ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/391800800

Arquitectura segura para la trazabilidad basada en IoT y blockchain

Conference Paper · June 2025

DOI: 10.5281/zenodo.15431367

CITATIONS
0

READS
30

3 authors:

Javier Alonso-Núñez
University of Valladolid
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Daniel López-Martínez
University of Valladolid
3 PUBLICATIONS **11** CITATIONS
SEE PROFILE

Diego R. Llanos
University of Valladolid
172 PUBLICATIONS **964** CITATIONS
SEE PROFILE

All content following this page was uploaded by Diego R. Llanos on 30 May 2025.

The user has requested enhancement of the downloaded file.

Arquitectura segura para la trazabilidad basada en IoT y blockchain

Javier Alonso-Núñez[1], Daniel López-Martínez[1], y Diego R. Llanos [1]

Resumen— **La integración del Internet de las Cosas (IoT) con la tecnología blockchain ofrece una solución innovadora para mejorar la trazabilidad, seguridad y eficiencia en la gestión de residuos. En este trabajo se propone una arquitectura híbrida que emplea el protocolo MQTT para la comunicación eficiente entre dispositivos IoT y blockchain para garantizar la inmutabilidad y trazabilidad de los datos almacenados en un Data Lake. Se presenta un caso de uso relacionado con el seguimiento y trazabilidad del destino final de residuos, donde la implementación de este sistema permite registrar y verificar la información en tiempo real mediante contratos inteligentes. Los resultados obtenidos validan la viabilidad de la propuesta, destacando su capacidad para facilitar auditorías transparentes, optimizar la supervisión operativa y asegurar la integridad de los datos. Estas conclusiones abren la puerta a futuras implementaciones en entornos urbanos e industriales, consolidando un modelo replicable.**

Palabras clave— **IoT, blockchain, MQTT, Trazabilidad, Seguridad de Datos, Contratos Inteligentes.**

I. Introducción

La creciente generación de residuos y su impacto ambiental han convertido la gestión eficiente de desechos en un desafío clave para ciudades e industrias. Asegurar la trazabilidad y el control de los residuos es fundamental para optimizar la recolección, reducir costos y garantizar el cumplimiento normativo. Sin embargo, los sistemas tradicionales de gestión de residuos presentan limitaciones en cuanto a la verificación de la información, la seguridad de los datos y la transparencia en el proceso.

La integración del Internet de las Cosas (IoT) con la tecnología blockchain ofrece una solución innovadora para abordar estos problemas. IoT permite la monitorización en tiempo real de los contenedores de residuos mediante sensores que recopilan información sobre ubicación, nivel de llenado o temperatura, mientras que blockchain proporciona un mecanismo seguro y descentralizado para registrar y verificar los datos, garantizando su inmutabilidad y facilitando auditorías confiables [1]. Estudios recientes han demostrado que el uso de estas tecnologías en cadenas de suministro y logística inversa ha mejorado significativamente la trazabilidad y la eficiencia operativa en la gestión de residuos y materiales reciclables [2].

Además, en el contexto de la economía circular, la combinación de blockchain e IoT está revolucionando los modelos de trazabilidad de residuos al permitir la verificación de la autenticidad de los datos en tiempo real, lo que facilita la toma de decisiones estratégicas para la reducción de desperdicios y el cumplimiento de normativas ambientales [3]. Implementaciones recientes han demostrado que estas tecnologías pueden mejorar la trazabilidad en la gestión de residuos y la eficiencia en el reciclaje y disposición final [4].

Este trabajo propone una arquitectura híbrida que combina MQTT para la comunicación eficiente entre dispositivos IoT y blockchain para asegurar la trazabilidad de los datos. La solución incluye el almacenamiento de información en un Data Lake, permitiendo su posterior análisis y verificación mediante contratos inteligentes. Como caso de uso, se plantea la trazabilidad del contenido de contenedores de residuos, a través del depósito en el contenedor de un dispositivo IoT que permita supervisar a dónde se dirigen dichos residuos. El objetivo de esta propuesta es mejorar de este modo la supervisión operativa, la transparencia y la seguridad de los datos, tanto en este como en otros contextos. Los resultados obtenidos demuestran la viabilidad del enfoque y su potencial para optimizar diferentes casos de uso en donde la trazabilidad sea importante [5].

II. Estado del arte

A. MQTT en sistemas IoT

El protocolo Message Queuing Telemetry Transport (MQTT) se ha convertido en un estándar ampliamente utilizado en aplicaciones de IoT debido a su bajo consumo de ancho de banda, eficiencia en la comunicación y compatibilidad con dispositivos de baja potencia [6]. Su arquitectura basada en el modelo publicador-suscriptor lo hace ideal para escenarios donde múltiples sensores envían información periódicamente a un sistema centralizado para su procesamiento.

En el contexto de la gestión de residuos, MQTT facilita la recolección de datos en tiempo real desde sensores instalados en contenedores inteligentes. Estos sensores pueden medir diversas variables clave. Por ejemplo, el nivel de llenado permite optimizar las rutas de recolección, reduciendo costos y mejorando la eficiencia del servicio [7]. La localización GPS facilita el seguimiento de los contenedores y ayuda a prevenir extravíos o robos. Además, algunos sensores pueden medir la temperatura y los gases emitidos, lo que resulta esencial en la identificación de riesgos asociados a residuos peligrosos o inflamables.

MQTT permite que estos datos sean transmitidos a un sistema de almacenamiento y análisis sin generar una sobrecarga en la red, garantizando actualizaciones periódicas o prácticamente en tiempo real con un consumo mínimo de energía [8].

Si bien estas ventajas no son exclusivas del ámbito

[1]Dpto. de Informática, Universidad de Valladolid, e-mail: {jalonso,daniel.lopezm,diego.llanos}@uva.es

de la gestión de residuos, es importante destacar algunos de los beneficios clave de MQTT en este contexto. Su eficiencia en la transmisión radica en su diseño ligero, que minimiza el consumo de recursos y permite operar en redes de baja potencia, algo fundamental para sensores IoT instalados en contenedores con conectividad intermitente [9]. Su escalabilidad permite gestionar múltiples sensores distribuidos en distintas ubicaciones sin afectar el rendimiento, gracias a su arquitectura descentralizada. Además, la fiabilidad del protocolo se ve reforzada por sus diferentes niveles de calidad de servicio (QoS), que garantizan la entrega de datos incluso en entornos con conectividad variable [10].

A pesar de sus ventajas, la implementación de MQTT en la gestión de residuos presenta ciertos desafíos. Uno de los principales es la seguridad de los datos, ya que MQTT no ofrece mecanismos nativos de autenticación y cifrado robustos, lo que hace necesaria la integración con tecnologías adicionales, como blockchain o protocolos de cifrado TLS [11]. Otro desafío es la escalabilidad en grandes redes, ya que en ciudades con miles de contenedores inteligentes, la gestión eficiente del tráfico MQTT puede requerir el uso de brokers distribuidos o técnicas de balanceo de carga [10]. Finalmente, la dependencia del bróker central puede convertirse en un problema, dado que una sobrecarga en este punto puede afectar la disponibilidad del sistema si no se optimiza adecuadamente [12].

En este trabajo, se propone el uso de MQTT como protocolo de comunicación para recolectar datos de los contenedores de residuos y almacenarlos en un Data Lake, donde posteriormente serán validados y registrados en blockchain para garantizar su trazabilidad y veracidad.

Recientemente, se han desarrollado extensiones y adaptaciones de MQTT con el objetivo de mejorar sus limitaciones y optimizar su rendimiento en redes IoT masivas. Entre estas mejoras se incluyen el uso de brókeres distribuidos y enfoques específicos para incrementar la tolerancia a fallos [13].

En conclusión, MQTT es una tecnología esencial en el ámbito del IoT gracias a su simplicidad y eficiencia. No obstante, su adopción a gran escala requiere abordar retos significativos en términos de seguridad y escalabilidad, aspectos centrales que este trabajo analiza y busca resolver en su propuesta.

B. Blockchain en la seguridad en integridad de datos

La tecnología blockchain ha emergido como una solución innovadora para garantizar la seguridad, integridad y trazabilidad de los datos en sistemas distribuidos. Originalmente desarrollada para respaldar criptomonedas como Bitcoin, blockchain ha expandido su aplicación a múltiples áreas gracias a su capacidad para operar como un libro de registro descentralizado, inmutable y transparente.

B.1 Comparación entre TSA y blockchain en la verificación de integridad

En sistemas tradicionales, la integridad de los datos suele garantizarse a través de una Autoridad de Sellado de Tiempo (TSA, Timestamping Authority), que actúa como un tercero de confianza encargado de generar y verificar marcas de tiempo digitales sobre documentos o registros de datos. Sin embargo, en aplicaciones donde la descentralización y la resistencia a manipulaciones son clave, blockchain se presenta como una alternativa más robusta.

- **TSA** (Autoridad de Sellado de Tiempo): Un servicio centralizado que genera sellos de tiempo sobre documentos o transacciones digitales, asegurando que la información existía en un momento determinado y no ha sido alterada. Su validez depende de la confiabilidad de la entidad que lo emite.
- **Blockchain**: Un sistema descentralizado en el que la integridad de los datos es garantizada mediante el consenso de múltiples nodos distribuidos. Al registrar información en blockchain, se obtiene un sello de tiempo inmutable que no depende de un único ente de confianza.

En la Tabla I se pueden observar las diferentes características entre TSA y blockchain.

B.2 Características clave de blockchain

Blockchain presenta diversas características clave que la convierten en una tecnología innovadora y confiable en múltiples aplicaciones. Una de sus propiedades más importantes es la descentralización, que elimina la necesidad de intermediarios al distribuir el control y la verificación de las transacciones entre múltiples nodos, aumentando así la seguridad y reduciendo la dependencia de una entidad central.

Otra característica fundamental es la inmutabilidad. Una vez que los datos han sido registrados en la cadena de bloques, no pueden ser alterados sin el consenso de la red. Esto garantiza la integridad de la información y evita modificaciones no autorizadas, ofreciendo un alto grado de confianza en los registros almacenados.

La trazabilidad es otro aspecto clave de blockchain. Cada transacción o dato almacenado se vincula de forma criptográfica con el bloque anterior, formando una cadena continua y verificable. Esto permite rastrear el origen y la evolución de cualquier información dentro de la red, lo que resulta especialmente útil en sectores como la logística, la cadena de suministro y la gestión de activos digitales.

Finalmente, blockchain emplea criptografía avanzada para proteger la información y autenticar a los participantes de la red. Gracias al uso de algoritmos criptográficos, se garantiza la seguridad y privacidad de los datos almacenados, asegurando que solo las personas autorizadas puedan acceder a la información confidencial [14] cuando sea necesario.

Característica	TSA (Centralizado)	Blockchain (Descentralizado)
Modelo de confianza	Basado en una entidad de confianza	Basado en consenso distribuido
Inmutabilidad	Depende del proveedor	Garantizada por la estructura criptográfica
Transparencia	Limitada, acceso controlado	Accesible y auditable públicamente
Resistencia a fallos	Vulnerable a ataques o fallos en la autoridad central	Alta disponibilidad en redes distribuidas
Coste	Bajo en implementaciones pequeñas	Mayor en blockchains públicas, optimizable en privadas
Automatización	Requiere integración con sistemas externos	Contratos inteligentes permiten automatización

Tabla I: Comparación entre TSA y blockchain para la integridad de datos

B.3 Aplicaciones de seguridad e integridad de datos

Blockchain ha demostrado ser una tecnología versátil con diversas aplicaciones en distintos sectores. Uno de sus usos más relevantes es en las auditorías, ya que permite mantener un historial confiable de operaciones que puede ser auditado en cualquier momento sin depender de una única fuente de verdad. Esto garantiza la transparencia y facilita la verificación de la información almacenada.

Otra aplicación clave es la protección contra manipulación. Gracias a su estructura inmutable, blockchain impide alteraciones no autorizadas en los datos registrados, lo que resulta especialmente valioso en sectores como la salud, las finanzas y el Internet de las Cosas (IoT). Esta característica refuerza la seguridad y la integridad de la información en entornos donde la precisión de los datos es crítica.

En el ámbito de las redes IoT, blockchain también puede contribuir significativamente a la autenticidad de los dispositivos. Su implementación permite garantizar que los datos provienen de dispositivos legítimos, reduciendo así los riesgos asociados a la suplantación de identidad o la falsificación de información en la red.

B.4 Avances y desafíos actuales

En los últimos años, la tecnología blockchain ha evolucionado para adaptarse a los requerimientos de los sistemas distribuidos, dando lugar a diversas soluciones innovadoras. Entre ellas, se encuentran las blockchain ligeras, diseñadas para dispositivos IoT con limitaciones de procesamiento y almacenamiento. Ejemplos de estas soluciones incluyen IOTA y las cadenas basadas en *Directed Acyclic Graphs* (DAG) [15], que optimizan el uso de recursos sin comprometer la seguridad y fiabilidad de la red.

Otro avance importante se encuentra en los consensos eficientes, donde algoritmos como *Proof of Stake* (PoS) o *Practical Byzantine Fault Tolerance* (pBFT) [16] han permitido reducir el consumo energético y mejorar el rendimiento de la red. Asimismo, han surgido las blockchain híbridas, que combinan características de las redes públicas y privadas para integrar transparencia con mayor control sobre el acceso y la gestión de los datos.

A pesar de estos avances, blockchain aún enfrenta varios desafíos. Uno de los principales es la escalabilidad, ya que el tiempo de procesamiento y el almacenamiento requerido aumentan a medida que crece la red y el volumen de datos. Este problema se acentúa en aplicaciones que requieren tiempos de respuesta rápidos, ya que la latencia de la blockchain puede dificultar su uso en escenarios de tiempo real, donde la confirmación de transacciones puede tomar demasiado tiempo.

Otro obstáculo importante es el coste computacional. La ejecución de algoritmos criptográficos avanzados puede resultar prohibitiva para dispositivos IoT de bajo rendimiento o de bajo consumo energético. Además, aunque la transparencia de blockchain es una de sus fortalezas, en algunas aplicaciones es necesario garantizar mayor privacidad, por lo que han surgido enfoques que buscan mejorar el anonimato o la protección de datos sensibles.

El coste de almacenamiento también representa un reto significativo, especialmente en entornos donde los dispositivos IoT generan grandes volúmenes de datos. Para mitigar este problema, una solución eficaz es el uso de árboles de Merkle, los cuales permiten agrupar múltiples firmas de datos en un único hash representativo de un bloque de información. En lugar de almacenar individualmente cada registro en la blockchain, se registra solo la raíz del árbol de Merkle, lo que reduce significativamente el consumo de espacio sin comprometer la verificabilidad e integridad de los datos.

III. Desarrollo del dispositivo IoT

El desarrollo de un dispositivo capaz de obtener y enviar información del entorno es clave para la correcta trazabilidad de residuos, por eso que gran parte de los esfuerzos de este artículo se han centrado en obtener un dispositivo funcional; los detalles del mismo se describen a continuación.

A. Especificaciones del dispositivo

Como primer prototipo, se ha desarrollado el dispositivo que se puede observar en la Figura 1. Este dispositivo incluye los siguientes componentes:

- La placa de desarrollo NodeMCU 1.0 basada en el chip ESP8266
- Módulo GPS GY NEO 6MV2 que recoge diversos valores relacionados con la posición geográfica.

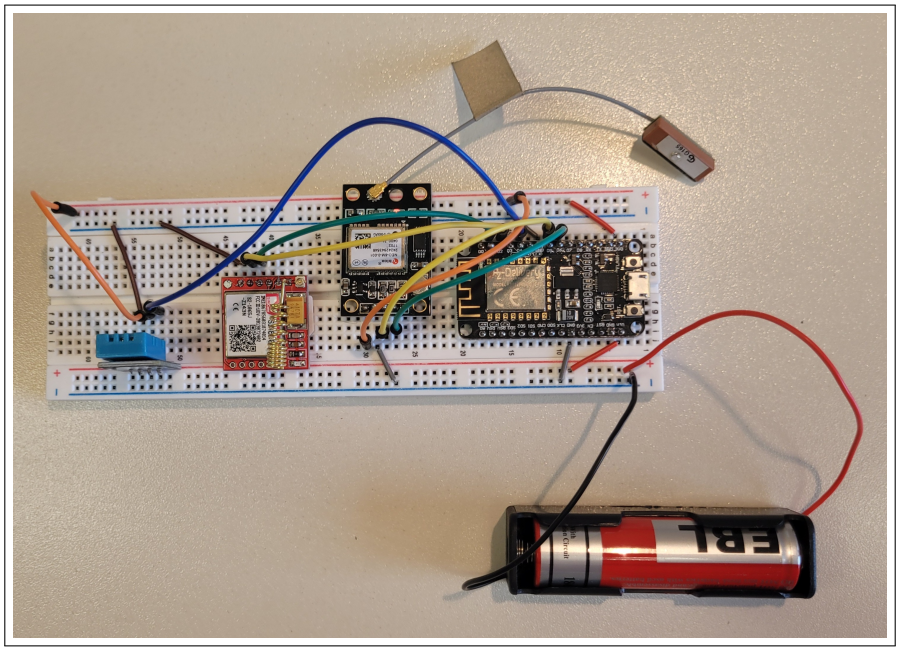

Fig. 1: Versión inicial del sistema IoT

- Módulo GPRS/GSM de banda cuádruple SIM800L para la conexión del dispositivo al servidor externo.
- Un sensor que mide la temperatura y humedad del ambiente, DHT11.

A mayores, se requieren otros recursos para la conexión y alimentación de los distintos componentes expuestos. En la Tabla II se pueden consultar estos componentes junto con su precio aproximado.

B. Detalles de la implementación

El ESP8266, a través de su módulo WiFi, permite la conexión del dispositivo a una red local, lo que facilita la transmisión de datos siempre que esta conexión esté disponible. Sin embargo, dado que el caso de uso principal limita la conexión WiFi, el módulo SIM800L establecerá la comunicación mediante tecnología GPRS en su lugar. A través de este módulo, el dispositivo puede transmitir los datos recopilados a un servidor remoto mediante diversos protocolos de comunicación. No obstante, en el presente trabajo se ha optado por centrar la implementación en el uso del protocolo MQTT, por su eficiencia y adecuación al entorno propuesto.

Por otro lado, la comunicación con el módulo GPS se realiza mediante la interfaz UART, lo que permite obtener coordenadas geográficas en tiempo real con una alta precisión. Además, los datos de temperatura y humedad ambiental proporcionados por el sensor DHT11 se recogen a través de un bus de datos digital y se almacenan temporalmente en la memoria del ESP8266 antes de ser procesados y transmitidos a los servidores correspondientes. Alimentación y gestión de energía: El dispositivo obtiene su energía a partir de una batería de litio de 3.7V, alimentando por un lado el dispositivo NodeMCU por su entrada Vin y por otro lado el dispositivo SIM800L. La entrada Vin del NodeMCU es capaz de traducir corrientes de diversos voltajes a corriente de 3.3V, alimentando así al módulo GPS GY NEO 6MV2.

Para lograr una mínima optimización del consumo energético, se han intercalado pausas de ejecución mediante la función `delay()` entre las recogidas de datos. Esto permite mantener el dispositivo en espera, reduciendo el gasto energético, aunque con un consumo residual. Como trabajo futuro se pretende implementar optimizaciones energéticas mayores mediante funciones que pongan a dormir a los distintos módulos del dispositivo, aunque esto dependerá del microprocesador utilizado en el sistema final. Firmware y lógica de operación: El código del microcontrolador ha sido desarrollado utilizando el framework de desarrollo Arduino, lo que facilita su programación, su ejecución en un ambiente controlado y compatibilidad con los distintos módulos del dispositivo.

Para garantizar una comunicación estable, se han implementado mecanismos de reconexión automática que permiten restablecer la conexión en caso de fallos en la red GPRS, asegurando así la continuidad en la transmisión de datos. Específicamente se comprueba el estado de la conexión al servidor MQTT antes de realizar cualquier envío de datos, ya que en algunos casos el tiempo entre envíos puede superar al tiempo de permanencia de la conexión MQTT.

Además, la estructura de datos se maneja en formato JSON, lo que facilita su integración con el servidor y permite una mayor compatibilidad con bases de datos NoSQL, optimizando el almacenamiento y procesamiento de la información recopilada. No obstante, ha sido necesario adaptar la librería de conexión MQTT a los tamaños de mensaje previstos, ya que estos suelen superar el límite de 215 bytes.

IV. Propuesta de arquitectura del servicio

La arquitectura propuesta en este artículo integra IoT, blockchain y sistemas de almacenamiento esca-

Nombre del componente	Descripción	Precio (€)	Precio acum. (€)
NodeMCU 1.0	Kit de desarrollo con transceptor WiFi	7,20	7,20
GY-NEO6MV2	Módulo GPS	8,44	15,64
SIM800L	Módulo GPRS/GSM	8,25	23,89
DHT11	Sensor de temperatura y humedad	1,65	25,54

Tabla II: BOM (*Bill of Materials* del prototipo de dispositivo IoT desarrollado

lables para garantizar la trazabilidad y seguridad de los datos transmitidos desde dispositivos IoT a través de MQTT. Se compone de varios módulos que interactúan para recolectar, procesar, almacenar y validar los datos en un entorno distribuido y seguro.

A. Componentes principales

Como se puede observar en la Figura 2, la solución está compuesta por los siguientes módulos, cada uno implementado con tecnologías específicas alineadas con los objetivos de escalabilidad, seguridad y eficiencia. Respecto del dispositivo IoT en sí, se trata de sensores conectados que envían datos a través del protocolo de comunicación MQTT, seleccionado por su eficiencia en entornos IoT con ancho de banda limitado. Para la gestión del bróker MQTT, se utilizará Eclipse Mosquitto, debido a su ligereza, bajo consumo de recursos y facilidad de instalación y configuración. Respecto de la API Gateway, actúa como punto central de acceso, manejando la autenticación, autorización y el enrutamiento del tráfico entre los distintos servicios. Para este módulo, se ha elegido Kong, dada su versatilidad y facilidad de configuración. Respecto del Collector (MQTT), está compuesto por un bróker y un proceso worker que puede escalar según la necesidad. Su función principal es recibir, validar y transformar los datos provenientes de los dispositivos IoT antes de su procesamiento. Este módulo estará desarrollado en Python, asegurando flexibilidad y escalabilidad en su integración con el sistema. Respecto del Back Service (API), expone puntos de acceso finales para la consulta y gestión de los datos procesados. Para optimizar el ancho de banda y minimizar las peticiones, se implementará una API GraphQL, permitiendo actualizaciones en tiempo real. Este servicio estará conectado a una base de datos para el tratamiento de los datos de las personas usuarias del sistema, dispositivos y sus relaciones, utilizando PostgreSQL como base de datos relacional. Respecto del Front Service, es una interfaz web para la visualización y consulta de los datos facilitados por el Back Service. Este módulo estará desarrollado en Next.js, permitiendo una interfaz moderna y eficiente que consume los datos procesados en el backend y muestra la información del dispositivo solicitada por la persona usuaria. La Figura 3 ilustra el flujo para la visualización de los datos a petición de una persona que utilice el sistema. El Persistence Controller es un módulo encargado del almacenamiento de los datos en un Data Lake y su posterior procesamiento y recuperación. Para este propósito, se utilizará Delta Lake, que ofrece compatibilidad con múltiples conexiones de almacenamiento, tanto local como en la nube. Como infraestructura de almacenamiento, se ha seleccionado AWS S3 por su facilidad de acceso, coste optimizado en las primeras etapas del sistema y la posibilidad de emular su comportamiento en local mediante MinIO. Además, este módulo se encargará de actualizar los datos una vez incluidos en la blockchain. El Blockchain Controller es un módulo encargado de interactuar con los contratos inteligentes en una red blockchain, garantizando la integridad de los datos almacenados y facilitando la recuperación del identificador de la transacción para su almacenamiento junto con los datos en el Data Lake. Para la implementación de la blockchain, se emplearán contratos inteligentes en Ethereum o Hyperledger Fabric, desarrollados en Solidity para garantizar la trazabilidad de los datos almacenados.

B. Persistencia de datos

En cuanto a la persistencia de los datos recibidos de los dispositivos IoT, estos se van a almacenar en dos niveles:

- **Data Lake** (AWS S3 con Delta Lake): Se almacenarán los datos en bruto de los dispositivos IoT para su procesamiento posterior, junto a la identificación de la transacción de la blockchain.
- **Blockchain** (Web3, Smart Contracts): Se almacenará la firma hash de los datos para así poder garantizar la integridad y evitar alteraciones malintencionadas. De esta forma, se pueden enlazar los datos con la identificación de la transacción del blockchain y la firma de los mismos.

El diagrama de la Figura 4 muestra el flujo que sigue la información del IoT desde su generación en el dispositivo, pasando por su recolección y almacenamiento en un Data Lake y en la blockchain.

C. Estructura del mensaje de datos

Para la transmisión de información en el sistema, se ha definido un formato de mensaje basado en JSON (*JavaScript Object Notation*), dada su flexibilidad, ligereza y compatibilidad con múltiples lenguajes de programación y plataformas. Este formato permite estructurar los datos de manera eficiente, facilitando su procesamiento y almacenamiento en los diferentes módulos del sistema.

El mensaje JSON diseñado contiene la información clave generada por los dispositivos IoT, asegurando la integridad y trazabilidad de los datos recopilados. En la Figura 5 se tiene un ejemplo de cómo sería el mensaje JSON generado y enviado por los dispositivos IoT.

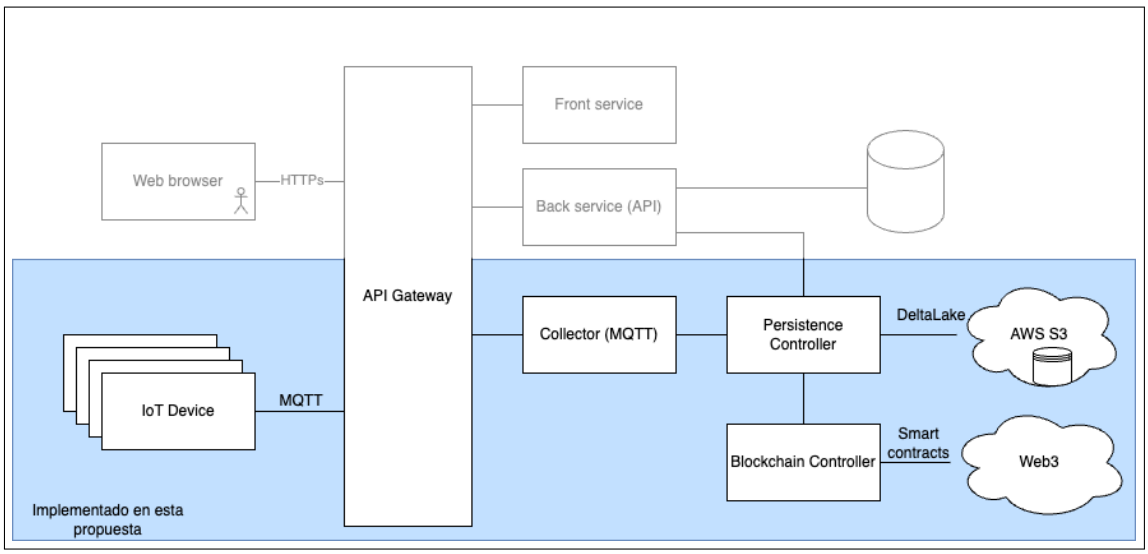

Fig. 2: Propuesta de arquitectura

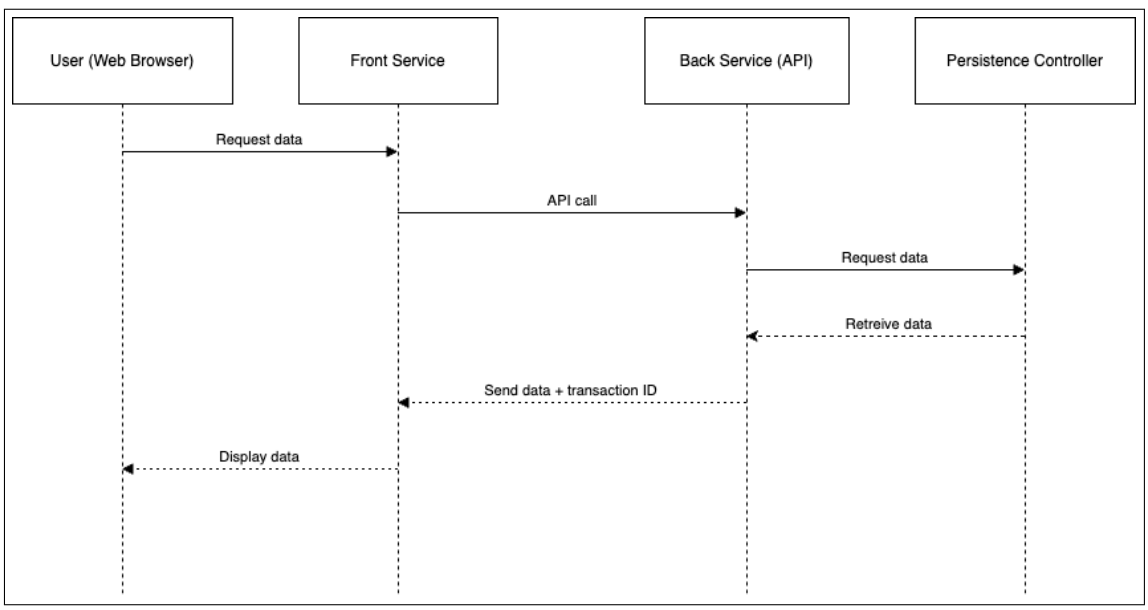

Fig. 3: Diagrama de secuencia de la recolección de datos por parte de la persona usuaria.

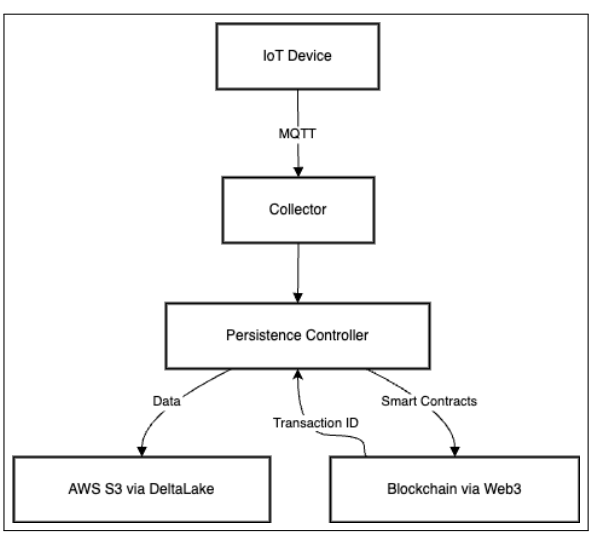

Fig. 4: Diagrama de recolección y persistencia del sistema

```
{
    "sha256": "e3b0c4...2b855",
    "mac": "1E:E5:89:44:FE:19",
    "imei": "123456789012345",
    "support": [
        "GPS",
        "TEMP"
    ],
    "data": [{
        "timestamp": "1736875157",
        "lat": "41.662739",
        "long": "-4.705044",
        "temp": "18"
    },
    {
        "timestamp": "1736876157",
        "lat": "41.662703",
        "long": "-4.705082",
        "temp": "18"
    }]
}
```

Fig. 5: Ejemplo de mensaje JSON generado por los dispositivos IoT.

Cada mensaje JSON se compone de los siguientes elementos:

- sha256: Un identificador hash SHA-256 genera-

do para garantizar la integridad del mensaje.
- mac: Dirección MAC del dispositivo IoT, permitiendo su identificación única dentro de la red.
- imei: Número IMEI del dispositivo en caso de utilizar conectividad móvil.
- support: Lista de funcionalidades soportadas por el dispositivo, como GPS o sensores de temperatura.
- data: Arreglo de registros con la información recopilada en distintos momentos, incluyendo:
  - timestamp: Marca de tiempo en formato UNIX, representando el momento exacto de la captura de datos.
  - lat, long: Coordenadas geográficas del dispositivo IoT en el momento del registro.
  - temp: Valor de temperatura medido en ese instante.

El uso de este esquema JSON facilita la integración de los datos en el Data Lake, su posterior procesamiento y su registro en la blockchain, asegurando la trazabilidad y veracidad de la información transmitida por los dispositivos IoT.

V. Implementación

A. Caso de uso: trazabilidad del contenido de contenedores con residuos

La gestión eficiente de residuos es un desafío clave en la actualidad, especialmente en sectores industriales y municipales, donde es fundamental garantizar la correcta disposición y trazabilidad de los desechos. Un sistema basado en IoT y blockchain puede proporcionar una solución robusta para asegurar la autenticidad y verificabilidad de los datos, mejorando la trazabilidad, pérdidas de información o manipulaciones indebidas.

B. Configuración del entorno experimental

El sistema que se propone implementaría una red de dispositivos IoT instalados en los contenedores de residuos, los cuales envían información periódica sobre variables clave, como la localización o la temperatura.

Estos dispositivos transmiten los datos mediante MQTT a la infraestructura central, donde son procesados y almacenados en un Data Lake para su análisis. Simultáneamente, las firmas digitales de estos datos son registradas en blockchain para garantizar su autenticidad.

Para asegurar que los datos almacenados en el Data Lake no han sido alterados y son verificables en blockchain, se sigue el siguiente proceso:

1. Recepción de Datos desde dispositivos IoT: Los sensores envían información estructurada a través del protocolo MQTT, garantizando la comunicación eficiente y en tiempo real. Esta estructura de mensaje ya viene con un hash criptográfico como resumen de los datos que envía el dispositivo.
2. Generación de huella digital (hash) de los Datos: Antes de almacenar la información en el Data Lake (AWS S3 con Delta Lake), se genera un hash criptográfico con el conjunto de datos, hora de recepción, identificador del mensaje, etc. Este hash actúa como una "firma digital" única para la información recibida.
   Para optimizar el almacenamiento y mejorar la eficiencia en la verificación de la integridad de los datos, se emplea un árbol de Merkle. Este mecanismo agrupa los datos en una estructura jerárquica de hashes, donde:
   - Cada dato individual se representa como una hoja del árbol, generando su correspondiente *hash*.
   - Los *hashes* de cada par de nodos se combinan para formar *hashes* intermedios, continuando hasta generar un único valor hash raíz, conocido como raíz del árbol de Merkle (Merkle Root).
   - El hash raíz del árbol de Merkle se almacena en la blockchain, permitiendo verificar posteriormente la autenticidad de cualquier dato sin necesidad de almacenar todos los registros en la red.

   Esta estructura reduce significativamente el espacio requerido en la red blockchain, ya que en lugar de registrar cada dato individualmente, solo se almacena el hash raíz del árbol de Merkle. Además, permite verificar la validez de un dato consultando únicamente los *hashes* intermedios necesarios en la prueba de inclusión (*Merkle Proof*), sin necesidad de acceder a toda la base de datos.
3. Registro del Hash en blockchain: La "firma digita" generada se almacena mediante un contrato inteligente desplegado en una blockchain pública o privada. En este caso concreto sobre la red Ethereum. Esto permitirá que, en cualquier momento, se pueda verificar si los datos en el Data Lake han sido modificados comparando su hash actual con el registrado en la red blockchain.
4. Validación y autoría de datos: Cuando se requiere auditar la trazabilidad de los residuos, se recalcula el hash de los datos almacenados en el Data Lake y se compara con el valor registrado en la red blockchain, pudiendo referirse a ellos a través del identificador de la transacción.
   Si estos valores coinciden, se garantiza que la información no ha sido alterada. En caso de discrepancia, se puede detectar manipulación o corrupción de los datos. Gracias al uso de árboles de Merkle, este proceso se realiza de manera eficiente, sin necesidad de recorrer toda la base de datos, sino verificando únicamente las pruebas de inclusión asociadas a la consulta.

C. Beneficios esperados del sistema

A partir del diseño propuesto y del caso de uso implementado, se identifican una serie de beneficios potenciales asociados a la integración de IoT, almacenamiento en Data Lake y tecnología blockchain para la trazabilidad de residuos. Aunque algunos de estos

beneficios requieren una validación más amplia mediante pruebas a gran escala, la arquitectura desarrollada muestra indicios prometedores en los siguientes aspectos:

- Mejoras en la trazabilidad y alteraciones de registros: El uso de funciones hash y el registro en blockchain incrementan la resistencia a manipulaciones no autorizadas de los datos.
- Mejora en la trazabilidad y transparencia: La arquitectura permite el seguimiento de los datos generados por los dispositivos IoT desde su origen hasta su almacenamiento, facilitando auditorías y verificaciones posteriores.
- Optimización de procesos operativos: La disponibilidad de datos estructurados y fiables abre la puerta a mejoras en la planificación y ejecución de las operaciones logísticas, aunque su impacto operativo deberá medirse en futuros despliegues reales.
- Soporte para el cumplimiento normativo: El sistema ofrece un marco técnico que puede apoyar la generación de evidencias verificables sobre el tratamiento de residuos, especialmente en sectores regulados.

Estos beneficios se alinean con los objetivos planteados en el diseño del sistema y constituyen una base para futuras evaluaciones empíricas que permitan cuantificar su impacto en entornos reales.

VI. Pruebas

Con el objetivo de evaluar distintas estrategias de recogida y envío de datos, consumo y almacenamiento, se han definido tres escenarios experimentales, cada uno con diferentes intervalos de recolección y transmisión. Esta variabilidad permite analizar el comportamiento del sistema en distintas condiciones operativas y facilitar la selección de la configuración más adecuada. Los casos de prueba son:

- **Caso 1**: Los datos se recogerán cada 10 segundos y se enviarán cada minuto. Como resultado, cada paquete de datos contendrá 6 registros de GPS, temperatura y humedad.
- **Caso 2**: Los datos se recogerán cada 60 segundos y se enviarán cada 10 minutos. En este caso, cada paquete incluirá 10 registros de GPS, temperatura y humedad.
- **Caso 3**: Los datos se recogerán cada 600 segundos y se enviarán cada hora. Así, cada paquete contendrá 6 registros de GPS, temperatura y humedad.

A. Pruebas del dispositivo IoT

Con el fin de evaluar el comportamiento del dispositivo IoT en distintas configuraciones de operación, se han definido los tres escenarios de prueba anteriormente presentados, y que varían en los intervalos de recolección y transmisión de datos. Estas pruebas permiten analizar el impacto de dichas configuraciones en el consumo de datos y en la eficiencia general del sistema.

Caso de prueba	Tamaño por envío	Bytes/hora
Caso 1	910	54600
Caso 2	1404	8424
Caso 3	909	909

Tabla III: Tabla de consumo de datos para los casos de prueba

Si bien el tamaño del paquete de datos en el Caso 1 y el Caso 3 es similar, la frecuencia de envío varía, lo que puede tener un impacto significativo en el consumo de datos del sistema. Un análisis de este consumo en cada uno de los casos definidos se presenta en la Tabla III. Dado que el tamaño del paquete varía en función de la longitud de los valores de algunas variables registradas por los sensores, se ha utilizado como referencia el tamaño promedio del paquete, obtenido tras varias ejecuciones de la misma prueba.

B. Pruebas sobre la arquitectura propuesta

Además de evaluar el comportamiento del dispositivo IoT, se ha analizado el rendimiento de la arquitectura completa en el procesamiento, almacenamiento y verificación de los mensajes recibidos. Para ello, se ha medido el tiempo de procesado de los mensajes, así como el tamaño de los datos enviados a la blockchain y en el Data Lake bajo dos configuraciones diferentes: utilizando árboles de Merkle y sin utilizarlos.

El uso de árboles de Merkle permite reducir significativamente el tamaño de la información registrada en la blockchain, ya que únicamente se almacena la raíz del árbol que resume todas las medidas incluidas en un mensaje. No obstante, esta optimización implica que, para garantizar la validación de cada medida individual, es necesario almacenar en el Data Lake las pruebas de inclusión (Merkle Proofs) asociadas.

Cuando se utiliza un árbol de Merkle para validar la integridad de múltiples medidas dentro de un único mensaje, es necesario almacenar adicionalmente las *proofs de inclusión* en el Data Lake. Estas pruebas permiten verificar que una medida pertenece al árbol cuya raíz ha sido registrada en la blockchain.

Para un árbol binario, cada *Merkle proof* consiste en un conjunto de hashes hermanos en el camino desde la hoja hasta la raíz. Por tanto, el número de hashes necesarios por prueba se aproxima a:

$$n_{\text{hashes}} = \lceil \log_2(N) \rceil$$

donde N es el número de medidas (hojas) en el árbol. Como cada hash SHA-256 ocupa 32 bytes, el tamaño de cada prueba es:

$$\text{Tamaño de la proof} = \lceil \log_2(N) \rceil \times 32 \text{ bytes}$$

Aplicando esta fórmula a los dos casos definidos:

- Para $N = 6$:

$$\lceil \log_2(6) \rceil = 3 \Rightarrow 3 \times 32 = 96 \text{ bytes por proof}$$

$$\text{Total por mensaje} = 6 \times 96 = 576 bytes$$

- Para $N = 10$:

$$\lceil \log_2(10) \rceil = 4 \Rightarrow 4 \times 32 = 128 \text{ bytes por proof}$$

$$\text{Total por mensaje} = 10 \times 128 = 1280 bytes$$

En la Tabla IV se puede observar un resumen de los tamaños para las mediciones de los casos de prueba.

Por el contrario, si se opta por registrar en la blockchain el hash individual de cada medida, se elimina la necesidad de almacenar pruebas adicionales en el Data Lake, pero se incrementa el volumen de datos en la red blockchain, afectando a su escalabilidad y coste.

A continuación, se presentan los resultados comparativos en la Tabla V, donde se puede observar el impacto del uso de árboles de Merkle en el tiempo de procesado y el tamaño de los datos enviados a la blockchain. Asimismo, la Figura 6 ilustra gráficamente la diferencia en el volumen de datos registrados en la blockchain en ambos enfoques.

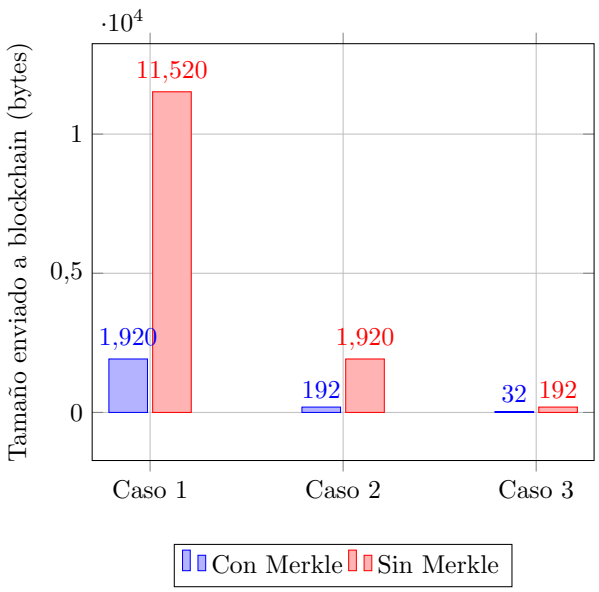

Fig. 6: Comparativa del tamaño de datos enviados a la red blockchain con y sin árboles de Merkle (SHA-256).

Del análisis de los resultados se desprende que el uso de árboles de Merkle con función hash de 256 bits reduce drásticamente el tamaño de la información almacenada en la blockchain, pasando de registrar múltiples hashes individuales por cada medida a almacenar únicamente la raíz del árbol generado. Esta optimización es especialmente relevante en mensajes que agrupan un mayor número de registros, como en el Caso 1, donde el ahorro supera los 9600 bytes por hora.

No obstante, esta reducción en el uso de la blockchain conlleva un ligero incremento en el tiempo de procesado y en el volumen de información almacenada en el Data Lake, debido a la necesidad de guardar las pruebas de inclusión (Merkle Proofs) para cada medida individual. Por tanto, la elección entre ambos enfoques debe contemplar el equilibrio entre la eficiencia en el almacenamiento en la blockchain y la complejidad operativa en el almacenamiento local. Conviene destacar que esta última resulta más asumible gracias a las capacidades actuales de los sistemas y plataformas de almacenamiento orientados a entornos *Big Data*.

En conjunto, los resultados validan que el uso de árboles de Merkle puede ser una estrategia eficaz para mejorar la escalabilidad del sistema sin comprometer la integridad de los datos.

VII. Conclusiones

La arquitectura propuesta demuestra que la combinación de IoT, blockchain y almacenamiento distribuido en Data Lakes es una estrategia efectiva para garantizar la trazabilidad y seguridad de los datos en sistemas distribuidos, además de permitir su tratamiento y análisis posterior. A través del uso de MQTT para la ingesta de datos y de blockchain para su verificación, se consigue un sistema escalable y resiliente frente a manipulaciones o alteraciones.

El caso de uso de trazabilidad del contenido de contenedores con residuos nos permite estudiar la efectividad del enfoque, buscando asegurar que la información registrada sea auténtica y verificable en todo momento. Al registrar las firmas digitales en contratos inteligentes, se logra una auditoría transparente y un control eficiente de los procesos de recolección y tratamiento de residuos.

Un aspecto especialmente relevante de la propuesta es el uso de árboles de Merkle para la verificación de la integridad de los datos. Esta técnica permite agrupar múltiples métricas asociadas a un mismo mensaje en una estructura jerárquica, almacenando únicamente su raíz en la blockchain. De este modo, se consigue una validación en bloque eficiente que reduce significativamente el volumen de datos registrados en la red, lo que se traduce en una disminución directa de los costes asociados al uso de blockchain, especialmente en redes públicas o de alto coste por transacción.

Entre los principales beneficios identificados destacan la mejora de la trazabilidad, la optimización de la gestión operativa y el cumplimiento normativo en sectores donde la cadena de custodia es crítica. Sin embargo, aún existen desafíos a considerar, como la escalabilidad de blockchain en redes de alto tráfico y la latencia en la verificación de transacciones.

Como futuras líneas de investigación, exploraremos alternativas de blockchain más ligeras, como IOTA o DAG, así como optimizar la integración con sistemas de inteligencia artificial para el análisis predictivo de los datos almacenados. Estas líneas permitirán escalar y adaptar la arquitectura propuesta a nuevos escenarios, consolidando el potencial del modelo presentado, tal y como se anticipó en el resumen. También se trabajará en reducir los costes y el tamaño del dispositivo desarrollado. En conjunto, este trabajo contribuye a la evolución de soluciones seguras y escalables para entornos IoT, promoviendo la transferencia tecnológica y la innovación en la industria.

Nº de medidas	Hashes por proof	Tamaño por proof (bytes)	Total por mensaje (bytes)
6	3	96	576
10	4	128	1280

Tabla IV: Tamaño adicional requerido en el Data Lake por las Merkle proofs

Caso	Merkle	T. procesado (ms)	Blockchain (bytes/mens.)	Blockchain (bytes/hora)	Data Lake extra (bytes)
1	Sí	372	32	1920	576
1	No	316	192	11520	0
2	Sí	413	32	192	1280
2	No	353	320	1920	0
3	Sí	397	32	32	576
3	No	322	192	192	0

Tabla V: Comparativa del procesamiento y almacenamiento con y sin uso de árboles de Merkle (SHA-256)

VIII. Disponibilidad del código

Con el objetivo de fomentar la reproducibilidad y facilitar la reutilización de esta solución, el código fuente asociado a esta contribución está disponible públicamente en los siguientes repositorios de GitHub:

- **Dispositivo IoT**: `https://github.com/danipequelangos/iot-trace-system`
- **Servicio**: `https://github.com/javalon/iot-trace-chain`

Ambos repositorios contienen instrucciones para su despliegue, dependencias requeridas y ejemplos de uso.

Agradecimientos

Esta investigación ha sido financiada en parte por el proyecto NATASHA (PID2022-142292NB-I00), del Ministerio de Ciencia, Innovación y Universidades de España.

Referencias

[1] Smita Bansod, Akash Lalitkumar Mankwana, Atik Zakirhusen Mujawar, and Lalit Sailesh Jain, "Smart waste management using iot and blockchain," in *Challenges and Solutions in Internet of Things-Based Smart Applications*. Chapman and Hall/CRC, 2025.

[2] Rovin Pathania, Yashwant Kumar, Megha Sehgal, and Amarjit Deshmukh, "Embracing sustainable value chains in reverse logistics," *International Journal of Research in Innovative Multidisciplinary Studies*, vol. 2, Feb. 2023.

[3] Ermolovskaya Olga, "The impact of digital technologies on sustainable consumption and production," *Reliability: Theory & Applications*, vol. 19, no. SI 6 (81), pp. 290–299, 2024.

[4] Deepak Gupta, Deepanshu Arora, Arun Kumar Chaudhary, Warshi Singh, Ajay Suryavanshi, Kanwal Preet Singh Attwal, Deepak Kumar, and Vikas Misra, "Integrating iob in sustainable supply chain management of composites for enhancing efficiency and reducing waste," in *Mapping Human Data and Behavior With the Internet of Behavior (IoB)*. 2025, pp. 301–336, IGI Global Scientific Publishing.

[5] Yang Zhang, Vijai Kumar Gupta, Keikhosro Karimi, Yajing Wang, Mohd Azman Yusoff, Hassan Vatanparast, Junting Pan, Mortaza Aghbashlo, Meisam Tabatabaei, and Ahmad Rajaei, "Synergizing blockchain and internet of things for enhancing efficiency and waste reduction in sustainable food management," *Trends in Food Science & Technology*, vol. 156, pp. 104873, Feb. 2025.

[6] José Varela-Aldás, Christian Junta, Elias Choque, and Guillermo Palacios-Navarro, "Influence of higher education on iot acceptance through hands-on learning," *TEM Journal*, pp. 528–539, Feb. 2025.

[7] Yeong-Hwa Chang, Feng-Chou Wu, and Hung-Wei Lin, "Design and implementation of esp32-based edge computing for object detection," *Sensors*, vol. 25, no. 6, pp. 1656, Jan. 2025.

[8] Yi-Hsuan Tseng, Chao Wang, Yu-Tse Wei, and Yu-Ting Chiang, "Cloud-edge mqtt messaging for latency mitigation and broker memory footprint reduction," *PeerJ Computer Science*, vol. 11, pp. e2741, Mar. 2025.

[9] Claudio Bartoli, Michele Bonanni, Francesco Chiti, and Laura Pierucci, "The alliance of sdn and mqtt for the web of industrial things," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2025.

[10] Azita Rezaei, Ali Broumandnia, and Seyed Javad Mirabedini, "Mls: A novel hybrid security framework utilizing the wiedemann algorithm and chaotic mapping for mqtt," *The Journal of Supercomputing*, vol. 81, no. 4, pp. 597, Mar. 2025.

[11] Seyed Ali Ghazi Asgar and Narasimha Reddy, "Analysis of misconfigured iot mqtt deployments and a lightweight exposure detection system," in *Network and Distributed System Security (NDSS) Symposium, Workshop on Security and Privacy in Standardized IoT (SDIoTSec)*, Mar. 2025.

[12] Lina Nachabe, Riham Ginzarly, Hassan Kanj, and Jean Taleb, "Adaptive hybrid energy system (ahes) for smart home: Lebanese case/application," *Euro-Mediterranean Journal for Environmental Integration*, Feb. 2025.

[13] Sujitha Lakshminarayana and P. Santhi Thilagam, "Next-generation ddos attacks on iot deployments: Targeting the advanced features of mqtt v5.0 protocol," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[14] "Desafíos de seguridad y privacidad en la tecnología blockchain," *El Derecho*, Nov. 2023.

[15] "All about directed acyclic graphs," https://blog.sui.io/all-about-directed-acyclic-graphs/.

[16] Rob Behnke, "What is practical byzantine fault tolerance in blockchain?," https://www.halborn.com/blog/post/what-is-practical-byzantine-fault-tolerance-in-blockchain, Oct. 2023.

View publication stats