2.2.2. Diagrama de dominio. En la Figura 33 se presenta el diagrama de dominio, el cual modela y permite entender las relaciones entre ellos los conceptos más importantes del sistema.

Figura 33.

Diagrama de dominio.

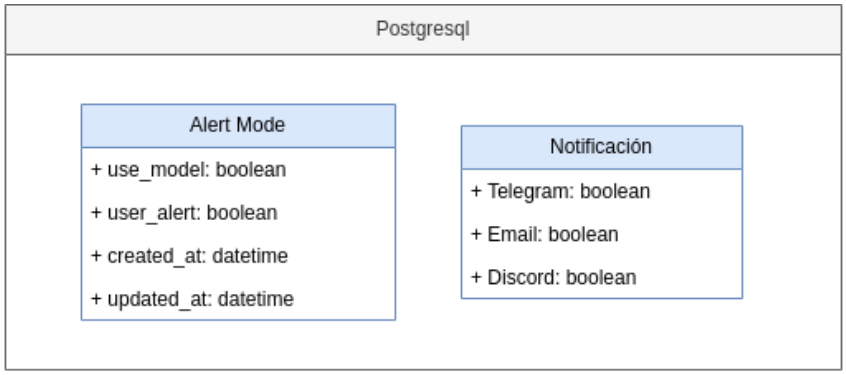


Dentro del diagrama de dominio se ilustra que el usuario o administrador con acceso puede configurar el sistema y modificar su modo de operación, incluyendo la activación o desactivación de canales comunicación, de esta forma cuando un sensor remite un registro al servidor este es procesado, almacenado y transmitido por notificación mediante los canales de comunicación activos.

4.3.2.2.3. Diagrama de clases. En la Figura 34 se puede conocer la estructura se almacenamiento de las bases de datos empleadas, PostgreSQL destinada a la configuración del sistema e InfluxDB para almacenar los datos generados por los sensores.

Figura 34.

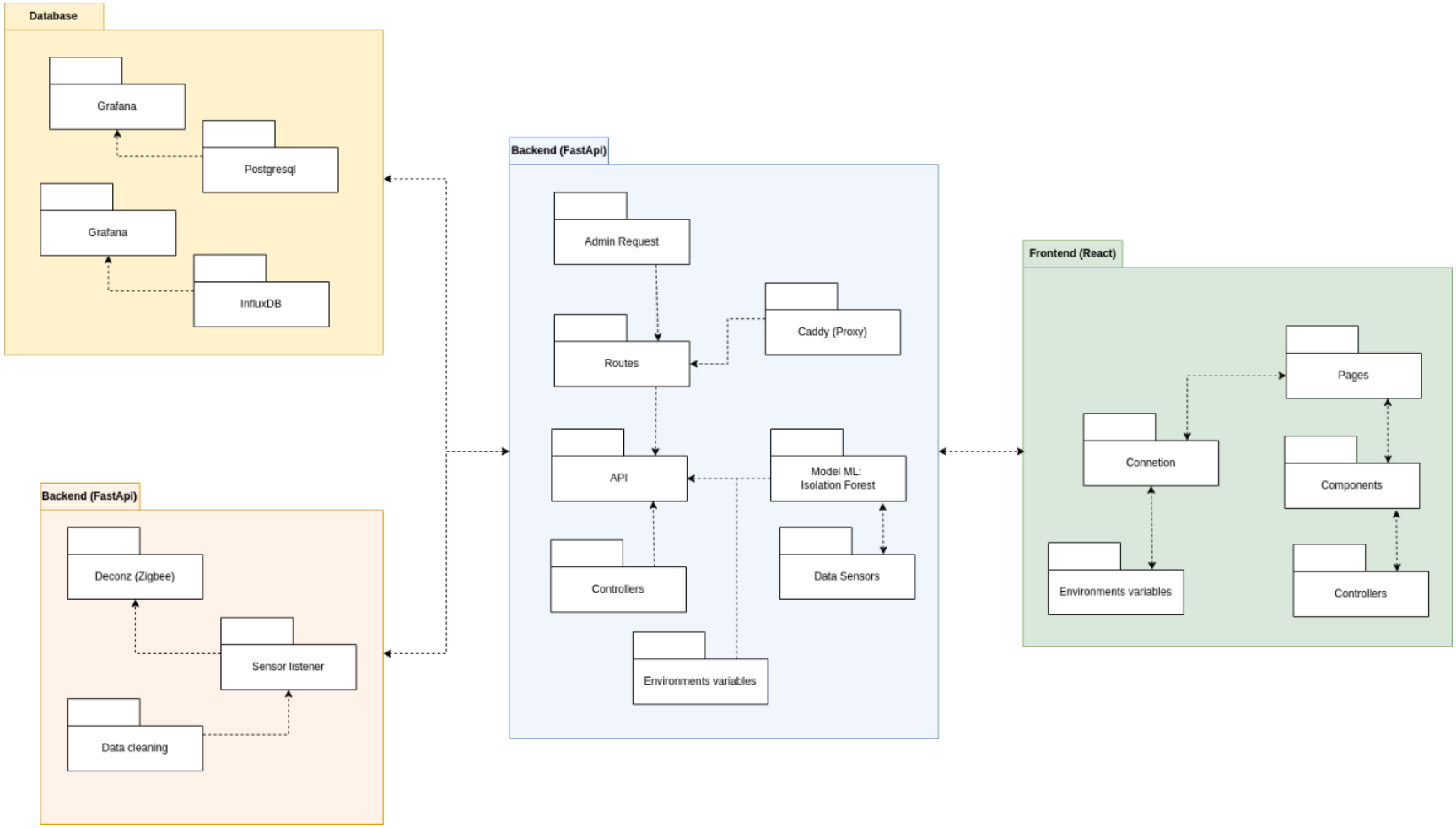
Diagramas de clases.



4.3.2.2.4. Diagramas de paquetes. En la Figura 35 se ilustra los componentes que conforman el sistema y como se orquestan entre sí.

Figura 35.

Diagrama de paquetes.



El primer sistema implementado (color naranja) se encarga de mantener una comunicación con los dispositivos ZigBee, registrar y depurar los eventos para enviarlos al segundo servidor.

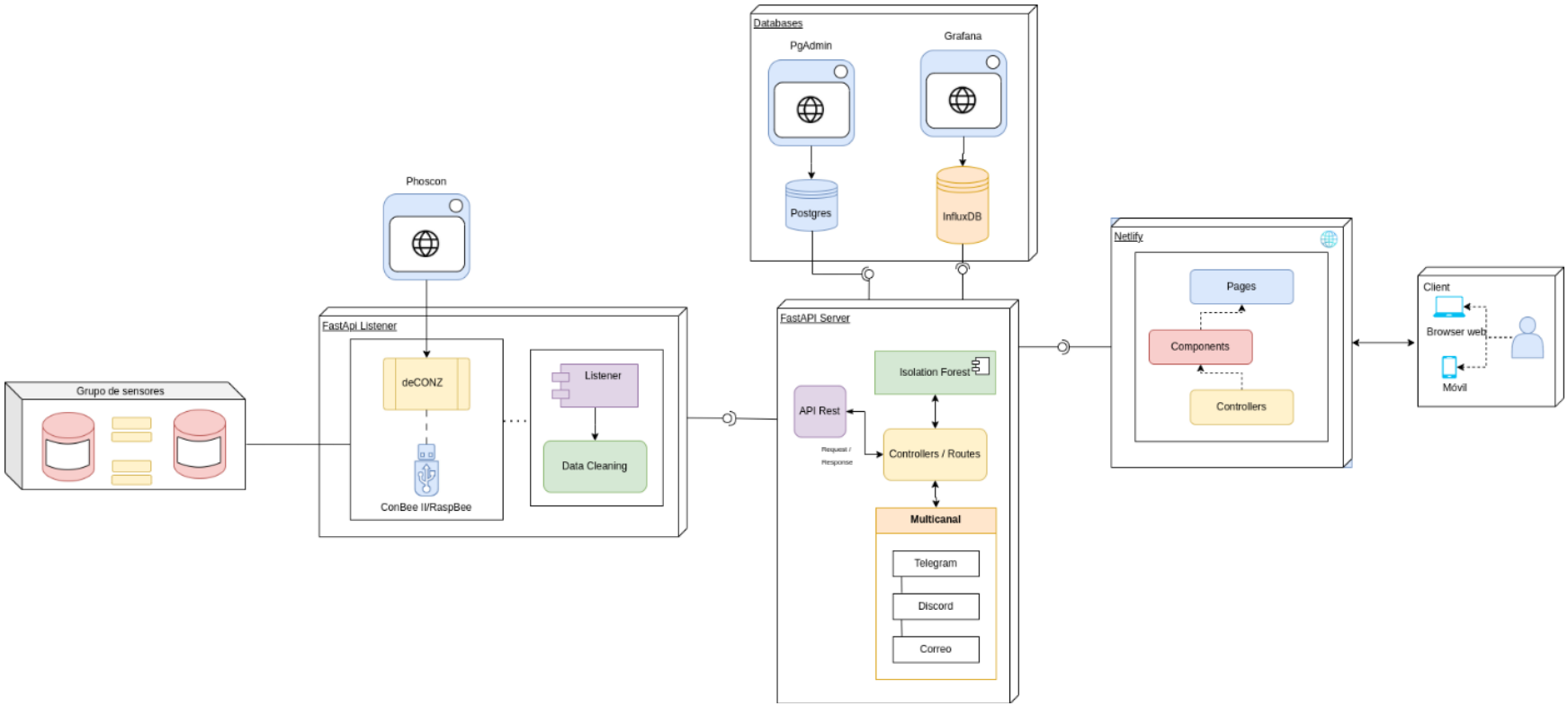
El segundo servidor (color azul) maneja las diferentes de rutas de comunicación, el uso del modelo de ML (Isolation Forest) y la conexiones a la base de datos con PostgreSQL e InfluxDB (color amarillo).

Finalmente, el servidor React (color verde) desplegado en Netlify y constituido por componentes que conjuntamente conforman paginas mantiene comunicación con el servidor principal (color azul) mediante el proxy de Caddy, lo que permite el acceso al sistema y la modificación de las preferencias del sistema.

***4.3.2.2.5. Diagrama de arquitectura*:** La Figura 36 presenta una visión general del sistema, compuesto por lo componentes importantes del sistema desde la comunicación de los sensores hasta el usuario que recibe la notificación.

Figura 36.

Diagrama de arquitectura.



A continuación, la Tabla 17 detalla el comportamiento de cada componente del diagrama mostrado en la Figura 36.

Tabla 17.

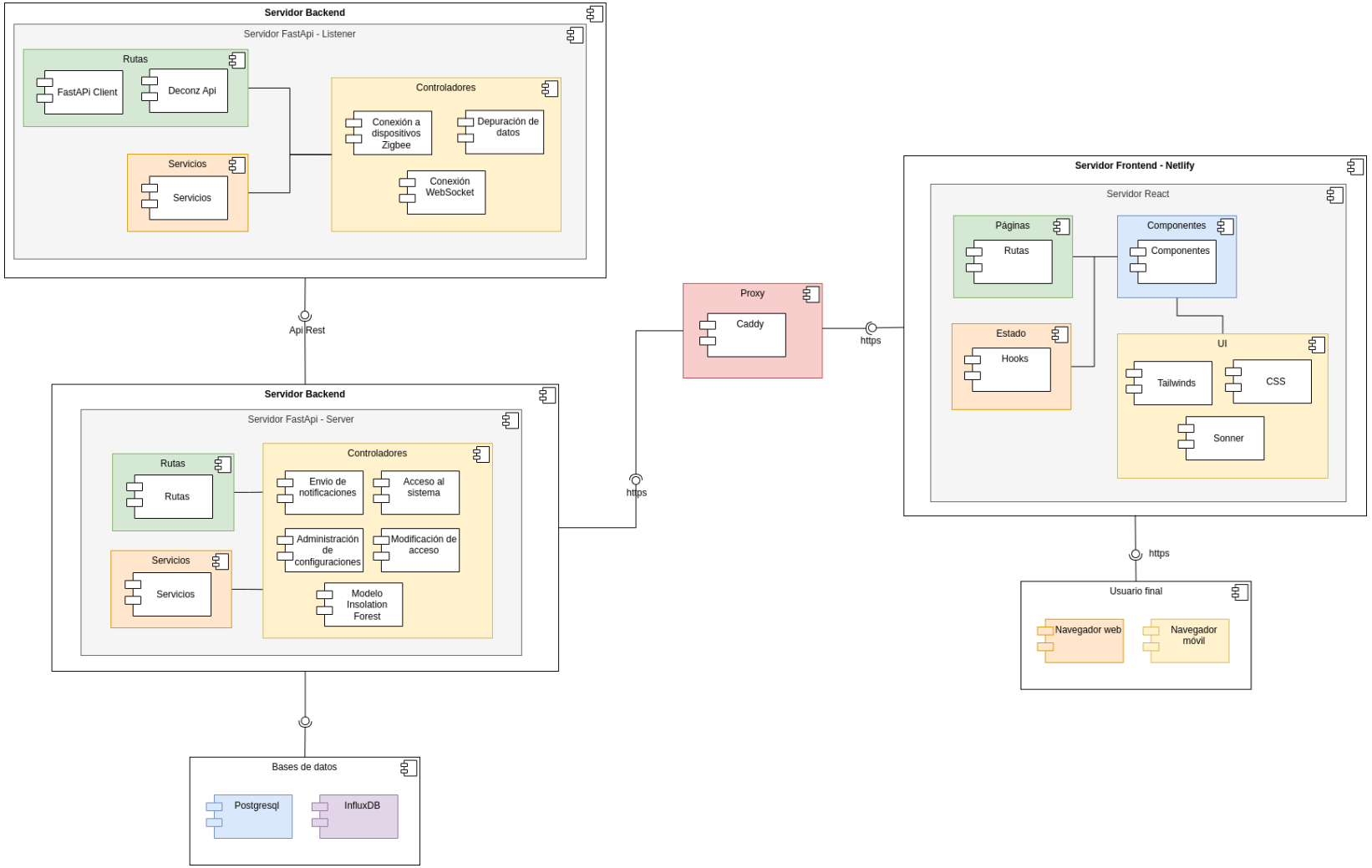
Componentes de la arquitectura del sistema.

N°	Nombre	Descripción	Tipo/Categoría
1	Grupo de sensores	Conjunto de sensores IoT que recopilan datos del entorno y los transmiten para su procesamiento	Hardware/IoT
2	FastAPI Listener	Servicio que recibe y procesa datos de los sensores, incluyendo componentes como deCONZ, ConBee II/Rasperry Pi y funciones de limpieza de datos	Microservicio/Backend
3	FastAPI Server	Servidor principal que maneja la lógica principal, APIs REST, enrutamiento y comunicación multicanal (Telegram, Discord, Correo)	Microservicio/Backend
4	Databases	Sistema de bases de datos que incluye PgAdmin (PostgreSQL) para datos estructurados y Grafana con InfluxDB para métricas y visualizaciones	Almacenamiento/Datos
5	Netlify	Plataforma de hosting para la aplicación frontend que incluye páginas web, componentes de interfaz y controladores	Frontend/Hosting
6	Cliente/Usuario	Usuario final que interacciona con el sistema a través del navegador web accediendo a la interfaz desplegada en Netlify	Actor/Usuario

4.3.2.2.6. Diagrama de componentes. La Figura 37 presenta la arquitectura modular del sistema mediante el diagrama de componentes, detallando la interconexión entre los componentes del sistema y el flujo de datos.

Figura 37.

Diagrama de componentes.



El primer componente (Servidor Listener) mantiene comunicación directa con los sensores mediante conexión WebSocket a dispositivos ZigBee, además en este microservicio se realiza la depuración de datos recibidos.

El segundo componente (Servidor) gestiona él envió de notificaciones multicanal, administra la configuración del sistema, controla el acceso e implementa el modelo de ML (Isolation Forest) para detectar anomalías.

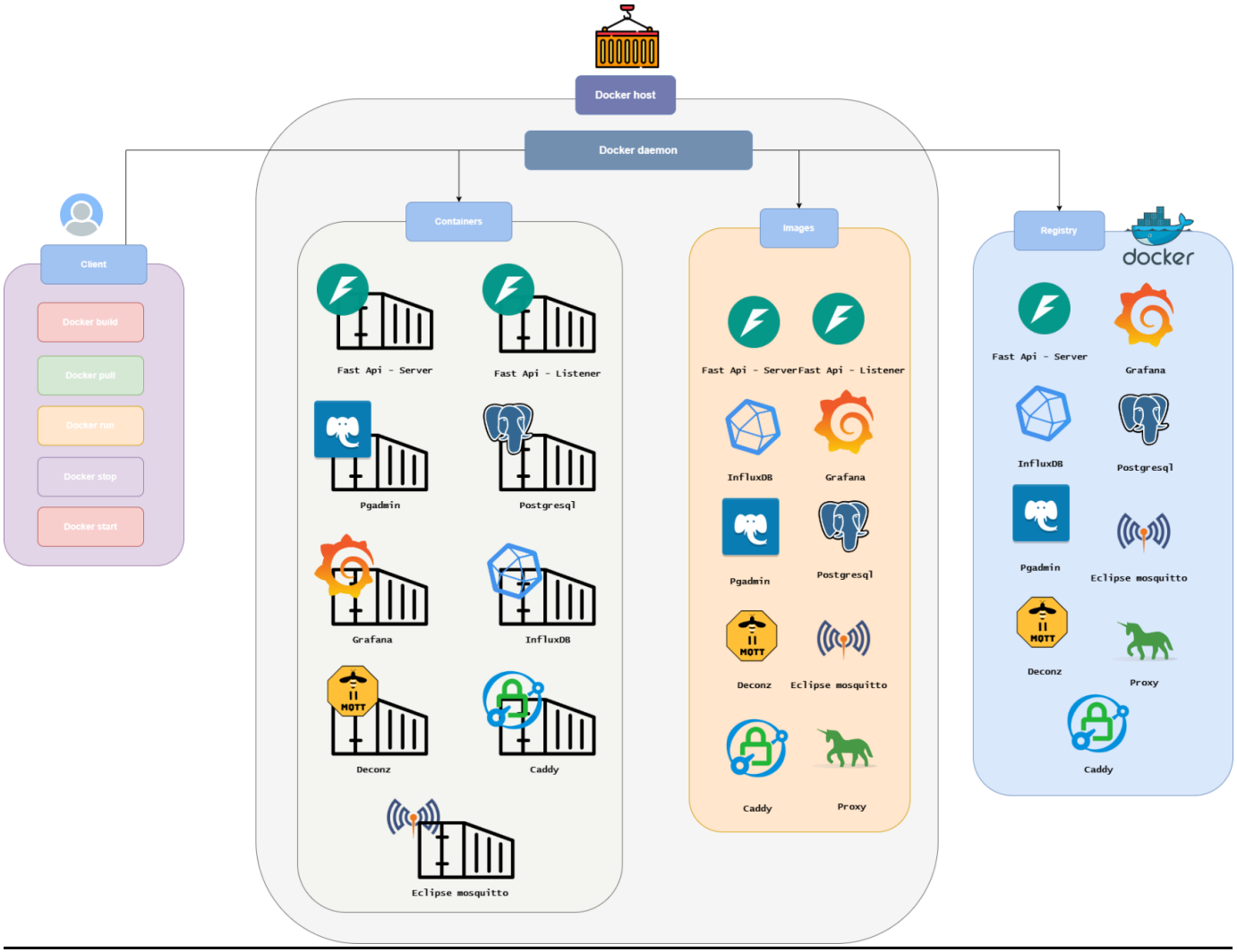
El tercer componente (React) proporciona una interfaz web responsiva accesible desde computadoras y dispositivos móviles, en ella se puede cambiar las preferencias del sistema.

Los últimos componentes como las bases de datos almacenan la configuración del sistema (PostgreSQL) y los datos de los sensores (InfluxDB), por otra parte, Caddy actúa como un Proxy reverso para mantener una comunicación entre el frontend desplegado en Netlify hasta el servidor Backend.

4.3.2.2.7. Diagramas de Dockers: La Figura 38 muestra la arquitectura del proyecto desde la perspectiva de Docker:

Figura 38.

Diagrama de Dockers.



Docker permite gestionar nueve contenedores: dos propios basados en FastAPI que contienen la lógica del proyecto y siete adicionales que proporcionan servicios de apoyo, estos contenedores se despliegan desde imágenes Docker que son paquetes inmutables conteniendo código, dependencias, librerías, configuración y el sistema operativo base, las imágenes se almacenan y distribuyen a través de un registry, el cual es un repositorio centralizado que actúa como biblioteca de imágenes Docker.

4.3.2.3. Fase 3: Codificación.

4.3.2.3.1. Programación e implementación. Dentro de esta etapa se desarrolla la implementación de las funcionalidades del sistema empleando las tecnologías definidas y manteniendo como marco los requerimientos funcionales y no funcionales, además del diseño del sistema realizado mediante la diagramación mostrada en la Fase 2: Diseño.

Dentro de Figura 39, Figura 40, Figura 41, Figura 42, se pueden apreciar la organización del proyecto, detallando la distribución de archivos, directorios y componentes que conforman el sistema.

La Tabla 18 presenta una descripción de los componentes principales del sistema.

Tabla 18.

Componentes principales del sistema empleado.

Nombre	Descripción
Cortex	Interfaz web desarrollada con React a la que accede el administrador para modificar las preferencias del sistema.
Deconz	Servicio dedicado para establecer la conexión con los dispositivos ZigBee.
NeuroAlert	Directorio que contiene dos microservicios: el primero escucha la transmisión de datos de los sensores y procesa eventos; el segundo actúa como servidor principal, se conecta a las bases de datos, mantiene comunicación con el frontend desplegado y administra accesos y configuraciones del sistema.
Failover	Servicio desarrollado para mantener una conexión redundante entre los diferentes métodos de comunicación a internet (Ethernet, Wifi, Datos móviles)

Figura 39.

Distribución de carpetas dentro de vscode.

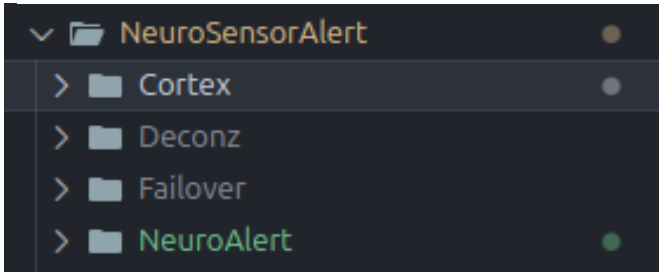


Figura 41.

Carpeta NeuroAlert - Vscode.

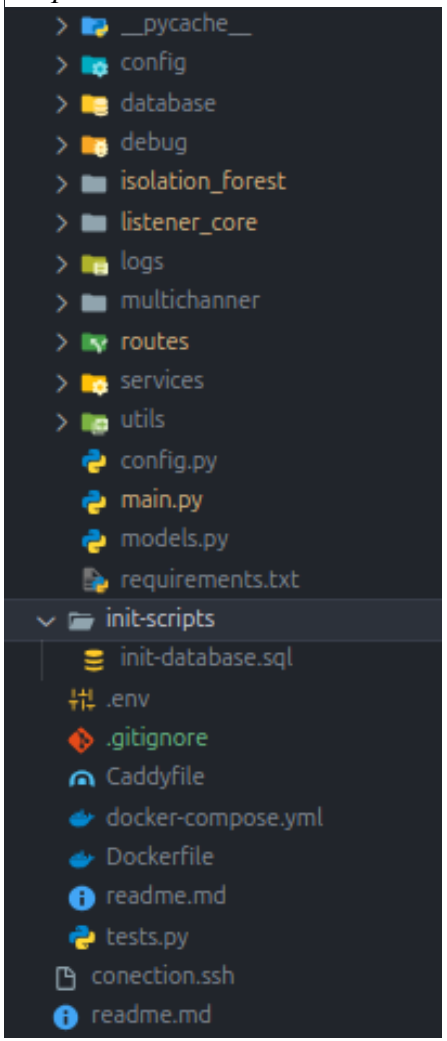


Figura 40.

Carpeta Cortex - Vscode.

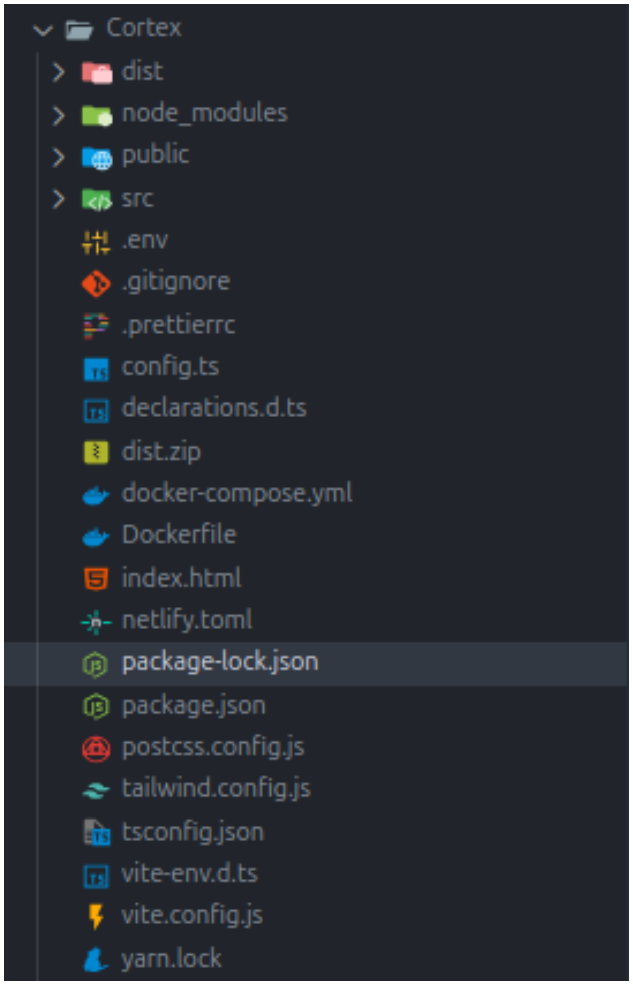
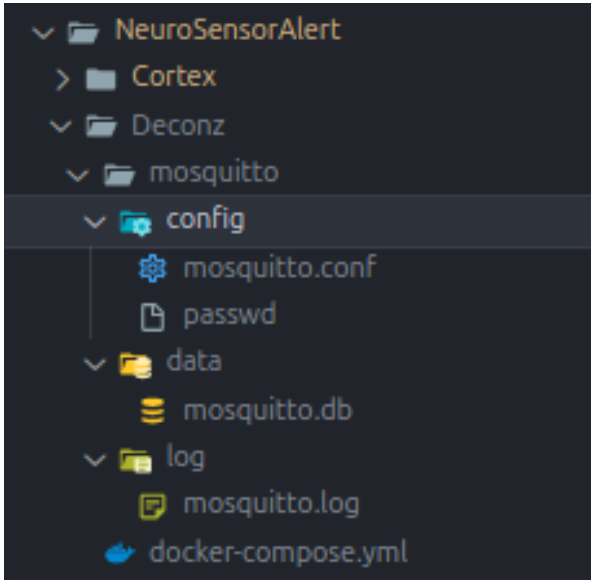


Figura 42.

Carpeta Deconz - Vscode.



La Tabla 19 describe los archivos más importantes del directorio trabajado en vscode.

Tabla 19.

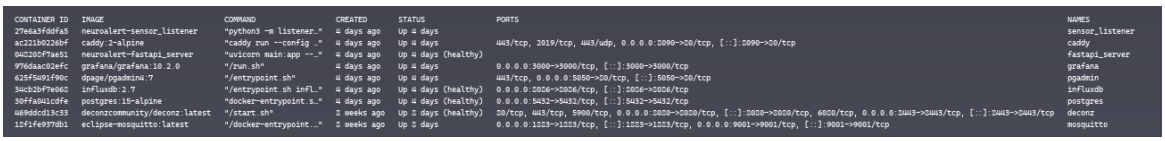
Estructura y descripción de los archivos del proyecto.

N°	Nombre	Descripción
1	node_modules	Directorio que contiene todas las dependencias y paquetes de Node.js instalados para el proyecto.
2	dist	Carpeta de distribución que almacena los archivos compilados y optimizados listos para producción.
3	public	Directorio de archivos estáticos públicos accesibles directamente por el navegador.
4	.env	Archivo de variables de entorno que almacena configuraciones sensibles del proyecto.
5	.gitignore	Archivo que especifica qué archivos y directorios debe ignorar Git en el control de versiones.
6	.prettierrc	Archivo de configuración para Prettier que define las reglas de formateo del código.
7	config.ts	Archivo TypeScript que contiene las configuraciones principales de la aplicación.
8	docker-compose.yml	Archivo que define y orquestas múltiples contenedores Docker para el proyecto.
9	Dockerfile	Archivo de instrucciones para construir la imagen Docker de la aplicación.
10	netlify.toml	Archivo de configuración específico para el despliegue y configuración en Netlify.
11	package-lock.json	Archivo que bloquea las versiones exactas de las dependencias de npm instaladas.
12	package.json	Archivo principal que define metadatos, dependencias y scripts del proyecto Node.js.
13	vite.config.js	Archivo de configuración para Vite, herramienta de construcción y desarrollo frontend.
14	yarn.lock	Archivo que asegura instalaciones consistentes de dependencias cuando se usa Yarn.
15	tailwind.config.js	Archivo de configuración para el framework CSS Tailwind personalizado al proyecto.
16	caddyfile	Archivo de configuración para el servidor web Caddy y sus reglas de proxy.
17	readme.md	Documento en formato Markdown que contiene información y documentación del proyecto.

4.3.2.3.2. Despliegue en entorno de producción. En la Figura 43 se puede observar cada uno de los contenedores en ejecución dentro del Rasperry Pi 5:

Figura 43.

Contenedores ejecutados con Docker.



Se puede observar que los contenedores tienen un ID de identificación, una imagen, el comando de ejecución, la fecha de creación, su estado actual, los puertos que ocupan y el nombre del contenedor.

Por otro lado, el despliegue de la interfaz gráfica del proyecto denominada Cortex y desarrollada con React, se realiza mediante Netlify para garantizar una comunicación directa y accesible con los servidores backend desde cualquier parte del mundo, se puede encontrar la explicación de uso en el **Anexo 1. Manual de usuario.** En las Figura 44, Figura 45, Figura 46, Figura 47, Figura 48 se muestra la apariencia del sistema.

- **Acceso al sistema:** https://cortexalert.netlify.app/

Figura 44.

Inicio de sesión en Cortex.

Figura 45.

Cambio de preferencias en Cortex.

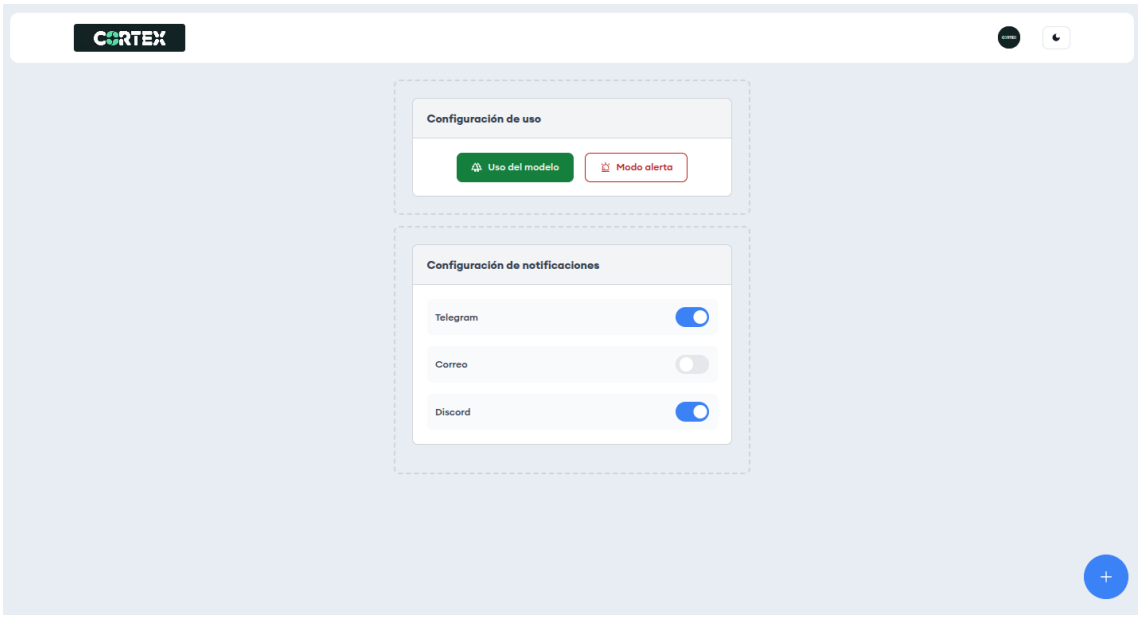


Figura 46.

Cambio de preferencias desde modo oscuro.

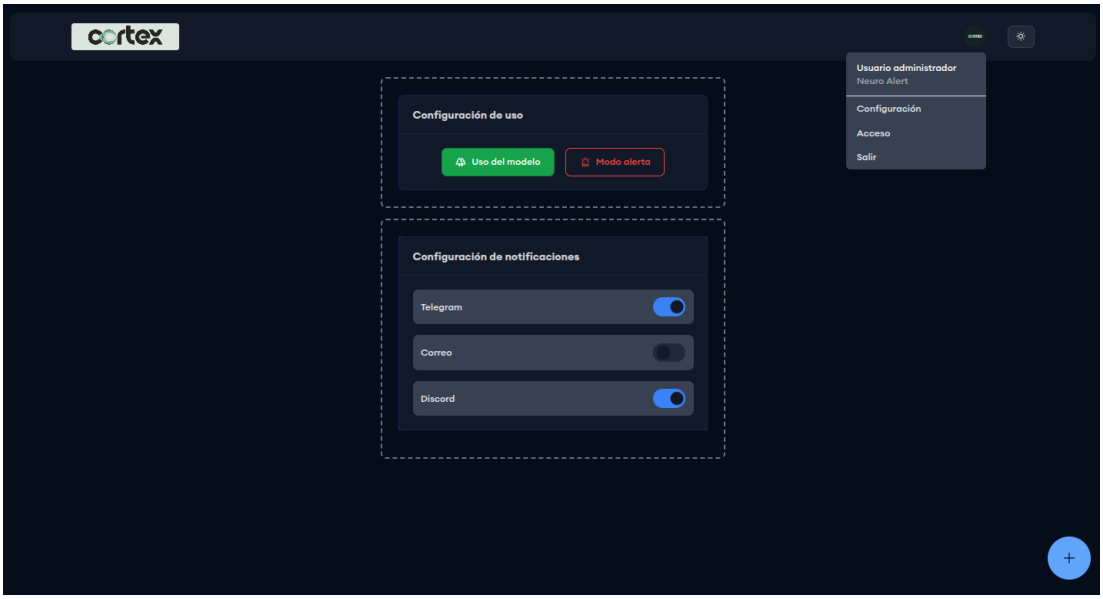


Figura 47.

Cambio de preferencias para dispositivos móviles.

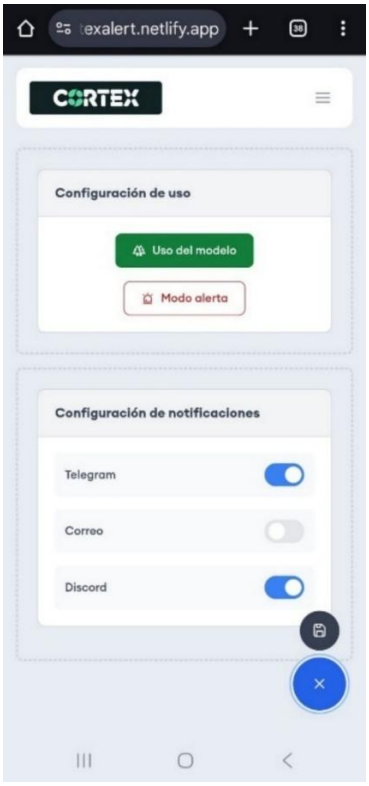
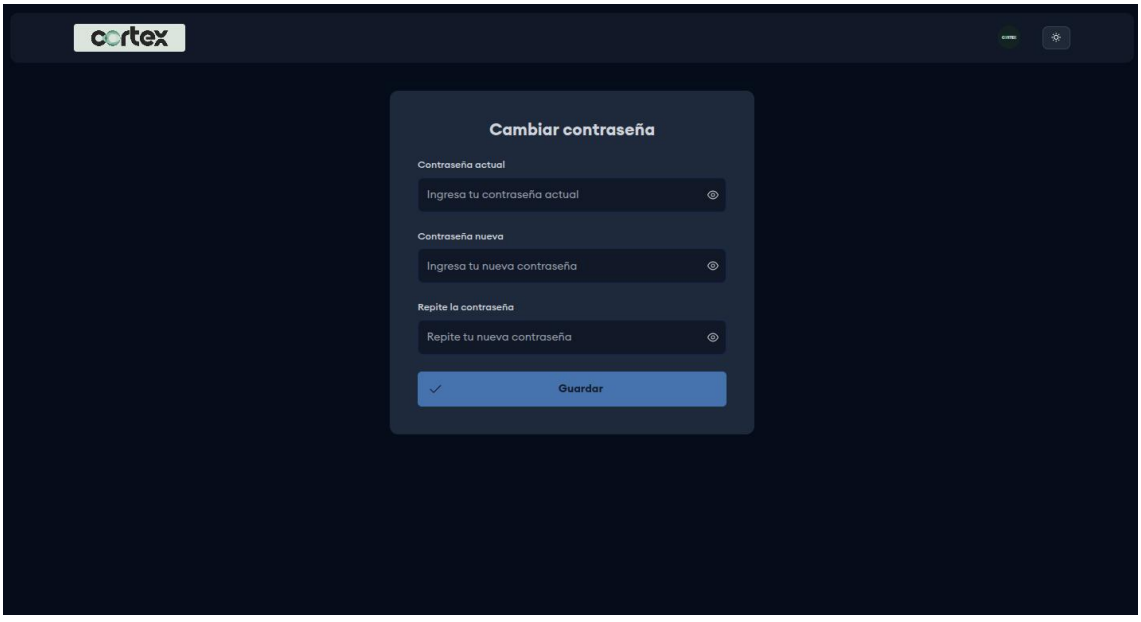


Figura 48.

Cambio de clave de acceso al sistema.



4.4. Evaluación integral del sistema

4.4.1. Desempeño de modelos de detección por ML

En las Figura 49, Figura 50 se presentan los resultados del proceso GridSearchCV para la optimización de hiperparámetros, donde los algoritmos evaluados obtuvieron accuracies promedio de 0.8158, 0.7772 y 0.6621 en validación cruzada (CV=3), así se destaca que EllipticEnvelope superó el umbral objetivo de 0.8, logrando un balance óptimo entre la detección de anomalías, la reducción de falsos positivos y la robustez ante outliers en los datos.

Figura 49.

Comparación de algoritmos de ML.

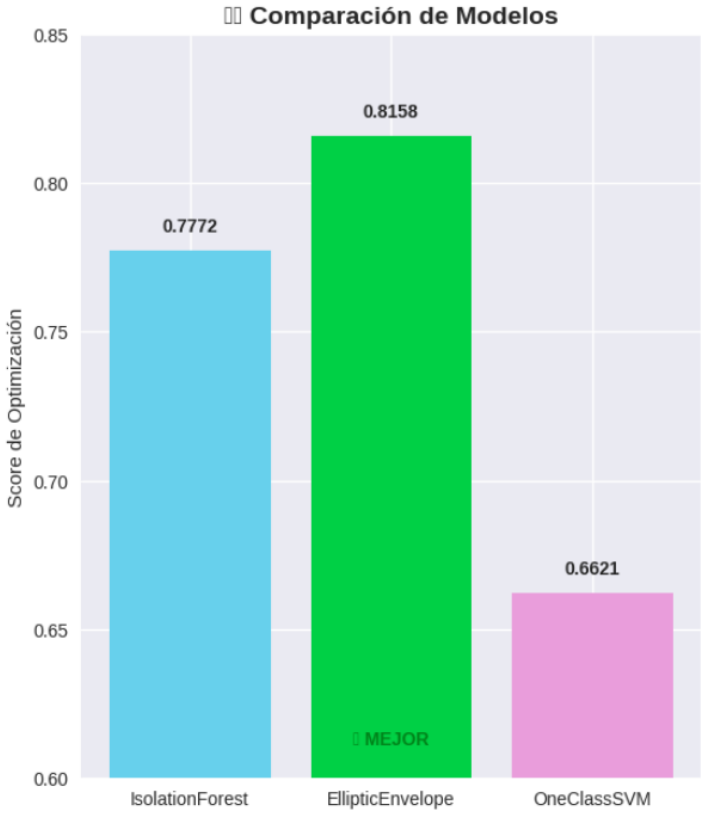
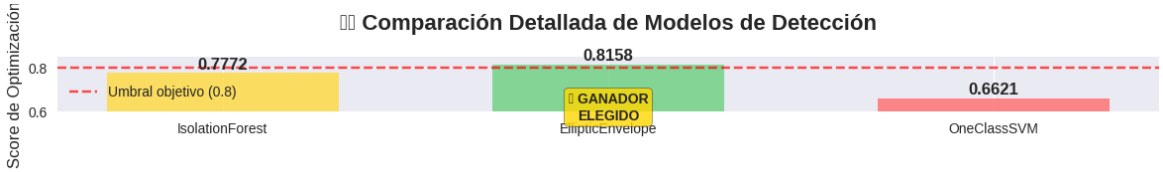


Figura 50.

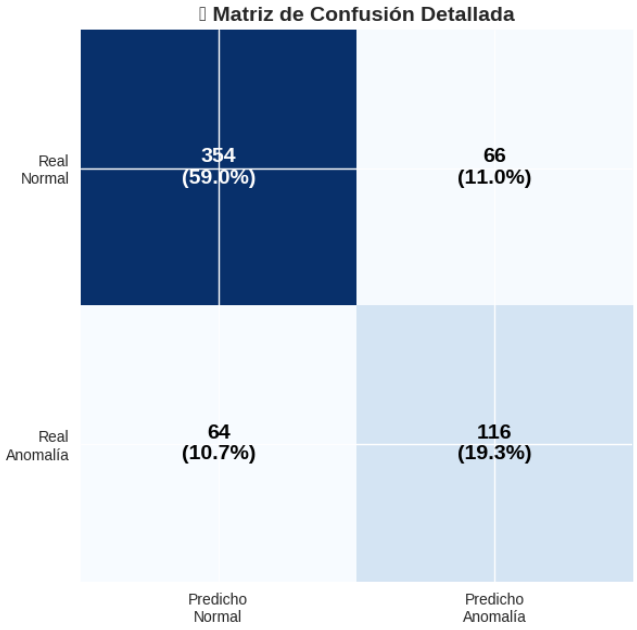
Resultados de algoritmos empleados.



La matriz de confusión presentada en la Figura 51 permitió evaluar el rendimiento del modelo de clasificación binaria, entre casos normales y anomalías.

Figura 51.

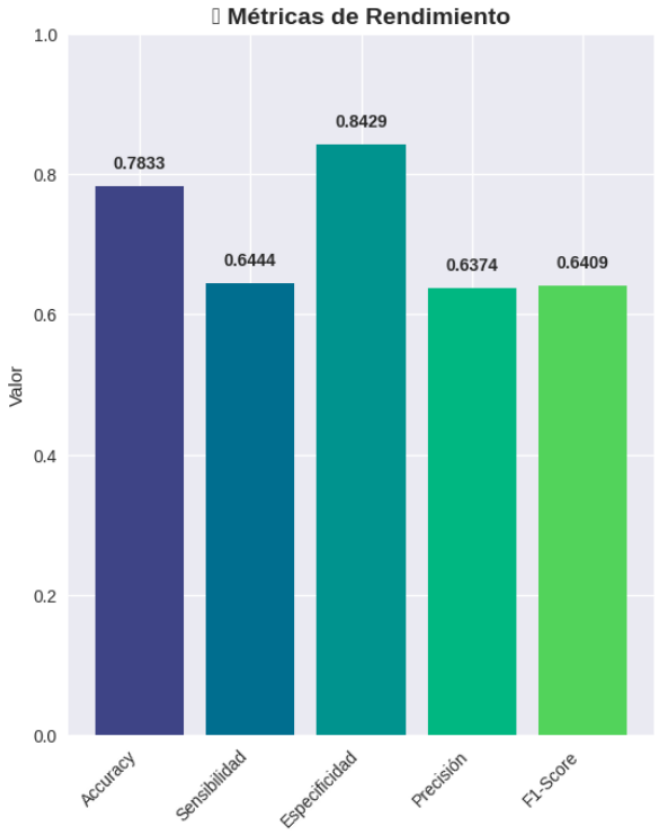
Resultados de matriz de confusión.



La matriz de confusión muestra que de 600 casos evaluados 470 fueron clasificados correctamente (354 normales y 116 anomalías), mientras que 130 resultaron en errores (66 falsos positivos y 64 falsos negativos), bajo estos resultados se calculan diferentes métricas específicas de rendimiento que se pueden observar en la Figura 52.

Figura 52.

Métricas de rendimiento de modelo empleado.



Según la matriz de confusión, el modelo alcanzó una precisión (Accuracy) del 78.33% en todas las predicciones, con un recall del 64.44% para la detección de anomalías reales, una especificidad del 84.29% para identificar correctamente casos normales, una precisión del 63.74% para las alertas de anomalía reales y un F1-Score que refleja un balance general entre precisión y sensibilidad.

4.4.2. Validación experimental con patrones de intrusión

Validación del modelo ante diferentes patrones de comportamiento de intrusos. Para validar el sistema de seguridad, se utiliza el modelo bajo diversos patrones de comportamiento, los cuales se detallan en la

.

Tabla 20.

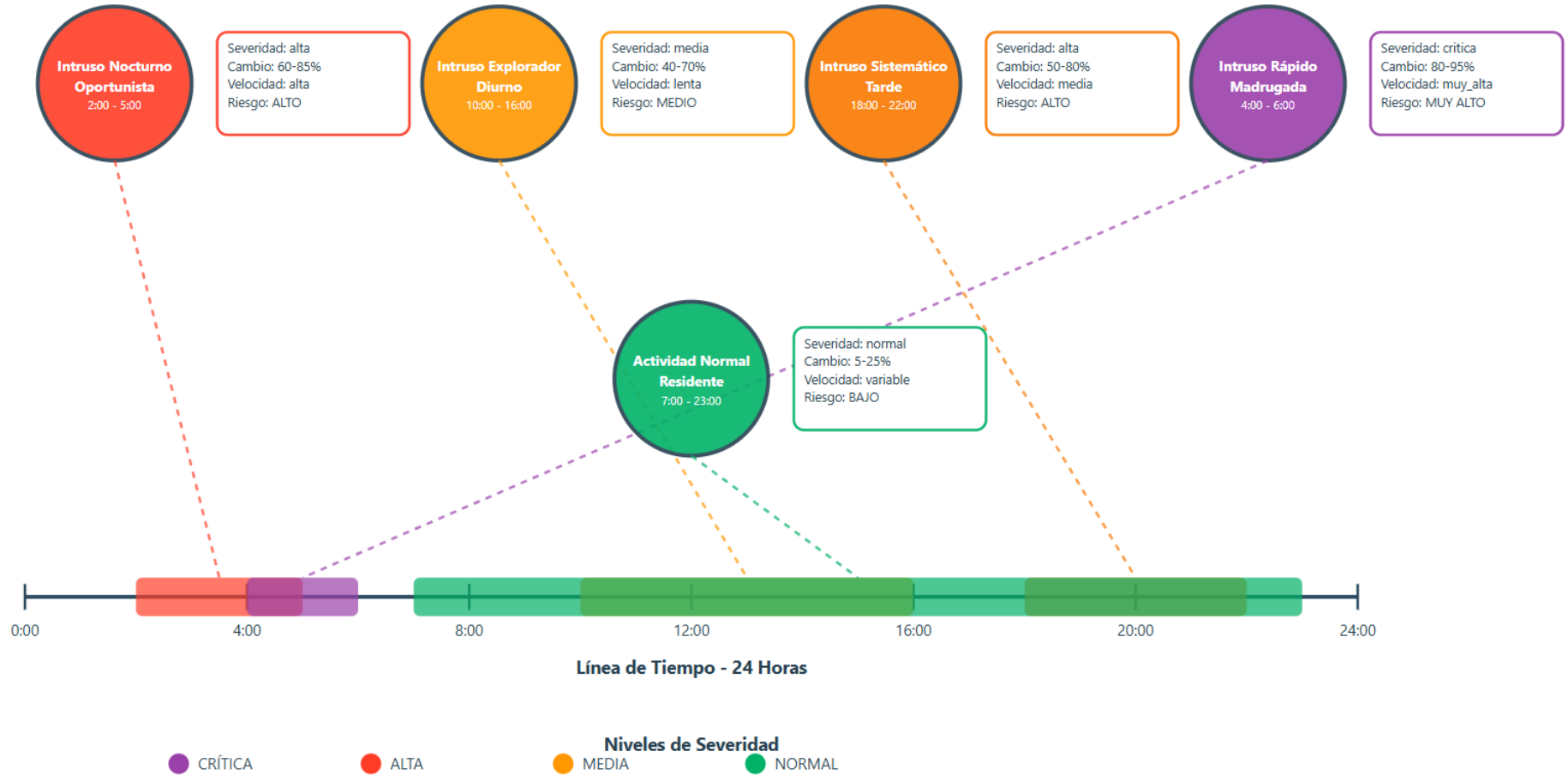
Patrones de comportamiento de intrusos.

Patrones de intrusos	Descripción
Intruso nocturno oportunista	Actúa entre las 02:00 y 05:00 con movimientos rápidos y oportunistas.
Intruso explorador diurno	Realiza reconocimientos lentos y discretos durante las horas laborales diurnas.
Intruso sistemático tarde	Realiza búsquedas metódicas en el horario vespertino (tarde-noche).
Intruso rápido madrugador	Accede rápidamente, con conocimiento previo de la vivienda en horas de la madrugada.
Patrón normal: Actividad normal residente.	Comportamiento típico de los habitantes.

En la Figura 53 se presenta el comportamiento de los diferentes patrones a evaluados.

Figura 53.

Comportamiento de los diferentes patrones empleados.



Los diferentes patrones son evaluados bajo las siguientes graficas, mostradas en la Figura 54, las cuales se explican en la Tabla 21.

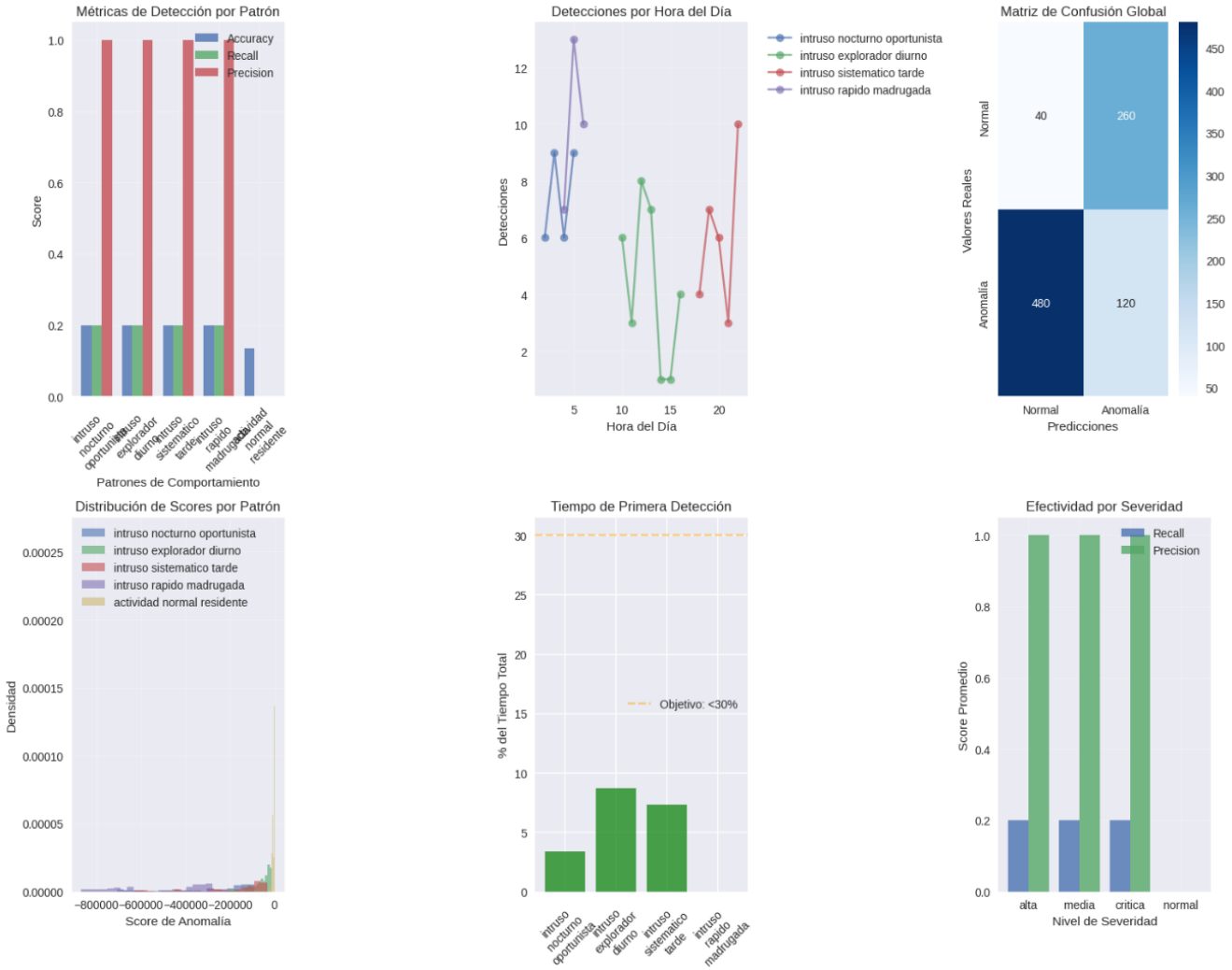
Tabla 21.

Propósito de las gráficas de análisis.

Gráfica	Propósito
Métricas de detección por patrón	Muestra las puntuaciones de precisión y recall para diferentes patrones de comportamiento, ayudando a evaluar el rendimiento del modelo por categoría.
Detecciones por hora del día	Ilustra la cantidad de detecciones de intrusos por hora, destacando patrones temporales de actividad según el tipo de intruso.
Matriz de confusión global	Representa las predicciones frente a los valores reales (normal vs. anómalo), mostrando la precisión general del modelo en clasificar datos.
Distribución de scores por patrón	Muestra la distribución de los scores de anomalía para cada patrón, ayudando a entender la variabilidad de los datos por categoría.
Tiempo de primera detección	Indica el porcentaje del tiempo total hasta la primera detección por tipo de intruso, comparado con un objetivo del 30%.
Efectividad por severidad	Evalúa las puntuaciones de precisión y recall según el nivel de severidad (alta, media, crítica, normal), mostrando el rendimiento por categoría de riesgo.

Figura 54.

Evaluación con diferentes patrones de comportamiento.



La Tabla 22 resume los resultados que se obtuvieron tras cada grafica.

Tabla 22.

Resumen de resultados por graficas de evaluación.

Gráfica	Qué muestra	Interpretación principal
Métricas de detección por patrón	Accuracy, Recall y Precision para cada tipo de intruso y actividad normal.	Alta precisión (cerca de 1), pero recall bajo, se detectan pocas anomalías en proporción a las reales.
Detecciones por hora del día	Número de detecciones a distintas horas del día por tipo de intruso.	Cada patrón de intruso tiene horarios característicos (ej. nocturno oportunista de madrugada, explorador diurno de día).
Matriz de confusión global	Comparación entre valores reales y predicciones (Normal vs Anomalía).	Buen desempeño general, pero aún hay falsos negativos (120 anomalías no detectadas) y falsos positivos (40 normales mal detectados).
Distribución de scores por patrón	Distribución de los scores de anomalía según el patrón.	Los intrusos tienen scores distintos a la actividad normal → útil para diferenciarlos, aunque hay cierta superposición.
Tiempo de primera detección	% del tiempo total que tarda en detectarse cada intruso.	Todos los intrusos se detectan antes del 30% del tiempo, cumpliendo el objetivo.
Efectividad por severidad	Recall y Precision según nivel de severidad (alta, media, crítica, normal).	La precisión es excelente en todos los niveles, pero el recall sigue siendo bajo (~0.2).

4.4.3. Análisis de los resultados de validación del modelo.

En la Tabla 23 se presentan diferentes métricas de evaluación por cada patrón de comportamiento analizado.

Tabla 23.

Métricas de evaluación por cada patrón de comportamiento.

Patrón	Severidad	Accuracy	Precision	Recall	Resultado
Intruso nocturno oportunista	Alta	0.200	1.000	0.200	Detecta solo 20% de casos. Detección temprana, pero deficiente recall.
Intruso explorador diurno	Media	0.200	1.000	0.200	Bajo recall, aunque logra detección en <10% del tiempo.
Intruso sistemático tarde	Alta	0.200	1.000	0.200	Misma problemática: recall bajo, aunque detección temprana.
Intruso rápido madrugada	Crítica	0.200	1.000	0.200	Preocupante porque es el patrón más crítico, pero con el mismo recall bajo.

Actividad normal residente	Normal	0.133	0.000	0.000	Alto nivel de falsos positivos, el sistema no diferencia adecuadamente actividad normal.

El modelo actual presenta limitaciones críticas para la seguridad doméstica, ya que ignora el 80% de las amenazas, incluidas las de severidad alta y crítica, clasifica erróneamente el 86.7% de los comportamientos normales como anomalías, generando un alto índice de falsas alarmas (86% de actividad normal), aunque muestra precisión en ciertos aspectos, falla en su objetivo principal de detectar de manera confiable todos los intentos de intrusión y no distingue correctamente los patrones temporales.

4.4.4. Documentación audiovisual del sistema

En el siguiente anexo se puede acceder a video de prueba del sistema dentro del entorno doméstico **Anexo 3. Demostración de funcionamiento del sistema.**

4.5. Discusión

En la revisión de literatura se identifican distintos enfoques para la implementación de sistemas de seguridad doméstica basados en IoT y ML, esto permite contrastar las mejores prácticas y tecnologías utilizadas, por ejemplo, (Long et al., 2021) presentan un análisis de técnicas de detección de intrusos en redes IoT donde destacan el uso de algoritmos como SVM y Random Forest por su capacidad de clasificación precisa, por otro lado (Alani & Awad, 2023) proponen un sistema de dos capas que combina aprendizaje profundo con IoT para lograr una alta precisión y bajo tiempo de procesamiento. Esta diferencia metodológica muestra cómo algunos estudios priorizan la eficiencia computacional (Long et al., 2021) , mientras que otros se enfocan en la robustez del modelo ante amenazas complejas (Alani & Awad, 2023).

La infraestructura IoT basada en ZigBee con sensores Aqara, coordinador ConBee II y un Rasperry Pi 5 como Gateway permitió la recolección de datos en tiempo real con baja latencia y bajo consumo energético, lo que concuerda con hallazgos de (Orfanos et al., 2023) sobre WSN e IoT quienes subrayan la comunicación de estos dispositivos en automatización doméstica, además las propiedades de eficiencia de ZigBee son consistentes con observaciones de (Casilari et al., 2010) que señala que su diseño está orientado al bajo

consumo. Por otra parte, la escalabilidad del sistema actual constituye un aspecto critico que requiere atención, coincidiendo con preocupaciones planteadas por (Ali & Zorlu Partal, 2022) que sugieren la necesidad de implementar redes hibridas para soportar mayores cargas de dispositivos como ZigBee y LoRa.

La arquitectura de microservicios desarrollada mantiene una comunicación en tiempo real mediante WebSocket y gestiona múltiples canales de notificación, alineada con las tendencias actuales que propone marcos basados en IA para detección de intrusiones en tiempo real y mitigación automatizada (Obaid et al., 2014), además el enfoque de procesamiento local en Rasperry Pi con comunicación continua es consistente con estudios que sugieren arquitecturas centralizadas donde los dispositivos IoT envían datos de sensores al nodo central para procesamiento con modelos de detección de anomalías (Stolojescu-Crisan et al., 2021), por otra parte, la interfaz web desarrollada junto con la arquitectura dockerizada establece una base que podría expandirse con aplicaciones móviles nativas y dashboards predictivos, los cuales se subrayan en los avances de detección de anomalías para IoT y las perspectivas futuras del campo (Kumar et al., 2019).

La validación del modelo de ML revelo resultados mixtos, el desempeño del algoritmo Elliptic Envelope con un accuracy promedio de 0.85% en validación cruzada supera los umbrales establecidos, sin embargo, contrasta con los ensayos de diferentes patrones de intrusos donde se ven problemas de clasificación, esto se alinea con investigaciones en detección de anomalías en hogares inteligentes donde se recalca que una detección confiable de anomalías requiere un dataset robusto y algoritmos especializados (Zohourian et al., 2023) sugiriendo que las limitaciones observadas podrían deberse a deficiencias en los datos o falta de optimización para patrones específicos de intrusión, por otra parte la alta tasa de clasificación del 86.7% de comportamiento normales como anomalías representa un desafío operacional que va en sintonía con observaciones de (Ramotsoela et al., 2022) que propone modelos hibrido de ML y DL para superar estas limitaciones en sistemas de detección de intrusos, así mismo las recomendaciones de implementar ensambles de algoritmos que combinen algoritmos como RF, KNN, SVM y más están respaldadas por estudios actuales que demuestran mejoras significativas en métricas de clasificación (Almotairi et al., 2024).

En el desarrollo de software la claridad en la definición de requisitos y la estructuración de la arquitectura son fundamentales para garantizar un proceso eficiente, es por eso que en este trabajo la elicitación de requisitos realizada en la Fase 1 y los diagramas UML elaborados en la Fase 2 proporcionaron una base sólida para alinear el desarrollo con los objetivos establecidos, estos resultados coinciden con (Montgomery et al., 2022) quienes concuerdan en que una documentación clara y bien definida reduce los errores en el desarrollo y optimiza los tiempos de implementación.

La decisión de utilizar Docker como herramienta de contenedorización en el desarrollo de software fue clave para asegurar un despliegue consistente en los entornos de desarrollo y producción, además Docker Compose simplificó la orquestación multicontenedor mejorando la escalabilidad y permitiendo adaptaciones rápidas a cambios en los requisitos, sin embargo su curva de aprendizaje inicial representó un reto, en este trabajo Docker eliminó las discrepancias entre configuraciones y previno errores de implementación lo cual coincide con (Boettiger, n.d.) donde se destaca que Docker simplifica la configuración de entornos y minimiza inconsistencias entre desarrollo y producción.

5. CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se implementó un sistema de seguridad doméstica inteligente, funcional, autónomo y escalable, basado en una infraestructura IoT, el cual permitió la integración de hardware, software y energía autónoma, asegurando el monitoreo en tiempo real, la gestión remota y la continuidad operativa del entorno doméstico.

El modelo de ML Elliptic Envelope alcanzó un desempeño sólido en la fase de validación en comparación a sus homólogos Isolation Forest, OneClass SVM, con una precisión promedio del 85%, estos resultados evidencian que los algoritmos de detección de anomalías pueden constituirse en una herramienta útil para la identificación de comportamientos atípicos en entornos de seguridad doméstica.

La arquitectura basada en microservicios y notificaciones en múltiples canales demostró ser práctica y flexible, ya que facilitó la interoperabilidad entre diversos dispositivos y ofreció al usuario un control centralizado desde su interfaz web Cortex y en tiempo real del sistema de seguridad.

El sistema desarrollado constituye un aporte tecnológico relevante y replicable, ya que integra de manera efectiva hardware, software, energía autónoma y ML en un entorno real. Aunque el modelo actual presenta limitaciones para despliegues críticos, la arquitectura, la documentación técnica y los componentes implementados ofrecen una base sólida para futuras investigaciones orientadas a optimizar la precisión, robustez y adaptabilidad de los sistemas inteligentes de seguridad IoT en el hogar.

5.2. Recomendaciones

Se recomienda continuar explorando nuevas tecnologías IoT y algoritmos de detección de intrusos que puedan optimizar aún más la eficiencia del sistema de seguridad doméstica, así sería útil realizar pruebas comparativas entre diferentes protocolos de comunicación y métodos de detección para identificar posibles mejoras en la cobertura, precisión y consumo energético del sistema, esto permitirá adaptar mejor el sistema a diversos entornos residenciales y aumentar su fiabilidad.

Para futuras implementaciones se recomienda evaluar la escalabilidad del protocolo ZigBee con la implementación de mesh networking que permita soportar una mayor cantidad de dispositivos sin comprometer el rendimiento, así mismo sería necesario desarrollar un sistema de redundancia con múltiples Gateway para garantizar la alta disponibilidad del sistema central.

Los algoritmos implementados pueden mejorarse significativamente extendiendo el periodo de entrenamiento a al menos seis meses, lo cual permite capturar variaciones estacionales y patrones de comportamiento más complejos, además se sugiere emplear un ensamble de algoritmos que combine Isolation Forest con LSTM Y One-Class SVM para reducir falsos positivos y mejorar la precisión en la detección de anomalías complejas.

Para ampliar la accesibilidad del sistema, se sugiere desarrollar una aplicación móvil nativa que complemente la interfaz web actual, proporcionando notificaciones push y control remoto más intuitivo donde se implemente un dashboard predictivo que muestre tendencias basadas en análisis de datos históricos que permita a los usuarios anticipar patrones y optimizar la configuración del sistema.

Se recomienda entrenar modelos especializados, dividiéndolos en submodelos según la severidad o el patrón de actividad de los intrusos, implementando un sistema de aprendizaje continuo que se adapte activamente a los patrones de comportamiento de los miembros de la vivienda, de esta forma incrementaría la cantidad y diversidad de los datos de entrenamiento para mantener una amplia gama de variaciones de comportamientos normales e intrusivos, además de establecer protocolos de recolección de datos más rigurosos que capturen con precisión la variabilidad real de los patrones de comportamiento con el fin aumentar la precisión en la identificación de anomalías.

6. BIBLIOGRAFÍA

Adhikary, A., Halder, S., Bose, R., Panja, S., Halder, S., Pratihar, J., & Dey, A. (2024). Design and Implementation of an IOT-based Smart Home Automation System in Real World Scenario. *EAI Endorsed Transactions on Internet of Things*, *10*. https://doi.org/10.4108/eetiot.6201

Ahmed, S. H., & Zeebaree, S. (2021). A survey on security and privacy challenges in smarthome based IoT. *International Journal of Contemporary Architecture*, *8*(2), 489–510.

Alani, M. M., & Awad, A. I. (2023). An Intelligent Two-Layer Intrusion Detection System for the Internet of Things. *IEEE Transactions on Industrial Informatics*, *19*(1), 683–692. https://doi.org/10.1109/TII.2022.3192035

Albornoz, R., & Soto, E. (2018). *Estudio del Estándar Zigbee*. http://profesores.elo.utfsm.cl/~agv/elo322/1s18/projects/reports/Zigbee.pdf

Ali, A. I., & Zorlu Partal, S. (2022). Development and performance analysis of a ZigBee and LoRa-based smart building sensor network. *Frontiers in Energy Research*, *10*, 933743. https://doi.org/10.3389/FENRG.2022.933743/BIBTEX

Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. *Systems Science & Control Engineering*, *12*(1), 2321381. https://doi.org/10.1080/21642583.2024.2321381

Boettiger, C. (n.d.). *An introduction to Docker for reproducible research*. Retrieved August 24, 2025, from https://gist.github.com/samth/9641364

Calatayud Sánchez, H. (2021). *Integración de un sistema de vigilancia mediante Telegram en Home Assistant*. Universitat Politècnica de València.

Casilari, E., Cano-García, J. M., & Campos-Garrido, G. (2010). Modeling of Current Consumption in 802.15.4/ZigBee Sensor Motes. *Sensors 2010, Vol. 10, Pages 5443-5468*, *10*(6), 5443–5468. https://doi.org/10.3390/S100605443

Chen, Z., Peng, L., & Fu, H. (2022a). Isolated forest-based ZigBee Device Identification Using Adaptive Filter Coefficients. *2022 7th International Conference on Computer and Communication Systems (ICCCS)*, 715–720. https://doi.org/10.1109/ICCCS55155.2022.9846363

Chen, Z., Peng, L., & Fu, H. (2022b). Isolated forest-based ZigBee Device Identification Using Adaptive Filter Coefficients. *2022 7th International Conference on Computer and Communication Systems (ICCCS)*, 715–720. https://doi.org/10.1109/ICCCS55155.2022.9846363

Dougal Ferguson, A. S. P. G. N. C. M. B. (2025). Identificación de pacientes con riesgo de cáncer de próstata mediante espectroscopia infrarroja por transformada de Fourier y aprendizaje automático. *International Society for Clinical Spectroscopy*.

Dresden elektronik ingenieurtechnik GmbH. (2025). *deCONZ – Software for ConBee II and RaspBee*. Https://Phoscon.de/En/Conbee2/Software#deconz.

Duque Quevedo Odalys Rashel. (2024). *PROPUESTA DE MEJORAS DE ALERTAS DE SEGURIDAD DE DISPOSITIVOS DE IOT MEDIANTE INTELIGENCIA ARTIFICIAL*.

elektronik ingenieurtechnik GmbH, D. (2025). deCONZ – Software for ConBee II and RaspBee. In *https://phoscon.de/en/conbee2/software#deconz*.

Farizi, W. S. Al, Hidayah, I., & Rizal, M. N. (2021). Isolation Forest Based Anomaly Detection: A Systematic Literature Review. *2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, 118–122. https://doi.org/10.1109/ICITACEE53184.2021.9617498

Group, P. G. D. (2025). PostgreSQL – The World's Most Advanced Open Source Relational Database. In *https://www.postgresql.org/*.

Hillar, G. C. (2017). *MQTT Essentials-A lightweight IoT protocol*. Packt Publishing Ltd.

INEC. (2024). *Estadísticas de Seguridad Integral*. https://www.ecuadorencifras.gob.ec/justicia-y-crimen/

InfluxData. (2025a). *InfluxDB – Time Series Database*. Https://Www.Influxdata.Com/.

InfluxData. (2025b). InfluxDB – Time Series Database. In *https://www.influxdata.com/*.

Iturbe-Araya, J. I., & Rifà-Pous, H. (2025). Enhancing unsupervised anomaly-based cyberattacks detection in smart homes through hyperparameter optimization.

International Journal of Information Security, *24*(1), 45. https://doi.org/10.1007/s10207-024-00961-6

Jacinto, A., Chico, A., Oviedo Galarza, J. E., Vicente, R., Paredes, G., Isaac, W., & Cruz, M. (2024). *Electronic prototype for the internet of things in smart homes*. *9*(2), 2024. https://doi.org/10.5281/zenodo.10951513

Khalid, B., Khan, A. M., Akram, M. U., & Batool, S. (2019). Person Detection by Fusion of Visible and Thermal Images Using Convolutional Neural Network. *2019 2nd International Conference on Communication, Computing and Digital Systems (C-CODE)*, 143–148. https://doi.org/10.1109/C-CODE.2019.8680991

Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, *6*(1), 1–21. https://doi.org/10.1186/S40537-019-0268-2/FIGURES/9

Li, S., Xu, L. Da, & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, *17*(2), 243–259. https://doi.org/10.1007/s10796-014-9492-7

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Long, J., Fang, F., & Luo, H. (2021). A Survey of Machine Learning-based IoT Intrusion Detection Techniques. *Proceedings - 2021 IEEE 6th International Conference on Smart Cloud, SmartCloud 2021*, 7–12. https://doi.org/10.1109/SMARTCLOUD52277.2021.00009

Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N. O., Guarnizo, J. D., & Elovici, Y. (2017). *Detection of Unauthorized IoT Devices Using Machine Learning Techniques*. https://arxiv.org/abs/1709.04647

Montgomery, L., Fucci, D., Bouraffa, A., Scholz, L., & Maalej, W. (2022). Empirical research on requirements quality: a systematic mapping study. *Requirements Engineering*, *27*(2), 183–209. https://doi.org/10.1007/S00766-021-00367-Z/TABLES/7

Obaid, T., Rashed, H., -Elnour, A. A., Rehan, M., Muhammad Saleh, M., & Tarique, M. (2014). Zigbee Technology and its Application in Wireless Home Automation Systems: A Survey. *International Journal of Computer Networks & Communications*, *6*(4), 115–131. https://doi.org/10.5121/IJCNC.2014.6411

Orfanos, V. A., Kaminaris, S. D., Papageorgas, P., Piromalis, D., & Kandris, D. (2023). A Comprehensive Review of IoT Networking Technologies for Smart Home Automation Applications. *Journal of Sensor and Actuator Networks 2023, Vol. 12, Page 30*, *12*(2), 30. https://doi.org/10.3390/JSAN12020030

Pedregosa, F. V. G. G. A. M. V. T. B. G. O. B. M. P. P. W. R. D. V. V. J. P. A. C. (2011). Scikit-learn: Machine Learning in Python. *Http://Jmlr.Org/Papers/V12/Pedregosa11a.Html*, 2825–2830.

pgAdmin Development Team. (2025). pgAdmin – PostgreSQL Tools. In *https://www.pgadmin.org/*.

Project, D. (2023). Debian “bookworm” Release Information. In *https://www.debian.org/releases/bookworm/*.

Ramírez, S. (2025). FastAPI – The modern, fast (high-performance) web framework for building APIs with Python. In *https://fastapi.tiangolo.com/*.

Ramotsoela, D., Abu-Mahfouz, A. M., Silva, B., Mujtaba Qureshi, U., Umair, Z., Butt, N., Shahid, A., Naseer Qureshi, K., Haider, S., Osman Ibrahim, A., Binzagr, F., & Arshad, N. (2022). Intelligent Deep Learning for Anomaly-Based Intrusion Detection in IoT Smart Home Networks. *Mathematics 2022, Vol. 10, Page 4598*, *10*(23), 4598. https://doi.org/10.3390/MATH10234598

Rosas Cruz, A. (2021). Sistema de automatización con IoT para el control de una vivienda con Nodemcu ESP8266. In *Sui Umsa*.

Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, *99*, 107810. https://doi.org/https://doi.org/10.1016/j.compeleceng.2022.107810

Samani, E., Khaledian, P., Aligholian, A., Papalexakis, E., Cun, S., Nazari, M. H., & Mohsenian-Rad, H. (2020). Anomaly Detection in IoT-Based PIR Occupancy Sensors to Improve Building Energy Efficiency. *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 1–5. https://doi.org/10.1109/ISGT45199.2020.9087681

Singh, S., Fernandes, S. V., Padmanabha, V., & Rubini, P. E. (2021). MCIDS-Multi Classifier Intrusion Detection system for IoT Cyber Attack using Deep Learning

algorithm. *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 354–360. https://doi.org/10.1109/ICICV50876.2021.9388579

Stolojescu-Crisan, C., Crisan, C., & Butunoi, B. P. (2021). An IoT-Based Smart Home Automation System. *Sensors (Basel, Switzerland)*, *21*(11), 3784. https://doi.org/10.3390/S21113784

Taiwo, O., Ezugwu, A. E., Oyelade, O. N., & Almutairi, M. S. (2022). Enhanced intelligent smart home control and security system based on deep learning model. *Wireless Communications and Mobile Computing*, *2022*(1), 9307961. https://doi.org/10.1155/2022/9307961

V, P., Sumaiya Thaseen, I., Reddy Gadekallu, T., K. Aboudaif, M., & Abouel Nasr, E. (2021). Robust Attack Detection Approach for IIoT Using Ensemble Classifier. *Computers, Materials & Continua*, *66*(3), 2457–2470. https://doi.org/10.32604/cmc.2021.013852

Velastegui Morales Jhoselyn Lizeth, & Cuzme Rodríguez Fabián Geovanny. (2024). *SISTEMA DE DETECCIÓN DE INTRUSOS APLICANDO INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE ATAQUES EN UNA RED DE SENSORES INALÁMBRICOS (WSN)*. http://repositorio.utn.edu.ec/handle/123456789/15493

Vera Romero, C. A., Barbosa Jaimes, J. E., & Pabón González, D. C. (2017). La Tecnología ZigBee estudio de las características de la capa física. *Scientia et Technica Año XXII*, *Vol. 22, No. 3,*(ZigBee Technology study of the characteristics of the physical layer). http://www.redalyc.org/pdf/849/84954626002.pdf

Xu, D., Wang, Y., Meng, Y., & Zhang, Z. (2017). An Improved Data Anomaly Detection Method Based on Isolation Forest. *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, *2*, 287–291. https://doi.org/10.1109/ISCID.2017.202

Xu, H., Pang, G., Wang, Y., & Wang, Y. (2023). Deep Isolation Forest for Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering*, *35*(12), 12591–12604. https://doi.org/10.1109/TKDE.2023.3270293

Yalçnkaya, F., Aydilek, H., Erten, M. Y., & \.Inanç, N. (2020). IoT based smart home testbed using MQTT communication protocol. *International Journal of*

Engineering Research and Development, *12*(1), 317–324. https://doi.org/10.29137/umagd.654056

Zohourian, A., Dadkhah, S., Neto, E. C. P., Mahdikhani, H., Danso, P. K., Molyneaux, H., & Ghorbani, A. A. (2023). IoT Zigbee device security: A comprehensive review. *Internet of Things*, *22*, 100791. https://doi.org/10.1016/J.IOT.2023.100791

7. ANEXOS

7.1. Anexo 1. Manual de usuario

Sistema Cortex empleado para modificar las preferencias del comportamiento del sistema NeuroAlert

Manual de usuario

1. Introducción

El presente documento constituye una guía para comprender el funcionamiento del sistema Cortex con la finalidad del aprovechamiento de sus funcionalidades.

2. Objetivo

El objetivo de este documento es detallar las funcionalidades del sistema con el fin de dejar esclarecido el comportamiento del sistema.

3. Alcance

El presente documento está dirigido al entendimiento del sistema y a cualquier usuario interesado en usar un sistema parecido al desarrollado en el presente trabajo.

4. Requisitos previos

Los requisitos previos son los siguientes:

- Dispositivo móvil o computador
- Acceso a internet
- Navegador web (Google Chrome, Microsoft Edge, Firefox, Brave)
- Acceso a: https://cortexalert.netlify.app/

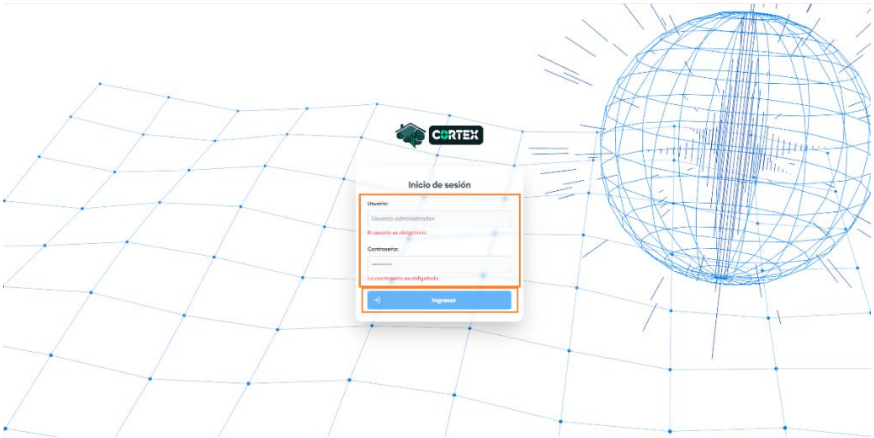
5. Funcionalidades

Las siguientes funcionalidades están descritas para el uso exclusivo del administrador.

5.1. Iniciar sesión

Para el ingreso al sistema es necesario:

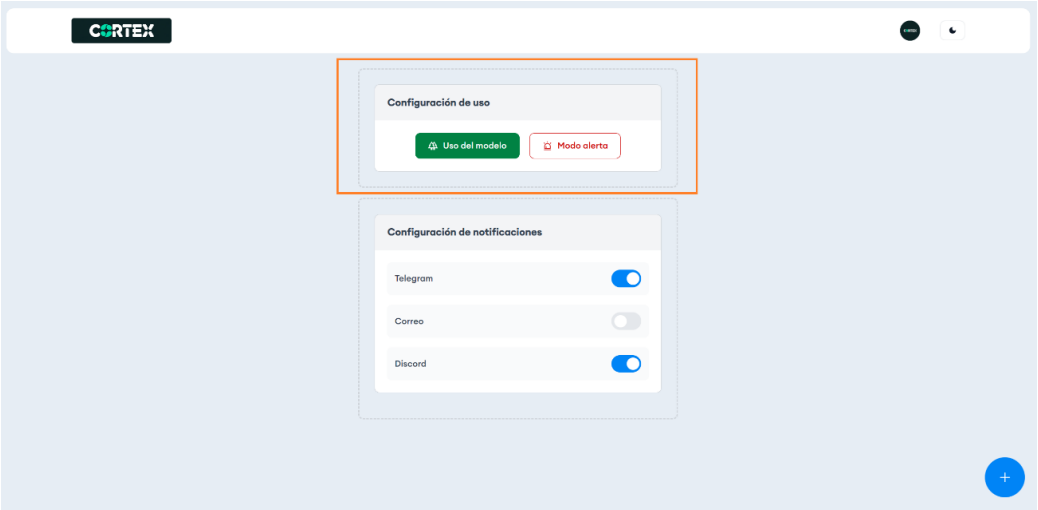
- Ingresar con las credenciales del administrador
- Seleccionar el botón “Ingresar”



5.2. Cambio del comportamiento del sistema

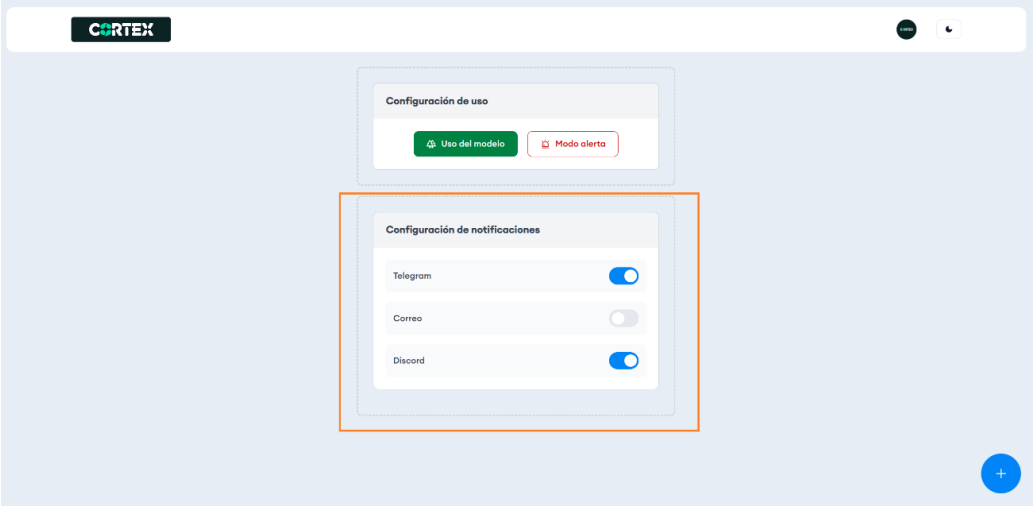
Cortex permite funcionar de dos formas diferentes.

- **Uso del modelo:** Los eventos obtenidos de los sensores se validarán primero por el modelo Insolación Forest y luego se decidirá si enviar o no una notificación.
- **Modo alerta:** Cada evento obtenido será enviado por notificación, especialmente útil cuando se requiere el monitoreo exhaustivo de casa.



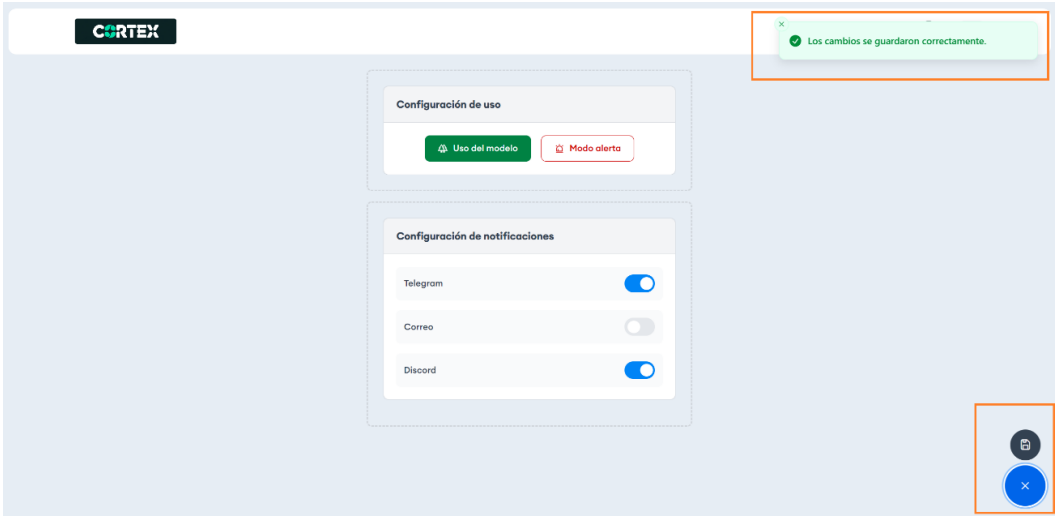
5.3 Configuración de notificaciones

Se puede activar o desactivar los diferentes medios de comunicación (Telegram, Correo, Discord)



5.4 Guardar preferencias de aplicación

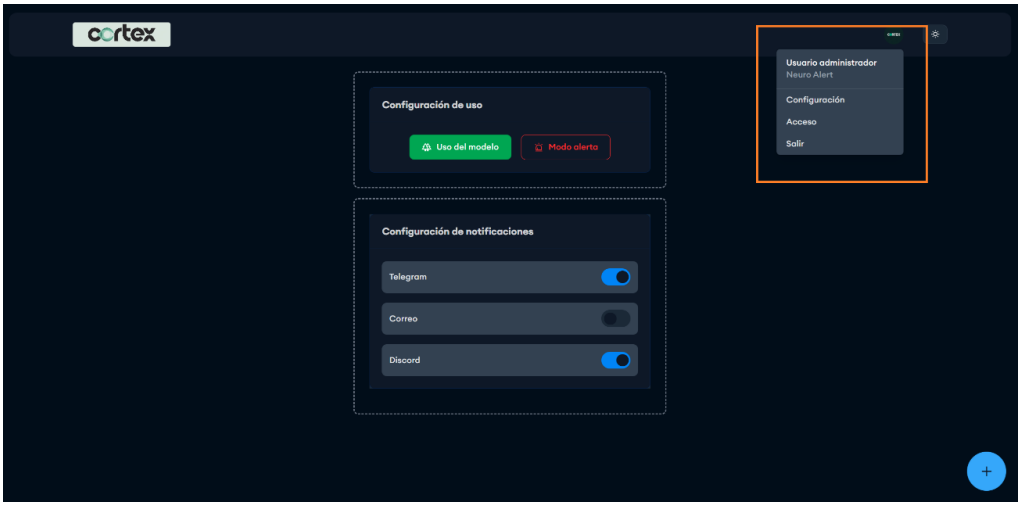
Los cambios se aplican después de seleccionar el botón azul y presionar guardar, posterior a ello se mostrará una notificación de confirmación tras la respuesta del servidor que actualiza la base de datos PostgreSQL.



5.5. Menú de opciones

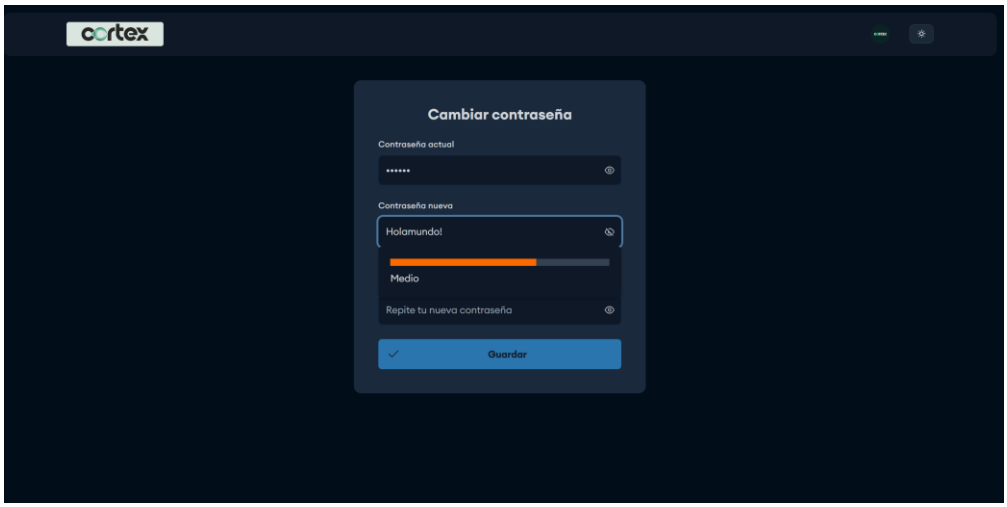
El menú de opciones permite navegar entre las diferentes opciones del sistema, se detallan las mismas a continuación:

- **Configuración:** Permite el modificar las preferencias del sistema.
- **Acceso:** Pantalla de cambio de clave.
- **Salir:** Cierra la sesión activa del sistema.



5.6. Cambio de clave

La siguiente interfaz permite cambiar la clave de acceso al sistema, se solicita la contraseña actual seguidamente de la nueva y una confirmación, posterior a ellos se debe seleccionar el botón “Guardar”, esta validación se realiza desde otro servidor y si todo es correcto se altera la clave guardada en PostgreSQL



7.2. Anexo 2. Repositorio del proyecto - GitHub

En la siguiente sección se expone el código fuente de todo el proyecto, mismo que puede ser recreado fácilmente bajo el entorno de Docker y el uso de sensores de la misma tecnología.

- **Enlace a repositorio del proyecto dentro de GitHub *(Repositorio Privado):*** https://github.com/NeuroAlertHome/System-Alert-Home

- **Solicitud de acceso a:** alertasdavcasa@gmail.com

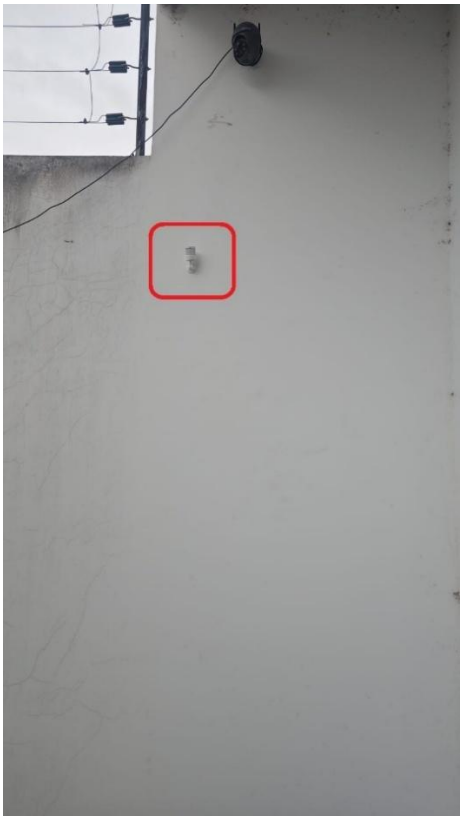
7.3. Anexo 3. Demostración de funcionamiento del sistema.

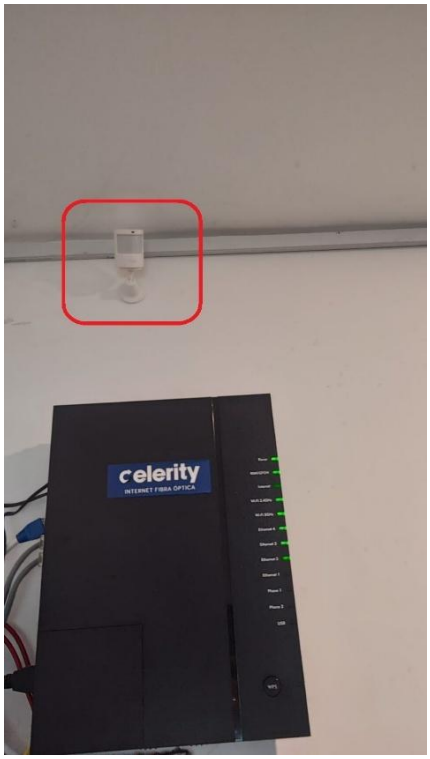
En la siguiente sección se prueba el funcionamiento del sistema desarrollado, mediante grabaciones que demuestran las diferentes funcionalidades implementadas:

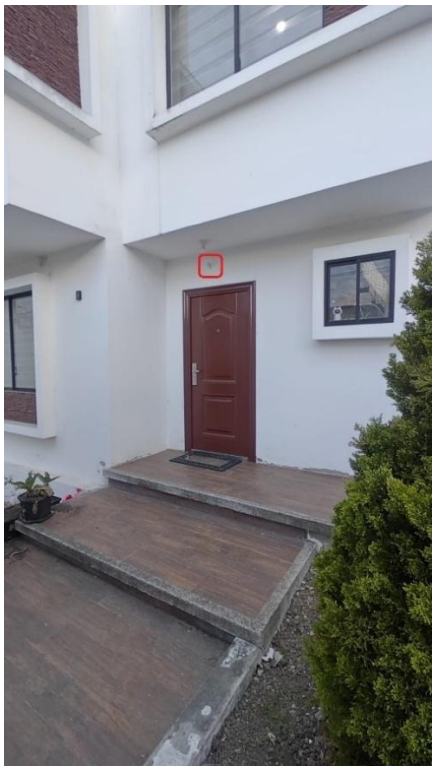
- **Enlace a video:**

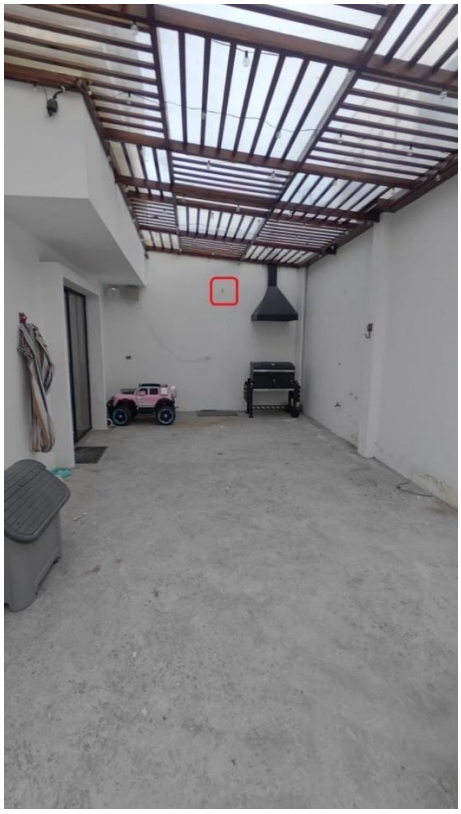
https://drive.google.com/drive/folders/1Ufq8ngDPYtB0P0M4IfjHbz1_bj3kaCnB?usp=drive_link

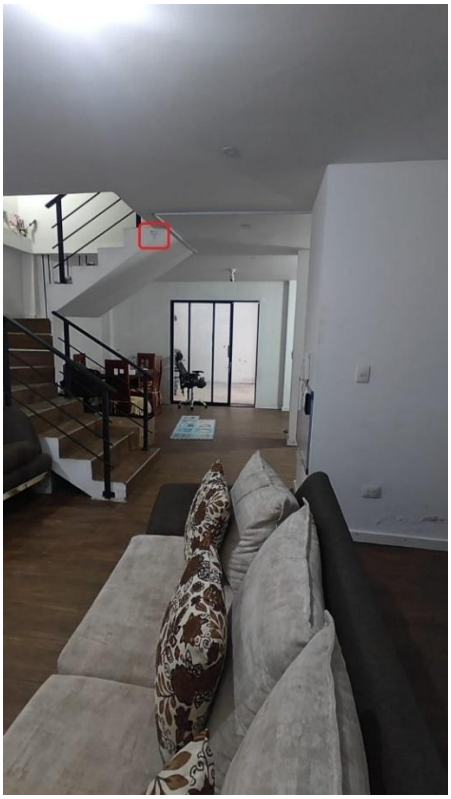
7.4. Anexo 4. Sensores instalados en vivienda

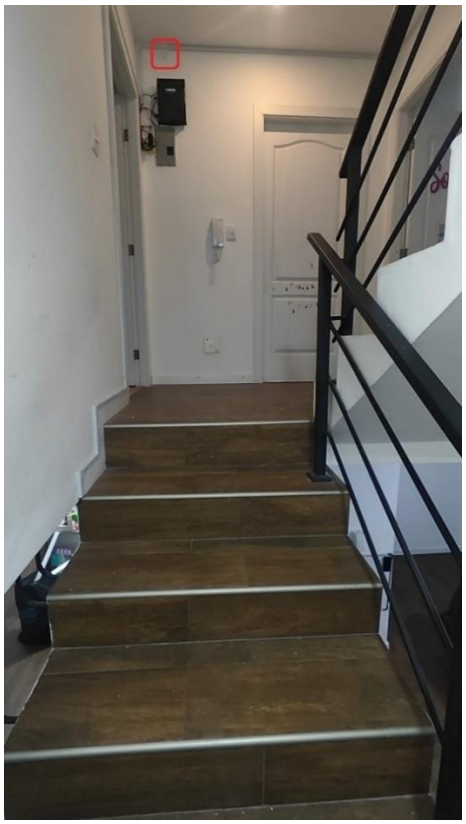


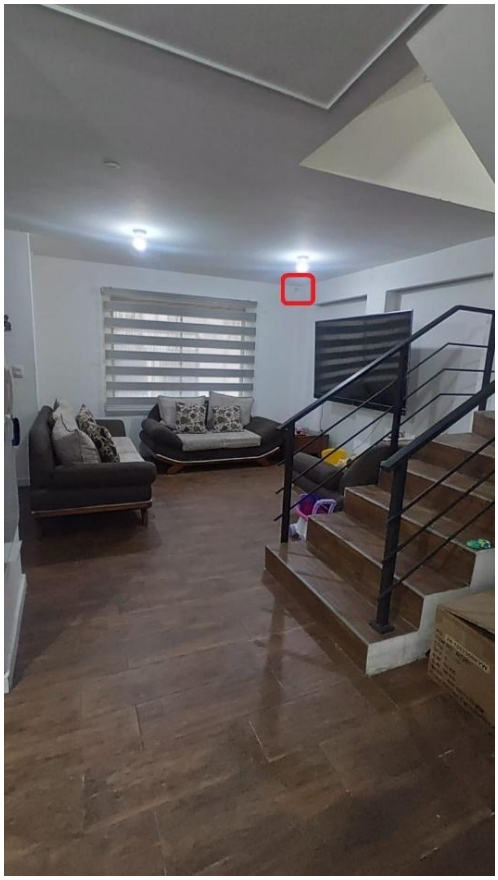

celerity
INTERNET FIBRA ÓPTICA











7.5. Anexo 5. Sistema de respaldo energético