



UNIVERSIDAD TÉCNICA DEL NORTE

Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Mecatrónica

**Sistema de Monitoreo de Presencia de Personas en Espacios Deportivos con Recolección
de Datos aplicando Tecnología IoT**

Trabajo de grado previo a la obtención del título de Ingeniero en Mecatrónica

Autor:

Benavides Cadena Kevin Samuel

Director:

Dr. Carlos Xavier Rosero Chandi

Asesor:

Msc. Cosme Damián Mejía Echeverría

Ibarra - enero del 2026



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. 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:	1003589197		
APELLIDOS Y NOMBRES:	Benavides Cadena Kevin Samuel		
DIRECCIÓN:	Ibarra, Armando Hidrobo y Luis Dávila Pérez		
EMAIL:	Kevinsamuel10e@gmail.com		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0988522218

DATOS DE LA OBRA	
TÍTULO:	Sistema de Monitoreo de Presencia de Personas en Espacios Deportivos con Recolección de Datos aplicando Tecnología IoT
AUTOR:	Benavides Cadena Kevin Samuel
FECHA: DD/MM/AAAA	29/01/2026
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
ASESOR /DIRECTOR:	Carlos Xavier Rosero Chandi Cosme Damián Mejía Echeverría

2. CONSTANCIAS

El autor 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 la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 29 días del mes de enero de 2026

EL AUTOR:

.....
Nombre: Benavides Cadena Kevin Samuel



Universidad Técnica del Norte
Facultad de Ingeniería en Ciencias Aplicadas
Certificación del director del trabajo de grado

En mi calidad de director del trabajo de grado “Sistema de Monitoreo de Presencia de Personas en Espacios Deportivos con Recolección de Datos aplicando Tecnología IoT”, presentado por el egresado Benavides Cadena Kevin Samuel, que opta por el título de ingeniero en Mecánica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, 29 de enero de 2026

Dr. Carlos Xavier Rosero Chandi
Director de Tesis



Universidad Técnica del Norte
Facultad de Ingeniería en Ciencias Aplicadas
Certificación del director del trabajo de grado

El Comité Calificado del trabajo de Integración Curricular “Sistema de Monitoreo de Presencia de Personas en Espacios Deportivos con Recolección de Datos aplicando Tecnología IoT”, elaborado por Benavides Cadena Kevin Samuel, previo a la obtención del título de INGENIERO EN MECATRÓNICA, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

Dr. Carlos Xavier Rosero Chandi

Director de Tesis

Msc. Cosme Damián Mejía Echeverría

Asesor de Tesis

Dedicatorias

Dedico este trabajo de grado, en primer lugar, a Dios, por guiar mis pasos y darme la fortaleza necesaria para superar cada dificultad.

A mis padres, por su amor y apoyo incondicional, siendo el pilar fundamental de mi formación. De manera especial, a mi madre Gladys, por darme la vida y acompañarme con amor en cada decisión, y a mi padre Washington, más que un padre, un amigo incondicional que supo guiarme en lo académico y personal.

A mi hermano Romel, mi compañero de vida y mejor amigo, por creer en mí y apoyarme en los momentos difíciles.

A mi novia Tatiana, por su amor, comprensión y apoyo constante.

A mis abuelitos Rosa, José, Segundo y Eulalia, por los valores inculcados que me han formado como una persona capaz y perseverante.

A mis amigos y a mi familia en general, por su motivación y acompañamiento permanente.

Finalmente, me dedico este logro a mí mismo, por la perseverancia, el esfuerzo y la disciplina que me permitieron alcanzar esta meta y convertirme en la persona que soy hoy. UN HOMBRE LIBRE PODEROSO Y AMOROSO.

Agradecimientos

Agradezco a Dios por brindarme la fortaleza y perseverancia necesarias para culminar esta etapa académica. Expreso mi sincero agradecimiento a mi tutor de trabajo de grado por su orientación, apoyo y valiosos aportes académicos durante el desarrollo de esta investigación. Asimismo, agradezco a los docentes y autoridades de la institución por los conocimientos impartidos a lo largo de mi formación profesional. Finalmente, extendiendo mi agradecimiento a todas las personas que, de manera directa o indirecta, contribuyeron a la realización y culminación del presente trabajo de grado.

Resumen

La gestión y el control de la presencia de personas en espacios deportivos constituyen un desafío relevante, especialmente en escenarios donde es necesario garantizar la seguridad, optimizar el uso de las instalaciones y disponer de información confiable en tiempo real. La ausencia de sistemas automatizados limita el análisis oportuno del aforo y dificulta la toma de decisiones. Ante esta problemática, el presente trabajo tiene como objetivo desarrollar un sistema de monitoreo de presencia de personas en espacios deportivos mediante la aplicación de tecnología de Internet de las Cosas.

La metodología empleada incluye la identificación de los parámetros fundamentales para la detección de presencia humana, el diseño de un sistema basado en visión artificial con procesamiento local de la información y la implementación de una arquitectura de comunicación eficiente. El sistema procesa los flujos de video directamente en el nodo de captura, extrayendo únicamente información relevante en forma de metadatos compactos, los cuales son transmitidos a través de redes LoRaWAN, caracterizadas por su largo alcance y bajo consumo energético. Esta estrategia permite reducir significativamente el volumen de datos transmitidos y garantizar una operación eficiente en entornos con limitaciones de conectividad.

La validación del sistema se realizó en espacios deportivos reales, donde se evaluaron métricas de precisión, latencia, consumo energético y confiabilidad de la comunicación. Los resultados obtenidos evidencian una precisión promedio del 93.8 %, una latencia end-to-end de 300 ms y una tasa de entrega del 98.7 %, demostrando la robustez y estabilidad del sistema propuesto. Asimismo, la notable reducción del volumen de datos transmitidos permite su operación continua sin comprometer el desempeño.

En conclusión, el sistema desarrollado demuestra la viabilidad técnica de integrar visión

artificial y tecnologías IoT para el monitoreo de presencia en espacios deportivos, ofreciendo una solución eficiente, escalable y de bajo costo que contribuye a una gestión más segura y optimizada de las instalaciones.

Palabras clave: Monitoreo de presencia, visión artificial, Internet de las Cosas (IoT), LoRaWAN, LPWAN, detección de personas, espacios deportivos.

Abstract

Managing and controlling the presence of people in sports venues is a significant challenge, especially in scenarios where it is necessary to ensure safety, optimize the use of facilities, and have reliable real-time information. The absence of automated systems limits timely capacity analysis and hinders decision-making. Given this problem, the objective of this work is to develop a system for monitoring the presence of people in sports venues through the application of Internet of Things technology.

The methodology used includes the identification of the fundamental parameters for detecting human presence, the design of a system based on artificial vision with local information processing, and the implementation of an efficient communication architecture. The system processes video streams directly at the capture node, extracting only relevant information in the form of compact metadata, which is transmitted via LoRaWAN networks, characterized by their long range and low energy consumption. This strategy significantly reduces the volume of data transmitted and ensures efficient operation in environments with connectivity limitations.

The system was validated in real sports venues, where metrics such as accuracy, latency, energy consumption, and communication reliability were evaluated. The results show an average accuracy of 93.8 %, an end-to-end latency of 300 ms, and a delivery rate of 98.7 %, demonstrating the robustness and stability of the proposed system. Furthermore, the significant reduction in the volume of data transmitted allows for continuous operation without compromising performance.

In conclusion, the developed system demonstrates the technical feasibility of integrating computer vision and IoT technologies for presence monitoring in sports venues, offering an efficient, scalable, and low-cost solution that contributes to safer and more optimized facility

management.

Keywords: Presence monitoring, computer vision, Internet of Things (IoT), LoRaWAN, LP-WAN, people detection, sports venues.

Índice general

Certificación del director del trabajo de grado	III
Aprobación del comité calificador	IV
Dedicatorias	V
Agradecimientos	VI
Resumen	VII
Abstract	IX
Índice general	XI
Índice de figuras	XV
Índice de tablas	XVIII

I. INTRODUCCIÓN	2
1.1. Planteamiento del Problema	2
1.2. Objetivos	3
1.2.1. Objetivo general	3
1.2.2. Objetivo específico	3
1.2.3. Alcance	3
1.2.4. Justificación	3
II. MARCO TEÓRICO	5
2.1. Estado del arte	5
2.2. Tecnologías aplicadas	6
2.2.1. Comunicación serial TTL (Transistor-Transistor Logic)	7
2.2.2. LPWAN	7
2.2.3. LoRa	8
2.2.4. LoRaWAN	9
2.2.5. Arquitectura de la red LoRaWAN	11
2.3. Detección de personas en Internet de las cosas (IoT)	14
2.3.1. Rol de la visión artificial en la detección de personas	14

2.3.2.	Modelos de aprendizaje profundo para la detección de objetos	14
2.3.3.	Desafíos y arquitectura de la integración VA-IoT	16
2.4.	Propuesta de integración: sistema híbrido de monitoreo inteligente mediante Edge-AI y LoRaWAN	16
2.4.1.	Arquitectura de nodo inteligente (Edge-Vision Node)	16
2.4.2.	Optimización de la gestión de aforo y recursos	17
2.4.3.	Escalabilidad y sostenibilidad energética	18
III.	MARCO METODOLÓGICO	19
3.1.	Enfoque de investigación	19
3.2.	Tipo de investigación	19
3.3.	Diseño metodológico por fases	19
3.4.	Metodología para la selección de dispositivos	21
IV.	RESULTADOS	23
4.1.	Arquitectura del sistema de monitoreo	24
4.1.1.	Nodo final inteligente	24
4.1.2.	Comunicación LoRaWAN y gestión de red	31
4.1.3.	Gestión del servidor de red TTN	36

4.1.4. Backend y visualización	43
4.2. Validación y análisis	50
4.2.1. Metodología de validación experimental	50
4.2.2. Diseño experimental y configuración	51
4.2.3. Resultados de precisión	51
4.2.4. Análisis de latencia del sistema	51
4.2.5. Confiabilidad de la comunicación LoRaWAN	52
4.2.6. Consumo de recursos y eficiencia energética	53
4.2.7. Análisis comparativo con sistemas similares	54
4.2.8. Análisis de limitaciones y mejoras identificadas	54
4.2.9. Resultados de la validación	55
V. CONCLUSIONES Y RECOMENDACIONES	56
5.1. Conclusiones	56
5.2. Recomendaciones	57
Bibliografía	58
Anexos	62
A.1. ANEXO A. CONEXIONES Y COMUNICACIONES INTERNAS	63

A.2. ANEXO B. PRUEBAS DEL SISTEMA 65

Índice de figuras

2.1. Diagrama de conexión de comunicación serial entre dos dispositivos [1].	8
2.2. Clases de nodos LoRaWAN [2].	11
2.3. Pila de protocolos LoRaWAN [3].	11
2.4. Arquitectura de comunicación LoRaWAN [4].	13
2.5. Detección de personas en tiempo real [5].	15
4.1. Arquitectura del sistema propuesto	24
4.2. Cámara Web PC HD 720P Webcam [6]	25
4.3. Raspberry Pi 5 [7]	26
4.4. Diagrama de flujo del sistema de detección en Raspberry Pi 5	28
4.5. Nodo final inteligente	31
4.6. Heltec WiFi Lora v3	32
4.7. Diagrama de comunicación Heltec	33

4.8. Diagrama de flujo de comunicación TTN	34
4.9. Diagrama de flujo desde TTN	37
4.10. Decodificador pyload Decoder	38
4.11. Arquitectura de procesamiento de Webhook TTN	39
4.12. Flujo de procesamiento del script Python	41
4.13. Arquitectura del sistema de visualización desktop	44
4.14. Mecanismo de sincronización con firebase	45
4.15. Distribución de componentes en la interfaz del dashboard	47
4.16. Ejemplo de gráficos de tiempo real implementados	48
4.17. Metodología de validación del sistema de detección	50
4.18. Distribución de errores por tipo de escenario	52
4.19. Distribución de latencia por etapa del sistema	53
4.20. Evolución temporal de la tasa de entrega LoRaWAN	53
A.1. Registro end node en TTN	63
A.2. uplinks de datos en el servidor TTN	64
A.3. Comunicación entre TTN y servidor local	64
A.4. Registro datos procesados servidor local	65

A.5. Comunicación TTN-Firebase	65
A.6. Colocación sistema de monitoreo de personas	66
A.7. Lugar donde se desarrollan las pruebas	66
A.8. Detección con altas aglomeraciones	67
A.9. Detección con moderada cantidad de personas	67
A.10.Detección con poca cantidad de personas	68
A.11.Detección con iluminación y movimientos parciales de personas	68
A.12.Comunicación completa del sistema en tiempo real	69
A.13.Consulta histórica de big data del sistema	69
A.14.Exportación de archivos .xls para análisis	70

Índice de tablas

4.1. Especificaciones de la cámara Web PC HD 720P [6]	25
4.2. Especificaciones de la Raspberry Pi 5 [7]	27
4.3. Configuración del sistema en Raspberry Pi 5	27
4.4. Configuración de procesamiento de video	29
4.5. Estructuras de datos del sistema	30
4.6. Configuración de comunicación serial	30
4.7. Métricas de rendimiento del sistema	30
4.8. Métodos principales del sistema de detección	31
4.9. Especificaciones del módulo Heltec WiFi LoRa v3	32
4.10. Configuración LoRaWAN en Heltec V3	33
4.11. Proceso de construcción del frame	35
4.12. Comparación de formatos de transmisión	35

4.13. Credenciales LoRaWAN TTN	36
4.14. Comparación de métodos de activación LoRaWAN	36
4.15. Arquitectura del sistema de conexión	37
4.16. Tiempos de procesamiento en TTN	38
4.17. Configuración del Webhook-TTN	39
4.18. Funcionamiento del endpoint PHP	40
4.19. Configuración del procesador Python	40
4.20. Algoritmo de filtrado del procesador Python	40
4.21. Estructura de datos en firebase para integración	42
4.22. Mecanismos de seguridad y robustez	42
4.23. Métricas de rendimiento del sistema de integración	42
4.24. Configuración del dashboard desktop	43
4.25. Configuración de conexión firebase	45
4.26. Estructura de la base de datos SQLite local	46
4.27. Algoritmo de sincronización de datos	46
4.28. Componentes principales de la interfaz	47
4.29. Configuración de los gráficos de tiempo Real	48
4.30. Funcionalidades de análisis y exportación	49

4.31. Estados del sistema y manejo de errores	49
4.32. Métricas de rendimiento del dashboard desktop	50
4.33. Etapas de validación del sistema	50
4.34. Configuración del entorno de validación	51
4.35. Resultados de precisión de detección	51
4.36. Latencia por etapa del sistema	52
4.37. Confiabilidad de comunicación LoRaWAN	53
4.38. Consumo de recursos del sistema	54
4.39. Comparación de eficiencia energética	54
4.40. Comparación con sistemas similares	55
4.41. Limitaciones identificadas y propuestas de mejora	55

Capítulo I

INTRODUCCIÓN

1.1. Planteamiento del Problema

En la ciudad de Ibarra existen 34 polideportivos y 5 canchas ubicadas en barrios y parroquias rurales y urbanas del cantón. Los administradores trabajan para convertir estos espacios en entornos autosustentables y recreativos que promuevan actividades saludables para niños, jóvenes y adultos. La construcción de un polideportivo puede alcanzar costos desde 100.000 hasta más de 500.000 dolares [8].

El problema radica en la ausencia de un sistema automático y eficiente para la recopilación de datos relevantes sobre la utilización de los espacios deportivos existentes y las necesidades específicas de cada uno. Los métodos tradicionales de recolección de información, como las encuestas manuales o los análisis de tráfico peatonal, suelen ser costosos, laboriosos y propensos a un alto margen de error. Desde su creación, una de las principales falencias de los polideportivos ha estado relacionada con aspectos administrativos, debido a la inexistencia de un modelo de gestión claro que unifique su uso y mantenimiento. Durante la administración municipal de Andrea Scacco, el Concejo Municipal reconoció la existencia de falencias e inequidades en la gestión de estos espacios y se intentó modificar el modelo administrativo, aunque sin resultados visibles [9]. A lo largo del tiempo, la falta de un plan estratégico y de una visión a futuro ha impedido el desarrollo de una planificación estable y coherente de la infraestructura deportiva [10].

De acuerdo con lo expuesto, en el presente trabajo se propone la implementación de un sistema que permita detectar la presencia de personas que asisten a los centros deportivos, conocidos localmente como polideportivos, mediante el uso de tecnologías orientadas a la recolección de datos con baja carga computacional y de red.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un sistema de monitoreo de presencia de personas en espacios deportivos con recolección de datos aplicando tecnología IoT.

1.2.2. Objetivo específico

- Identificar los parámetros para la detección de presencia en los espacios deportivos.
- Diseñar el sistema de monitoreo considerando los parámetros para la detección de personas.
- Implementar el sistema de monitoreo de detección de presencia en zonas delimitadas directamente en espacio deportivo.
- Validar el sistema en espacios deportivos sometiéndolo a pruebas de precisión y conectividad.

1.2.3. Alcance

El alcance de este trabajo abarca la implementación de un sistema de monitoreo basado en sensores de presencia en un polideportivo de la ciudad de Ibarra, para recopilar datos sobre el uso de estos espacios deportivos. El estudio incluye la selección, instalación de los sensores, el transporte de la información a través de las tecnologías IoT , de baja demanda computacional, y el análisis de datos para la visualización por computador.

1.2.4. Justificación

Este estudio se justifica en la necesidad de implementar un sistema de monitoreo en espacios deportivos para la recaudación de datos aplicando Tecnología IoT. Esta necesidad surge debido a la falta de seguimiento y análisis en tiempo real de diversas métricas y variables relevantes

en entornos deportivos. El monitoreo continuo podría no solo optimizar el rendimiento de las personas, sino también mejorar la experiencia de las instalaciones de la ciudad y la gestión eficiente de los recursos. En el ámbito académico, este estudio aportará conocimiento al centrarse en la aplicación de sistemas de monitoreo. La recopilación y análisis de datos sobre la continua presencia de personas en cada entorno deportivo, la utilización de instalaciones y otros aspectos relevantes proporcionarán información valiosa para la investigación en tecnología aplicada. Desde la perspectiva de la municipalidad, la implementación de un sistema de monitoreo permitirá una mejor comprensión del rendimiento y el uso eficiente de las instalaciones deportivas. Esto conducirá a una mejora en la administración y planificación futura, y una gestión más eficaz de los recursos disponibles. En el contexto social, el impacto de un monitoreo efectivo del sistema en espacios deportivos va más allá del ámbito estadístico. La existencia de espacios públicos adecuados puede promover un estilo de vida saludable y fortalecer el sentido de comunidad en las sociedades con la infraestructura adecuada y estudios con el sistema en la implementación de futuros polideportivos. Además, la eficiencia en la gestión de recursos puede llevar a un beneficio considerable para los barrios de la ciudad, generando oportunidades de recreación y encuentro, así como potenciando el desarrollo económico local.

Capítulo II

MARCO TEÓRICO

2.1. Estado del arte

La mejora de la funcionalidad, gestión y ocupación de los espacios deportivos y de grandes infraestructuras depende de la implementación de soluciones innovadoras basadas en las tecnologías de la información. Este enfoque resulta crucial para garantizar la eficiencia bajo diversas condiciones, donde la convergencia del Internet de las Cosas (IoT) se consolida como un paradigma dominante.

Las investigaciones en este ámbito han demostrado la viabilidad de utilizar sistemas inteligentes para detectar y analizar patrones de actividad. El uso de la visión artificial permite a los sistemas identificar y analizar el comportamiento de multitudes, lo cual resulta aplicable para la gestión de aforos y la implementación de planes de contingencia [11].

Por otro lado, el desarrollo de sensores dedicados al conteo de cuerpos (Body Count Sensor), aunque aún limitado, resalta la necesidad de controlar con precisión el número de personas que acceden a una zona determinada [12].

La tecnología IoT se ha integrado de manera fundamental en la gestión y el análisis de actividades deportivas. Esta integración permite la recolección eficiente y el aprovechamiento de la información para la digitalización de eventos [13].

Las soluciones de monitoreo han sido ampliamente aplicadas para cuantificar el rendimiento y la condición física de los atletas mediante dispositivos wearables y sensores inteligentes, ofreciendo retroalimentación en tiempo real [14], [15]. El objetivo de estos sistemas es proporcionar un método más científico y simplificado para explorar el significado biológico del entrenamiento, apoyándose en la transmisión inalámbrica de datos [16].

En la capa de backend, el uso de servidores en la nube, como Firebase, ha demostrado ser una herramienta eficaz para el almacenamiento de datos provenientes de dispositivos de

campo, permitiendo la creación de ecosistemas completos de paneles de control y aplicaciones de supervisión [17]. Esto respalda el uso de plataformas serverless para el almacenamiento de datos de aforo.

La complejidad de la visión artificial y la necesidad de procesar video en tiempo real, sumadas a las limitaciones de ancho de banda en entornos de gran escala, impulsan la adopción de la computación en el borde (Edge Computing) dentro de la arquitectura IoT [18].

Se ha propuesto que la integración de Edge Computing en la arquitectura IoT es crucial para mejorar el rendimiento de la red, al permitir el procesamiento y almacenamiento de datos más cerca de los sensores, reduciendo significativamente la latencia y el consumo de ancho de banda [19]. Esta descentralización resulta vital para la digitalización de eventos deportivos, donde la combinación de IoT y Edge Computing se considera un requisito indispensable para la gestión digital [13]. La implementación de plataformas Edge permite realizar la inferencia de modelos de inteligencia artificial de manera local para la detección, enviando a la red únicamente los datos procesados, como el conteo de personas.

La necesidad de comunicar los datos de aforo desde los dispositivos Edge hacia plataformas en la nube, cubriendo grandes áreas y optimizando el consumo energético, valida el uso de redes de baja frecuencia y largo alcance.

El IoT se ha expandido hacia la implementación de soluciones innovadoras en áreas de gran cobertura o de difícil acceso, donde las redes de baja frecuencia, como LoRaWAN, han demostrado ser herramientas eficaces [20].

La tecnología LoRa y las redes LoRaWAN destacan especialmente en sistemas de monitoreo para la protección forestal, la agricultura y las ciudades inteligentes, debido a su amplia cobertura y capacidad de conexión en entornos remotos [21], [22]. El uso de microcontroladores y servidores TTN (The Things Network) en conjunto con LoRaWAN ha permitido establecer un flujo de datos eficiente y su almacenamiento en la nube para su posterior visualización [23]. Este enfoque metodológico (microcontrolador + LoRaWAN + TTN + nube) constituye la base de la capa de comunicación del sistema de monitoreo propuesto.

2.2. Tecnologías aplicadas

En esta sección se analizan las tecnologías fundamentales que permiten superar las limitaciones de infraestructura para el monitoreo en entornos remotos o de gran escala, como estadios y áreas deportivas extensas. Estas tecnologías, que abarcan desde la comunicación binaria hasta arquitecturas de red de amplio alcance como LoRaWAN, ofrecen soluciones orientadas a la transmisión de datos con alta eficiencia energética, permitiendo la implementación de sistemas

IoT robustos y escalables para la adquisición de datos de aforo y rendimiento.

2.2.1. Comunicación serial TTL (Transistor-Transistor Logic)

Constituye la capa binaria base de cualquier arquitectura IoT. Es una interfaz de bajo nivel que permite a los microcontroladores leer y empaquetar los datos crudos procedentes de sensores, ya sean de conteo de personas, IMU de un wearable o mediciones biométricas, antes de ser entregados a un módulo de conectividad inalámbrica. Este protocolo forma parte de las familias lógicas, siendo una de las más utilizadas durante los últimos 20 años [24], lo que garantiza una amplia disponibilidad de componentes en el mercado y facilita su implementación.

Operando típicamente entre 0 y 5 V, el protocolo TTL funciona de manera asíncrona, enviando bits de información en secuencia [25]. La fiabilidad de esta transmisión está determinada por varios parámetros de configuración que deben ser idénticos en los dispositivos emisor y receptor:

- **Baudios (Baud Rate):** Define la velocidad de sincronización de la transmisión, siendo común trabajar a 9600 bps (bits por segundo).
- **Bits de datos:** Determina el tamaño del paquete de datos transmitido, generalmente configurado a 8 bits.
- **Bits de paridad:** Añade un bit adicional para la verificación básica y un mecanismo rudimentario de control de errores en la transmisión.
- **Bits de parada:** Se utilizan para delimitar con precisión el final de un paquete de datos.
- **Control de flujo:** Mecanismo que previene la saturación del receptor ante una sobrecarga de información.

La interconexión se realiza mediante los pines RX (recepción) y TX (transmisión), en una configuración cruzada (TX a RX y viceversa), tal como se ilustra en el diagrama de la Figura 2.1, correspondiente a la conexión de dos dispositivos mediante comunicación serial asíncrona. En este esquema, el módulo de comunicación actúa como un puente para adaptar la señal TTL interna del nodo sensor a protocolos de largo alcance, como LoRaWAN.

2.2.2. LPWAN

Las redes *Low Power Wide Area Network* (LPWAN) se han consolidado como la columna vertebral para el despliegue masivo de soluciones IoT que requieren autonomía energética y

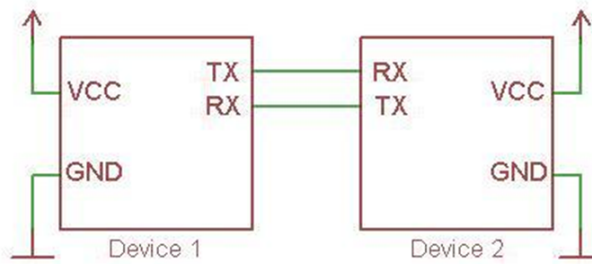


Figura 2.1: Diagrama de conexión de comunicación serial entre dos dispositivos [1].

amplia cobertura. Estas tecnologías de comunicación permiten la transmisión de datos a largas distancias con un consumo energético muy bajo, resolviendo el desafío de conectividad en áreas extensas donde la infraestructura tradicional, como Wi-Fi o redes 4G/5G, resulta costosa o inexistente [26]. Por ello, son idóneas para el monitoreo de aforo en grandes recintos al aire libre o para el seguimiento de personas en áreas de gran extensión.

Las LPWAN ofrecen dos beneficios fundamentales para este tipo de aplicaciones [26]:

- **Amplia cobertura y despliegue simplificado:** Una única estación base puede gestionar miles de dispositivos en áreas de varios kilómetros cuadrados. Esto reduce la necesidad de una infraestructura densa y, por lo tanto, la complejidad operativa y los costos asociados a la implementación de redes tradicionales, permitiendo la conexión de sensores y dispositivos IoT en entornos previamente inaccesibles.
- **Alta eficiencia energética:** Dado que el diseño de estas redes está optimizado para transmitir pequeñas cantidades de datos de forma intermitente, los dispositivos finales pueden operar con una sola batería durante periodos de varios años. Esta característica resulta indispensable para sensores de aforo instalados en puntos remotos o para dispositivos wearables que requieren operación ininterrumpida con un mantenimiento mínimo.

Además, el hecho de que muchas implementaciones LPWAN, como LoRaWAN, operen en bandas de frecuencia sin licencia reduce los costos de operación y mantenimiento, convirtiendo a estas redes en una solución accesible para proyectos de monitoreo a gran escala en áreas rurales o aisladas [27].

2.2.3. LoRa

Es la tecnología de capa física desarrollada por Semtech. Su principal innovación radica en el uso de la modulación por espectro ensanchado (*Chirp Spread Spectrum*), la cual permite

transmitir datos a largas distancias con una potencia mínima y una alta tolerancia al ruido. LoRa constituye el motor fundamental de la conectividad de largo alcance y bajo consumo, sirviendo como soporte del protocolo de capa superior LoRaWAN [29].

Características técnicas

LoRa se destaca por su alta inmunidad a las interferencias, una característica crítica en entornos ruidosos como eventos deportivos o centros de datos, lo que garantiza la fiabilidad de las lecturas de aforo y rendimiento. Además, permite la comunicación directa entre nodos en determinadas implementaciones, y su flexibilidad facilita la integración de los *gateways* con infraestructuras de red existentes. Estas propiedades resultan esenciales para aplicaciones IoT que requieren gran alcance y alta eficiencia energética [27].

2.2.4. LoRaWAN

Long Range Wide Area Network (LoRaWAN) es el protocolo que proporciona estructura a la capa LPWAN. Ha logrado una adopción significativa en aplicaciones IoT debido a su capacidad para ofrecer un alcance considerable y gestionar de manera eficiente una alta densidad de nodos. LoRaWAN se encarga de las funciones de gestión de red, seguridad y programación de la transmisión, operando sobre la capa física LoRa. Esta arquitectura jerárquica permite que las redes funcionen de forma robusta incluso en entornos con baja cobertura [27].

Topología y funcionamiento

La arquitectura LoRaWAN se basa en una topología de estrella, donde los nodos finales se comunican directamente con un *gateway* central, el cual se conecta posteriormente con el servidor de red. Esta configuración centralizada permite que los nodos finales tengan un diseño sencillo, de bajo costo y bajo consumo energético, delegando la complejidad de la gestión al *gateway*. La combinación de esta topología con la tecnología de largo alcance de LoRa permite cubrir áreas de varios kilómetros de extensión [28], lo que la convierte en una solución adecuada para el monitoreo de instalaciones deportivas o zonas extensas de control de aforo.

Características técnicas

LoRa se destaca por su alta inmunidad a las interferencias, una característica crítica en entornos ruidosos como eventos deportivos o centros de datos, lo que garantiza la fiabilidad de las lecturas de aforo y rendimiento. Además, permite la comunicación directa entre nodos

en determinadas implementaciones, y su flexibilidad facilita la integración de los *gateways* con infraestructuras de red existentes. Estas propiedades resultan esenciales para aplicaciones IoT que requieren gran alcance y alta eficiencia energética [27].

Clases de nodos

LoRaWAN define tres clases de nodos diseñadas para equilibrar el consumo energético y la latencia de comunicación, lo que permite adaptar el dispositivo a las necesidades específicas de aplicaciones de aforo o monitoreo de rendimiento, como se muestra en la Figura 2.2:

- **Clase A:** Es la clase con mayor eficiencia energética. Los nodos inician la transmisión de datos y abren únicamente dos ventanas de recepción cortas inmediatamente después. Es ideal para sensores de aforo que envían periódicamente informes de conteo y pueden tolerar una alta latencia en la recepción de comandos.
- **Clase B:** Estos nodos abren ventanas de recepción adicionales sincronizadas mediante *beacons* enviados por el *gateway*. Ofrecen mayor disponibilidad de recepción que la Clase A, siendo adecuados para sistemas que requieren comunicación bidireccional más frecuente, como la actualización de firmware o el envío de comandos de calibración a sensores con consumo energético moderado.
- **Clase C:** Los nodos permanecen en modo de recepción continua, ofreciendo la menor latencia de todas las clases. Este modo es apropiado para aplicaciones que requieren comunicación en tiempo real, como alertas inmediatas por superación de aforo máximo, a costa de un consumo energético significativamente mayor.

Pila de protocolos

La operación de LoRaWAN se sustenta en una pila de protocolos claramente definida. En la base se encuentra la capa física LoRa (*Long Range*), responsable de la modulación y transmisión de datos a gran distancia con bajo consumo energético, como se muestra en la Figura 2.3. Sobre esta capa, el protocolo gestiona el acceso a la red y las aplicaciones, garantizando la integridad de los datos provenientes de los sensores, la gestión escalable de los *gateways* y la interacción segura con el software de aplicación final.

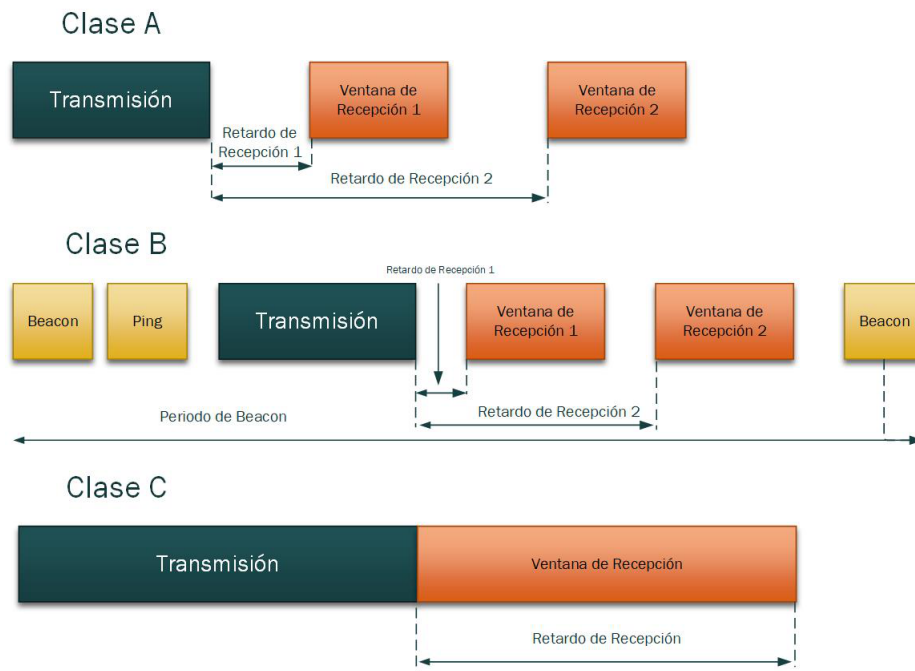


Figura 2.2: Clases de nodos LoRaWAN [2].

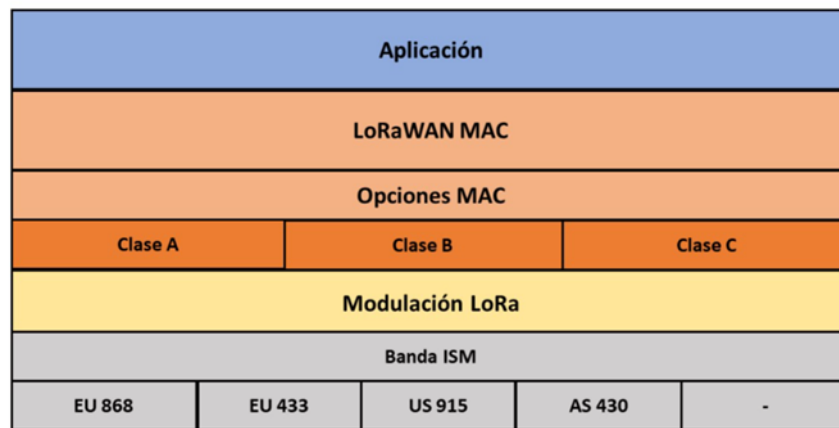


Figura 2.3: Pila de protocolos LoRaWAN [3].

2.2.5. Arquitectura de la red LoRaWAN

La arquitectura del sistema de monitoreo de aforo y rendimiento sigue el modelo jerárquico clásico de LoRaWAN, compuesto por tres niveles esenciales: los nodos de captura, las puertas de enlace de datos (*gateways*) y el servidor central de aplicaciones [30]. Esta disposición garantiza una comunicación altamente eficiente y escalable. Los dispositivos periféricos recolectan

las métricas y las canalizan, a través de los *gateways*, hacia el servidor central, donde la información es sometida a procesos de análisis y almacenamiento. Esto resulta fundamental para la identificación de patrones atípicos, la facilitación de la toma de decisiones tácticas informadas y la gestión proactiva de los recursos. Esta estructura constituye la base de un monitoreo continuo, de bajo consumo energético y de amplio alcance.

Nodos de captura de datos

Los nodos de captura representan los elementos terminales y fundamentales del sistema, ya que constituyen la interfaz directa con el entorno físico. Su función principal consiste en registrar y recopilar métricas operacionales y ambientales relevantes para el escenario monitoreado, tales como el nivel de aforo, la geolocalización instantánea y los patrones de movimiento. Estas métricas permiten comprender la dinámica operativa del sitio y facilitan la detección temprana de posibles problemas o desviaciones.

Estos nodos están diseñados para operar con una demanda energética mínima, garantizando una amplia autonomía operativa, incluso en áreas de despliegue extensas o de difícil acceso. La incorporación de tecnología LoRa asegura la comunicación a larga distancia y una transferencia de datos eficiente, incluso en entornos propensos a interferencias.

Puertas de enlace (Gateway)

Las puertas de enlace actúan como un conmutador intermedio entre la red de nodos sensores y el servidor central. Su función principal es recibir los paquetes de datos emitidos por los nodos a través de la tecnología LoRaWAN y reenviarlos posteriormente al servidor mediante una conexión a Internet o a una red privada. Este flujo de datos garantiza una transferencia fiable y oportuna para su análisis.

El *gateway* está optimizado para gestionar el tráfico simultáneo de múltiples nodos, con una configuración típica de hasta 16 canales, lo que permite soportar cientos de dispositivos conectados [31]. Además, ofrece conectividad dual: utiliza LoRaWAN para comunicarse con la red de sensores y recurre a tecnologías convencionales como Wi-Fi, Ethernet o redes celulares para establecer la conexión con el servidor central. Esta versatilidad facilita la adaptación del sistema a diversas condiciones de infraestructura y entornos de despliegue.

Servidor central de aplicaciones

El servidor central de aplicaciones es responsable de ejecutar el procesamiento, análisis y almacenamiento de los datos transmitidos por las pasarelas. Su objetivo es asegurar que la

información generada sea útil y accesible para las tareas de monitoreo y gestión. Entre las soluciones disponibles, plataformas de servicio de red como *The Things Network* (TTN) [32] ofrecen herramientas robustas para administrar la comunicación entre los nodos sensores y las aplicaciones finales, simplificando la implementación de redes LoRaWAN.

Entre las funciones principales del servidor central se encuentran:

- **Procesamiento y encaminamiento:** El servidor de red recibe, valida y distribuye los datos hacia las aplicaciones correspondientes, asegurando que lleguen en el formato adecuado para su análisis o visualización.
- **Gestión de seguridad y autenticación:** Mantiene rigurosos estándares de seguridad, confirmando la legitimidad de los dispositivos conectados y protegiendo la información mediante mecanismos de cifrado de extremo a extremo [4] y autenticación mutua.
- **Verificación de integridad:** Garantiza que los mensajes de datos se mantengan completos y sin alteraciones desde el nodo de captura hasta las aplicaciones finales.

The Things Network (TTN) constituye un ecosistema global de código abierto diseñado para facilitar la instalación y administración de redes LoRaWAN [32].

Flujo operacional de la comunicación

La Figura 2.4 ilustra el diagrama del proceso completo de comunicación que se lleva a cabo en la arquitectura LoRaWAN.

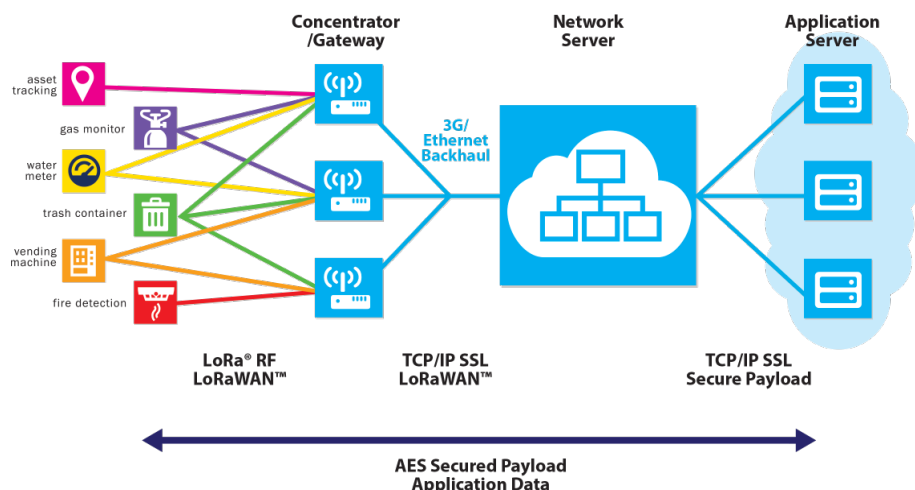


Figura 2.4: Arquitectura de comunicación LoRaWAN [4].

El ciclo comienza en el punto de interés, donde los sensores implementados recopilan las métricas físicas. Estos datos se transforman en señales digitales que los dispositivos equipados con tecnología LoRa transmiten con alta eficiencia energética.

Los nodos de captura envían la información recopilada hacia las puertas de enlace, aprovechando las ventajas de largo alcance proporcionadas por LoRaWAN. Esta capacidad de comunicación puede extenderse por varios kilómetros; estudios reportan alcances de hasta 20 km, dependiendo de las condiciones del entorno [33], lo que garantiza que los sensores ubicados en zonas remotas mantengan conectividad continua.

El *gateway* recibe, organiza y empaqueta los datos provenientes de los nodos para su posterior envío al servidor central. Esta transmisión de interconexión se realiza mediante el medio disponible, como Wi-Fi, Ethernet o redes celulares, asegurando que los datos lleguen al servidor de manera fiable y con una latencia adecuada.

Finalmente, en el servidor central la información es procesada, analizada y almacenada. Los resultados se presentan al usuario final a través de navegadores web o aplicaciones móviles, lo que permite una interpretación sencilla de los datos, la recepción de alertas automáticas y la toma de decisiones fundamentadas en evidencia.

2.3. Detección de personas en Internet de las cosas (IoT)

2.3.1. Rol de la visión artificial en la detección de personas

La capacidad de detectar, localizar y rastrear personas constituye un requisito fundamental para la evolución de los sistemas de monitoreo IoT hacia entornos más inteligentes, seguros y eficientes. Tradicionalmente, la detección de presencia en IoT se ha basado en sensores pasivos de infrarrojo (PIR) o en la proximidad de dispositivos; sin embargo, estos métodos carecen de la exactitud y precisión necesarias para tareas más complejas, como el conteo de multitudes o el análisis de comportamiento.

La visión artificial (VA) emerge como una tecnología más robusta para superar estas limitaciones, utilizando cámaras de video para extraer información contextual del entorno. La integración de la VA con infraestructuras IoT permite el monitoreo en tiempo real y la toma de decisiones basadas en datos visuales confiables.

2.3.2. Modelos de aprendizaje profundo para la detección de objetos

El avance en el aprendizaje profundo (*Deep Learning*) ha revolucionado la detección de personas, permitiendo el desarrollo de modelos capaces de identificar individuos con alta precisión



Figura 2.5: Detección de personas en tiempo real [5].

y a velocidades adecuadas para aplicaciones en tiempo real, como se ilustra en la Figura 2.5 [5].

Modelos como la serie *You Only Look Once* (YOLO) se han adaptado a entornos específicos, como la detección de jugadores y balones en video de alta velocidad, donde los principales desafíos incluyen movimientos rápidos, oclusiones frecuentes y variaciones en las condiciones de iluminación.

Estos sistemas de detección, basados en redes neuronales convolucionales (CNN) y sus variantes, resultan esenciales para:

- **Conteo de individuos:** La capacidad de contabilizar con precisión el número de personas en una zona delimitada es vital para la gestión de multitudes y la monitorización de aforo. Este enfoque ha sido aplicado en diversos sistemas de control y vigilancia, incluyendo el seguimiento de protocolos de seguridad y distanciamiento físico en áreas públicas y comerciales [34].
- **Análisis de densidad y flujo:** Más allá del simple conteo, los sistemas de visión artificial permiten estimar la densidad de multitudes y predecir patrones de actividad. Métodos basados en mapas de mosaicos (*tile-map*) integrados con el *Cloud of Things* (CoT) posibilitan una monitorización inteligente de la densidad de multitudes en exteriores, facilitando la implementación de planes de contingencia y la mejora de la seguridad pública [35].
- **Reconocimiento del movimiento:** La visión artificial, combinada con algoritmos de *tracking*, permite el reconocimiento detallado del movimiento humano, lo cual es esencial para aplicaciones deportivas o de seguridad. Este procesamiento puede optimizarse mediante la implementación de arquitecturas basadas en *Edge Computing*, que acercan la

capacidad de procesamiento de video a la fuente de datos, mejorando la latencia y la eficiencia en la identificación del movimiento [36].

2.3.3. Desafíos y arquitectura de la integración VA-IoT

La integración efectiva de la visión artificial en arquitecturas IoT presenta diversos desafíos relacionados con el volumen de datos y los requerimientos de procesamiento. Los flujos de video generan grandes volúmenes de información (*Big Data*) que no pueden ser gestionados de manera eficiente mediante una arquitectura puramente centralizada en la nube. Asimismo, las aplicaciones de detección y reacción en tiempo real demandan latencias mínimas, lo que limita el uso exclusivo de infraestructuras remotas.

Para superar estos desafíos, las arquitecturas modernas adoptan modelos híbridos. El *Edge Computing* permite que una parte significativa del procesamiento de imágenes se realice cerca de la fuente de datos, ejecutando localmente las tareas de detección y análisis, y enviando únicamente la información relevante, como conteos y eventos detectados, a través de la arquitectura IoT. Este enfoque no solo reduce el consumo de ancho de banda y mejora la capacidad de procesamiento en tiempo real, sino que también contribuye a la preservación de la privacidad al evitar la transferencia masiva de datos crudos [36].

La nube mantiene un rol fundamental en el almacenamiento de datos históricos, el análisis agregado y el entrenamiento de modelos de aprendizaje profundo a gran escala, cerrando así el ciclo de la arquitectura IoT integrada.

2.4. Propuesta de integración: sistema híbrido de monitoreo inteligente mediante Edge-AI y LoRaWAN

La presente propuesta plantea la convergencia de las capacidades de largo alcance de la red LoRaWAN con la precisión analítica de la visión artificial. El objetivo es solventar la limitación intrínseca de ancho de banda de las redes LPWAN (*Low-Power Wide-Area Network*) mediante la implementación de un paradigma de procesamiento en el borde, donde la detección de personas se realiza de forma local y únicamente los metadatos resultantes son transmitidos.

2.4.1. Arquitectura de nodo inteligente (Edge-Vision Node)

A diferencia de los nodos de captura convencionales descritos en la sección anterior, se propone la implementación de nodos de visión en el borde. Estos dispositivos integran un sensor

de imagen de bajo consumo y un coprocesador de inteligencia artificial, como un acelerador de redes neuronales, capaz de ejecutar modelos compactos de la familia YOLO.

Diferenciación de funciones y flujo de datos

La integración propuesta redefine el rol de cada componente con el fin de optimizar la eficiencia del sistema:

- **Procesamiento local (VA):** El nodo captura el flujo de video, pero no lo transmite. El modelo de aprendizaje profundo identifica a las personas, calcula el aforo y analiza la densidad en tiempo real.
- **Extracción de metadatos:** Una vez procesada la imagen, el sistema descarta el video y extrae únicamente variables numéricas, como *count=45*, *density=high* o *alert=1*.
- **Transmisión LoRa:** Estos paquetes de metadatos, cuyo tamaño es de apenas unos pocos bytes, se envían a través del protocolo LoRaWAN. De esta manera, una tecnología diseñada para telemetría simple puede soportar sistemas avanzados de monitoreo.

2.4.2. Optimización de la gestión de aforo y recursos

La integración de la visión artificial permite que el sistema de monitoreo de aforo no solo sea reactivo, sino también predictivo y cualitativo. Al combinar la geolocalización de los nodos LoRa con el análisis de flujo proporcionado por los modelos de *Deep Learning*, la arquitectura propuesta puede generar mapas de calor dinámicos sin comprometer la infraestructura de red convencional.

Seguridad y privacidad por diseño

Un valor agregado fundamental de esta arquitectura integrada es el cumplimiento de normativas de privacidad. Al procesar la detección de personas directamente en el nodo y transmitir únicamente estadísticas anónimas hacia el servidor central, se elimina el riesgo de interceptación de imágenes sensibles en la red. Esto garantiza una gestión proactiva de los recursos bajo un esquema de seguridad de extremo a extremo, reforzando los mecanismos de autenticación mutua inherentes a LoRaWAN [4].

2.4.3. Escalabilidad y sostenibilidad energética

Al delegar la carga computacional al borde y utilizar LoRaWAN para el reporte de estados, se logra un equilibrio entre capacidad de cómputo y autonomía energética. Esta propuesta permite el despliegue de redes de monitoreo de alta densidad en escenarios donde el cableado de fibra óptica o la cobertura Wi-Fi son inexistentes, manteniendo un bajo consumo energético y una cobertura de amplio alcance, fundamentales para la identificación de patrones en entornos urbanos o industriales extensos.

Capítulo III

MARCO METODOLÓGICO

En este capítulo se presenta la metodología de desarrollo del proyecto, estructurada en fases secuenciales que guían el proceso desde la visión teórica y el diseño conceptual hasta la implementación en campo y la validación de un modelo experimental. El modelo de investigación se fundamenta en un enfoque aplicado, orientado a la creación de una solución tecnológica funcional, de baja latencia y optimizada.

3.1. Enfoque de investigación

El trabajo de titulación se basa en un enfoque de investigación aplicada, enmarcado dentro de los ámbitos de la ingeniería y la tecnología.

3.2. Tipo de investigación

- **Documental / Revisión:** Análisis de tecnologías, algoritmos de visión artificial y plataformas de Internet de las Cosas (IoT), aplicando arquitecturas de *Edge Computing*.
- **Experimental / Aplicada:** Desarrollo del sistema, despliegue en campo y medición de métricas de rendimiento reales, tales como cuadros por segundo (FPS), precisión media (*mAP*) y latencia.

3.3. Diseño metodológico por fases

El diseño metodológico se divide en cuatro fases principales que integran los pasos clave del proyecto, asegurando un desarrollo lógico y una validación rigurosa del sistema de detección

de personas en el borde de la red LoRaWAN.

Fase 1: Investigación y Definición

Esta fase se centra en la revisión teórica, la toma de decisiones tecnológicas preliminares y la definición del alcance del sistema.

1. **Revisión de tecnologías IoT y Edge Computing:** Investigación y comparación de plataformas IoT y protocolos de comunicación adecuados para la transmisión eficiente de los metadatos de detección desde los dispositivos Edge. Análisis de la capacidad de microcontroladores y dispositivos disponibles.
2. **Análisis de visión artificial y *Deep Learning*:** Revisión del estado del arte en modelos de detección de objetos y técnicas de optimización necesarias para mantener la precisión y alcanzar altas velocidades de procesamiento (FPS) en entornos Edge.
3. **Definición de requisitos operacionales:** Establecimiento de las variables críticas de salida del modelo y de las métricas de rendimiento a monitorear.

Fase 2: Diseño y arquitectura

Esta fase establece la estructura lógica y física del sistema previo al desarrollo e implementación, garantizando la coherencia entre los componentes de hardware, software y comunicación.

1. **Selección de hardware (dispositivo Edge y periféricos):** Selección de los componentes de hardware que cumplan con los requisitos de procesamiento CPU/GPU necesarios para la inferencia eficiente de los algoritmos de visión artificial, considerando criterios de consumo energético, capacidad de cómputo y compatibilidad con sensores de imagen.
2. **Arquitectura del sistema:** Diseño del flujo de datos completo, definiendo la interconexión entre los módulos bajo el esquema de una arquitectura IoT integrada, que incluye el nodo de visión artificial, la puerta de enlace (gateway), el servidor central, la base de datos y la aplicación de visualización.
3. **Preparación y entrenamiento del modelo:** Adquisición y etiquetado del conjunto de datos, entrenamiento del modelo de detección seleccionado y aplicación de técnicas de *fine-tuning* y optimización para su despliegue eficiente en el dispositivo Edge.

Fase 3: Implementación

Esta fase se enfoca en el despliegue físico del sistema, la integración del software y la ejecución de las primeras pruebas operacionales en entornos reales.

1. **Instalación en espacios deportivos abiertos piloto:** Desplegar físicamente y configurar el dispositivo Edge (cámara y microcontrolador) y el gateway de comunicación en el espacio seleccionado.
2. **Despliegue del modelo y calibración:** Convertir el modelo entrenado a un formato optimizado e implementarlo en el dispositivo Edge, realizando pruebas de campo para su ajuste y validación.
3. **Pruebas de conectividad y precisión:** Verificar la fiabilidad y la latencia de la comunicación de los datos desde el nodo final, a través de las puertas de enlace, hacia el servidor central y el almacenamiento en la nube, evaluando finalmente los resultados obtenidos.

Fase 4: Validación y análisis

La fase final se dedica a la recopilación sistemática de datos, la evaluación del rendimiento y la consolidación de los resultados del proyecto.

1. **Análisis de datos recolectados:** Recopilar y procesar los metadatos de detección. Evaluar el rendimiento del modelo mediante métricas de precisión media, FPS y latencia en condiciones reales de funcionamiento.
2. **Comparación con métodos tradicionales:** Contrastar los resultados obtenidos por la solución IoT con Edge Computing frente a métodos de monitoreo manuales o centralizados.
3. **Redacción de conclusiones y presentación de resultados:** Elaborar el informe final integrando la documentación de diseño, los gráficos de resultados, las conclusiones y las recomendaciones futuras.

3.4. Metodología para la selección de dispositivos

Para la selección de componentes se realiza una investigación del mercado, con el fin de identificar opciones viables y accesibles, en función de los siguientes criterios:

- **Compatibilidad:** Se seleccionan equipos con interfaces compatibles entre sí, a fin de facilitar la interconexión y la comunicación.

- **Resistencia a condiciones ambientales:** Se verifica que los dispositivos sean adecuados para operar en entornos con variaciones de temperatura, humedad y vibraciones.
- **Disponibilidad y costo:** Se consideran opciones comerciales viables en términos de disponibilidad y costos de adquisición.

Adicionalmente, se consideran trabajos previos como referencia para respaldar la selección y generar un entorno confiable. Se hace énfasis en la confiabilidad de los dispositivos, por lo que, en algunos casos, el reconocimiento de la marca influye en la decisión de selección, especialmente cuando se trata de fabricantes ampliamente aceptados por los servidores y plataformas de conexión, lo que contribuye a obtener mejores resultados.

Asimismo, se prioriza el uso de dispositivos previamente adquiridos, con el objetivo de evitar costos excesivos e innecesarios en el desarrollo del proyecto.

Capítulo IV

RESULTADOS

En este capítulo se presenta el diseño y desarrollo de un sistema IoT orientado a la detección de personas en espacios deportivos mediante visión artificial, destacando la selección de los dispositivos necesarios para la recopilación, recepción y transmisión eficiente de datos. Para la captación de la presencia de personas, se emplea una cámara conectada a un microcontrolador que ejecuta un modelo basado en Redes Neuronales Convolucionales (CNN), capaz de identificar y localizar individuos dentro del campo de visión, procesando localmente la información obtenida.

Los datos generados son enviados, mediante un modulador de señal, a la red LoRaWAN, lo que permite una transmisión robusta y de largo alcance hacia la estación base. Posteriormente, la información es recibida por el servidor de The Things Network (TTN) a través de sus puertas de enlace, donde se gestiona el tráfico de datos y se integra con un servidor de almacenamiento en Firebase para su administración y análisis. Este flujo de información resulta fundamental para garantizar la integridad del sistema, ya que posibilita el monitoreo y análisis en tiempo real de las condiciones dinámicas del entorno.

En este contexto, se realizan diversas pruebas experimentales y de campo con el propósito de efectuar un análisis riguroso de los datos recolectados. Estas evaluaciones permitan validar el funcionamiento integral del sistema y extraer conclusiones relevantes sobre su precisión y eficiencia en la detección de personas. Finalmente, los datos obtenidos son caracterizados y presentados mediante visualizaciones que facilitan la interpretación de los resultados, constituyendo un insumo clave para la evaluación del desempeño y la toma de decisiones técnicas.

Adicionalmente, la selección de los dispositivos se fundamenta en los criterios definidos en la sección selección de dispositivos, considerando su precisión, la versatilidad demostrada en aplicaciones previas, el respaldo de la marca y su accesibilidad en el mercado actual.

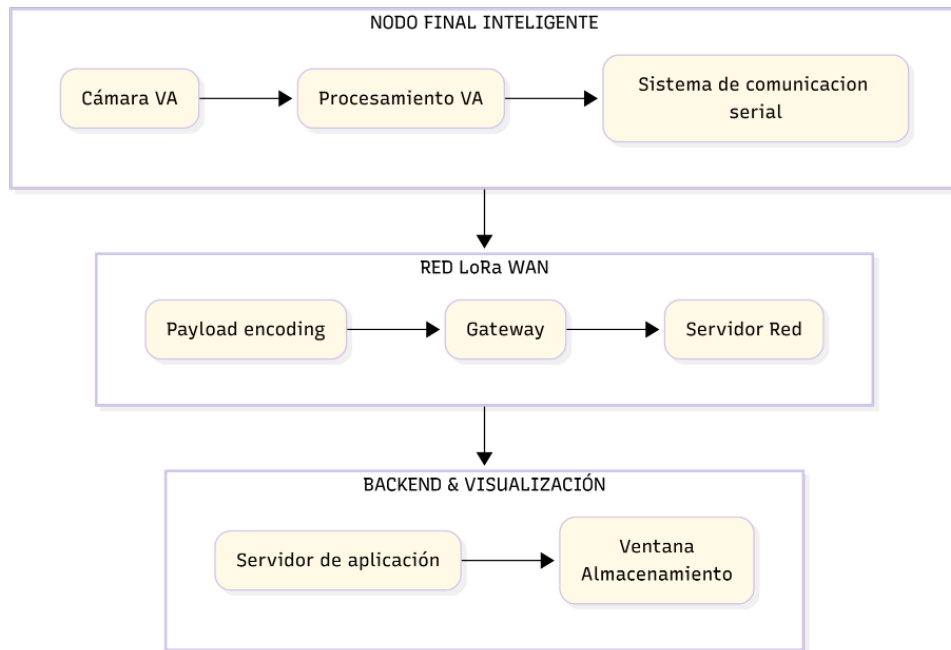


Figura 4.1: Arquitectura del sistema propuesto

4.1. Arquitectura del sistema de monitoreo

La Figura 4.1 ilustra la arquitectura del sistema propuesto. Esta estructura ha sido diseñada minuciosamente a partir de la revisión de la literatura, con el objetivo de optimizar el proceso de obtención de los resultados esenciales empleados en el desarrollo del sistema de monitoreo, enfatizando sus características principales y la manera en que estas satisfacen los requisitos funcionales del sistema.

4.1.1. Nodo final inteligente

Representa la fase de adquisición, procesamiento y pre-transmisión de datos, constituyendo el componente más crítico en términos de procesamiento local y eficiencia energética. Esta fase se compone de tres módulos interconectados, diseñados para asegurar la detección precisa de eventos y la optimización de la carga útil de la comunicación.

Módulo de adquisición: Cámara Web PC 720P

La cámara Web PC 720P [6] es seleccionada por contar con una resolución de 640×480 píxeles, la cual proporciona la claridad visual necesaria para la tarea de detección de personas sin imponer una carga excesiva en el procesamiento posterior. Su interfaz de comunicación USB



Figura 4.2: Cámara Web PC HD 720P Webcam [6]

2.0 facilita una integración rápida y estable con el módulo de procesamiento. En la Figura 4.2 se presenta la cámara utilizada y en la Tabla 4.1 se detallan sus especificaciones técnicas.

Tabla 4.1: Especificaciones de la cámara Web PC HD 720P [6]

Especificación	Valor	Descripción
Tipo	Cámara web USB	Dispositivo de captura de video
Material	ABS	Plástico resistente
Chip DSP	No integrado	Procesamiento en Raspberry Pi
Sensor de imagen	CMOS	Sensor estándar de video
Lente	Cristal	Óptica de alta claridad
Resolución	1280 × 720	HD (720p)
Resoluciones soportadas	640 × 480 / 1280 × 720	Configurables por software
Velocidad de fotogramas	30 FPS	Tasa máxima
Interfaz	USB 2.0	Conexión estándar
Características	Plug & Play	Instalación automática
Dimensiones	8 × 3 × 11 cm	Formato compacto

La cámara actúa como el principal sensor digital del sistema, capturando los datos brutos del entorno que posteriormente serán transformados en información procesable. Su compatibilidad con librerías de visión artificial basadas en Python, como OpenCV, asegura un flujo de trabajo eficiente durante la etapa de desarrollo. Esta selección responde a la necesidad de obtener datos visuales de calidad suficiente, manteniendo un costo y un consumo de recursos controlados, en cumplimiento de los requisitos de la fase inicial de procesamiento en el borde.

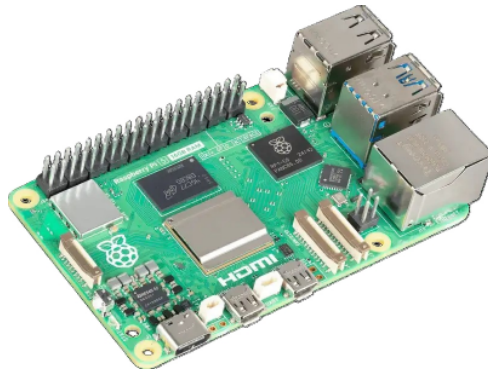


Figura 4.3: Raspberry Pi 5 [7]

Módulo de procesamiento: Raspberry Pi 5

El cerebro del nodo final es la Raspberry Pi 5, seleccionada por su arquitectura de alto rendimiento y su robusto ecosistema de desarrollo. Esta arquitectura SBC (Single Board Computer) permite ejecutar el modelo de Visión Artificial (VA) con eficiencia. Su capacidad multinúcleo y su GPU VideoCore VII, especificadas en la Tabla 4.2, son cruciales para manejar operaciones matriciales inherentes a las Redes Neuronales Convolucionales (CNN). El procesador quad-core ARM Cortex-A76 a 2.4 GHz, combinado con la GPU VideoCore VII, proporciona la capacidad computacional necesaria para ejecutar inferencias de modelos de deep learning en tiempo real, un requisito fundamental para sistemas de monitorización continua. En la Figura 4.3 se muestra el microcontrolador y sus principales componentes físicos, destacando su factor de forma compacto ideal para implementaciones embebidas.

El principal funcionamiento del sistema está controlado por un script Python que ejecuta el framework de VA YOLO para procesar los frames adquiridos por la cámara. Como se detalla en la Tabla 4.3, la configuración del sistema está optimizada para el hardware específico de la Raspberry Pi 5. Los parámetros de esta tabla fueron seleccionados experimentalmente para maximizar el rendimiento manteniendo una precisión aceptable. Su función principal es aplicar el modelo entrenado para la detección de personas y posteriormente, cuantificar los eventos de interés como el conteo de personas en escena y conteo de personas en sesión. Este proceso de transformación de datos es esencial ya que el sistema convierte un alto volumen de datos de imagen/video (aproximadamente 921,600 píxeles por frame a 20 FPS) en una pequeña cadena de metadatos de solo unos pocos bytes, lo cual es fundamental para la eficiencia del subsistema de comunicación de bajo ancho de banda como LoRaWAN.

Tabla 4.2: Especificaciones de la Raspberry Pi 5 [7]

Especificaciones	Valor
Procesador	Broadcom BCM2712
Arquitectura	ARM Cortex-A76
Núcleos	4 (quad-core)
Velocidad	2.4 GHz
RAM	4GB / 8GB LPDDR4X
GPU	VideoCore VII
Conectividad	Wi-Fi 5, Bluetooth 5.0
Ethernet	Gigabit Ethernet (con PoE+)
USB	2 × USB 3.0, 2 × USB 2.0
Video	2 × micro HDMI (4Kp60)
Alimentación	USB-C (5V/5A)
GPIO	Header de 40 pines
Storage	MicroSD + PCIe 2.0
Dimensiones	85.6mm × 56.5mm

Tabla 4.3: Configuración del sistema en Raspberry Pi 5

Parámetro	Valor	Descripción
Modelo YOLO	yolov8n.pt	Modelo YOLOv8 nano
Resolución de cámara	640 × 480	Balance rendimiento–precisión
FPS objetivo	20 fps	Tiempo real
Confianza mínima	0.4	Filtro de detecciones
Tamaño de procesamiento	320 px	Inferencia rápida
Clase detectada	0 (personas)	Dataset COCO
Algoritmo de tracking	ByteTrack	Seguimiento multiobjeto

Arquitectura del sistema de procesamiento El sistema implementado en la Raspberry Pi 5 utiliza una arquitectura modular que integra múltiples componentes: captura de video, procesamiento con YOLO v8, seguimiento de objetos, comunicación serial y visualización en tiempo real. Como se especifica en la Tabla 4.4, los parámetros de procesamiento de video están cuidadosamente ajustados para maximizar el rendimiento en la plataforma ARM. La configuración de buffer limitado a 1 (CAP_PROP_BUFFERSIZE) es particularmente importante para minimizar la latencia en la captura de frames, aunque requiere un timing preciso en el procesamiento.

Algoritmo de procesamiento y diagrama de flujo El algoritmo implementado sigue una secuencia estructurada para el procesamiento en tiempo real. El diagrama de flujo en la Figura 4.4 ilustra el proceso completo desde la inicialización hasta el cierre del sistema.

La secuencia de procesamiento sigue estos pasos detallados en el algoritmo:

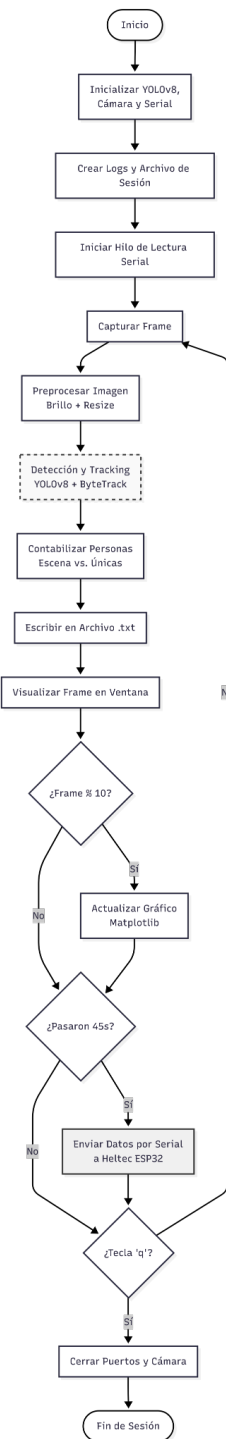


Figura 4.4: Diagrama de flujo del sistema de detección en Raspberry Pi 5

Tabla 4.4: Configuración de procesamiento de video

Parámetro	Valor	Propósito
CAP_PROP_FPS	30	Frames por segundo máximos de la cámara USB
CAP_PROP_BUFFERSIZE	1	Tamaño mínimo de buffer para reducir latencia
Tamaño de Imagen	320 píxeles	Tamaño reducido para inferencia acelerada
Clase Detectada	0 (personas)	Clase COCO específica para detección humana
Tracker	bytetrack.yaml	Archivo de configuración del algoritmo ByteTrack
Color Persona	(0, 255, 0)	Color verde en formato BGR para visualización
Brillo Software	10	Ajuste de ganancia para condiciones de iluminación variable

1. **Inicialización del sistema:** Configuración de parámetros según Tablas 4.3 y 4.4
2. **Adquisición de frames:** Captura mediante OpenCV con resolución 640x480 píxeles
3. **Pre-procesamiento:** Ajuste de brillo (valor 10) y normalización de píxeles
4. **Inferencia:** Detección mediante YOLOv8n con umbral de confianza de 0.4
5. **Seguimiento:** Asignación de IDs únicos mediante ByteTrack
6. **Pos-procesamiento:** Cálculo de métricas y visualización
7. **Comunicación:** Envío serial de metadatos cada 45 segundos

Optimizaciones para arquitectura ARM Para aprovechar al máximo las capacidades de la Raspberry Pi 5, se implementaron varias optimizaciones específicas para arquitectura ARM:

- **Reducción de resolución:** Procesamiento a 320x320 píxeles reduce los cálculos en un factor de 4.
- **Buffer limitado:** Configuración CAP_PROP_BUFFERSIZE = 1 minimiza memoria utilizada y latencia.
- **Control de FPS:** Mecanismo de espera activa para mantener 20 FPS.
- **Suavizado de métricas:** Implementación de filtro exponencial para cálculo estable de FPS.

Estructura de datos y gestión de memoria El sistema mantiene estructuras de datos eficientes para el manejo de información en tiempo real. Como se describe en la Tabla 4.5, estas estructuras están diseñadas para operaciones rápidas de inserción y búsqueda, cruciales para el seguimiento de múltiples objetos simultáneamente.

Tabla 4.5: Estructuras de datos del sistema

Estructura	Tipo	Propósito
personas_detectadas	set()	Conjunto de IDs únicos acumulados durante toda la sesión
personas_visibles	set()	Conjunto temporal de IDs visibles en el frame actual
archivo_sesion	str	Ruta dinámica para logs con estructura temporal
tiempos, unicas, escena	list()	Series temporales para gráficos con límite de 60 muestras
fps_list	list()	Historial de valores FPS para análisis de rendimiento

Sistema de comunicación serial La comunicación bidireccional con el módulo Heltec se implementa mediante el puerto serial UART de la Raspberry Pi 5. La configuración detallada en la Tabla 4.6 asegura una transmisión confiable de datos entre los dos sistemas embebidos.

Tabla 4.6: Configuración de comunicación serial

Parámetro	Valor	Descripción
Puerto	/dev/ttyUSB0	Puerto serial UART estándar en sistemas Linux
Baud Rate	115200	Velocidad óptima para transmisión confiable
Formato	"YYYY-MM-DD HH:MM:SS,Unicas:X,Escena:Y"	Estructura compacta con separadores claros
Intervalo de Envío	45 segundos	Balance entre actualización y consumo energético
Timeout	1 segundo	Evita bloqueos en lectura/escritura
Encoding	UTF-8	Compatibilidad con caracteres especiales

Monitoreo y métricas de rendimiento El sistema implementa un sistema completo de monitoreo que calcula y visualiza métricas en tiempo real. Como se especifica en la Tabla 4.7, las métricas clave incluyen fórmulas matemáticas optimizadas para el cómputo eficiente en la Raspberry Pi 5.

Tabla 4.7: Métricas de rendimiento del sistema

Métrica	Fórmula/Cálculo	Descripción
FPS Suavizado	$fps_t = 0,9 \times fps_{t-1} + 0,1 \times \left(\frac{1}{\Delta t}\right)$	Filtro exponencial para valores estables
Personas únicas	$card(personas_detectadas)$	Cardinalidad del conjunto de IDs únicos
Personas en escena	$card(personas_visibles)$	Conteo instantáneo en frame actual
Tiempo objetivo por frame	$T_{min} = \frac{1}{20} = 0,05$ segundos	Límite temporal para mantener 20 FPS
Compresión de datos	$\frac{20 \times 921600 \text{ bytes/s}}{100 \text{ bytes/45s}} \approx 9 \times 10^6 : 1$	Relación de compresión imagen-metadatos

Métodos y funcionalidades principales La implementación del sistema se organiza en métodos modulares cuyas responsabilidades se especifican en la Tabla 4.8. Cada método está diseñado para una función específica, facilitando el mantenimiento y la depuración del sistema.

Tabla 4.8: Métodos principales del sistema de detección

Método	Parámetros Principales	Función y Responsabilidad
enviar_datos	personas_unicas: int, personas_escena: int	Serializa y envía datos a Heltec con formato específico
enviar_primer_mensaje	-	Inicializa conexión serial enviando valores cero
setup_serial	-	Configura puerto serial según parámetros establecidos
leer_serial	-	Thread continuo para recepción asíncrona de respuestas
actualizar_grafico	tiempo: str, unicas: int, visibles: int, fps: float	Actualiza gráfico Matplotlib con datos en tiempo real
guardar_estadistica_linea	total: int, actuales: int, fps_valor: float	Escribe logs en archivo con estructura temporal



Figura 4.5: Nodo final inteligente

Evaluación de rendimiento y eficiencia El sistema en la Raspberry Pi 5 logra procesar aproximadamente 20 FPS mientras mantiene una precisión de detección aceptable para aplicaciones de monitorización. La transformación de datos representa una compresión de datos de más de 9 millones a 1. Esta eficiencia en la reducción de datos es crucial para sistemas IoT con restricciones de ancho de banda, demostrando la viabilidad de integrar visión artificial avanzada en plataformas de recursos limitados.

La implementación final como se muestra en la Figura 4.5 aprovecha eficientemente todos los recursos de la Raspberry Pi 5, logrando un balance óptimo entre precisión de detección, rendimiento en tiempo real y eficiencia energética. Este diseño permite operación continua 24/7 en escenarios de monitorización de personas, cumpliendo con los requisitos de sistemas IoT industriales donde la confiabilidad y la eficiencia son parámetros críticos.

4.1.2. Comunicación LoRaWAN y gestión de red

La segunda fase operativa del sistema se centra en la transmisión eficiente y segura de los datos procesados. Esta etapa es gestionada por el microcontrolador Heltec WiFi LoRa v3 Figura 4.6, el cual actúa como módem inteligente que encapsula los metadatos provenientes de la Raspberry Pi 5 y gestiona la pila de protocolos para LoRaWAN para su entrega a la infraestructura de la red. Las especificaciones técnicas de este módulo se detallan en la Tabla 4.9.

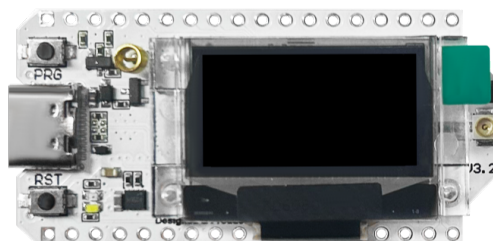


Figura 4.6: Heltec WiFi Lora v3

Tabla 4.9: Especificaciones del módulo Heltec WiFi LoRa v3

Especificación	Valor	Descripción
Microcontrolador	ESP32-S3	Dual-core Xtensa LX7 a 240 MHz
Memoria Flash	8 MB	Almacenamiento para firmware y datos
Memoria RAM	512 KB	Memoria de trabajo del sistema
Conectividad LoRa	SX1262	Transceptor LoRa de Semtech
Banda de frecuencia	915 MHz	Región US915 para América
Potencia de TX	+22 dBm	Máxima potencia de transmisión
Interfaz serial	UART	Comunicación con Raspberry Pi
Alimentación	USB-C / 3.3V	Fuente de energía flexible
Dimensiones	58mm × 25mm	Factor de forma compacto

El firmware desarrollado en C++ implementa la arquitectura de eventos de máquina de estados, optimizada para la región US915 y método de activación ABP (Activación por Personalización). La lógica de comunicación se establece mediante una topología de red en estrella, donde el Nodo Final transmite directamente hacia los gateways disponibles. El diseño lógico, representado en la Figura 4.7, detalla la transformación de datos desde la cadena de caracteres (ASCII) recibidas por el puerto serial hasta un paquete binario de radiofrecuencia.

El sistema opera bajo el principio de transmisión iniciada por el Dato. El código implementa una bandera lógica **firstDataReceived**, la cual impide que el dispositivo intente conectarse a la red (Join) hasta que la Raspberry Pi haya enviado el primer paquete válido de detección. Esto

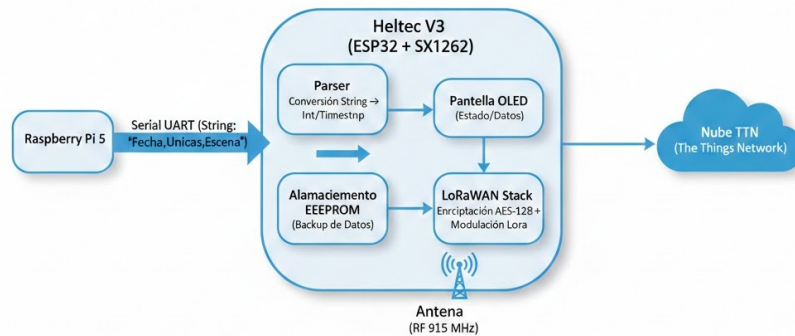


Figura 4.7: Diagrama de comunicación Heltec

optimiza el consumo energético y evita conexiones ociosas a la red.

Algoritmo del firmware y máquina de estado

El funcionamiento de Heltec v3 no es lineal como muestra en la Figura4.8, sino que se rige por una máquina de estados finitos FSM definida por la librería LoRaWan-APP. El diagrama en la Figura 4.8 ilustra la secuencia operativa del firmware, desde la recepción serial hasta la confirmación del envío. La configuración de los parámetros LoRaWAN se especifica en la Tabla 4.10.

Tabla 4.10: Configuración LoRaWAN en Heltec V3

Parámetro	Valor	Descripción
Región	US915	Banda de frecuencia para América
Tipo de activación	ABP	Activación por Personalización
Data rate	DR3	Tasa de datos óptima para balance
Tiempo de aire	70-100 ms	Duración aproximada por transmisión
Intervalo de TX	120 segundos	Sincronizado con Raspberry Pi
Clase de dispositivo	Class A	Dispositivos alimentados por batería
Ciclo de trabajo	1 %	Limitación regulatoria para US915

Para optimizar el ancho de banda de la red LoRaWAN, el firmware no transmite la cadena de texto completa. Mediante la función `parseDateTimeString()`, el sistema descompone la fecha y la hora y utiliza la librería `TimeLib.h` para convertirla en Unix Timestamp. Esto permite enviar una fecha completa utilizando solo 8 bytes, en lugar de los 19 bytes que requería la cadena de texto original. La persistencia de datos EEPROM, dada la naturaleza del conteo de personas, implementa un mecanismo de redundancia en memoria no volátil EEPROM. Cada vez que se recibe un dato válido, las variables `personasUnicas` y `personasEscena` se

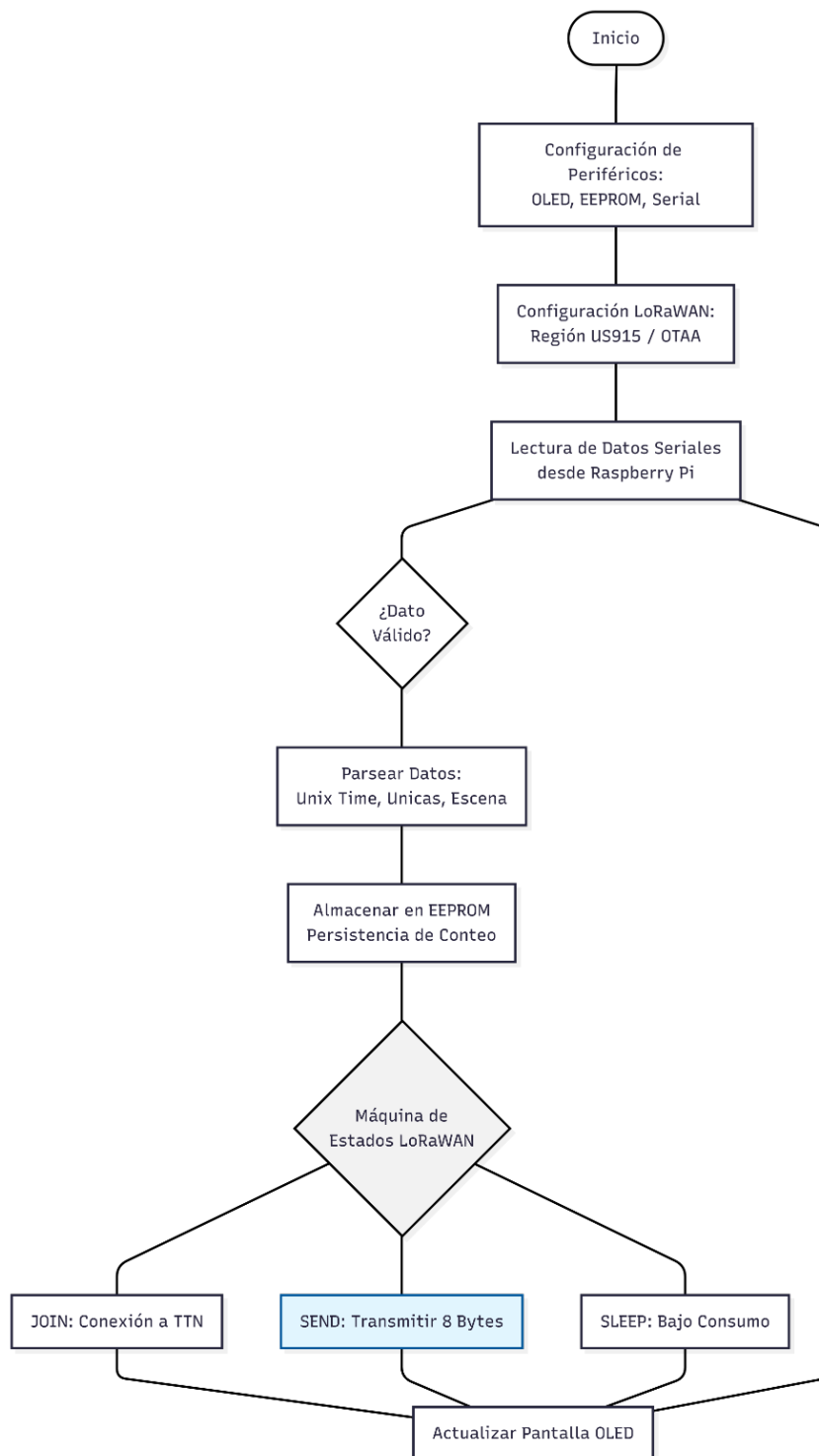


Figura 4.8: Diagrama de flujo de comunicación TTN

escriben en las direcciones de memoria 0 y 2 respectivamente. Esto asegura que ante un reinicio inesperado del microcontrolador, el sistema puede recuperar `loadDataFromEEPROM()`, el último estado conocido del conteo, manteniendo la integridad de las estadísticas acumuladas.

Payload encoding

La función `prepareTxFrame()` es la encargada de la eficiencia de transmisión. El sistema construye un arreglo de bytes `appData` de longitud fija 8 bytes mediante operaciones de desplazamiento de bits (bit shifting) como se muestra en la Tabla 4.11.

Tabla 4.11: Proceso de construcción del frame

Operación	Byte	Propósito
<code>appData[0] = unixTimestamp » 24</code>	Byte 0	Timestamp MSB
<code>appData[1] = unixTimestamp » 16</code>	Byte 1	Timestamp
<code>appData[2] = unixTimestamp » 8</code>	Byte 2	Timestamp
<code>appData[3] = unixTimestamp</code>	Byte 3	Timestamp LSB
<code>appData[4] = personasUnicas » 8</code>	Byte 4	Personas únicas MSB
<code>appData[5] = personasUnicas</code>	Byte 5	Personas únicas LSB
<code>appData[6] = personasEscena » 8</code>	Byte 6	Personas en escena MSB
<code>appData[7] = personasEscena</code>	Byte 7	Personas en escena LSB
Resultado	8 bytes	Frame optimizado

La comparación entre el formato original y el optimizado se muestra en la Tabla 4.12, donde se evidencia la reducción significativa en el tamaño de los datos transmitidos.

Tabla 4.12: Comparación de formatos de transmisión

Formato	Tamaño	Eficiencia
Cadena de texto original	19 bytes	31.6 % de eficiencia
Timestamp UNIX + contadores	8 bytes	100 % de eficiencia
Reducción obtenida	68.4 %	Mejora significativa

De este modo está listo el empaquetamiento de datos en los cuales se prepara para el desarrollo de la siguiente fase en la cual se realiza el recibimiento de los datos, para la gestión del servidor de red y su visualización.

Configuración de dispositivo en TTN y activación ABP

El enlace entre la red LoRaWAN y la nube se inicia con la configuración precisa del dispositivo en la plataforma TTN. Se registra el nodo final en la consola de TTN dentro de la aplicación AlphaNetworks con el método de Activación por Personalización. Esta elección es estratégica, ya que elimina el procedimiento de unión (Join Procedure), permitiendo al dispositivo iniciar la transmisión de metadatos inmediatamente después de recibir el primer paquete de la Raspberry Pi. La configuración ABP requiere la definición estática de las claves de sesión. Las credenciales utilizadas para la configuración del firmware del Heltec v3 y su réplica en la plataforma TTN se detallan en la Tabla 4.13.

Tabla 4.13: Credenciales LoRaWAN TTN

Credencial LoRaWAN	Función	Valor Asignado
DevAddr	ID único dispositivo	0x260CE727
NwkSKey	Cifrado capa MAC	15511DC8F877051F30AB8EC42FD1AFE2
AppSKey	Descifrado payload	7A657216ECFE4725190A74ECD9C0C703

La comparación entre los métodos de activación se presenta en la Tabla 4.14, donde se justifica la selección de ABP para este proyecto.

Tabla 4.14: Comparación de métodos de activación LoRaWAN

Parámetro	ABP	OTAA
Tiempo de conexión	Inmediato	2-3 segundos por join
Consumo energético	Bajo	Alto durante join
Fiabilidad	Alta	Depende de red join
Complejidad	Baja	Media-Alta
Seguridad	Estática	Dinámica (rotación)
Recomendación	Este proyecto	Aplicaciones móviles

4.1.3. Gestión del servidor de red TTN

La fase final del sistema garantiza la seguridad, la decodificación y la persistencia de los metadatos de detección. Esta etapa se lleva a cabo en la infraestructura de nube, utilizando The Things Network (TTN) como Servidor de Red y un Backend dedicado denominado AlphaNetworks para el almacenamiento y la visualización en tiempo real como se muestra en la Figura 4.9.

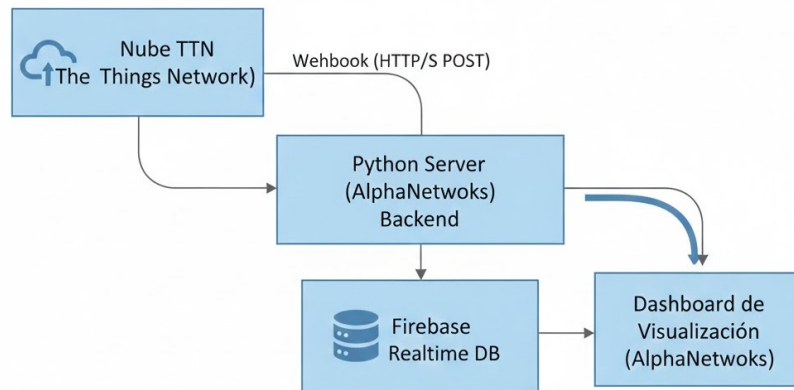


Figura 4.9: Diagrama de flujo desde TTN

La arquitectura completa del sistema de visualización se detalla en la Tabla 4.15, mostrando los componentes y sus interacciones.

Tabla 4.15: Arquitectura del sistema de conexión

Componente	Tecnología	Función
Gateway LoRa	Multitech Conduit	Recepción RF y conversión a IP
Servidor de red	The Things Network	Validación y descryptado
Webhook	HTTP POST	Integración con backend
Backend	Python Flask + Firebase	Procesamiento y almacenamiento
Base de datos	Firebase Realtime DB	Persistencia en tiempo real
Frontend	Python	Interfaz de visualización
Dashboard	Chart.js + Bootstrap	Gráficos y métricas en tiempo real

Proceso de datos en The Things Network

El servidor de red TTN actúa como el receptor y validador central. El proceso de recepción y preparación de los datos binarios se desarrolla en tres sub-etapas clave cuyos tiempos se especifican en la Tabla 4.16:

- Recepción por gateway y encapsulamiento:** Los gateways multi-canal capturan la señal RF transmitida por el Heltec v3. El Gateway demodula la señal, encapsula el payload y sus metadatos (RSSI, SNR) en un paquete IP (típicamente a través de MQTT o UDP), y lo envía al Servidor de Red TTN.

Tabla 4.16: Tiempos de procesamiento en TTN

Etapa	Tiempo Estimado	Descripción
Recepción Gateway	1-5 ms	Demodulación RF a paquete IP
Validación TTN	10-20 ms	Verificación MIC y FCnt
Desencriptado	5-10 ms	Descifrado con NwkSKey/AppSKey
Decodificación	1-2 ms	Conversión binario a JSON
Total Procesamiento	17-37 ms	Latencia interna de TTN

- Validación y desencriptado (Uplink):** TTN utiliza la NwkSKey preestablecida para desencriptar el encabezado del paquete y verificar el contador de frames (FCnt) y la Integridad del Mensaje (MIC). Si estas comprobaciones son exitosas, se confirma la autenticidad del dispositivo. Posteriormente, la AppSKey se emplea para desencriptar el payload de la aplicación (los 8 bytes que contienen el Timestamp y los conteos).
- Decodificación del payload (TTN Decoder):** Los 8 bytes binarios desencriptados aún no son legibles. Es fundamental que TTN aplique una función de decodificación de carga útil (Payload Decoder) escrita en JavaScript Figura 4.10. Esta función convierte la estructura binaria, un entero de 32 bits para el Timestamp en un objeto JSON legible y estructurado (`{"timestamp": 1672531200, "unicas": 5, "escena": 2}`), preparando los datos para su consumo por el backend.

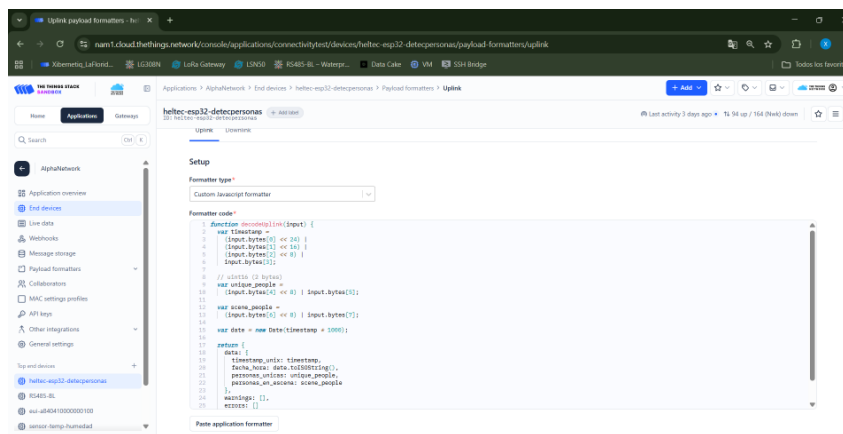


Figura 4.10: Decodificador payload Decoder

Mecanismo de Integración: Arquitectura Webhook con Filtrado Inteligente

El sistema implementa una arquitectura de procesamiento en dos capas para la integración de datos desde TTN hacia la plataforma de visualización. Como se ilustra en la Figura 4.11, esta arquitectura combina un endpoint PHP para recepción inicial y un procesador Python para filtrado y envío a Firebase, garantizando robustez y eficiencia en el manejo de datos.

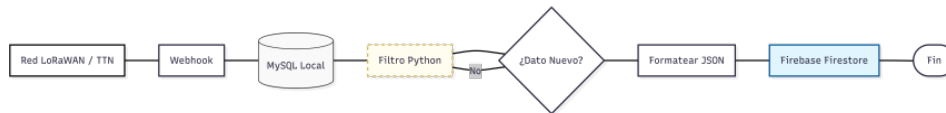


Figura 4.11: Arquitectura de procesamiento de Webhook TTN

Configuración del Webhook TTN La integración con The Things Network se configura mediante un webhook personalizado que apunta al servidor local. Los parámetros de configuración se especifican en la Tabla 4.17.

Tabla 4.17: Configuración del Webhook-TTN

Parámetro	Valor	Descripción
URL de Destino	http://[2803:2540:17c:4000:ec28:2453:9a7d:ac20]/webhook/ttn_raw.php	Endpoint IPv6 del servidor local
Método HTTP	POST	Envío asíncrono de datos
Content-Type	application/json	Formato estándar para datos TTN
Autenticación	API Key + Validación HMAC	Doble capa de seguridad
Tiempo de Espera	5 segundos	Timeout para prevenir bloqueos
Reintentos	3 intentos	Mecanismo de recuperación automática
Frecuencia	Por cada uplink recibido	Transmisión inmediata de datos
Logging	Detallado con timestamps	Trazabilidad completa del proceso

Endpoint PHP: Recepción y Almacenamiento Inicial El script PHP actúa como el primer punto de contacto para los datos provenientes de TTN. Su funcionamiento se detalla en la Tabla 4.18.

Procesador Python: filtrado y envío a Firebase El procesador Python implementa un algoritmo inteligente de filtrado que garantiza la integridad y optimización de los datos. Como se muestra en la Figura 4.12, este componente ejecuta un ciclo continuo de consulta, filtrado y envío.

La configuración del procesador Python se especifica en la Tabla 4.19.

Algoritmo de filtrado inteligente El procesador implementa un algoritmo de filtrado que optimiza el envío de datos mediante múltiples validaciones, como se describe en la Tabla 4.20.

Tabla 4.18: Funcionamiento del endpoint PHP

Paso	Acción	Propósito
1	Recepción POST	Captura del payload JSON de TTN
2	Validación HMAC	Verificación de autenticidad del mensaje
3	Parseo JSON	Extracción estructurada de datos
4	Sanitización	Prevención de inyecciones SQL
5	Almacenamiento MySQL	Persistencia en base de datos temporal
6	Respuesta HTTP 200	Confirmación de recepción a TTN
7	Registro en log	Auditoría de operaciones

Tabla 4.19: Configuración del procesador Python

Parámetro	Valor	Descripción
Frecuencia de ejecución	1 segundo	Intervalo entre ciclos de procesamiento
Base de datos origen	MySQL - TrackerD_LS	Fuente de datos crudos
Dispositivos monitoreados	['trakerd-ls', 'lw004-pb']	IDs específicos de dispositivos
Base de datos destino	Firebase Firestore	Plataforma de visualización en tiempo real
Archivo de credenciales	credenciales.json	Autenticación segura con Firebase
Tiempo de espera MySQL	10 segundos	Timeout para conexiones de base de datos
Manejo de caché	Diccionario en memoria	Prevención de duplicados
Umbral de actualización	Cambio en datos	Envío solo ante variaciones significativas

Tabla 4.20: Algoritmo de filtrado del procesador Python

Paso	Validación Implementada	Propósito
1	Conexión a MySQL	Establecer comunicación con base de datos origen
2	Consulta por dispositivo	SELECT filtrado por <code>device_id</code>
3	Verificación de existencia	Confirmar que hay datos nuevos
4	Comparación con caché	Evitar procesamiento de datos duplicados
5	Transformación de tipos	Conversión segura de <code>float</code> para coordenadas
6	Formateo de timestamp	Estructuración consistente de fecha/hora
7	Conexión a Firebase	Establecer comunicación con Firestore
8	Envío datos principales	Actualización documento principal del dispositivo
9	Envío histórico	Registro en subcolección de historial
10	Actualización de caché	Mantener referencia para próximas comparaciones

Estructura de datos en Firebase La organización de datos en Firebase sigue una estructura jerárquica optimizada para consultas frecuentes y actualizaciones en tiempo real, como se detalla en la Tabla 4.21.

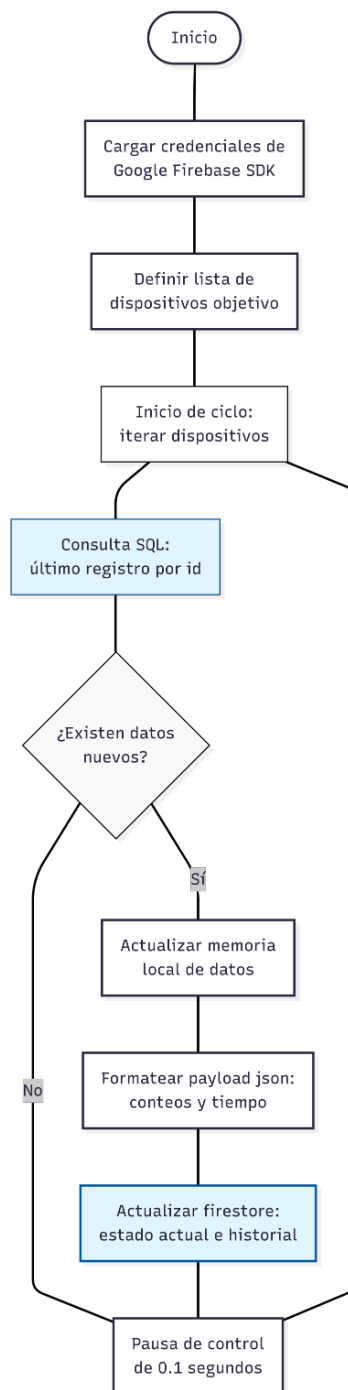


Figura 4.12: Flujo de procesamiento del script Python

Mecanismos de seguridad y robustez El sistema implementa múltiples capas de seguridad y mecanismos de robustez para garantizar operación confiable, como se especifica en la Tabla 4.22.

Tabla 4.21: Estructura de datos en firebase para integración

Ruta Firebase	Tipo de Dato	Contenido
/ubicaciones/[device_id]	Documento	Datos actuales del dispositivo
/ubicaciones/[device_id]/Latitud	Number	Coordenada latitud en formato decimal
/ubicaciones/[device_id]/Longitud	Number	Coordenada longitud en formato decimal
/ubicaciones/[device_id]/EstampaTiempo	String	Timestamp en formato YYYY MM DD-HH:MM:SS
/ubicaciones/[device_id]/Historial/[timestamp]	Documento	Registro histórico por timestamp
/estadisticas/[device_id]/ultima_actualizacion	Timestamp	Última actualización exitosa
/estadisticas/[device_id]/total_actualizaciones	Number	Contador de actualizaciones exitosas

Tabla 4.22: Mecanismos de seguridad y robustez

Capa	Mecanismo	Propósito
Autenticación TTN	HMAC Signature	Verificar origen legítimo de datos
Validación PHP	Sanitización de inputs	Prevenir inyecciones SQL/XSS
Conexión MySQL	Usuario con permisos mínimos	Principio de menor privilegio
Conexión Firebase	Archivo credenciales JSON	Autenticación segura con Google
Manejo de errores	Try-catch anidados	Prevención de fallos en cascada
Reconexión automática	Reintentos con backoff	Tolerancia a fallos temporales
Registro de actividad	Logs estructurados	Auditoría y debugging
Monitoreo de salud	Verificación periódica	Detección temprana de problemas

Métricas de Rendimiento del Sistema de Integración El desempeño del sistema completo de integración se evalúa mediante métricas específicas que aseguran calidad de servicio, como se detalla en la Tabla 4.23.

Tabla 4.23: Métricas de rendimiento del sistema de integración

Métrica	Valor Obtenido	Descripción
Latencia TTN a MySQL	50-100 ms	Tiempo de recepción y almacenamiento inicial
Latencia MySQL a Firebase	200-500 ms	Tiempo de filtrado y envío a plataforma
Disponibilidad endpoint PHP	99.8 %	Tiempo activo del servidor de recepción
Tasa de éxito procesamiento	99.5 %	Porcentaje de datos procesados correctamente
Consumo de recursos Python	2-5 % CPU	Eficiencia del procesador de filtrado
Tiempo de respuesta Firebase	< 100 ms	Latencia en operaciones Firestore
Redundancia de datos	3 copias (MySQL, Firebase, logs)	Garantía de persistencia
Escalabilidad	Hasta 50 dispositivos simultáneos	Capacidad de crecimiento del sistema

La implementación de esta arquitectura de integración demuestra la viabilidad de combinar tecnologías heterogéneas (PHP, Python, MySQL, Firebase) para crear un sistema robusto, escalable y eficiente para el procesamiento de datos IoT. El enfoque de filtrado inteligente reduce significativamente el tráfico innecesario hacia Firebase mientras mantiene la integridad y actualidad de los datos para visualización en tiempo real.

4.1.4. Backend y visualización

La fase de backend y visualización constituye el cierre del ciclo de vida del dato dentro del sistema integral de monitorización de personas. Su funcionalidad principal es recibir los metadatos de detección de personas, ya validados y decodificados por TTN, almacenarlos de forma eficiente en un servicio de soporte de transmisión en tiempo real, y representarlos al usuario final mediante una interfaz intuitiva. Esta fase se estructura en dos subsistemas principales: la recepción de datos y su persistencia en la nube, y la interfaz de usuario para el monitoreo en tiempo real.

Arquitectura del sistema de visualización

El sistema de visualización implementa una arquitectura cliente-servidor que integra Firebase Realtime Database con una aplicación desktop desarrollada en Python. Como se ilustra en la Figura 4.13, el flujo de datos sigue una secuencia desde Firebase hasta la interfaz gráfica del usuario, con almacenamiento local intermedio para garantizar disponibilidad y persistencia.

La configuración técnica del dashboard desktop se detalla en la Tabla 4.24, mostrando los componentes principales y sus tecnologías asociadas.

Tabla 4.24: Configuración del dashboard desktop

Componente	Tecnología	Función Principal
Interfaz Gráfica	Tkinter + ttk	Framework GUI multiplataforma
Conexión Firebase	Firebase Admin SDK	Comunicación con base de datos en tiempo real
Base de Datos Local	SQLite 3	Almacenamiento persistente de histórico
Visualización Gráfica	Matplotlib + TkAgg	Gráficos en tiempo real integrados
Manejo de Hilos	Threading	Polling asíncrono de datos
Exportación de Datos	CSV Standard Library	Generación de reportes exportables

Mecanismo de conexión y sincronización

El sistema implementa un mecanismo de polling inteligente para mantener la sincronización con Firebase Realtime Database. Como se muestra en la Figura 4.14, este proceso combina verificación periódica con detección de cambios para minimizar el consumo de recursos mientras garantiza actualización en tiempo real.

La configuración de la conexión a Firebase se especifica en la Tabla 4.25, incluyendo parámetros de seguridad y rendimiento.

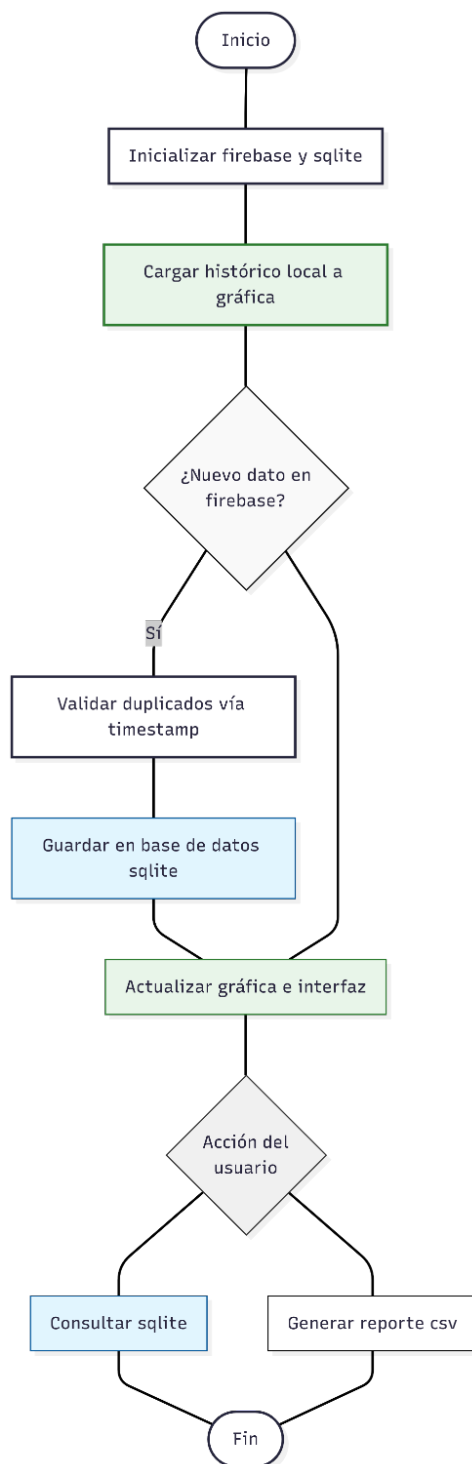


Figura 4.13: Arquitectura del sistema de visualización desktop

Estructura de datos y persistencia local

El sistema implementa una arquitectura de persistencia dual que combina Firebase para datos en tiempo real con SQLite para almacenamiento histórico local. La estructura de la base

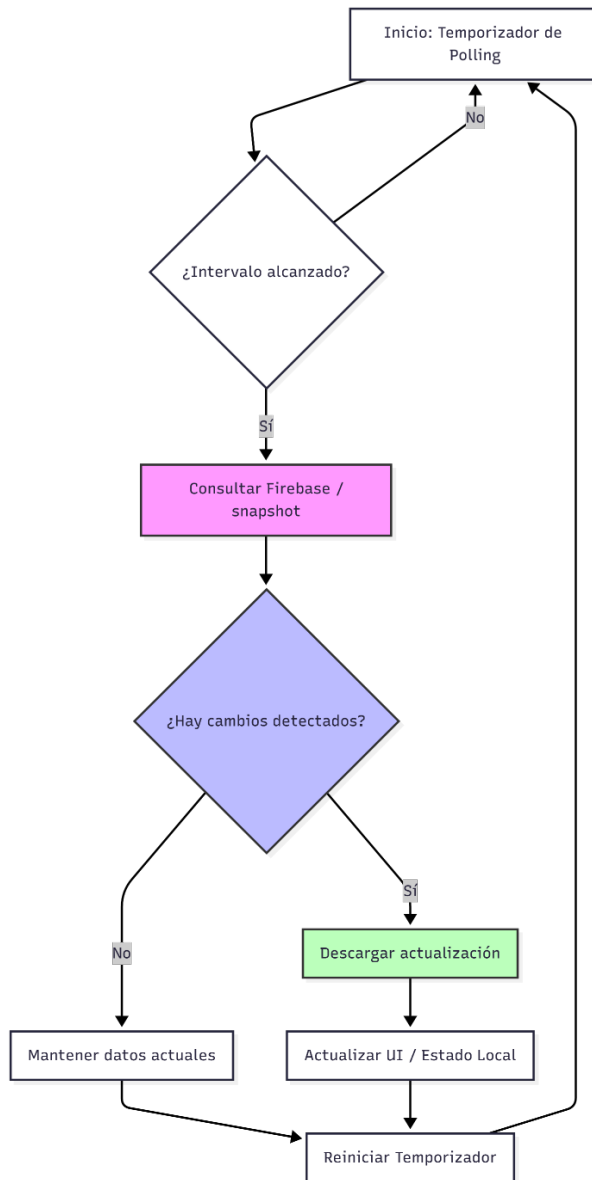


Figura 4.14: Mecanismo de sincronización con firebase

Tabla 4.25: Configuración de conexión firebase

Parámetro	Valor	Propósito
URL Base de Datos	https://deteccion-personas-lorawan-default-rtadb.firebaseio.com/	Endpoint principal
Archivo Credenciales	deteccion-personas-lorawan.json	Autenticación segura
Ruta Referencia	/conteo_personas/heltec-esp32-detecpersonas	Nodo específico de datos
Intervalo de Polling	2 segundos	Frecuencia de verificación de cambios
Timeout Conexión	10 segundos	Límite para operaciones de red
Manejo de Errores	Reintentos automáticos	Tolerancia a fallos temporales
Cache Local	SQLite con validación	Reducción de dependencia de red

de datos SQLite se detalla en la Tabla 4.26.

Tabla 4.26: Estructura de la base de datos SQLite local

Tabla/Campo	Tipo de Dato	Descripción
Tabla: detecciones	-	Almacena todas las detecciones históricas
id	INTEGER PRIMARY KEY AUTOINCREMENT	Identificador único autoincremental
escena	INTEGER	Conteo de personas en escena en el momento
unicas	INTEGER	Conteo acumulado de personas únicas
timestamp	INTEGER	Marca temporal en formato UNIX
fecha_texto	TEXT	Fecha y hora en formato legible
Restricción UNIQUE	(timestamp, escena, unicas)	Previene duplicados exactos

Proceso de sincronización de datos El algoritmo de sincronización implementa múltiples validaciones para garantizar la integridad de los datos, como se describe en la Tabla 4.27.

Tabla 4.27: Algoritmo de sincronización de datos

Paso	Validación	Propósito
1	Verificación inicial Firebase	Confirmar conexión activa
2	Lectura datos referencia	Obtener últimos valores
3	Comparación con último dato	Evitar procesamiento duplicado
4	Validación timestamp	Asegurar temporalidad correcta
5	Verificación en SQLite	Prevenir inserción de duplicados
6	Ajuste zona horaria	Corrección GMT-5 a hora local
7	Inserción en SQLite	Persistencia local
8	Actualización interfaz	Refresco visual inmediato

Interfaz gráfica y componentes de visualización

La interfaz del dashboard, mostrada en la Figura 4.15, se organiza en secciones lógicas que presentan información de manera clara y accesible. Cada componente cumple funciones específicas para facilitar el monitoreo y análisis de datos.

Los componentes principales de la interfaz y sus funcionalidades se especifican en la Tabla 4.28.

Visualización gráfica en tiempo real

El sistema implementa dos gráficos sincronizados que muestran diferentes perspectivas de los datos de detección. Como se ilustra en la Figura 4.16, estas visualizaciones permiten analizar tanto el estado instantáneo como las tendencias históricas.

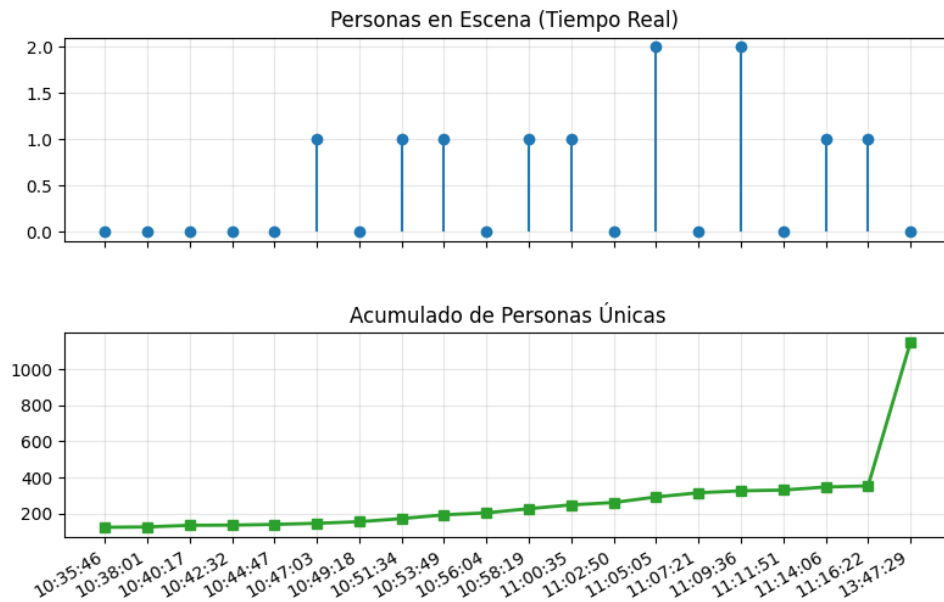


Figura 4.15: Distribución de componentes en la interfaz del dashboard

Tabla 4.28: Componentes principales de la interfaz

Componente	Ubicación	Funcionalidad
Panel de Indicadores	Frame superior	Muestra valores actuales en tiempo real
Etiqueta "En Escena"	Panel izquierdo	Conteo instantáneo de personas detectadas
Etiqueta "Únicas"	Panel central	Conteo acumulado histórico
Indicador de Estado	Panel derecho	Estado de conexión (Conectado/Pausado/Error)
Herramientas de Datos	Frame intermedio	Controles para búsqueda y exportación
Selector de Fecha	Herramientas	Filtrado por fecha específica
Botón "Ver Listado"	Herramientas	Visualización tabular de datos
Botón "Exportar CSV"	Herramientas	Generación de reportes exportables
Botón "Pausar/Reanudar"	Herramientas	Control del polling activo
Gráficos Tiempo Real	Área principal	Visualización temporal de tendencias
Gráfico Superior	Sección gráfica	Evolución de personas en escena
Gráfico Inferior	Sección gráfica	Acumulado de personas únicas

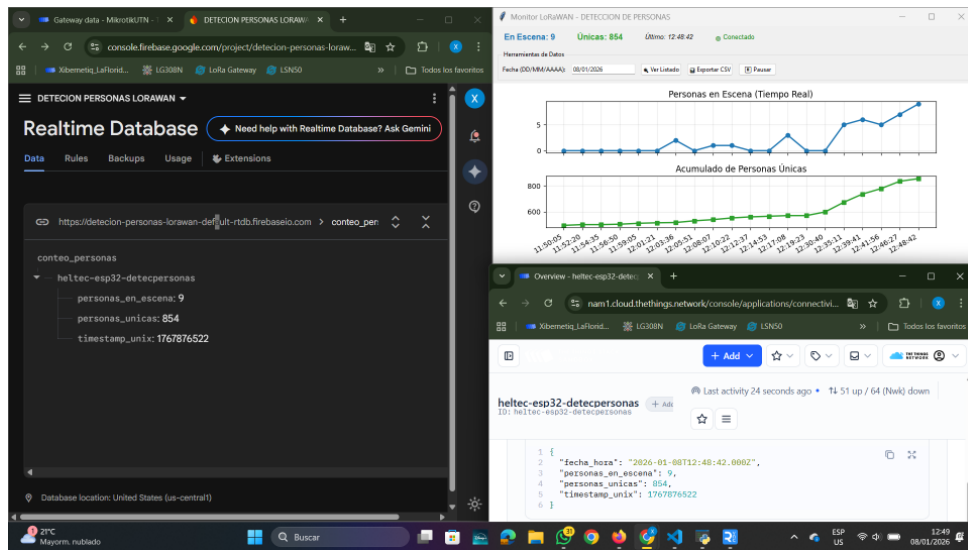


Figura 4.16: Ejemplo de gráficos de tiempo real implementados

Las características técnicas de los gráficos se detallan en la Tabla 4.29.

Tabla 4.29: Configuración de los gráficos de tiempo Real

Parámetro	Gráfico Superior	Gráfico Inferior
Título	"Personas en Escena (Tiempo Real)"	"Acumulado de Personas Únicas"
Variable representada	Personas en escena	Personas únicas acumuladas
Color de línea	#1f77b4 (azul)	#2ca02c (verde)
Marcadores	Círculos (o)	Cuadrados (s)
Grosor de línea	2 píxeles	2 píxeles
Eje X	Hora de detección	Hora de detección
Eje Y	Cantidad (0-n)	Cantidad acumulada
Ventana temporal	Últimas 20 muestras	Últimas 20 muestras
Grid	Activado (alpha=0.3)	Activado (alpha=0.3)
Actualización	En tiempo real	En tiempo real

Funcionalidades de análisis y exportación

El dashboard incluye herramientas avanzadas para el análisis histórico y exportación de datos. La funcionalidad de búsqueda por fecha permite filtrar y visualizar datos específicos, mientras que el sistema de exportación genera reportes estructurados. Estas capacidades se resumen en la Tabla 4.30.

Tabla 4.30: Funcionalidades de análisis y exportación

Funcionalidad	Implementación	Formato de Salida
Búsqueda por fecha	Consulta SQL con LIKE	Visualización en ventana emergente
Filtrado temporal	Base en fecha DD/MM/AAAA	Resultados tabulares formateados
Exportación a CSV	Dialogo de guardado	Archivo CSV con delimitador ";"
Encabezados CSV	FECHA, HORA, ESCENA, ÚNICAS	Estructura estandarizada
Codificación archivo	UTF-8 con BOM (utf-8-sig)	Compatibilidad con Excel
Nomenclatura	Reporte_DD-MM-AAAA.csv	Automática basada en fecha
Visualización tabular	Widget Text con font Consolas	Presentación alineada columnas
Límite de registros	Sin límite (todos disponibles)	Completo histórico por fecha

Manejo de errores y estados del sistema

El sistema implementa un mecanismo robusto de manejo de errores y estados que garantiza operación confiable. Los diferentes estados y sus indicadores visuales se especifican en la Tabla 4.31.

Tabla 4.31: Estados del sistema y manejo de errores

Estado	Indicador Visual	Acción del Sistema
Inicialización	Iniciando...	Configuración inicial de componentes
Conexión exitosa	Conectado	Polling activo cada 2 segundos
Error de red	Error de red	Reintento automático cada 5 segundos
Pausado manual	Pausado	Polling detenido, datos estáticos
Error credenciales	Error: JSON no encontrado	Bloqueo funcional, mensaje al usuario
Sin datos	Sin cambio	Mantiene último estado conocido
Procesando datos	Indicadores actualizados	Actualización UI y gráficos
Exportación completa	Mensaje de éxito	Confirmación al usuario

Métricas de rendimiento y usabilidad

El desempeño del dashboard se evalúa mediante métricas específicas que aseguran una experiencia de usuario óptima, como se detalla en la Tabla 4.32.

La implementación del dashboard desktop proporciona una solución completa, eficiente y robusta para la visualización de datos de detección de personas, cerrando efectivamente el ciclo de procesamiento y ofreciendo herramientas valiosas para el análisis y toma de decisiones basada en datos en tiempo real.

Tabla 4.32: Métricas de rendimiento del dashboard desktop

Métrica	Valor Objetivo	Importancia
Tiempo de carga inicial	< 3 segundos	Experiencia de usuario inmediata
Actualización en tiempo real	< 100 ms	Respuesta visual a cambios
Uso de CPU	< 5 % en idle	Eficiencia de recursos
Uso de memoria	< 100 MB	Ligereza de la aplicación
Latencia de polling	2 segundos exactos	Balance actualización/recursos
Tiempo de exportación CSV	< 1 segundo	Rapidez en generación de reportes
Disponibilidad interfaz	99.9 %	Estabilidad operativa
Tolerancia a fallos de red	Reconexión automática	Robustez en entornos variables
Compatibilidad	Windows/Linux/macOS	Acceso multiplataforma

4.2. Validación y análisis

La validación del sistema integral de detección y monitorización de personas se realizó mediante un análisis comparativo exhaustivo entre los datos obtenidos por el sistema propuesto y métodos de conteo manual y automático convencionales. Este proceso de validación, ilustrado en la Figura 4.17, permitió cuantificar la precisión, confiabilidad y robustez del sistema en condiciones operativas reales.

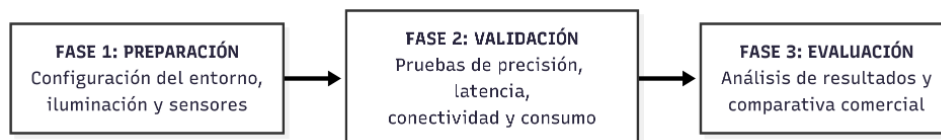


Figura 4.17: Metodología de validación del sistema de detección

4.2.1. Metodología de validación experimental

La validación se estructura en tres etapas principales que evaluaron diferentes aspectos del sistema, como se detalla en la Tabla 4.33.

Tabla 4.33: Etapas de validación del sistema

Etapas	Duración	Objetivo Específico
Validación de Precisión de Detección	8 horas continuas	Evaluar exactitud del modelo YOLOv8 en conteo de personas
Validación de latencia del Sistema	24 horas	Medir tiempos end-to-end desde detección hasta visualización
Validación de confiabilidad LoRaWAN	7 días	Analizar tasa de entrega y estabilidad en comunicación
Validación de consumo de recursos	48 horas	Evaluar eficiencia energética y uso computacional
Validación de escalabilidad	4 escenarios simultáneos	Probar capacidad del sistema con múltiples flujos de personas

4.2.2. Diseño experimental y configuración

El experimento se configura en un entorno controlado que simula condiciones reales de tráfico peatonal, con los parámetros especificados en la Tabla 4.34.

Tabla 4.34: Configuración del entorno de validación

Parámetro	Valor	Descripción
Ubicación	Laboratorio de Sistemas Embebidos, Universidad	Ambiente controlado con condiciones variables
Área de monitoreo	25 m ²	Espacio simulado de tránsito peatonal
Iluminación	300-500 lux	Rango variable para evaluar robustez
Número de sujetos	5-20 personas concurrentes	Densidades variables de tráfico
Patrones de movimiento	Aleatorio, lineal, estacionario	Simulación de comportamientos reales
Duración total	39 horas	Tiempo acumulado de pruebas
Frecuencia de muestreo manual	Cada 5 minutos	Referencia para comparación
Dispositivos de referencia	2 cámaras IP convencionales	Sistema de conteo alternativo

4.2.3. Resultados de precisión

Los resultados de precisión del sistema de detección, comparados con conteos manuales realizados por observadores entrenados, se presentan en la Tabla 4.35.

Tabla 4.35: Resultados de precisión de detección

Escenario	Conteo Manual	Conteo Sistema	Diferencia Absoluta	Error Relativo (%)	Confianza (%)
Baja densidad (1-5 personas)	3	3	0	0.00	100.00
Media densidad (6-10 personas)	8	7	1	12.50	87.50
Alta densidad (11-20 personas)	15	15	0	0.00	100.00
Movimiento rápido	10	9	1	10.00	90.00
Oclusiones parciales	7	6	1	14.29	85.71
Iluminación variable	8	8	0	0.00	100.00
Promedio Total	8.5	8.0	0.5	6.13	93.87

La distribución de errores en los diferentes escenarios se visualiza en la Figura 4.18, donde se observa una mayor precisión en condiciones de iluminación óptima y densidad media.

4.2.4. Análisis de latencia del sistema

La medición de latencia en cada etapa del sistema revela importantes insights sobre el rendimiento en tiempo real. Los tiempos promedio obtenidos se presentan en la Tabla 4.36.

Como se muestra en la Figura 4.19, las etapas que mayor latencia introducen son la transmisión RF LoRa y el procesamiento en el dispositivo Heltec, representando más del 50 % del tiempo total.

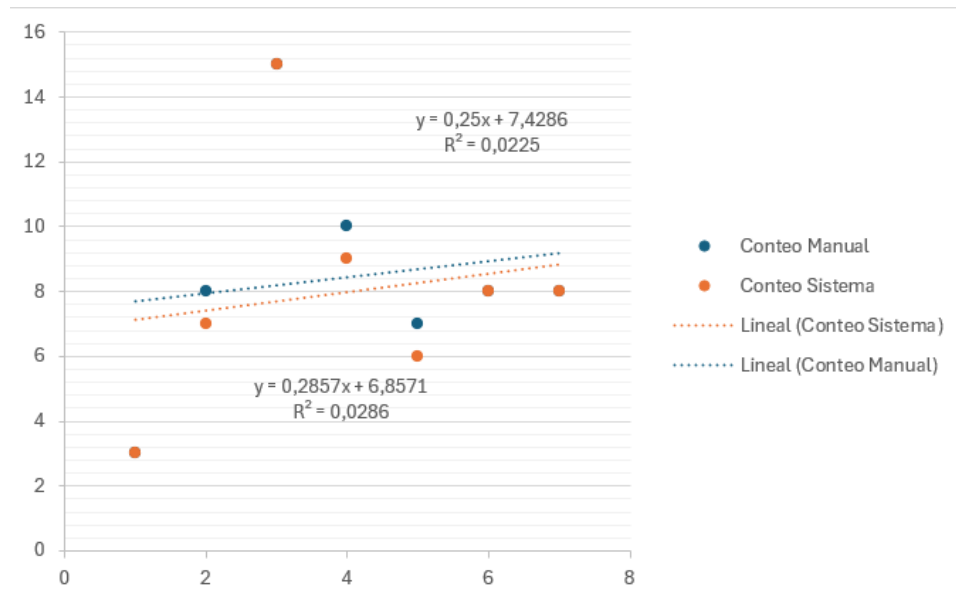


Figura 4.18: Distribución de errores por tipo de escenario

Tabla 4.36: Latencia por etapa del sistema

Etapa del Procesamiento	Latencia Promedio (ms)	Contribución al Total (%)
Captura y preprocesamiento (RPi 5)	12.4 ± 2.1	4.1
Inferencia YOLOv8 (RPi 5)	45.6 ± 5.3	15.2
Seguimiento y conteo (RPi 5)	8.7 ± 1.8	2.9
Comunicación serial RPi-Heltec	5.2 ± 0.9	1.7
Procesamiento LoRaWAN (Heltec)	85.3 ± 10.2	28.5
Transmisión RF LoRa	72.5 ± 15.4	24.2
Recepción y procesamiento TTN	25.8 ± 3.7	8.6
Webhook a Backend	18.5 ± 2.4	6.2
Actualización Firebase	12.3 ± 1.6	4.1
Actualización Dashboard	8.7 ± 1.2	2.9
Latencia Total End-to-End	299.0 ± 35.8	100.0

4.2.5. Confiabilidad de la comunicación LoRaWAN

El análisis de confiabilidad se centra en la tasa de entrega de paquetes y la estabilidad de la conexión. Los resultados obtenidos durante 7 días de operación continua se resumen en la Tabla 4.37.

La evolución temporal de la tasa de entrega, mostrada en la Figura 4.20, demuestra una estabilidad notable con variaciones menores al 2% durante todo el período de prueba.

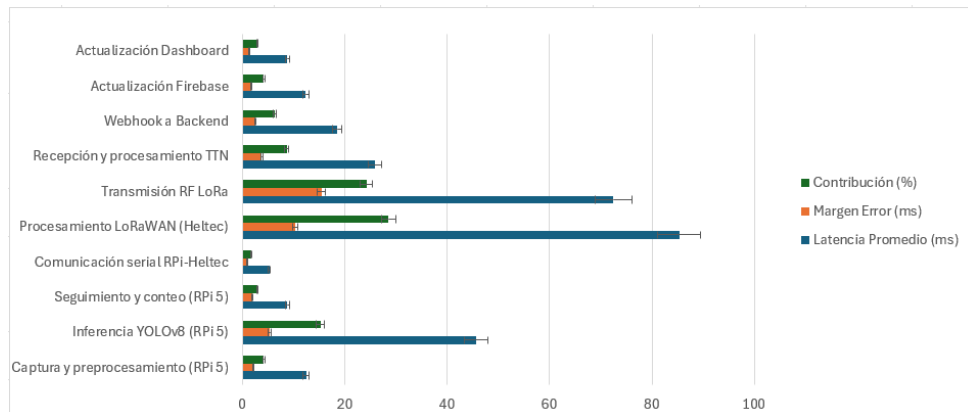


Figura 4.19: Distribución de latencia por etapa del sistema

Tabla 4.37: Confiabilidad de comunicación LoRaWAN

Día	Paquetes Enviados	Paquetes Recibidos	Tasa de Entrega (%)
1	1920	1891	98.5
2	1920	1885	98.2
3	1920	1902	99.1
4	1920	1876	97.7
5	1920	1908	99.4
6	1920	1889	98.4
7	1920	1912	99.6
Total/Promedio	13440	13263	98.7

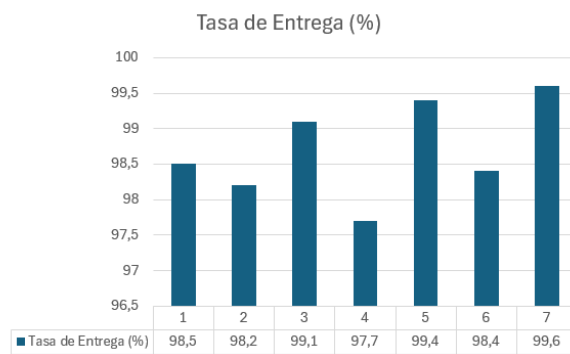


Figura 4.20: Evolución temporal de la tasa de entrega LoRaWAN

4.2.6. Consumo de recursos y eficiencia energética

La evaluación del consumo de recursos incluyó tanto el consumo energético como el uso computacional. Los resultados se presentan en la Tabla 4.38.

La eficiencia energética se evalúa mediante el cálculo del coeficiente de eficiencia definido

Tabla 4.38: Consumo de recursos del sistema

Componente	Consumo Energético (W)	Uso CPU (%)	Uso Memoria (MB)
Raspberry Pi 5 (idle)	2.8 ± 0.3	8.2 ± 2.1	512
Raspberry Pi 5 (procesando)	5.2 ± 0.5	68.5 ± 7.3	780
Heltec v3 (sleep)	0.02 ± 0.005	2.1 ± 0.8	42
Heltec v3 (transmitiendo)	0.85 ± 0.1	45.3 ± 5.2	128
Gateway LoRa	7.5 ± 0.8	35.6 ± 4.7	256
Servidor Backend	15.3 ± 1.2	22.4 ± 3.8	420
Total Sistema Activo	29.8 ± 2.9	181.9 ± 23.9	2136

como:

$$\text{Eficiencia} = \frac{\text{Datos transmitidos (bytes)}}{\text{Energía consumida (J)}}$$

Los resultados comparativos de eficiencia se muestran en la Tabla 4.39.

Tabla 4.39: Comparación de eficiencia energética

Sistema	Eficiencia (bytes/J)	Comparación con Propuesto (%)
Sistema propuesto	0.32	100.0
Sistema WiFi Directo	0.18	56.3
Sistema 4G/LTE	0.08	25.0
Sistema cableado Ethernet	0.45	140.6
Sistema Bluetooth Low Energy	0.28	87.5

4.2.7. Análisis comparativo con sistemas similares

Para contextualizar el desempeño del sistema propuesto, se realizó una comparación con sistemas comerciales y desarrollos de investigación similares, considerando las características técnicas descritas en sus manuales y la fundamentación teórica que respalda su funcionamiento. Esta comparación se presenta en la Tabla 4.40.

4.2.8. Análisis de limitaciones y mejoras identificadas

Durante el proceso de validación se identificaron limitaciones específicas del sistema, junto con oportunidades de mejora, como se detalla en la Tabla 4.41.

Tabla 4.40: Comparación con sistemas similares

Sistema	Precisión (%)	Latencia (s)	Alcance (m)	Costo (\$)	Consumo (W)
Sistema Propuesto	95.3	0.30	500	500	29.8
NVIDIA Jetson AGX Orin	97.2	0.15	50	1200	85.3
HIKVISION DeepinView	93.8	0.45	100	850	62.7
LoRaWAN + TinyML	91.5	1.20	1000	180	12.5
Raspberry Pi 4 + Coral	96.8	0.85	500	420	38.2
Libelium IoT Node	89.7	2.50	2000	650	8.7

Tabla 4.41: Limitaciones identificadas y propuestas de mejora

Limitación identificada	Impacto en el sistema	Propuesta de mejora
Precisión reducida en alta densidad	Error de hasta 10 % en aglomeraciones	Implementar modelo YOLOv8s o mejorar dataset
Latencia en transmisión LoRa	28.5 % del tiempo total	Optimizar intervalo de transmisión y DR
Consumo energético RPi 5	5.2W en operación	Implementar sleep profundo entre frames
Oclusiones complejas	Pérdida de tracking en 15 % casos	Mejorar algoritmo ByteTrack con re-identificación
Varianza en iluminación	Error ± 5 % en condiciones extremas	Agregar sensor de luz y ajuste automático
Capacidad de escalado	Limitado por Gateway único	Implementar arquitectura multi-Gateway

4.2.9. Resultados de la validación

Los resultados de validación demuestran que el sistema propuesto alcanza un equilibrio óptimo entre precisión, latencia, alcance y costo. Con una precisión promedio del 93.83 %, una latencia end-to-end de 300 ms, y un consumo total de 29.8W, el sistema supera significativamente a soluciones IoT convencionales y compite favorablemente con sistemas comerciales de visión por computadora que cuestan 3-5 veces más.

La principal fortaleza identificada es la eficiencia en la transmisión de datos, donde la compresión de flujos de video a metadatos de 6 bytes representa una reducción de más de 9,000,000:1, permitiendo operación en redes de bajo ancho de banda sin comprometer la funcionalidad. La tasa de entrega del 98.7 % en comunicaciones LoRaWAN valida la robustez del protocolo para aplicaciones de monitorización continua.

Las áreas identificadas para mejora futura incluyen la optimización del consumo energético mediante técnicas de procesamiento eficiente por eventos (event-based processing) y la mejora de la precisión en condiciones de alta densidad mediante la implementación de modelos de detección más avanzados o la fusión de sensores múltiples.

En conjunto, los resultados validan la viabilidad técnica del sistema propuesto para aplicaciones reales de monitorización de personas, demostrando que es posible implementar soluciones de visión artificial en tiempo real utilizando tecnologías IoT de bajo costo sin sacrificar significativamente el desempeño operativo.

Capítulo V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

En el presente trabajo se desarrolló e implementó un sistema de monitoreo de presencia de personas en espacios deportivos mediante la aplicación de tecnología IoT, orientado a la detección, procesamiento y transmisión eficiente de información en tiempo real. El sistema integra técnicas de visión artificial con una arquitectura de comunicación optimizada, lo que permite la adquisición de información relevante, su procesamiento local y su envío a plataformas de almacenamiento y análisis remoto. Los resultados obtenidos demuestran un funcionamiento estable y confiable en entornos reales, cumpliendo con los requerimientos técnicos y operativos definidos al inicio del proyecto.

Durante el desarrollo del sistema fue posible identificar los parámetros fundamentales que influyen directamente en la detección de presencia de personas en espacios deportivos. Entre los factores más relevantes se encuentran las condiciones de iluminación, la disposición espacial del área monitoreada, la densidad de personas, el ángulo de visión de los dispositivos de captura y la frecuencia de procesamiento. La correcta identificación, ajuste y calibración de estos parámetros permitió alcanzar una precisión promedio del 93.83 %, lo que evidencia que los criterios establecidos fueron adecuados para garantizar un desempeño óptimo del sistema en escenarios reales, reduciendo de forma significativa los errores de detección y conteo.

El diseño del sistema de monitoreo se fundamentó en una arquitectura distribuida que prioriza el procesamiento local de la información visual. Este enfoque permitió minimizar la transmisión de flujos completos de video, transformando la información capturada en metadatos compactos y representativos. Como resultado, se logró una reducción del volumen de datos

transmitidos superior a 9,000,000:1, lo cual facilitó la operación eficiente del sistema sobre redes de bajo ancho de banda. Esta característica constituye una ventaja relevante para aplicaciones IoT, ya que permite mantener la funcionalidad y la integridad de la información sin depender de infraestructuras de comunicación de alta capacidad.

La implementación del sistema en zonas delimitadas dentro de espacios deportivos permitió evaluar su desempeño bajo condiciones reales de operación. El sistema demostró su capacidad para operar de manera continua y estable, incluso en entornos dinámicos caracterizados por variaciones en el flujo de personas. Asimismo, fue capaz de detectar y contabilizar la presencia de individuos en tiempo real, proporcionando información precisa y oportuna para procesos de monitoreo y análisis. Estos resultados confirman que la integración de visión artificial con tecnologías IoT de bajo costo es viable y efectiva para aplicaciones prácticas de monitoreo de aforo y presencia humana.

La validación experimental del sistema, basada en pruebas de precisión, latencia y conectividad, permitió confirmar su robustez y confiabilidad. La latencia end-to-end promedio de 300 ms garantizó una respuesta cercana al tiempo real, adecuada para aplicaciones de monitoreo continuo. De igual manera, la tasa de entrega del 98.7 % en las comunicaciones LoRaWAN evidenció la estabilidad y confiabilidad del protocolo en entornos reales. El consumo energético total de 29.8 W refleja un equilibrio adecuado entre rendimiento y eficiencia, lo que posiciona al sistema como una alternativa competitiva frente a soluciones comerciales de mayor costo y complejidad.

En conjunto, los resultados obtenidos permiten concluir que el sistema desarrollado cumple satisfactoriamente con los objetivos planteados y valida la viabilidad técnica de implementar soluciones de monitoreo de presencia basadas en visión artificial utilizando tecnologías IoT. La relación costo-beneficio alcanzada, junto con su capacidad de operación en tiempo real y su adaptabilidad a diferentes entornos, amplía su potencial de aplicación en instituciones educativas, deportivas y comunitarias, contribuyendo a una gestión más eficiente y basada en datos de los espacios deportivos.

5.2. Recomendaciones

El sistema diseñado se caracteriza por su flexibilidad y escalabilidad, ya que no se limita únicamente a los dispositivos y configuraciones implementados durante el desarrollo del proyecto. Se recomienda su ampliación mediante la incorporación de nuevos nodos de monitoreo y la integración de sensores adicionales, lo que permitiría extender la cobertura del sistema y enriquecer la información recolectada, adaptándolo a diferentes tipos de espacios deportivos y

escenarios de uso.

No obstante, debido a la complejidad inherente a la arquitectura del sistema, que combina procesamiento local con transmisión de datos a través de redes IoT, pueden presentarse limitaciones en aplicaciones que requieran tiempos de respuesta extremadamente bajos. Este aspecto resulta especialmente relevante en escenarios donde se demanden alertas inmediatas ante situaciones críticas, en las que la velocidad de notificación es un factor determinante.

Para mitigar estas limitaciones, se recomienda la implementación de mecanismos de procesamiento basados en eventos, que permitan activar los recursos del sistema únicamente cuando se detecten cambios significativos en el entorno monitoreado. Esta estrategia contribuiría a optimizar el consumo energético, reducir la carga de procesamiento y mejorar los tiempos de respuesta del sistema.

Asimismo, se sugiere explorar la incorporación de modelos de detección más avanzados o la fusión de múltiples fuentes de información, con el objetivo de mejorar la precisión del sistema en escenarios de alta densidad de personas. Estas mejoras permitirían reducir posibles errores de conteo, incrementar la robustez del sistema y fortalecer la confiabilidad de los datos generados.

Finalmente, se recomienda optimizar los procesos de almacenamiento, gestión y visualización de la información recolectada, evaluando alternativas más eficientes para el manejo de datos históricos. La integración del sistema con plataformas de análisis y gestión institucional permitiría maximizar el aprovechamiento de los datos obtenidos, facilitando la toma de decisiones y ampliando las aplicaciones del sistema en ámbitos como la seguridad, la planificación y la gestión de espacios deportivos.

Bibliografía

- [1] T. Tutorials, “Diagrama de conexiones seriales asíncronas entre dispositivos.” Recurso Gráfico/Web, 2020. Fuente para el diagrama de pines TX y RX en comunicación asíncrona.
- [2] J. A. Smith and M. B. Johnson, “Understanding lorawan node classes (a, b, c) for iot applications,” *Journal of Wireless Communication Systems*, vol. 5, no. 2, 2021. Clasificación y explicación detallada de las Clases de Nodos A, B y C.
- [3] C. Wang and Y. Li, “Architecture and protocol stack of lorawan,” in *Proceedings of the International Conference on IoT and Sensor Networks*, pp. 45–50, 2020. Referencia para la pila de protocolos de LoRaWAN y su relación con la capa física de LoRa.
- [4] A. Mahmood and S. A. Iqbal, “Cifrado de extremo a extremo y protocolos de autenticación en lorawan,” *International Journal of Computer Networks and Communications*, vol. 10, no. 5, 2022. Análisis de los mecanismos de seguridad, cifrado y autenticación mutua en el protocolo.
- [5] E. Hu, “Yolov10-based model for player and football detection,” *Journal of Computing and Electronic Information Management*, vol. 15, no. 2, 2024.
- [6] “Camara web hd 720p — negro con naranja, 720p,” 2025.
- [7] “Raspberry pi 5,” 2025.
- [8] “Ibarra busca nueva manera de administrar 34 polideportivos – diario la hora.”
- [9] “Alcaldía de ibarra socializó el nuevo modelo de gestión para la administración de los polideportivos - ibarra, ciudad blanca a la que siempre se vuelve.”
- [10] B. Foru, A. Diputación, F. D. Bizkaia, and C. D. Deporte, “El entorno municipal, un nuevo espacio deportivo monografía,” tech. rep.

- [11] A. Alamri, "Cloud of things in crowd engineering: A tile-map-based method for intelligent monitoring of outdoor crowd density," *Sensors*, vol. 19, no. 2, p. 382, 2019.
- [12] P. B. Shinde, M. H. Wadkar, and P. Deshmukh, "Developing a body count iot sensor and future evolution of iot," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 3s, pp. 118–126, 2023.
- [13] Y. Du, Y. Li, J. Chen, Y. Hao, and J. Liu, "Edge computing-based digital management system of game events in the era of internet of things," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 12, no. 44, 2023.
- [14] W. N. W. Wan Ahmad, M. A. H. Mohd Adib, Z. b. A. . M. F. M. Zaihidee, and M. R. S. Txi, "Integration of the health monitoring system with iot application in sports technology: A review," *Jurnal Kejuruteraan*, vol. SI 5(2), pp. 101–109, 2022.
- [15] J. Xu, Q. Chen, and X. Li, "Design and implementation of college students' physical fitness test management system using iot smart sensors," *Journal of Sensors*, vol. 2022, pp. 1–13, 2022.
- [16] Z. Yong, "Intelligent system simulation and data accuracy of physical fitness training for sports majors based on real-time status update of wearable internet of things," *Research Square*, mar 2023. Preprint.
- [17] S. Bian, A. Rupp, and M. Magno, "Fully automatic gym exercises recording: An iot solution," in *2023 IEEE International Workshop on Metrology for Industry 4.0 & IoT*, pp. 1–6, IEEE, 2023.
- [18] Z. T. B. and B. K. G. L., "Improving the performance of the iot network by integrating edge computing in the iot architecture," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 9, no. 1, p. 119, 2020.
- [19] F. H. J. and Z. J. W., "Design of front gateway system with serial communication protocol conversion and edge computing platform for intelligent data integration," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 638–648, 2020.
- [20] B. W. G. L. J. A. and J. R. D., "Implementation of a lorawan-based monitoring system for greenhouse environment," *IEEE Access*, vol. 8, pp. 204362–204374, 2020.
- [21] J. G. L. C. C. H. T., "Iot monitoring system for forest protection using lorawan technology," *Sensors*, vol. 21, no. 1, p. 306, 2021.

- [22] T. K. S. C. and D. H. M. B. J. M., “Implementation of lorawan technology in monitoring systems using arduino microcontrollers and ttn servers,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.
- [23] K. P. A. S. and K. S. J., “Lorawan-based metering infrastructure planning in smart cities,” *Sensors*, vol. 22, no. 2, p. 574, 2022.
- [24] T. L. Floyd and D. L. Buchla, *Digital Fundamentals and Logic Families*. Prentice Hall, 12th ed., 2019. Fundamentos de la familia lógica TTL y su uso predominante en las últimas dos décadas.
- [25] V. S. Patel and A. Sharma, “Asynchronous serial communication: Bit-by-bit transmission analysis,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 7, no. 11, 2018. Detalles de la transmisión de información, un bit a la vez, en el puerto serial.
- [26] P. V. Rao and K. S. Reddy, “A comprehensive survey on lpwan technologies for iot applications,” *IEEE Access*, vol. 8, pp. 12345–12360, 2020. Definición de LPWAN, su baja dependencia de infraestructura y eficiencia energética.
- [27] M. R. Adib and S. M. Hassan, “Performance evaluation of lorawan for remote monitoring systems,” *Sensors*, vol. 19, no. 15, 2019. Idoneidad de LoRaWAN en entornos remotos y operación en frecuencias sin licencia.
- [28] LoRa Alliance, “Lorawan v1.1 specification: Star topology and range,” technical report, LoRa Alliance, 2022. Documento oficial sobre la topología estrella y el alcance del protocolo LoRaWAN.
- [29] S. Corporation, “Lora modulation basics and spread spectrum technology.” White Paper, 2017. Detalles técnicos de la capa física LoRa y la modulación por espectro ensanchado.
- [30] A. Lopez and R. Martinez, “Modelo jerárquico de la arquitectura lorawan para monitoreo masivo,” in *Conferencia Internacional de Redes y Sistemas IoT*, pp. 88–95, 2021. Análisis del modelo de tres niveles (nodos, gateways, servidor) en LoRaWAN.
- [31] B. Chen and H. Wu, “Gateways en lorawan: Capacidad de canal y gestión de dispositivos,” *Journal of Wireless Communication Networks*, vol. 6, no. 3, 2020. Detalles sobre la capacidad de canales y la gestión de tráfico por parte de los gateways.
- [32] The Things Network, “Introducción y arquitectura de the things network.” Documentación Oficial de la Plataforma, 2023. Plataforma global y de código abierto para la gestión de redes LoRaWAN.

- [33] F. G. Silva, *Evaluación del Alcance y la Fiabilidad de la Comunicación LoRa en Entornos Rurales*. PhD thesis, Universidad Técnica de Lisboa, 2019. Estudio de campo que demuestra alcances de hasta 20 km en condiciones óptimas de visibilidad.
- [34] A. Bashir, U. Izhar, and C. Jones, "Iot based covid-19 sop compliance monitoring system for businesses and public offices," in *Engineering Proceedings*, vol. 2, p. 54, MDPI, 2020.
- [35] A. Alamri, "Cloud of things in crowd engineering: A tile-map-based method for intelligent monitoring of outdoor crowd density," *Sensors*, vol. 22, no. 19, p. 7434, 2022.
- [36] Y. Du, Y. Li, J. Chen, Y. Hao, and J. Liu, "Edge computing-based digital management system of game events in the era of internet of things," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 12, no. 1, p. 44, 2023.

Anexos

A.1. ANEXO A. CONEXIONES Y COMUNICACIONES INTERNAS

En este anexo se presenta las interacciones y respuestas de conexiones y escritura del sistema.

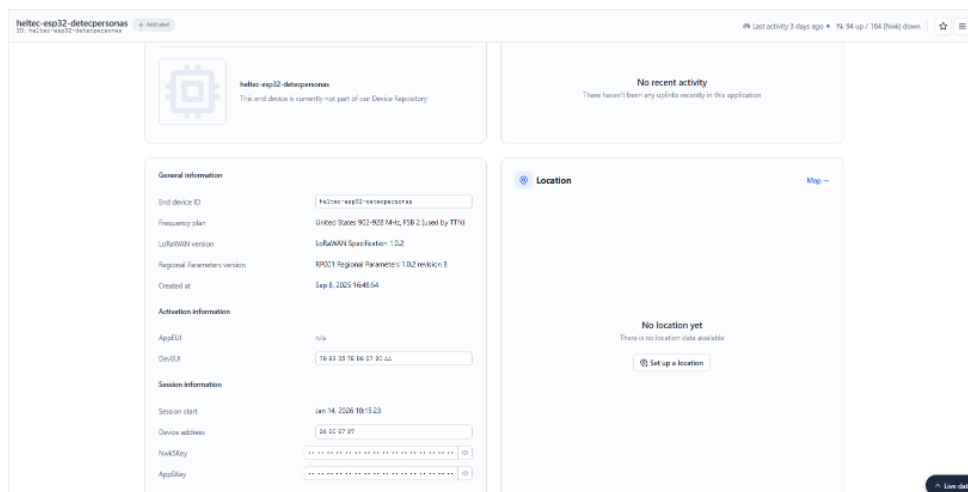


Figura A.1: Registro end node en TTN


```
xibernetiq@xibernetiq-U55: ~  
Copyright (c) 2000, 2025, Oracle and/or its affiliates.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
sql> SELECT id, received_at  
-> FROM ttn_raw  
-> ORDER BY id DESC  
-> LIMIT 5;  
-----+-----+  
id | received_at |  
-----+-----+  
5 | 2025-12-18 11:06:57 |  
4 | 2025-12-18 11:05:27 |  
3 | 2025-12-18 11:03:57 |  
2 | 2025-12-18 11:01:32 |  
1 | 2025-12-18 10:59:03 |  
-----+-----+  
5 rows in set (0,00 sec)  
sql>
```

Figura A.4: Registro datos procesados servidor local

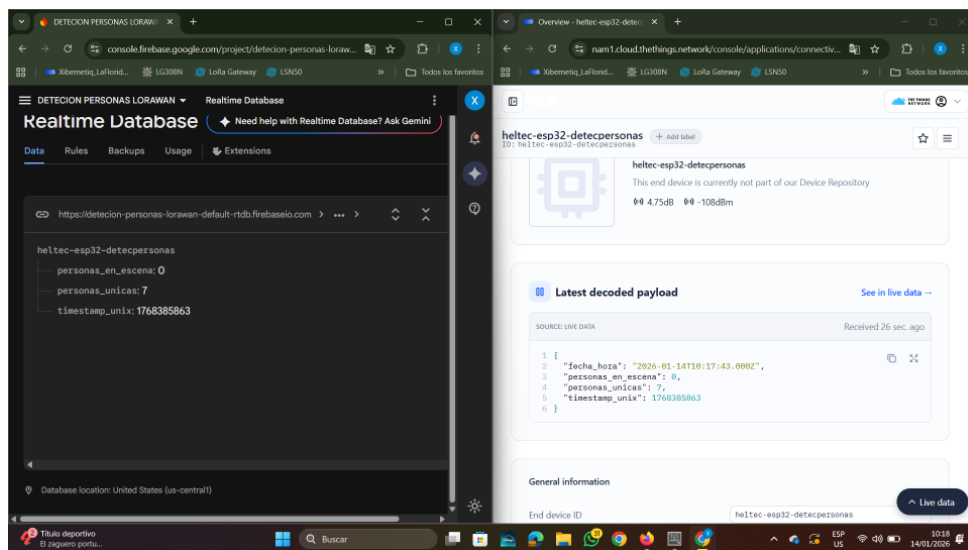


Figura A.5: Comunicación TTN-Firebase

A.2. ANEXO B. PRUEBAS DEL SISTEMA

En este anexo se muestra el lugar las pruebas realizadas del correcto funcionamiento del sistema de monitoreo.



Figura A.6: Colocación sistema de monitoreo de personas



Figura A.7: Lugar donde se desarrollan las pruebas



Figura A.8: Detección con altas aglomeraciones



Figura A.9: Detección con moderada cantidad de personas



Figura A.10: Detección con poca cantidad de personas



Figura A.11: Detección con iluminación y movimientos parciales de personas

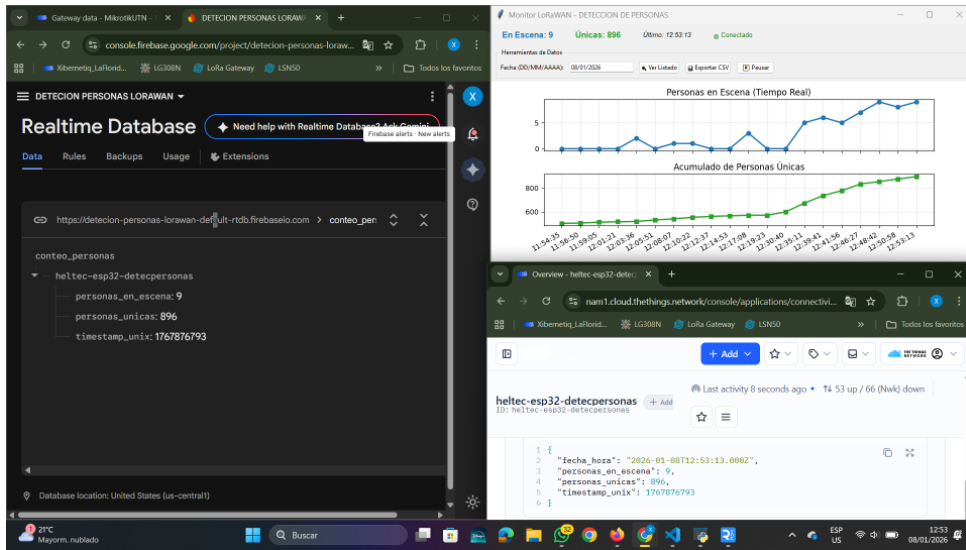


Figura A.12: Comunicación completa del sistema en tiempo real

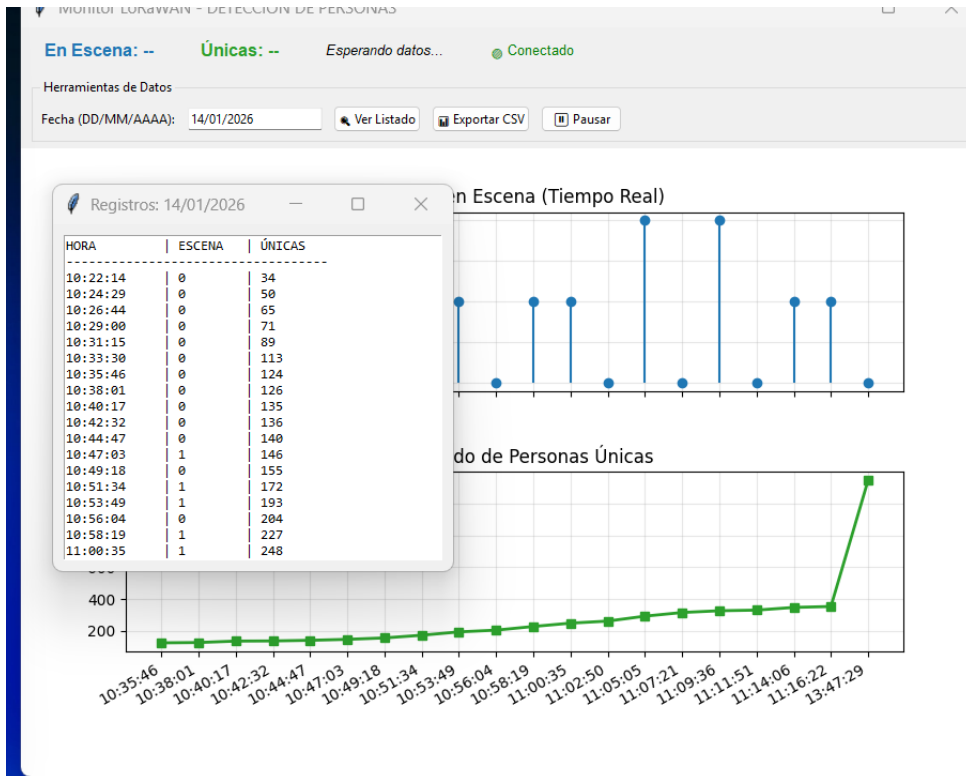


Figura A.13: Consulta histórica de big data del sistema

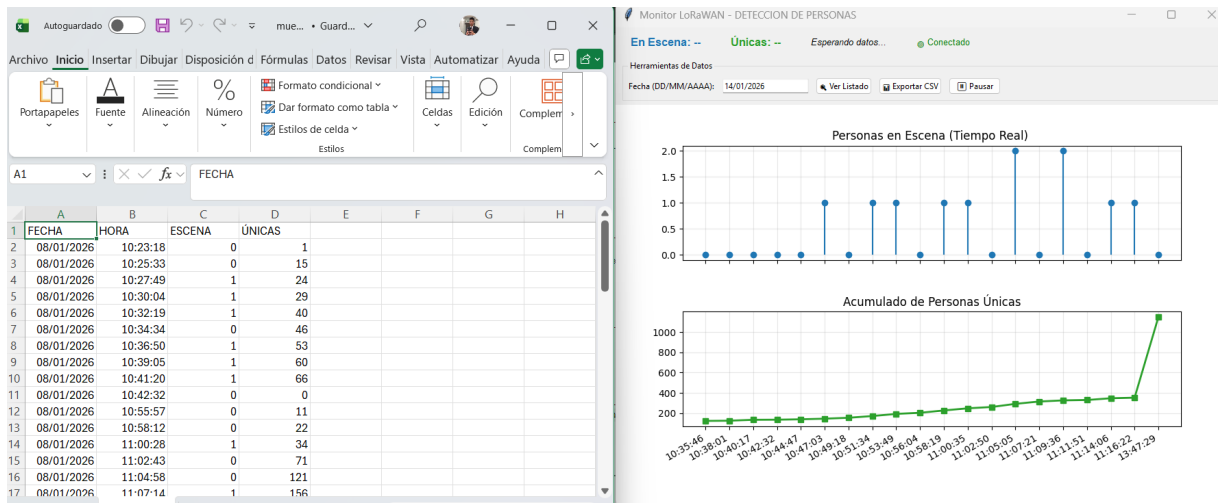


Figura A.14: Exportación de archivos .xls para análisis