Universidad de Huelva

Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática

Nuevos paradigmas para la configuración y recuperación automática de equipos de red mediante dispositivos portátiles, gestión en la nube y blockchain

Memoria para optar al grado de doctor
presentada por:

Juan Diego Morillo Reina

Fecha de lectura: 28 de mayo de 2025

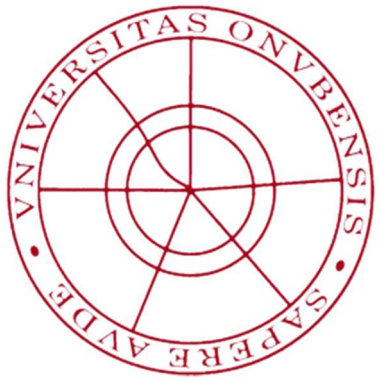
Bajo la dirección del doctor:

Tomás de Jesús Mateo Sanguino

Huelva, 2025

Universidad de Huelva

Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática



NUEVOS PARADIGMAS PARA LA CONFIGURACIÓN Y RECUPERACIÓN AUTOMÁTICA DE EQUIPOS DE RED MEDIANTE DISPOSITIVOS PORTÁTILES, GESTIÓN EN LA NUBE Y BLOCKCHAIN

Memoria para optar al grado de doctor
presentada por:

Juan Diego Morillo Reina

Bajo la dirección del doctor:

Tomás de J. Mateo Sanguino

Fecha de lectura:

Huelva, 2025

UNIVERSITY OF HUELVA

Doctoral Program in Industrial and Environmental Science and Technology

Research line

Electrical, electronic, control and robotics engineering

NEW PARADIGMS FOR AUTOMATIC CONFIGURATION AND RECOVERY OF NETWORK EQUIPMENT USING MOBILE DEVICES, CLOUD MANAGEMENT AND BLOCKCHAIN

Ph. D. Thesis:

Juan Diego Morillo Reina

Supervisor:

Tomás de J. Mateo Sanguino

Date:

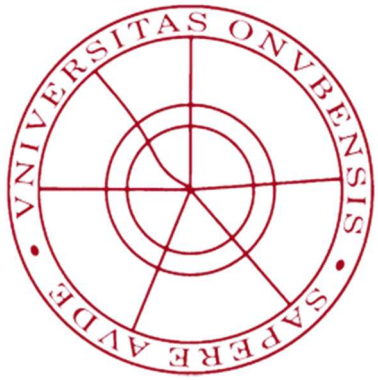
Huelva, 2025

UNIVERSIDAD DE HUELVA

Programa de Doctorado en Ciencia y Tecnología Industrial y Ambiental

Línea de investigación

Ingeniería eléctrica, electrónica, de control y robótica



NUEVOS PARADIGMAS PARA LA CONFIGURACIÓN Y RECUPERACIÓN AUTOMÁTICA DE EQUIPOS DE RED MEDIANTE DISPOSITIVOS PORTÁTILES, GESTIÓN EN LA NUBE Y BLOCKCHAIN

Doctorando:

Juan Diego Morillo Reina

Director:

Tomás de J. Mateo Sanguino

Fecha:

Huelva, 2025

D. Tomás de Jesús Mateo Sanguino, Profesor Titular de Universidad de la Escuela Técnica Superior de Ingeniería de la Universidad de Huelva,

CERTIFICA:

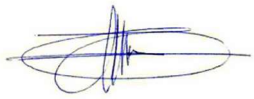
Que D. Juan Diego Morillo Reina, Graduado en Ingeniería Informática y Máster en Ingeniería Informática por la Universidad de Huelva, ha realizado bajo mi dirección y dentro del Programa de Doctorado en Ciencia y Tecnología Industrial y Ambiental (CyTIA) y en la línea de investigación Ingeniería Eléctrica, Electrónica, de Control y Robótica el trabajo correspondiente a su tesis doctoral titulada:

NUEVOS PARADIGMAS PARA LA CONFIGURACIÓN Y RECUPERACIÓN AUTOMÁTICA DE EQUIPOS DE RED MEDIANTE DISPOSITIVOS PORTÁTILES, GESTIÓN EN LA NUBE Y BLOCKCHAIN

Revisado el presente trabajo, estimo que puede ser presentado al Tribunal que ha de juzgarlo.

Y para que así conste a efectos de lo establecido en el Real Decreto 99/2011 y por la normativa Reguladora del título de Doctor en la Universidad de Huelva, autorizo la presentación de este trabajo en la Universidad de Huelva.

Huelva, 6 de Marzo de 2025



Director: Dr. Tomás de J. Mateo Sanguino

Doctorando: Juan Diego Morillo Reina

Agradecimientos

En primer lugar, quiero expresar mi más sincero agradecimiento a mi director de tesis, el Dr. Tomás de Jesús Mateo Sanguino, por su paciencia, sus valiosas enseñanzas y su dedicación constante durante todo este proceso. Sus consejos, motivación y confianza han sido fundamentales para alcanzar esta meta, especialmente considerando el desafío adicional de compaginar este trabajo con mi vida laboral. Gracias por creer en mí y por enseñarme a confiar en mis capacidades.

A mis padres, Pedro y Juana, por su amor incondicional, sacrificio permanente y por insistirme en no elegir el camino más sencillo, animándome siempre a mejorar y nunca rendirme. Ellos han sido y siempre serán mi mayor ejemplo de esfuerzo, perseverancia y compromiso. Gracias por estar siempre a mi lado.

A mis hermanos, Alejandro y Francisco, por siempre contagiarme con su alegría y brindarme su apoyo sincero. Su compañía y ayuda han sido fundamentales para crear momentos valiosos que me permitieron dedicarme plenamente a este trabajo. Gracias por mantenerme feliz y acompañado durante todo este recorrido.

Quiero agradecer de manera especial a Esperanza por ser mi apoyo incondicional y mi mayor inspiración. Gracias por tu paciencia infinita y tu comprensión sin límites a lo largo de este camino. Contar contigo ha sido clave para lograr esta meta, ya que haces que todo sea siempre mucho más sencillo.

Finalmente, quiero dar las gracias a todas las personas que, directa o indirectamente, han colaborado en este trabajo. Su contribución ha sido esencial en este importante capítulo de mi vida y estoy profundamente agradecido por haber contado con su ayuda.

ÍNDICE GENERAL

LISTA DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1. Planteamiento General

1.1 Resumen

La tesis doctoral tiene como objetivo investigar en torno a un sistema de tipo hardware/software utilizado para recuperar de forma automática equipos de red tales como enrutadores y conmutadores, de aquí en adelante DRACSC. El sistema, objeto de una invención, tiene como principal ventaja la simplificación de tareas de recuperación manual de sistemas operativos (SO) embebidos en dichos equipos, facilitando así la gestión del procedimiento de recuperación. La necesidad de llevar a cabo tareas de recuperación se produce cuando un equipo presenta un error de arranque motivado por un fallo en el SO tal como la pérdida de la imagen de arranque, memoria Flash dañada o cambio de contraseña de acceso, entre otros. La mayoría de equipos de red disponen de un registro de configuración que les obliga a detenerse en estado de espera durante el arranque. En este caso, se requiere enviar manualmente una secuencia de interrupción a través de una línea de comunicación para iniciar el protocolo de recuperación del equipo. En un entorno de trabajo con un número de equipos significativo, se torna esencial poder actuar en ellos con agilidad sin preocuparse por recordar a cuál corresponde cada credencial de acceso o sin necesidad de usar varias aplicaciones para administrar su recuperación.

Este aspecto es sumamente importante ya que supone una de las situaciones de fallo más comunes en infraestructuras de red, lo que a su vez se traduce en grandes pérdidas económicas para las empresas. Como dato significativo, los centros de procesamiento de datos (CPD) están clasificados en niveles TIER que van desde 1 a 4 con una disponibilidad garantizada entre 99,671 % y 99,995 %. Esto quiere decir que sus instalaciones y equipos de red están certificados por el Uptime Institute para soportar un máximo de entre 28,8 horas y 0,8 horas de interrupción al año. Por ejemplo, los CPD de BBVA y Telefónica disponen de la última calificación TIER IV Gold con 26 min/año de interrupción máxima garantizada para alrededor de 10.000 procesadores de alta gama, una capacidad de procesamiento equivalente a 35 millones de portátiles de última generación.

Considerando la problemática actual, la tesis doctoral propone una solución que integra hardware y software para recuperar de forma automática equipos en redes de datos y facilitar las labores de administración típicas. Además de la importancia señalada de su utilidad, se aporta una solución al estado del arte actual que no disponen los sistemas con propósitos similares. El ámbito de aplicación de esta solución es doble. Por un lado, se aplica a entornos docentes formados por profesores, alumnos y técnicos de laboratorio. Por otro, se aplica a entornos profesionales en el ámbito de las TIC. Para ello, los objetivos científico-técnicos marcados son: *i*) presentar un prototipo de grandes prestaciones, *ii*) interoperable

con independencia de fabricantes o marcas, y *iii*) proponerlo como software de código abierto para uso generalizado. La metodología llevada a cabo incluye acciones para mejorar el prototipo desarrollado, incremento de servicios y funciones, experimentación del sistema en los entornos de aplicación, explotar resultados y transferir. Esta metodología ha sido desarrollada en un laboratorio de investigación y validada en aulas docentes mediante equipos de red de distintos fabricantes. Como resultado de este marco de pruebas en el que se ha llevado a cabo la tesis doctoral, tanto a nivel teórico como práctico, se ha obtenido un impacto traducido en publicaciones científicas de excelencia y de transferencia tecnológica, siendo protegido dicho sistema mediante patente internacional.

En cuanto a las soluciones desarrolladas durante la tesis doctoral, en primer lugar, se presenta un dispositivo portable que pretende simplificar la administración de configuraciones y SO de equipos de red. El primer prototipo fue implementado mediante una solución hardware/software basada en Raspberry Pi 2, el cual ha sido probado sobre sistemas embebidos incluyendo conmutadores y enrutadores de los fabricantes Cisco y MikroTik, extensibles a otros. Además de la recuperación de sistemas ante desastres, DRACSC permite clonar configuraciones y desplegarlas en una infraestructura de red en tan solo unos pasos. Las pruebas de rendimiento han mostrado que mientras mayor sea la tarea de administración, contabilizada por el número de sentencias y caracteres introducidos por el usuario, mayor es la ventaja de usar las funciones automáticas de DRACSC. En el caso extremo de desplegar equipos de red con configuraciones desde cero, el ahorro de tiempo es significativamente mayor.

La segunda solución presentada en la tesis doctoral consiste en una actualización de DRACSC con nuevas funcionalidades, optimización de software y rendimiento mejorado (i.e., Raspberry Pi 4 Model B). También incluye la creación de una plataforma en la nube que permite la sincronización de funciones de alto nivel entre diferentes dispositivos DRACSC. Esto se realiza además proporcionando libertad de elección del fabricante del equipo de red gestionado. Para su validación se llevaron a cabo diferentes análisis estadísticos sobre distintos escenarios con estudiantes, profesores y profesionales TIC, revelando una alta puntuación por parte de todos los grupos de usuarios. El estudio encontró tendencias similares en cuatro áreas relativas a *i*) conocimiento/aprendizaje, *ii*) interés/motivación, *iii*) usabilidad / practicidad, y *iv*) resultados/viabilidad. Como resultado, el estudio confirmó el potencial de DRACSC para ahorrar tiempo y esfuerzo a los usuarios en entornos educativos y profesionales sin necesidad de conocimiento experto.

La tercera solución presentada en la tesis doctoral consiste en un sistema desarrollado para el almacenamiento de eventos de los equipos de red gestionados mediante blockchain, el cual supera las limitaciones de los métodos encontrados en el estado del arte. Tradicionalmente, el registro de eventos se ha utilizado para detectar comportamientos erróneos o visualizar operaciones de red. Sin embargo, su uso en un posible proceso pericial podría resultar no adecuado al plantear dudas sobre el cumplimiento de los principios de integridad, inmutabilidad y no repudio de los datos. Para hacer frente a la manipulación de los archivos de registro, la tesis pretende proporcionar una solución para crear sistemas seguros de almacenamiento de eventos. Para ello, se presentan la metodología y los experimentos utilizando dos cadenas de bloques diferentes para transacciones con y sin metadatos.

En resumen, estas contribuciones tienen como fin presentar un dispositivo diseñado para simplificar las tareas de administración de sistemas y equipos de red, proporcionando soluciones innovadoras que enriquecen el estado del arte actual. Las soluciones presentadas no solo abordan las limitaciones existentes, sino que también establecen una base sólida para continuar esta línea de investigación con miras a mejorar la eficiencia y el rendimiento de los profesionales del sector TIC.

1.2 Innovaciones aportadas por la tesis doctoral

En este apartado se destacan las novedades aportadas por cada trabajo incluido en la tesis doctoral en referencia a lo publicado anteriormente en el estado del arte.

En primer lugar, se presenta una solución portátil de hardware y software dedicada a unificar la administración de SO, archivos de configuración y servicios de red en equipos de comunicación. Esto incluye la recuperación automática de firmware en condiciones de fallo relativas a la pérdida de la imagen de arranque, la corrupción de memoria Flash y el olvido de contraseñas, entre otros. Esta innovación se ha publicado tanto en patente internacional como en revista internacional de alto impacto.

En segundo lugar, se realizan mejoras significativas en la solución propuesta para superar las limitaciones de hardware y software más avanzadas, fusionando ventajas como la portabilidad, el diseño de funciones MACRO automatizadas, y un repositorio basado en la nube para sincronizar distintos DRACSC y compartir funciones de alto nivel dentro de un mismo entorno de trabajo. Esta nueva versión se ha publicado en un artículo de revista internacional de alto impacto.

Por último, se aporta una solución para el almacenamiento seguro de eventos de equipos gestionados, que permite su inclusión en un entorno productivo de forma sencilla sin necesidad de cambiar la infraestructura subyacente. Esta contribución ha sido publicada también a través de un artículo de revista internacional de alto impacto.

1.3 Aportaciones científicas de la tesis doctoral

Este apartado pretende destacar el contexto científico en el que se ha desarrollado la tesis doctoral y los principales resultados obtenidos. Con esta intención, se detallarán las publicaciones científicas indexadas en revistas internacionales, así como las aportaciones realizadas a congresos internacionales y nacionales. Por último, se detallará el documento de propiedad intelectual publicado. En resumen, el objetivo principal de este apartado es mostrar el soporte científico y los resultados llevados a cabo.

1.3.1 Publicaciones en revistas internacionales

A continuación, se enumeran los trabajos publicados en revistas científicas indexadas en Journal Citation Reports (JCR) durante la realización de la tesis doctoral. Los artículos se presentan por orden cronológico de publicación.

Artículo 1. *Portable Device for Easy Management and Automatic Recovery of Networking Systems*

Autores: J.D. Morillo Reina, T.J. Mateo Sanguino
Revista: IEEE Latin America Transactions
Referencia: 8863310
Año: 2019
DOI: 10.1109/TLA.2019.8863310
Indicios de calidad: Revista incluida en JCR, posición 145/156 (Q4) en la categoría “Computer Science, Information Systems” y posición 234/266 (Q4) en la categoría “Engineering, Electrical & Electronic”
Factor de impacto: 0.782
Número de citas: 2 (Scopus), 6 (ResearchGate) y 5 (Google Scholar)

Este artículo presenta una solución portátil de hardware y software dedicada a unificar la administración de SO, archivos de configuración y servicios de red en equipos de comunicación [1]. Esto incluye la recuperación automática del firmware en condiciones de fallo relativas a la pérdida de la imagen de arranque, la corrupción de la memoria flash y el olvido de contraseñas, entre otros.

El trabajo describe que el prototipo, probado principalmente con enrutadores y conmutadores de Cisco y MikroTik, puede aplicarse también a otros equipos como puntos de acceso, cortafuegos o dispositivos embebidos, ya que busca la universalidad e independencia con dispositivos y fabricantes. Como ventaja, esto facilita el mantenimiento y despliegue de infraestructuras de red, ahorrando así costes y tiempo a los ingenieros.

Por último, una descripción de la experimentación llevada a cabo muestra la importante ventaja que supone el tiempo empleado en realizar algunas funciones automatizadas en lugar de ejecutarlas manualmente. Se trata de una potente herramienta que, gracias a ello, ha sido patentada y ampliada mediante un tratado de cooperación internacional (PCT).

Artículo 2. *Cloud-Based Automatic Configuration and Disaster Recovery of Communication Systems Applied in Engineering Training*

Autores: J.D. Morillo Reina, T.J. Mateo Sanguino
Revista: MDPI Electronics
Referencia: 4203
Año: 2024
DOI: https://doi.org/10.3390/electronics
Indicios de calidad: Revista incluida en JCR, posición 115/250 (Q2) en la categoría “Computer Science, Information Systems” y posición 157/353 (Q2) en la categoría “Engineering, Electrical & Electronic”
Factor de impacto (2023): 2.6
Número de citas: 1 (Scopus), 0 (ResearchGate) y 1 (Google Scholar)

En este artículo se realizan distintas mejoras al prototipo propuesto en la tesis doctoral para superar las limitaciones de hardware y software existentes, fusionando además la ventaja de su portabilidad con funciones MACRO automatizadas y un repositorio basado en la nube como características principales [2].

Esta nueva versión del sistema ha sido probada con 89 usuarios durante el curso 2020/21, incluyendo estudiantes y profesores de centros educativos, y profesionales TIC. Para la validación del sistema se ha realizado una experimentación simulando escenarios de trabajo típicamente utilizados en entornos TIC con dos equipos de red gestionados. Un análisis estadístico aplicando la prueba t de Welch de dos muestras sobre una encuesta de opinión encontró una alta valoración por parte de todos los grupos de usuarios con tendencias similares en

cuatro áreas relativas a "conocimiento/aprendizaje", "interés/motivación", "usabilidad/practicidad", y "resultados/viabilidad".

Como resultado, el estudio confirmó el potencial del sistema propuesto para ahorrar tiempo y esfuerzo a los usuarios en entornos educativos y profesionales sin necesidad de conocimientos expertos.

Artículo 3. *Decentralized and Secure Blockchain Solution for Tamper-proof Logging Events*

Autores: J.D. Morillo Reina, T.J. Mateo Sanguino
Revista: MDPI Future Internet
Referencia: 108
Año: 2025
DOI: 10.3390/fi17030108
Indicios de calidad: Revista incluida en JCR, posición 110/252 (Q2) en la categoría “Computer Science, Information Systems”
Factor de Impacto (2024): 2.8
Número de citaciones: 0 (Scopus), 0 (ResearchGate) y 0 (Google Scholar)

La contribución principal de este artículo es la creación de una nueva mejora para el dispositivo bajo estudio en la tesis doctoral, añadiendo un sistema para el almacenamiento seguro de eventos de equipos de red en producción de forma sencilla sin necesidad de cambiar la infraestructura subyacente [3].

La publicación describe tanto el software desarrollado como la arquitectura implementada en el sistema garantizando la integridad, inmutabilidad y no repudio de los datos a través de diferentes soluciones de registro público basadas en blockchain. Este enfoque ofrece una capa adicional de seguridad a través de un registro descentralizado y resistente a la manipulación.

Finalmente, se lleva a cabo una experimentación para demostrar la eficacia en diversos contextos mediante dos cadenas de bloques diferentes basadas en transacciones con y sin metadatos. Los resultados para transacciones sin metadatos sugieren que los tiempos de respuesta de Solana la hacen muy adecuada para entornos con registros moderadamente críticos que requieren certificación. Por el contrario, Cardano muestra tiempos de respuesta más elevados, lo que la hace adecuada para eventos menos frecuentes con metadatos que requieren legitimidad.

1.3.2 Publicaciones en conferencias nacionales

En este subapartado se incluyen otras publicaciones científicas menores en el ámbito realizadas durante la tesis doctoral. En concreto, tres contribuciones en congresos nacionales.

Artículo 4. *WADC. Enfoque Open Source para la Automatización de Sistemas y Redes*

Autores: J.D. Morillo Reina, T.J. Mateo Sanguino
Evento: III Jornadas ScienCity, Fomento de la Cultura Científica, Tecnológica y de Innovación en Ciudades Inteligentes
Publicación: Libro de Actas de las III Jornadas ScienCity, pp. 37-38
Fecha: 27-29 noviembre 2021. Huelva (España)

Artículo 5. *Evaluación de un Sistema de Gestión para la Configuración y Recuperación de Equipos de Red*

Autores: J.D. Morillo Reina, T.J. Mateo Sanguino
Evento: IV Jornadas ScienCity, Fomento de la Cultura Científica, Tecnológica y de Innovación en Ciudades Inteligentes
Publicación: Libro de Actas de las IV Jornadas ScienCity, pp. 11-14
Fecha: 27-29 noviembre 2022. Huelva (España)

Artículo 6. *Uso de Blockchain en la Gestión de Sistemas de Red*

Autores: J.D. Morillo Reina, T.J. Mateo Sanguino
Evento: V Jornadas ScienCity, Fomento de la Cultura Científica, Tecnológica y de Innovación en Ciudades Inteligentes
Publicación: Libro de Actas de las V Jornadas ScienCity, pp. 28-30
Fecha: 27-29 noviembre 2023. Huelva (España)

1.3.3 Propiedad intelectual

Este subapartado pretende recoger la publicación internacional de la patente que protege el sistema descrito en el primer artículo de la tesis doctoral. Esta patente complementa las aportaciones realizadas en revistas y congresos estableciendo un nuevo enfoque para la recuperación de errores de equipos hardware que componen las redes de datos.

Patente 1. *Device and System for the Recovery of Communication Equipment*

Autores: T.J. Mateo Sanguino, J.D. Morillo Reina

Referencia internacional: WO2016055682 A1
Referencia española: ES2569414 B1
Estado actual: Adjudicado desde el 24 de marzo de 2017

1.4 Justificación e hipótesis

Los centros de datos son vitales en la actualidad debido a la creciente dependencia de la tecnología digital en todas las áreas de la sociedad y los negocios. Son fundamentales para almacenar, procesar y distribuir grandes volúmenes de datos de manera eficiente y segura.

Uno de los desafíos más importantes y recurrentes que encontramos es garantizar la disponibilidad y la fiabilidad en la operación de los centros de datos, ya que afectan directamente a la capacidad de las empresas para acceder a los datos y mantener sus operaciones en funcionamiento.

En este contexto, la hipótesis de esta investigación sostiene que un dispositivo portable y de bajo coste puede simplificar de manera sustancial la recuperación y configuración de equipos de red, además de garantizar la integridad y el no repudio de los eventos registrados en la infraestructura administrada, permitiendo auditorías transparentes en todas las operaciones ejecutadas en la misma. Para ello, dicho dispositivo aprovecharía la conectividad a una plataforma en la nube con el fin de sincronizar, respaldar y compartir de forma segura la información relativa a la configuración, estado y modificaciones realizadas en cada equipo gestionado.

Los objetivos establecidos para la demostración de la hipótesis de esta investigación son los siguientes:

Objetivo 1. Crear un prototipo portable y funcional que permite la simplificación en las tareas de recuperación y configuración de equipos de red.

Objetivo 2. Mejorar el prototipo a nivel hardware y optimizar el software, además de incluir funciones de alto nivel independientes del fabricante, y crear una plataforma en la nube para facilitar y centralizar el uso de estas funciones.

Objetivo 3. Añadir una funcionalidad para almacenar eventos en cadenas de bloques públicas para aquellos equipos que componen la red en la que se encuentra el DRACSC.

1.5 Metodología

Para alcanzar los objetivos definidos en el apartado anterior, se ha seguido una metodología de investigación en esta tesis doctoral. Esta puede resumirse brevemente en un procedimiento genérico de acuerdo con los siguientes puntos:

1) Estudio del estado del arte.
2) Definición de los requisitos de la solución.
3) Desarrollo de nuevas soluciones mediante una aplicación incremental hasta llegar a la solución final.
4) Recogida de datos y análisis de resultados mediante procedimientos objetivos.
5) Generación de las conclusiones a partir de los resultados obtenidos.

Esta metodología se ha aplicado a la investigación realizada en la tesis doctoral del siguiente modo, dando lugar posteriormente a las publicaciones científicas que se presentan en el capítulo 4:

Fase 1. El procedimiento consistió en revisar el estado de la técnica para conocer los pros y los contras de otras soluciones en cuanto a la administración de redes de datos. A continuación, se definieron los requisitos del sistema propuesto. Una vez desarrollado el sistema, se probó su funcionamiento en un entorno real y se validó mediante técnicas cuantitativas. Por último, se obtuvo una conclusión general sobre el sistema para el que se identificaron las limitaciones y posibles mejoras.

Fase 2. Se revisó el estado del arte para conocer los diferentes enfoques utilizados en la gestión de redes de computadores. Después, se establecieron las ventajas del uso de soluciones tanto hardware como software que cubren el problema específico a resolver. A continuación, se implementaron de forma incremental distintas soluciones para solventar las limitaciones encontradas en los sistemas estudiados. Por último, se analizaron los resultados con distintos grupos de usuarios mediante métodos estadísticos obteniendo las conclusiones generales.

Fase 3. De forma similar al primer y segundo objetivo, la primera tarea realizada fue un estudio del estado del arte para conocer las ventajas e inconvenientes de otras soluciones disponibles. Después, se implementó la nueva característica para solventar las limitaciones encontradas en otros estudios. Una vez desarrollada, se realizaron pruebas de estrés y se validaron los resultados mediante técnicas cuantitativas. Por último, se obtuvo una conclusión general sobre el sistema para el que se identificaron las limitaciones y posibles mejoras.

1.6 Estructura

Con respecto a la estructura de la tesis doctoral, esta se ha organizado en cinco capítulos según el siguiente orden:

El capítulo 1, *Planteamiento General*, pretende situar al lector en el contexto de esta tesis doctoral. En él se ofrece una visión general de la tesis, se describen las innovaciones aportadas y las contribuciones científicas realizadas, se muestran los objetivos y la metodología seguida, además de presentar la estructura de la propia tesis doctoral.

El capítulo 2, *Estado del Arte*, hace una descripción de los distintos enforques en la administración de redes de computadores. Este capítulo hace un recorrido por las diferentes soluciones propuestas en la literatura para aumentar la eficiencia y la seguridad en la gestión de este tipo de redes.

En el capítulo 3, *Recursos de Investigación*, se detallan los diferentes elementos hardware y software utilizados para el desarrollo de las soluciones aportadas en los diferentes trabajos que componen la tesis doctoral.

El capítulo 4*, Resultados y Discusión*, incluye el conjunto de contribuciones científicas que sustentan la tesis doctoral. En concreto, este capítulo recopila tres contribuciones en revistas indexadas de alto impacto.

En el capítulo 5*, Conclusiones Generales*, se muestran las conclusiones más relevantes obtenidas en la tesis doctoral, las limitaciones del estudio y las diferentes líneas de investigación actualmente abiertas que pueden llevarse a cabo en futuros trabajos.

Capítulo 2. Estado del Arte

2.1 Introducción

En este capítulo se sitúa al lector en el campo de investigación de la presente tesis doctoral. En primer lugar, se hace un recorrido por las diferentes soluciones software propuestas en el estado del arte para la administración y recuperación de sistemas de comunicación. A continuación, se presentan las soluciones hardware utilizadas para este mismo fin. Por último, se revisa el estado actual de las diversas soluciones que hacen uso de tecnología blockchain para el almacenamiento inmutable de registros del sistema.

2.2 Soluciones software para la gestión de elementos de red

Las siguientes aplicaciones están diseñadas para gestionar y configurar redes de manera eficiente. Facilitan la administración de equipos de red tales como conmutadores y enrutadores, permitiendo configuraciones remotas y simplificadas. Ofrecen funciones para monitorizar el estado de la red, automatizar tareas, editar configuraciones, actualizar firmware y realizar auditorías. En conjunto, proporcionan una interfaz centralizada para optimizar la operación y mantenimiento de redes, ya sea en entornos domésticos o empresariales.

Dentro de este tipo de soluciones se pueden citar Cisco Web Browser User Interface, una interfaz web suministrada con los propios conmutadores y enrutadores de la compañía que requiere una configuración adicional para trabajar con ellos [4]; Linksys Smart Wi-Fi, un conjunto de herramientas con funciones similares a Cisco Connect Express y Cisco Connect Cloud para administrar redes domésticas formadas por puntos de acceso y enrutadores Wi-Fi de forma remota y sencilla [5]; Colibri NetManager, antiguamente TeldaGES de la empresa Teldat, es una plataforma de gestión de enrutadores que aúna la auditoría, visión de red, acceso a equipos, recuperación de históricos de configuraciones y firmware a través de la nube [6]; Network Configuration Manager de la empresa ManageEngine, anteriormente llamado DeviceExpert, se trata de una solución muy potente que centraliza la administración, configuración, automatización de tareas y monitorización de elementos de una red (e.g., conmutadores, enrutadores, cortafuegos, etc.) desde una interfaz web [7]; Network Configuration Manager de la empresa SolarWinds, un referente en este campo que aúna la monitorización y configuración de elementos de red [8]; y WhatsUp Gold de Progress, una herramienta que facilita la automatización de la gestión de dispositivos de red con funcionalidades como auditoría, copias de seguridad de configuraciones y cumplimiento de regulaciones como PCI, SOX y HIPAA [9].

Con objeto de comparar las capacidades y características de las soluciones anteriores respecto a DRACSC se presenta la Tabla 1. Mientras el software de Cisco o Teldat son solo compatibles con sus marcas, tanto el de ManageEngine como el de SolarWinds han sido diseñados para ser compatibles con el mayor número posible de equipos (e.g., Cisco, Juniper, HP, Dell, Brocade, F5, Aruba, Ruckus, etc.). Ello supone una ventaja sobre los demás. Respecto a las funciones, todos los sistemas se rigen por lo que puede llamarse como "camino lógico" dirigido a simplificar la administración de elementos de red, compartiendo por tanto características similares. Al respecto, en la tabla se ha establecido el nivel medio cuando el dispositivo implementa automatización de funciones y alto cuando además incluye monitorización de red o copias de seguridad entre otros. Respecto al coste, cabe mencionar que las licencias de ManageEngine se otorgan por equipo administrado y el paquete básico de 25 equipos supone 1.995 $ más un importe de 399 $ para soporte al año. De forma similar, la adquisición del producto básico de SolarWinds comienza desde 2320 €. Para finalizar, se puede concluir que además de otras ventajas, solo DRACSC puede manejar equipos de red a través del puerto de consola.

2.3 Soluciones hardware para la gestión de elementos de red

Las siguientes soluciones están diseñadas para proporcionar acceso y control a equipos de red a través de puertos serie, Telnet o SSH. Funcionan como terminales o servidores, permitiendo la administración remota o local de equipos mediante una interfaz de consola. Algunos modelos ofrecen características adicionales como cortafuegos, servidores DHCP y VPN. También existen soluciones más económicas que emulan estas funciones mediante hardware más accesible, como Raspberry Pi, que se utiliza para crear servidores de consola de bajo coste, especialmente en entornos educativos.

Respecto a las soluciones hardware que funcionan a modo de terminales o servidores de consola podemos encontrar Cisco Terminal Server [10], el cual puede adquirirse a partir de 2.500 € para el modelo 2610XM-16TS. Entre otras se citan IOLAN SCS48 DAC, un dispositivo del fabricante Perle con precio alrededor de 4.540 € que permite administrar distintos equipos utilizando conexión por consola, Telnet y SSH [11]; Dominion® SX, un dispositivo de Raritan similar al anterior que proporciona acceso, monitorización y control a través del puerto serie [12]; Opengear IM7200, un dispositivo más enfocado a servidor de consola que ofrece mayor número de funciones como cortafuegos, servidor DHCP, VPN, etc. con administración Telnet/SSH sobre puerto serie [13]. Con el objetivo de reducir el

coste nacen algunas soluciones que tratan de emular dicho hardware como Raspisco, un servidor de consola basado en Raspberry Pi, módem 3G, adaptador SerialToUSB y cable de consola para administración remota o local de dispositivos mediante concentrador USB y adaptadores [14]. También en este sentido, pero enfocado al ámbito docente, se usa Raspberry Pi como dispositivo de bajo coste que emula un servidor de consola por software [15]. Por último, DIGI Connect EZ 8 es un servidor de consola versátil que admite comunicación RS-232/422/485, lo que la hace adecuada para una variedad de aplicaciones industriales. El dispositivo es fácil de configurar a través de una interfaz web intuitiva y proporciona distintas características de seguridad como cifrado SSL/TLS y SSH para la transmisión segura de datos [16].

La Tabla 2 presenta una comparativa de las capacidades y características de las soluciones anteriores respecto a la basada en DRACSC. Cabe mencionar que Dominion® SX es capaz de interactuar con distintos fabricantes como HP, Dell, Cisco o IBM. Salvo Raspisco, que es un prototipo aún en desarrollo, la mayoría de las soluciones se comunican a través del servicio Shell-in-a-box. Esto permite encapsular cualquier comunicación a través de un túnel SSH y tener acceso a cualquier equipo. Respecto a la solución DRACSC, cabe destacar la portabilidad y capacidad de recuperación de desastres más allá de la simple restauración de la configuración. Otra de las características que dispone DRACSC es el uso de un espacio de almacenamiento para guardar datos de conmutadores y enrutadores cuyo protocolo de intercambio de ficheros es TFTP (RFC1350) y FTP (RFC1123).

TABLA 1
CARACTERÍSTICAS DE SOLUCIONES SOFTWARE PARA GESTIÓN DE DISPOSITIVOS DE RED

Nombre	Gestión de equipos en paralelo	Fabricantes admitidos	Grado de funcionalidad	Control por consola	Control por Telnet/SSH	Función de recuperación del sistema	Repositorio en la nube	Coste
CISCO Web Browser User Interface	Uno	Cisco	Medio	No	No	No	No	Gratuito
Linksys Smart Wi-Fi	Varios	Cisco	Medio	No	No	No	✓	Gratuito
Colibri NetManager	Muchos	Teldat	Alto	No	✓	✓	✓	-
Solarwinds Network Configuration Manager	Muchos	Muchos	Alto	No	✓	No	✓	Alto
Network Configuration Manager	Muchos	Muchos	Alto	No	✓	No	No	Alto
WhatsUp Gold	Muchos	Muchos	Alto	No	No	✓	✓	Alto
DRACSC	Varios	Varios	Medio	✓	✓	✓	✓	Bajo

TABLA 2
CARACTERÍSTICAS DE SOLUCIONES HARDWARE PARA GESTIÓN DE DISPOSITIVOS DE RED

Nombre	Gestión de equipos en paralelo	Fabricantes admitidos	Portabilidad	Grado de funcionalidad	Control por consola	Control por Telnet/SSH	Función de recuperación del sistema	Repositorio en la nube	Coste
Aten SN0116CO	Varios	Varios	No	Medio	✓	✓	No	No	Alto
IOLAN SCS48 DAC	Varios	Varios	No	Medio	✓	✓	No	No	Alto
Dominion® SX II	Varios	Varios	No	Alto	✓	No	No	No	Alto
Opengear IM7200	Varios	Varios	No	Alto	✓	✓	No	No	Alto
Raspisco	One	Varios	✓	Bajo	✓	No	No	No	Free
Servidor de consola basado en Raspberry Pi	Varios	Varios	No	Bajo	✓	No	No	No	Free
Digi Connect EZ® 8	Varios	Varios	No	Alto	✓	✓	No	No	Alto
DRACSC	Varios	Varios	✓	Medio	✓	✓	✓	✓	Bajo

2.4 Soluciones para el almacenamiento inmutable de registros del sistema

Existen diversos estudios en el estado del arte centrados en el uso de cadenas de bloques para la trazabilidad de operaciones. En el contexto de esta tesis doctoral destacan aquellas soluciones centradas en la persistencia de registros de actividad de los sistemas informáticos. Los ficheros de registro generados por dichos sistemas cubren un amplio abanico de posibilidades, incluyendo accesos de usuarios, incidentes de seguridad y eventos generados tanto por aplicaciones como por equipos de red. Uno de los principales usos se encuentra en los procesos de auditoría, en los que se requiere persistencia y procesamiento de ficheros log para conocer el grado de cumplimiento que una entidad tiene respecto a una normativa específica [17].

La primera solución analizada en este trabajo propone un sistema auditable utilizando Exonum, una cadena de bloques privada [18]. Entre las características a destacar se encuentra el uso del algoritmo de consenso Byzantine Fault Tolerance (BFT), la capacidad de ejecutar contratos inteligentes y el uso de cadenas de bloques públicas para establecer fiabilidad adicional a través de Bitcoin. Este sistema se integra con una solución de gestión de eventos e información de seguridad (SIEM) para adquirir eventos a través del estándar Syslog. Su arquitectura híbrida garantiza la inmutabilidad e integridad de los registros almacenándolos tanto en la cadena, mediante hashes, como fuera de ella en un clúster local. Al mismo tiempo, permite un acceso eficaz y una gestión adecuada de los eventos de seguridad. No obstante, el nivel de transparencia y fiabilidad del sistema se ve comprometido al almacenar parte de los datos fuera de la cadena de bloques pública. Además, el uso de contratos inteligentes complica su implementación.

Un trabajo diferente utiliza Hyperledger [19], una cadena de bloques privada implementada para almacenar archivos de registro cifrados por los usuarios antes de ser enviados. Almacenar los archivos de registro directamente en la cadena de bloques plantea problemas a la hora de garantizar la integridad de los datos, ya que resulta imposible verificar que los datos no han sido manipulados antes de incluirlos en la cadena. Además, de forma similar al escenario anterior, implementar una cadena de bloques privada reduce la transparencia y fiabilidad del sistema.

Otro trabajo propone un sistema que ayuda a los auditores a verificar el cumplimiento de la normativa utilizando Bitcoin [20], una cadena de bloques pública que utiliza el mecanismo de consenso proof of work (PoW). Los datos se almacenan tanto dentro como fuera de la cadena, pero el sistema no está diseñado

para almacenar eventos generados por equipos de red. Además, ni el precio de las tasas ni el número de transacciones por segundo de la red Bitcoin son escalables para su uso en este contexto.

Por otra parte, BlockAudit propone un sistema seguro y transparente de auditoría de registros encargado de almacenar las acciones ejecutadas en el sistema [21]. Registra los datos a través de la cadena de bloques privada Hyperledger, por lo que carece de transparencia y fiabilidad en comparación con las cadenas de bloques públicas.

En otra solución, se propone un sistema autónomo de gestión de almacenamiento de logs para dispositivos IoT que utiliza una cadena de bloques tanto pública como privada mediante Ethereum e Hyperledger, respectivamente [22]. El sistema almacena el contenido de los registros en la cadena de bloques privada y sus firmas en la cadena de bloques pública, cuyas transacciones se gestionan mediante contratos inteligentes. Un resumen de estas transacciones se firma y almacena en la cadena de bloques pública, proporcionando mayor transparencia y fiabilidad. Sin embargo, sigue existiendo el riesgo de manipulación de los bloques en la cadena de bloques privada antes de que se envíen a la cadena de bloques pública. Además, el sistema está sujeto a la volatilidad del precio de las transacciones impuesta por Ethereum.

En otro trabajo, se introduce el sistema de gestión segura de registros basado en cadena de bloques para computación en la nube (BCALS) utilizando Multichain, una cadena de bloques privada que almacena mensajes de registro [23]. Los datos contenidos en los registros de eventos se envían a Elasticsearch para su posterior explotación, un proyecto de código abierto utilizado para buscar, analizar y visualizar grandes cantidades de información en tiempo real. Como en los casos anteriores, el uso de una cadena de bloques privada implica un compromiso en el nivel de transparencia y fiabilidad del sistema.

Por otro lado, se propone el uso de Hyperledger como cadena de bloques privada, donde los contratos inteligentes gestionan las transacciones y la información asociada a los eventos se almacena directamente en la cadena de bloques [24]. Este proyecto de código abierto está diseñado específicamente para dispositivos IoT y se centra en su seguridad. Sin embargo, la transparencia y fiabilidad de una red privada difieren significativamente de las de una cadena de bloques pública.

Otro trabajo trata sobre un servicio basado en cadena de bloques para gestión de suministros y logística [25], también conocido como Logchain Logistics as a

Service (LCaaS). Este servicio se divide en dos niveles jerárquicos para lograr la escalabilidad. El primero almacena las firmas en una cadena de bloques que puede ser pública (e.g., Ethereum) o privada (e.g., cadena de bloques de IBM), mientras que los datos correspondientes a las firmas se almacenan tanto fuera como dentro de la cadena. Sin embargo, el servicio hace imposible certificar que dichos datos no han sido modificados antes de su inclusión cuando los archivos de registro se almacenan en la cadena de bloques elegida.

En el siguiente estudio se ha presentado un sistema escalable de gestión de registros basado en cadena de bloques, que utiliza Multichain para almacenar hashes, marcas de tiempo y datos de propiedad, a la vez que emplea el InterPlanetary File System (IPFS) para almacenar los registros completos [26]. También cuenta con un mecanismo de consulta que permite una recuperación y verificación eficiente de los registros. Esto aborda de manera efectiva las limitaciones que a menudo se encuentran en este tipo de soluciones, que típicamente carecen de capacidades de búsqueda estructurada. Sin embargo, al igual que en ejemplos anteriores, depender de una cadena de bloques privada puede disminuir la transparencia y fiabilidad del sistema.

Otro trabajo diferente propone una red descentralizada de almacenamiento de datos utilizando una cadena de bloques personalizada —desarrollada en Go— para evitar la manipulación de datos [27]. Este sistema emplea un mecanismo de consenso PoW para mantener la integridad de los datos y usa un protocolo Gossip para la reparación automática de bloques, permitiendo a los nodos detectar inconsistencias y recuperarse a un estado válido. Los datos se almacenan en la cadena para garantizar la inmutabilidad, mientras una API Gateway facilita las transacciones de registros a través de endpoints RESTful. No obstante, el uso de una blockchain privada y personalizada podría generar preocupaciones sobre la transparencia y fiabilidad.

Por último, otro trabajo propone un sistema de auditoría basado en cadena de bloques para garantizar la integridad de los datos en archivos de registro [28]. El sistema asume que tanto los registradores como los auditores pueden no ser confiables y mitiga estos riesgos utilizando contratos inteligentes dentro de Hyperledger Fabric. Este método permite generar pruebas de integridad en la cadena utilizando tokens no fungibles (NFT) con archivos de registro almacenados fuera de la cadena mediante el uso de IPFS. Si bien este sistema mejora la escalabilidad y seguridad, está limitado a una cadena de bloques privada, lo que puede reducir la transparencia en comparación con las soluciones públicas.

La Tabla 3 presenta una comparación de las soluciones anteriores con el enfoque presentado en este documento. Cabe destacar que solo la solución propuesta en esta tesis doctoral y la presentada en [18] utilizan el estándar Syslog como fuente de datos para garantizar la compatibilidad con herramientas y aplicaciones diseñadas para trabajar con este estándar. También cabe destacar que la mayoría de los trabajos propuestos utilizan redes privadas debido al coste que supone almacenar los datos en cadena de bloques públicas, lo que implica una menor transparencia en el proceso. Por otro lado, solo [18], [24] y [25] utilizan algunas herramientas adicionales para visualizar la información almacenada de forma amigable, como en la solución propuesta. Por último, cabe destacar el carácter de código abierto del sistema propuesto junto con la solución descrita en [24], que permite la transparencia y facilita la liberación del proyecto para su reproducibilidad por parte de la comunidad científica. El objetivo último es proponer un estándar industrial para el almacenamiento de registros mediante cadena de bloques.

Finalmente, la Tabla 4 presenta una visión comparativa de las cadenas de bloques usadas para enfatizar los aspectos clave discutidos en esta sección. Las cadenas de bloques públicas, incluyendo Bitcoin, Ethereum, Cardano y Solana, se caracterizan por su participación abierta y gobernanza descentralizada. Mientras que Bitcoin utiliza PoW con aproximadamente siete transacciones por segundo (TPS), Ethereum ha migrado a PoS con alrededor de 15 TPS. Cardano ofrece un modelo de tarifas deterministas con 250 TPS y Solana mejora el rendimiento con una combinación de PoH y PoS, alcanzando hasta 65.000 TPS. Por otro lado, las cadenas de bloques privadas como Multichain, Exonum, Hyperledger Fabric e IBM Blockchain están diseñadas para aplicaciones empresariales con mecanismos de consenso configurables y estructuras de tarifas administradas.

TABLA 3
CARACTERÍSTICAS DE LAS SOLUCIONES BLOCKCHAIN PARA ALMACENAMIENTO SEGURO DE EVENTOS

Referencia	Almacenamiento de los datos	Tipo de Blockchain	Algoritmo de Consenso	Trasferencia	Especifica para logs	Visualización de los datos	Multi blockchain	Open Source	Año
[18]	Off-chain / On-chain	Privada	BFT	Directa	✓	Web app	No	No	2019
[19]	On-chain	Privada	BFT	Directa	✓	No	No	No	2018
[20]	Off-chain / On-chain	Pública	PoW	Directa	✓	No	No	No	2017
[21]	On-chain	Privada	BFT	Directa	✓	No	No	No	2018
[22]	Off-chain / On-chain	Ambas	PoW	Contratos inteligentes	✓	Web app	No	No	2020
[23]	On-Chain	Privada	PBFT	Directa	✓	Elasticsearch	No	No	2021
[24]	On-Chain	Privada	PBFT	Contratos inteligentes	No	No	No	No	2022
[25]	Off-chain / On-chain	Privada	BFT	Directa	✓	No	✓	✓	2018
[26]	Off-chain / On-chain	Privada	Permission-based mining	Directa	✓	Web app	No	No	2022
[27]	On-chain	Privada	PoW	Directa	✓	No	No	No	2024
[28]	Off-chain / On-chain	Privada	PBFT	Contratos inteligentes	✓	No	No	No	2024
[3]	Off-chain / On-chain	Publica	Varios*	Directa	✓	Web app	✓	✓	2024

* PoH+PoS para Solana y PoS para Cardano

TABLA 4
COMPARATIVA ENTRE DIFERENTES CADENAS DE BLOQUES DEL ESTADO DEL ARTE

Cadena de bloques	Tipo	Algoritmo de Consenso	TPS	Coste de transacción	Costes deterministicos	Contratos inteligentes	Año de lanzamiento
Bitcoin	Pública	PoW	7	Alto	No	Limitado	2019
Ethereum	Pública	PoS	100**	Alto	No	✓	2018
Multichain	Privada	Varios	*	*	✓	Limitado	2017
Exonum	Privada	BFT	*	*	✓	✓	2018
Hyperledger Fabric	Privada	Varios	*	*	✓	✓	2020
Cardano	Pública	PoS	250	Bajo	✓	✓	2021
IBM Blockchain	Privada	Varios	*	*	✓	No	2022
Solana	Pública	PoH + PoS	4.000**	Bajo	No	✓	2018

* Estas características de las cadenas de bloques privadas dependen de la configuración
** Datos actualizados con respecto a los publicados en el artículo original [3]

Capítulo 3. Recursos para la Investigación

3.1 Introducción

En este capítulo se presentan los recursos utilizados durante el desarrollo de la investigación llevada a cabo en la tesis doctoral. Estos se han agrupado en función de su naturaleza (i.e., elementos software y hardware). Asimismo, el software utilizado se ha dividido en tres categorías: *i*) software utilizado para el desarrollo del dispositivo DRACSC; *ii*) software utilizado para el desarrollo del repositorio de macros en la nube; y *iii*) software utilizado en el registro de eventos en la cadena de bloques a prueba de manipulaciones. Para ello, el capítulo comienza exponiendo el hardware utilizado durante el desarrollo del DRACSC. A continuación, se describen el software utilizado en el mismo. El siguiente apartado contiene el software utilizado en la creación del repositorio de macros en la nube. Por último, se detallan los elementos de la solución basada en la cadena de bloques.

3.2 Elementos hardware

Durante la presente tesis doctoral se ha diseñado, implementado y validado el dispositivo DRACSC (Figura 1).

Con objeto de conseguir un prototipo basado en componentes de bajo coste y ampliamente disponibles en el mercado, el sistema ha sido implementado con Raspberry Pi, un hardware comúnmente utilizado en otros proyectos de investigación que no requiere elevados recursos.

Otro factor decisivo para su elección fue la amplia comunidad que lo respalda, lo que repercute notablemente tanto en los tiempos de desarrollo como en la compatibilidad con los periféricos. En concreto, el hardware utilizado ha sido Raspberry Pi 4 Model B que incluye un procesador Broadcom BCM2711 Quadcore Cortex-A72 (ARM v8) 64-bit SoC de 1.5GHz y 4 GB de LPDDR4-3200 SDRAM con un rendimiento aceptable en aplicaciones similares.

La funcionalidad de DRACSC ha sido extendida con una pantalla TFT resistiva de 5 pulgadas (800 x 480 píxeles), una carcasa diseñada e impresa en 3D con un filamento de plástico ABS (Figura 2), una memoria micro SD Kingston de 32 GB (clase 10, 45 MB/s) cuya función es almacenar imágenes del sistema operativo de los equipos gestionados, un cable de consola de tipo serial a RJ45 para gestionar los equipos a través del puerto de consola, un adaptador USBtoRS232 y

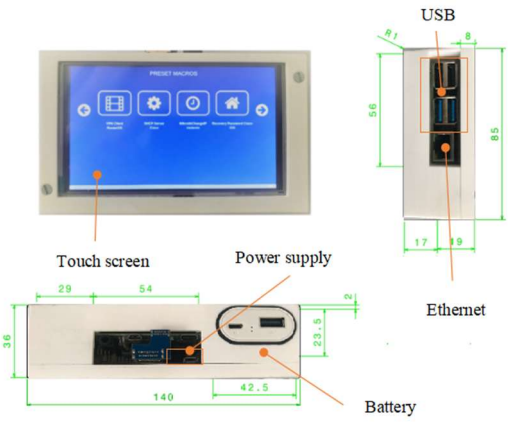


Figura 1. Dispositivo DRACSC

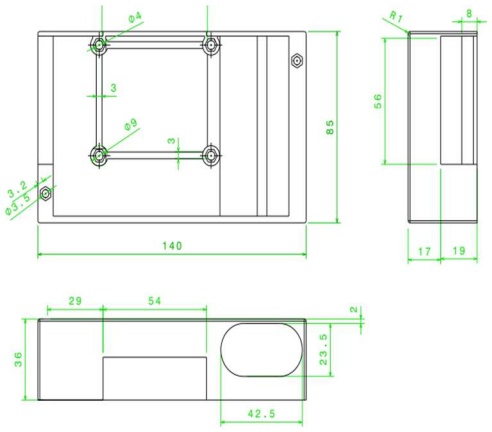


Figura 2. Plano de la carcasa

una batería modelo Contact LXBA4000U de 4000 mAh que permite una autonomía aproximada de 4 horas con un consumo de 1000 mAh.

Por otra parte, se han utilizado varios dispositivos de red, divididos en enrutadores y conmutadores, en los que se han apoyado los desarrollos e investigaciones de esta tesis.

Entre los enrutadores se encuentra el modelo Cisco 827 (Figura 3), el cual ofrece especificaciones de hardware diseñadas para entornos de pequeñas oficinas y oficinas domésticas (SOHO), incluyendo 16 MB de memoria DRAM ampliable hasta 32 MB, 8 MB de memoria Flash con posibilidad de expansión a 16 MB y un procesador RISC que opera a 80 MHz. Para el fabricante Mikrotik, se ha utilizado el modelo hAP lite (Figura 4), que es un enrutador inalámbrico compacto que incorpora un procesador QCA9533 a 650 MHz, 32 MB de memoria RAM y 16 MB de memoria Flash. Por último, el modelo D-Link DSR-1000N (Figura 5) es un enrutador empresarial que integra capacidades avanzadas de enrutamiento, seguridad y conectividad inalámbrica. Cuenta con un procesador Cavium CNS3420 a 300 MHz, 256 MB de memoria RAM y 32 MB de memoria Flash. Además, provee dos puertos WAN Gigabit, lo que facilita el balanceo de carga y la redundancia en la conexión a Internet.

En lo que respecta a los conmutadores, se utilizó el modelo Cisco Catalyst 2960 (Figura 6), un conmutador de nivel empresarial. Este dispone de 24 puertos Gigabit Ethernet (10/100/1000 Mbps), además de puertos uplink dedicados, que pueden ser interfaces combinadas RJ-45/SFP o puertos SFP de tipo Gigabit

Figura 3. Enrutador Cisco 827

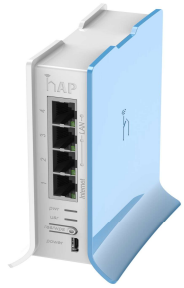


Figura 4. Enrutador Mikrotik Hap Lite

Figura 5. Enrutador D-Link DSR-1000N

Figura 6. Conmutador Cisco Catalyst 2960

Ethernet. Incorporan una CPU basada en arquitectura RISC, 64 MB de memoria DRAM y 32 MB de memoria Flash para almacenar el SO y las configuraciones. Dispone de soporte para montaje en rack de 19 pulgadas y LED frontales para el diagnóstico rápido del estado de los puertos y del sistema.

3.3 Elementos software

En este apartado se detallan los lenguajes de programación, librerías y otros elementos de software utilizados en el desarrollo de las distintas soluciones que conforman el sistema propuesto. Cada una de estas herramientas ha sido seleccionada para satisfacer las necesidades técnicas y operativas de cada componente, siendo la elección e implementación del software un punto clave para garantizar tanto la funcionalidad como la fiabilidad de todo el sistema en conjunto.

3.3.1 Software del dispositivo DRACSC

Respecto al SO se ha seleccionado Raspberry Pi OS 5.10 por ser una versión de Debian optimizada para el hardware de Raspberry Pi. Además, al ser oficial está

respaldada por la comunidad de Debian y la Fundación Raspberry Pi, ofreciendo actualizaciones regulares y soporte a largo plazo.

Sobre ella se han desarrollado las aplicaciones del sistema mediante Ruby 2.3.2, un lenguaje de programación interpretado y orientado a objetos distribuido

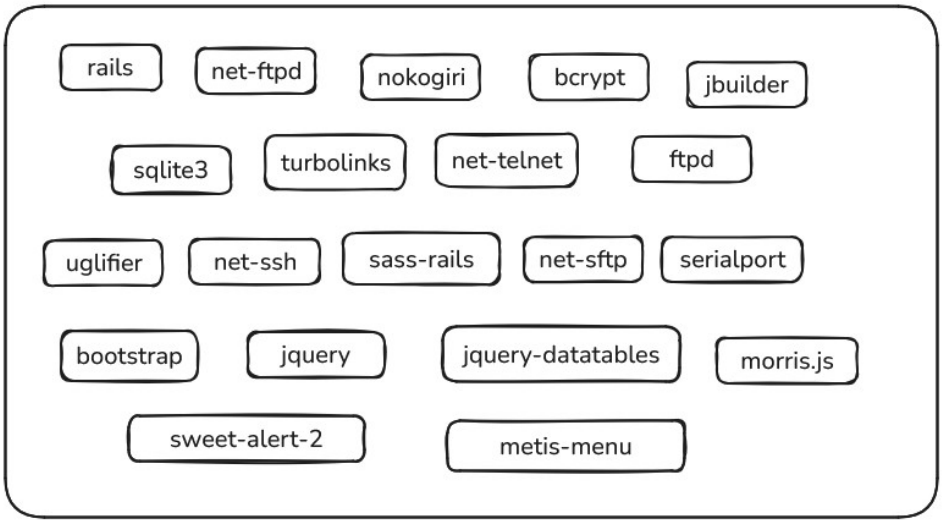


Figura 7. Librerías usadas en el software del dispositivo DRACSC

bajo software libre. Ruby cuenta con una amplia comunidad y un ecosistema rico en bibliotecas y herramientas, facilitando la implementación de funcionalidades complejas.

Para gestionar las dependencias y entornos de proyecto en Ruby, se ha utilizado Bundler. Esta herramienta permite definir y administrar de manera eficiente las librerías —denominadas gemas en este lenguaje— necesarias para el proyecto, garantizando la consistencia entre diferentes entornos de desarrollo y producción.

Entre las gemas utilizadas (Figura 7) se pueden destacar:

- **rails:** Es el entorno de trabajo principal utilizado para el desarrollo de la aplicación web. Rails sigue el patrón modelo-vista-controlador (MVC) y ofrece una estructura robusta para crear aplicaciones escalables y mantenibles.

- **sqlite3**: Permite la interacción con la base de datos SQLite utilizada en el sistema. Facilita las operaciones de almacenamiento y recuperación de datos de forma eficiente.
- **nokogiri:** Biblioteca para el análisis y manipulación de documentos XML y HTML. Se utiliza para procesar archivos de configuración y gestionar datos estructurados necesarios en el sistema.
- **sass-Rails:** Permite el uso de hojas de estilo dinámicas y mantenibles a través de la integración de Rails con SASS (Syntactically Awesome Style Sheets).
- **uglifier:** Herramienta que comprime y optimiza archivos JavaScript, mejorando el rendimiento de la aplicación al reducir el tamaño de los archivos servidos al cliente.
- **turbolinks:** Mejora la velocidad de navegación dentro de la aplicación web al evitar recargas completas de página, lo que resulta en una experiencia de usuario más fluida.
- **jbuilder:** Facilita la creación de respuestas JSON, esencial para construir APIs y servicios web que interactúan con otros sistemas o aplicaciones.
- **bcrypt:** Utilizado para cifrar contraseñas y manejar autenticación segura. Implementa el algoritmo bcrypt, proporcionando una capa adicional de seguridad para los datos de los usuarios.
- **ftpd:** Implementa un servidor FTP en Ruby, permitiendo gestionar transferencias de archivos a través del protocolo FTP dentro del sistema DRACSC.
- **net-tftp:** Proporciona funcionalidades de cliente TFTP, utilizado para transferencias simples de archivos, especialmente en entornos donde se requiere un protocolo ligero.
- **net-sftp:** Ofrece capacidades de cliente SFTP, permitiendo transferencias de archivos seguras sobre SSH.
- **net-telnet:** Biblioteca para implementar clientes Telnet, utilizada para establecer conexiones remotas y controlar dispositivos a través de este protocolo.
- **net-ssh:** Proporciona funcionalidades para establecer conexiones SSH, permitiendo comunicaciones seguras y cifradas con otros sistemas.
- **serialport:** Permite el acceso y control de puertos serie, esencial para la comunicación con equipos de hardware que se conectan a través de interfaces seriales.

La interfaz de usuario de la aplicación web se ha desarrollado utilizando tecnologías estándar como HTML, JavaScript y CSS, garantizando compatibilidad y accesibilidad en múltiples dispositivos y navegadores. Ello ha permitido implementar versiones de interfaz similares que corresponden a pantallas de escritorio y pantallas móviles, proporcionando una amplia gama de componentes y estilos predefinidos que aceleran el desarrollo y aseguran una estética profesional. Algunas de las librerías utilizas son:

- **Bootstrap:** Este potente entorno de trabajo de diseño web permite crear interfaces adaptables al tamaño y tipo de pantalla del dispositivo del usuario.
- **jQuery:** Biblioteca de JavaScript que simplifica la manipulación del DOM (Document Object Model), el manejo de eventos, animaciones y peticiones AJAX. Ha sido fundamental para desarrollar interacciones complejas de manera más sencilla y eficiente.
- **jQuery DataTables:** Plugin para jQuery que facilita la creación de tablas dinámicas y altamente personalizables. Ofrece funcionalidades avanzadas como paginación, búsqueda, ordenación y manejo de grandes volúmenes de datos, mejorando significativamente la experiencia del usuario al interactuar con tablas en la aplicación.
- **SweetAlert2:** Librería que proporciona alertas y cuadros de diálogo personalizados y atractivos. Mejora la comunicación con el usuario al presentar mensajes y notificaciones de manera más amigable e intuitiva, sustituyendo las alertas estándar del navegador por diseños más modernos y flexibles.
- **Morris.js:** Librería de gráficos que permite visualizar datos de forma sencilla y estética. Facilita la creación de gráficos de líneas, áreas, barras y donuts, lo cual es esencial para representar visualmente información relevante en el sistema y ayudar al usuario en la interpretación de datos.
- **MetisMenu:** Plugin de jQuery utilizado para crear menús desplegables y anidados en la interfaz web. Ha permitido implementar una navegación más organizada y accesible dentro de la aplicación, mejorando la usabilidad y facilitando el acceso a las diferentes secciones del sistema.

Con respecto al servidor de aplicaciones web, se ha seleccionado Phusion Passenger. Se trata de un servicio de código abierto diseñado para funcionar de manera óptima con aplicaciones desarrolladas en Ruby on Rails (RoR) y, por tanto, compatible con el sistema. Este servidor basado en Nginx es capaz de manejar peticiones HTTP, administrar procesos y recursos con menor carga para el sistema

que otros servidores web (e.g., GlassFish), así como permite la monitorización y diagnóstico de problemas.

Por otro lado, se ha diseñado una base de datos mediante SQLite debido a que soporta una estructura de datos relacional que consume pocos recursos y es compatible con el modelo ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Esto significa que controla la atomicidad, consistencia, aislamiento y durabilidad de los datos en una biblioteca escrita en lenguaje C que ocupa tan solo ~500 KB. Como ventaja adicional, SQLite se enlaza con el programa principal pasando a ser parte integral del mismo, reduciendo así la latencia de acceso al contrario que otros gestores de base de datos como PostgreSQL, MySQL o MariaDB.

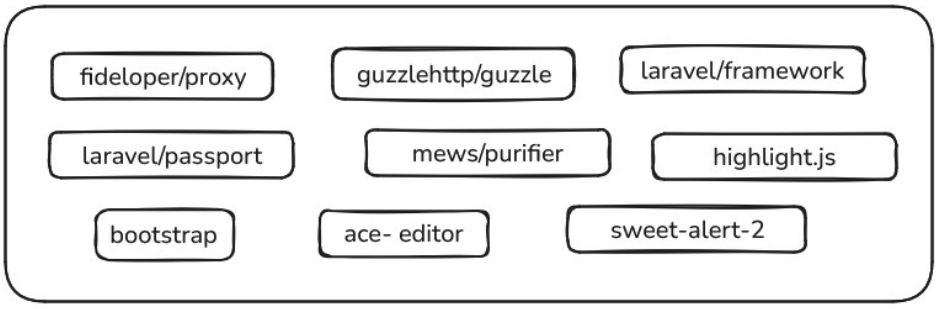


Figura 8. Librerías usadas en el repositorio de macros en la nube

Respecto a los terminales de conexión, se ha utilizado el servicio Ser2net para encapsular la conexión de consola por el puerto serie en Telnet.

3.3.2 Software del repositorio de macros en la nube

El repositorio ha sido desarrollado mediante Laravel, un entorno de trabajo de código abierto para el desarrollo de aplicaciones y servicios web con PHP. En concreto, se ha utilizado la versión 7.4 que, al igual que Ruby, es un lenguaje interpretado y distribuido bajo software libre.

Para la gestión de librerías y dependencias se ha utilizado Composer, el gestor de paquetes estándar en PHP. Composer facilita la instalación y actualización de paquetes, asegurando la compatibilidad y evitando conflictos entre dependencias. Entre las librerías más relevantes (Figura 8) utilizadas se encuentran:

- **fideloper/proxy:** Es esencial cuando la aplicación se ejecuta detrás de un servidor intermediario o proxy (como Nginx o Apache configurado como

proxy inverso), ya que permite gestionar correctamente las cabeceras de las solicitudes HTTP y asegurar que la aplicación obtiene la información correcta del cliente, como la dirección IP real.

- **guzzlehttp/guzzle:** Cliente HTTP para PHP que simplifica la realización de solicitudes HTTP y la integración con servicios web externos. Se utiliza para facilitar la integración con APIs de terceros, manejar peticiones y respuestas HTTP de manera eficiente. Resulta fundamental para la comunicación con otros servicios web desde el repositorio.
- **laravel/framework:** Entorno de trabajo que incluye todas las funcionalidades base que ofrece Laravel para el desarrollo de aplicaciones web como el sistema de enrutamiento, controladores, vistas, manejo de bases de datos con Eloquent ORM y otros componentes esenciales para la aplicación.
- **laravel/passport:** Paquete oficial de Laravel que proporciona una implementación completa de OAuth2 para la autenticación de usuarios y la gestión de tokens de acceso. Se utiliza para asegurar las APIs del repositorio, permitiendo que DRACSC y otras aplicaciones puedan autenticarse y comunicarse de forma segura con el repositorio mediante tokens.
- **mews/purifier:** Integra HTMLPurifier con Laravel para la sanitización de entradas HTML. Se utiliza para limpiar y purificar los datos proporcionados por los usuarios, eliminando cualquier código malicioso o no deseado. Esto mejora la seguridad de la aplicación al prevenir ataques como XSS (Cross-Site Scripting).

La interfaz de usuario del repositorio web se ha desarrollado utilizando tecnologías estándar como HTML, CSS y JavaScript, garantizando compatibilidad y accesibilidad en múltiples dispositivos y navegadores. Para mejorar la experiencia de usuario con diseño adaptable, se han integrado las siguientes herramientas y librerías:

- **Bootstrap:** Entorno de trabajo de diseño web que permite crear interfaces adaptadas de forma dinámica al tamaño y tipo de pantalla del dispositivo del usuario.
- **Highlight.js**: Librería para resaltar la sintaxis de código en diferentes lenguajes de programación, mejorando la legibilidad y comprensión de los fragmentos de código en el repositorio.
- **SweetAlert2:** Proporciona alertas y cuadros de diálogo personalizados y estéticamente agradables, mejorando la comunicación con el usuario y la interacción general con la aplicación.

- **Ace Editor:** Ofrece un editor de código enriquecido directamente en el navegador, permitiendo a los usuarios visualizar y editar macros de forma interactiva con soporte para resaltado de sintaxis y otras funcionalidades avanzadas.

Para la gestión de datos se ha utilizado MariaDB, un sistema gestor de bases de datos relacional derivado de MySQL y distribuido bajo licencia GPL. Otra gran ventaja es la gran comunidad que lo respalda, que ofrece un amplio soporte y actualizaciones regulares que mejoran la seguridad y funcionalidad del sistema.

El repositorio ha sido alojado en Hosting Cloud Linux. Específicamente, se ha utilizado una distribución Debian con rendimiento escalable, 2.5GB de RAM, certificado SSL, protección contra DDoS y georredundancia para evitar problemas de disponibilidad.

La última característica destacable es que dispone de una API REST, lo que permite la comunicación entre aplicaciones a través del protocolo HTTP. Esta interfaz es esencial para la integración con los sistemas DRACSC, ya que permite que los dispositivos se comuniquen con el repositorio en la nube, facilitando la sincronización y actualización de macros de forma automatizada. Utiliza una clave de acceso o API-KEY para autenticar las solicitudes, asegurando que solo los sistemas autorizados puedan acceder a los recursos y proporcionando así una autenticación segura.

3.3.3 Software para la solución basada en cadena de bloques

Esta sección describe los componentes utilizados en el sistema, cuya interacción se resume en la Figura 9 mediante cuatro pasos: 1) el equipo de red gestionado envía un evento log al servidor Syslog alojado en el sistema DRACSC, 2) los eventos se van encolando en el gestor de mensajes (bróker), para ser procesados posteriormente por el módulo de logs, 3) dependiendo de la cadena de bloques utilizada, el módulo de logs llevará a cabo unas tareas determinadas como obtener el hash de la información del evento, conectarse a un nodo de la cadena de bloques y enviar la correspondiente transacción a la red, 4) cuando se dispone del hash obtenido en el paso anterior y se firma, se envían al servidor de gestión de logs para su almacenamiento y posterior explotación.

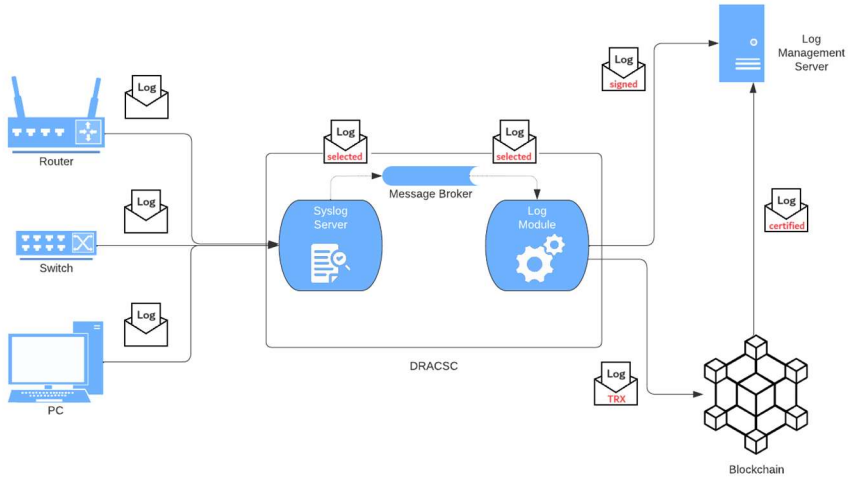


Figura 9. Elementos de DRACSC que intervienen en la certificación de logs

3.3.3.1 Servidor Syslog

Este componente tiene como objetivo recibir, transformar, filtrar, almacenar o redirigir los registros log generados por los equipos de red gestionados. Para ello, los equipos de red gestionados deberán tener activado un servicio de notificación de eventos Syslog. Al ser un estándar muy extendido, basado en el documento RFC 5424 [11], la configuración suele ser sencilla. Por otra parte, el servidor Syslog se encontraría instalado en el sistema DRACSC. Para esta Tesis Doctoral se ha optado por utilizar la versión syslog-ng de código abierto, más concretamente la versión 4.1.1. Los datos que llegan al servidor Syslog son reenviados hacia el bróker de mensajes mediante el protocolo Advanced Message Queuing Protocol (AMQP) a la espera de ser procesados en las siguientes etapas.

3.3.3.2 Message Broker

El objetivo de este componente es almacenar de forma temporal los mensajes de eventos mientras son atendidos por el módulo log. El bróker de mensajes se ha implementado en DRACSC mediante el sistema de intermediación de mensajes de código abierto RabbitMQ, más concretamente la versión 3.10.5. Está basado en AMQP, que tiene como principales ventajas la escalabilidad y la asincronía. El uso de este software aporta beneficios adicionales, como el desacoplamiento de los servicios del sistema y la garantía de que los eventos se conservarán, incluso si ocurren fallos en el software encargado de procesarlos.

3.3.3.3 Módulo Log

La función principal de este servicio, alojado en el sistema DRACSC, es recibir los registros de log filtrados a través del message broker. Una vez recibidos, este módulo procesa los registros para adaptarlos tanto al formato requerido por la cadena de bloques como al del servidor de gestión de logs. Esto garantiza que la información de registro se almacene y gestione de manera eficiente y segura en los distintos sistemas de registro y auditoría.

El lenguaje de programación utilizado ha sido Node.js, específicamente la versión 18.21. La elección viene motivada por su alto rendimiento y asincronía al estar basado en eventos. Además, cuenta con una extensa comunidad y un rico ecosistema de módulos y paquetes disponibles a través de NPM (Node Package Manager), lo que acelera el desarrollo y permite incorporar funcionalidades adicionales con facilidad.

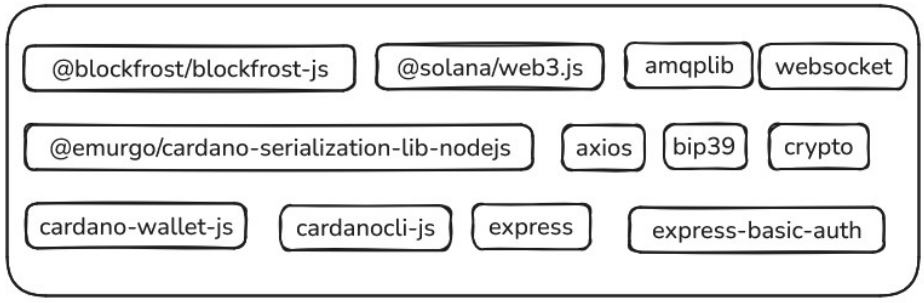


Figura 10. Librerías usadas en módulo log

Entre las librerías destacadas (Figura 10) se encuentran:

- **@blockfrost/blockfrost-js:** Permite interactuar con la API de Blockfrost, un servicio que proporciona una interfaz sencilla para acceder a datos de la cadena de bloques de Cardano.
- **@emurgo/cardano-serialization-lib-nodejs:** Proporciona herramientas para la serialización y deserialización de transacciones y datos específicos de la cadena de bloques de Cardano. Es esencial para construir y firmar transacciones, así como para manejar estructuras de datos de Cardano dentro del Módulo Log.
- **@solana/web3.js:** Librería para interactuar con la cadena de bloques de Solana. Permite al Módulo Log enviar registros a la cadena de bloques

de Solana, proporcionando soporte para múltiples cadenas de bloques, flexibilidad y alcance del sistema.

- **Amqplib**: Es una implementación del protocolo AMQP para Node.js. Se utiliza para conectarse al message broker, permitiendo que el Módulo Log reciba los registros filtrados de manera eficiente y confiable.
- **Axios:** Es un cliente HTTP basado en promesas para Node.js. Se utiliza para realizar solicitudes HTTP/HTTPS a servicios externos, como APIs de cadena de bloques o servidores de gestión de logs, facilitando la comunicación y transferencia de datos.
- **bip39:** Implementa el estándar BIP39 para la generación de mnemónicos y derivación de claves. Es útil para la gestión segura de claves y direcciones en criptomonedas, asegurando que las operaciones con la cadena de bloques se realicen de manera protegida.
- **cardano-wallet-js:** Proporciona una interfaz para interactuar con una cartera de Cardano, permitiendo gestionar direcciones, transacciones y fondos dentro de la cadena de bloques de Cardano desde el Módulo Log.
- **cardanocli-js:** Es una interfaz de JavaScript para el CLI de Cardano. Permite ejecutar comandos de Cardano directamente desde el código Node.js, facilitando operaciones avanzadas y una integración más profunda con la cadena de bloques de Cardano.
- **crypto:** Módulo de Node.js para operaciones criptográficas. Proporciona funcionalidades como generación de hashes, cifrado y descifrado esenciales para asegurar la integridad y confidencialidad de los datos procesados por el Módulo Log.
- **express:** Entorno de trabajo minimalista y flexible para aplicaciones web en Node.js. Es ampliamente utilizado para crear aplicaciones web y APIs debido a su simplicidad y robustez. Express facilita la gestión de solicitudes HTTP, el enrutamiento, el middleware y la integración de funcionalidades adicionales mediante una amplia variedad de paquetes disponibles en el ecosistema de Node.js.
- **express-basic-auth**: Middleware para Express que proporciona autenticación básica HTTP. Se emplea para proteger rutas específicas de la aplicación, asegurando que solo usuarios autorizados puedan acceder a ciertas funcionalidades o información sensible.
- **Websocket:** Permite establecer comunicaciones bidireccionales en tiempo real entre el servidor y clientes, lo cual es útil para transmitir registros o notificaciones de eventos de manera instantánea.

3.3.3.4 Blockchain

A grandes rasgos, una cadena de bloques es una tecnología de contabilidad distribuida (DLT). Se trata de una base de datos que puede ser compartida mediante intercambio directo entre pares y que permite almacenar información de forma inmutable y ordenada. Esto es, una vez que una transacción se ha producido, generalmente no puede alterarse ni eliminarse. Ello implica un alto nivel de integridad y transparencia en el manejo de datos.

En esta Tesis Doctoral se han seleccionado dos cadenas de bloques diferentes, una que permite incluir metadatos en la transacción y otra que no. Dada esta distinción, el módulo log interactuará con la cadena de bloques de una determinada manera. Por un lado, Cardano ofrece la inclusión de metadatos en las transacciones, lo que facilita el almacenamiento de información adicional directamente en la cadena de bloques. Por otro lado, Solana proporciona una alta velocidad de procesamiento y eficiencia en el manejo de transacciones, aunque no permite la inclusión de metadatos en las transacciones.

Uno de los elementos fundamentales es el algoritmo de consenso utilizado, el cual se encarga de validar la información que se agrega a la cadena de bloques entre todos los nodos que la componen, así como asegurar que todas las transacciones son correctas. Existen multitud de algoritmos de consenso como los dos relacionados con esta Tesis Doctoral (i.e., Proof of History y Proof of Stake).

Es importante señalar que, una vez que la transacción es enviada a la cadena de bloques, esta no es persistida inmediatamente. Previamente, la transacción será validada entre los nodos que componen la red y su información añadida a un bloque después de un número de transacciones establecido por la red.

3.3.3.5 Servidor de gestión de logs

Se trata de un servicio web desarrollado para recibir información del sistema DRACSC a través de una API Restful y almacenarla para su posterior explotación. El servidor es también responsable de comunicarse con la cadena de bloques para recuperar la marca de tiempo del bloque generado que registra la transacción de forma permanente (i.e., la certificación de un evento log). Por otro lado, este servidor

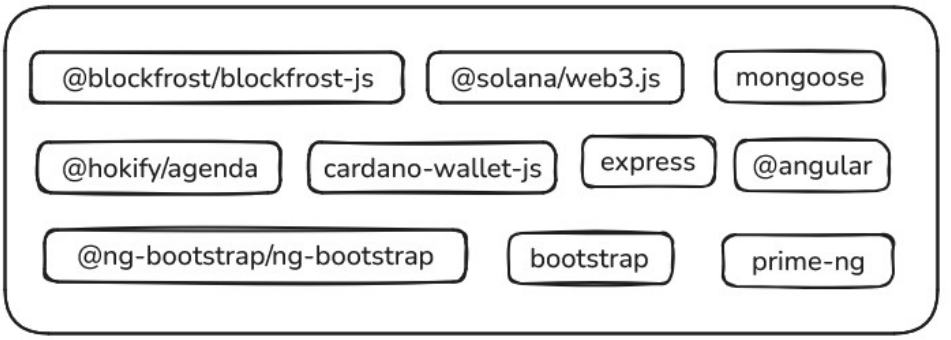


Figura 11. Librerías usadas en el servidor de gestión de logs

ofrece una interfaz intuitiva con la que acceder de forma sencilla a la información almacenada, la cual puede verse en la Figura 3.

Para el desarrollo del servidor, se utiliza la versión 18.21 Node.js, así como npm como gestor de dependencia. Entre las librerías utilizadas (Figura 11) se pueden destacar:

- **@blockfrost/blockfrost-js:** Permite interactuar con la API de Blockfrost, un servicio que proporciona una interfaz sencilla para acceder a datos de la cadena de bloques de Cardano.
- **@hokify/agenda:** Se usa para la programación y gestión de tareas en Node.js utilizando MongoDB para su almacenamiento. Se utiliza para programar tareas recurrentes o programadas, como la sincronización periódica con la cadena de bloques.
- **@solana/web3.js:** Usada para interactuar con la cadena de bloques de Solana, permitiendo acciones como el envío transacciones, consultar cuentas o gestionar claves, entre otras.
- **cardano-wallet-js:** Proporciona una interfaz para interactuar con la cartera de Cardano, permitiendo gestionar direcciones, transacciones y fondos dentro de la cadena de bloques de Cardano desde el servidor.
- **express:** Entorno de trabajo web minimalista y flexible para Node.js. Es el núcleo de la aplicación del servidor, manejando rutas, solicitudes HTTP, middleware y la lógica de negocio de la API RESTful. Express permite construir una API robusta y escalable, facilitando la integración con otros sistemas y servicios.
- **mongoose:** Librería de modelado de datos para MongoDB y Node.js. Proporciona una solución basada en esquemas para modelar los datos de

la aplicación, incluyendo validación, conversión de tipos (casting) y capas intermedias de procesamiento (middleware). Se utiliza para definir esquemas, validar datos y realizar operaciones CRUD (del inglés crear, leer, actualizar y borrar) en la base de datos, garantizando la integridad y consistencia de los datos almacenados.

En el desarrollo de la interfaz gráfica de usuario se pueden destacar las siguientes librerías:

- **@angular:** Incluye el núcleo del entorno de trabajo Angular y sus módulos esenciales, proporcionando las funcionalidades necesarias para construir aplicaciones web modernas basadas en componentes.
- **@ng-bootstrap/ng-bootstrap:** Integra componentes de Bootstrap en Angular, proporcionando una serie de componentes de interfaz de usuario (como modales, pestañas, alertas) adaptados para funcionar de manera nativa con Angular sin depender de jQuery.
- **Bootstrap:** Entorno de trabajo de CSS que facilita el diseño adaptable y estilizado de la aplicación, proporcionando estilos y componentes predefinidos que mejoran la apariencia y usabilidad.
- **PrimeNG:** Conjunto de componentes ricos de interfaz de usuario para Angular. Incluye una amplia variedad de elementos como tablas avanzadas, calendarios, gráficos, paneles y más, lo que acelera el desarrollo y mejora la funcionalidad de la aplicación.

Con respecto a la base de datos, se ha utilizado MongoDB, la cual es una base de datos NoSQL orientada a documentos que almacena la información en formato BSON (Binary JSON). Ello permite manejar estructuras de datos flexibles y jerárquicas, facilitando la adaptación a diferentes tipos de datos y la incorporación de nuevos campos sin afectar a los registros existentes, así como un desarrollo muy ágil.

Capítulo 4. Resultados y Discusión

Resultados y discusión. Artículo 1.

Debido a restricciones relativas a derechos de autor, el artículo “Portable Device for Easy Management and Automatic Recovery of Networking Systems”, ha sido retirado de la tesis. En sustitución del mismo ofrecemos la siguiente información: referencia bibliográfica, resumen y palabras claves.

- Morillo Reina, J. D., & Mateo Sanguino, T. de J. (2019). Portable Device for Easy Management and Automatic Recovery of Networking Systems. IEEE Latin America Transactions, 17(03), 401–408. https://doi.org/10.1109/tla.2019.8863310

Enlace al texto completo: https://doi.org/10.1109/tla.2019.8863310

RESUMEN:

Configure communication equipment is a critical task that typically requires handling advanced concepts aimed at managing and troubleshooting networking issues, thus demanding high analytical and problem-solving skills to ICT technicians. To reduce this gap, this paper presents a portable hardware and software solution devoted to unify the administration of operating systems, configuration files and network services in communication devices. This includes the automatic recovery of firmware under failure conditions concerning the loss of boot image, corruption of Flash memory and forgotten passwords, among others. The prototype, mainly tested on routers and switches from Cisco and MikroTik, can be also applied to other equipment such as APs, firewalls or embedded devices since it looks for universality and independence from devices and manufacturers. As advantage, this facilitates the easy maintenance and quick deployment of network infrastructures, thus saving cost and time to engineers. So the xperimentation carried out shows the significant advantage concerning the time taken to perform some automated functions rather than running manually. This stands for a powerful tool that, as a result, has been successfully patented and extended through an international cooperation treaty (PCT)

4.1 Artículo 1

Portable Device for Easy Management and Automatic Recovery of Networking Systems

Morillo Reina, J.D., Mateo Sanguino, T.J.

Publicado en:

Revista: IEEE Latin America Transactions
Editorial: IEEE
Editor-in-Chief: Ilse Cervantes
Referencia: Volume 17, NO.3 MARCH 2019, 8 pages
Año: 2019
ISSN: 1548-0992
DOI: 10.1109/TLA.2019.8863310

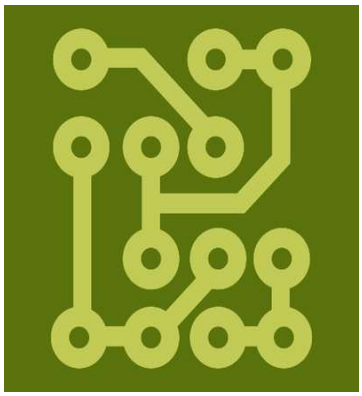
Categoría	**Posición/ Total**	**Cuartil**
Computer Science, Information Systems	145/156	Q4
Engineering, Electrical & Electronic	234/266	Q4
Factor de Impacto (2019)	0.782	

4.2 Artículo 2

Cloud-Based Automatic Configuration and Disaster Recovery of Communication Systems Applied in Engineering Training

Morillo Reina, J.D., Mateo Sanguino, T.J.

Publicado en:



Revista: Electronics
Editorial: MDPI
Editor-in-Chief: Flavio Canavero
Reference: Volume 13, Issue 21, Article ID 4203
Año: 2024
ISSN: 2079-9292
DOI: 10.3390/electronics13214203

Categoría	**Posición/ Total**	**Cuartil**
Computer Science, Information Systems	115/250	Q2
Engineering, Electrical & Electronic	157/353	Q2
Factor de Impacto (2023)	2.6	

electronics

MDPI

Article

Cloud-Based Automatic Configuration and Disaster Recovery of Communication Systems Applied in Engineering Training

J. D. Morillo Reina and T. J. Mateo Sanguino *

Department of Electronics Engineering, Computer Systems, and Automatics, University of Huelva, Avda. de las Artes S/N, 21007 Huelva, Spain; juandiego.morillo@alu.uhu.es

* Correspondence: tomas.mateo@diesia.uhu.es

Abstract: Network management and troubleshooting require not only a grasp of advanced concepts but also the development of analytical and problem-solving skills. To bridge this gap, this paper introduces a novel network administration system, DRACSC (Spanish acronym for device for automatic recovery and configuration of communication systems), designed for the automatic configuration and disaster recovery of communication equipment. This system transcends the limitations of current hardware and software solutions by combining their advantages, boasting portability, automated functions, and a cloud-based repository as its main features. The DRACSC system, undergoing a comprehensive large-scale evaluation involving diverse user groups across multiple institutions, was tested with 89 users, including students and teachers at educational centers and ICT (Information and Communication Technology) professionals. The benefits of the system were evaluated through a training program based on simulated real-world ICT environments, focusing on both quantitative results on the reduction in time to complete user tasks, as well as qualitative results on the interface and usability of the system. Statistical analysis, including Welch's t-test on opinion surveys, indicated a significant increase in knowledge and understanding, demonstrating the system's potential to enhance education and practice. Moreover, the evaluation shed light on the user experience, with positive impacts observed for learning and teaching implications. As a result, the study has verified that the system has the potential to significantly influence network management practices, enhancing both learning and professional application through improved efficiency and usability.

Keywords: automatic recovery; Cisco; communication equipment; IOS; Mikrotik; Raspberry Pi; networking; engineering education

Citation: Morillo Reina, J.D.; Mateo Sanguino, T.J. Cloud-Based Automatic Configuration and Disaster Recovery of Communication Systems Applied in Engineering Training. *Electronics* **2024**, *13*, 4203. https://doi.org/10.3390/electronics13214203

Academic Editors: Erik Kučera, Oto Haffner and Peter Drahoš

Received: 15 September 2024
Revised: 22 October 2024
Accepted: 24 October 2024
Published: 26 October 2024

1. Introduction

Communication devices used to provide connectivity to data networks—such as switches or routers—that contained proprietary machine instruction programs (i.e., firmware) created to manage the physical, link, and network layers of the OSI reference model [1]. The functions—depending on the device where they are performed—include the configuration of communication ports, routing tables, virtual private networks (VPNs), access control lists (ACLs), network address translations (NATs), or dynamic host configuration protocols (DHCPs), among others [2]. The information related to these functions is usually stored in files located in non-volatile memory that, together with the firmware located on the computer's internal storage unit, forms the operating system (OS) and profile loaded during the computer's boot process. Occasionally, the boot system or the configuration files are compromised by an equipment malfunction. In these cases, most manufacturers provide manual recovery procedures that require a deep understanding of the different line commands and specific connections to be used [3].

To address this problem, we propose a hardware/software system called DRACSC (i.e., Spanish acronym for device for automatic recovery and configuration of communication systems). DRACSC was introduced in a network laboratory at the University

of Huelva, Spain, as part of a larger educational project widely described in [4–10]. At present, the device's hardware has been upgraded with up-to-date low-cost options and a housing to improve its ergonomics and aesthetics. Also, the software has been enhanced to include new functions and optimize the user experience. As the main advance, the device has been complemented with a web repository of MACRO functions centralized in the cloud. To validate the proposed solution, teachers and students from vocational education and training institutions were actively involved in the evaluation of DRACSC over an academic year. The selection of vocational participants was based on its dual pedagogical approach, which integrates theoretical and practical training to enhance the development of professional skills. For comparative purposes, the system has also been tested with professionals from the ICT sector, thus obtaining a double perspective of both the educational and professional fields.

In summary, the contributions pursued by this work are the following: (i) to present a novel solution for the centralized management of communication equipment; (ii) to examine its applicability and advantages for managing not only equipment configurations but also network services; and (iii) to provide network engineers with a time- and effort-saving tool for managing network infrastructures. To this end, a complete, portable, and universal hardware/software system based on Raspberry Pi 4 Model B and web applications was developed. This solution was patented [11] and extended through an international patent cooperation treaty [12].

Research Hypotheses and Objectives

The main goal of this study is to analyze the impact of a new system devoted to the instruction of IT technicians. We hypothesize that participants can improve their user experience on repetitive and time-consuming tasks through a system that has been previously proven to facilitate IT learning [5]. Additionally, we aim to explore the feasibility of this system in a professional IT environment. This rationale is based on the fact that a significant amount of time is spent on recurring tasks in practical training for both students and teachers, as well as IT professionals. To this end, this manuscript aims to provide empirical evidence through an exhaustive analysis based on user perceptions and their performance during the training and development of IT technicians.

The manuscript is structured as follows. Section 2 presents the work related to software and hardware dedicated to managing communication equipment. Section 3 describes the complete system architecture of DRACSC. Section 4 presents the experimentation carried out. Section 5 analyzes the solution's impact on the educational and professional environment. Finally, the article provides the results achieved and discusses future work.

2. Related Work

The management and recovery of communication equipment have been solved by solutions classified into software applications and hardware devices [13,14]. On the one hand, software applications usually contain a graphical user interface (GUI) to manage communication equipment remotely. The GUI reduces administration time and makes it easier for the user to perform basic tasks without knowing the entire repertoire of commands. While these applications are widely used, they have the disadvantage that users often perceive limited control due to the reduced configuration possibilities compared to a command-line-based interface. On the other hand, hardware devices—also known as console terminals or servers—can communicate with switches, routers, and other devices via the console port. Although they have more considerable configuration potential, such devices have the disadvantage of being costly.

In addition to combining the facilities of related hardware and software solutions, the proposed system brings two main contributions, as described throughout the paper: portability and a template-based structure for creating high-level MACRO functions.

2.1. Network Device Management Software

Solutions found in the state of the art include Cisco Web Browser User Interface, a web interface supplied with the company's own switches and routers that requires additional configuration to work with them [2]; Linksys Smart Wi-Fi, a set of tools with functions similar to Cisco Connect Express and Cisco Connect Cloud to manage home networks consisting of access points and Wi-Fi routers remotely and easily [15]; Colibri NetManager, formerly Teldat's TeldaGES, a router management platform that brings together auditing, network vision, device access, configuration history retrieval, and firmware through the cloud [16]; ManageEngine's Network Configuration Manager, formerly DeviceExpert, a powerful solution that centralizes the administration, configuration, task automation, and monitoring of network elements (e.g., switches, routers, firewalls, etc.) from a web interface [17]; and SolarWinds' Network Configuration Manager, a benchmark in this field that combines the monitoring and configuration of network elements [18]. WhatsUp Gold by Progress is a tool that facilitates the automation of network device management, with functionalities such as auditing, configuration backup, and compliance with regulations like PCI, SOX, and HIPAA [19].

In order to compare the capabilities and features of the above solutions concerning DRACSC, Table 1 is presented. While the Cisco (https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/fundamentals/configuration/xe-16/fundamentals-xe-16-book/fundamentals-xe-16-book_chapter_01000.html (23 October 2024)) or Teldat (https://www.teldat.com/solutions/advanced-networking/net-cloud-administration-network-infrastructure-network-management/ (23 October 2024)) software are only compatible with their brands, ManageEngine, SolarWinds, or WhatsUp Gold have been designed to be compatible with as many devices as possible (e.g., Cisco, Juniper, HP, Dell, Brocade, F5, Aruba, or Ruckus). This compatibility is an advantage over others. In terms of functions, all systems follow what can be called a "logical path" to simplify the administration of network elements and, therefore, share similar features. In this respect, the table establishes the medium level when the device brings functions for automating tasks, and the high level when it also includes network monitoring or backups.

Table 1. Characteristics of network device management software solutions.

Name	Parallel Device Management	Manufacturers Admitted	Level of Functionality	Control via Console	Control via Telnet/SSH	System Recovery Function	Cloud Repository	Cost
CISCO Web Browser User Interface	One	Cisco	Medium	No	No	No	No	Free
Linksys Smart Wi-Fi	Several	Cisco	Medium	No	No	No	✔	Free
Colibri NetManager	Many	Teldat	High	No	✔	✔	✔	-
Solarwinds Network Configuration Manager	Many	Many	High	No	✔	No	✔	High
Network Configuration Manager	Many	Many	High	No	✔	No	No	High
WhatsUp Gold	Many	Many	High	No	No	✔	✔	High
DRACSC	Several	Several	Medium	✔	✔	✔	✔	N/A

Regarding cost, it is worth mentioning that ManageEngine licenses are granted per managed device (e.g., basic package of 25 devices and two users costs EUR 1134 plus EUR

227 of support per year). Similarly, the basic purchase of SolarWinds starts from EUR 1377 per user. Besides the lower cost and other advantages, only DRACSC can manage network devices via the console port.

2.2. Hardware for Network Device Management

Existing state-of-the-art hardware solutions that function as terminals or console servers, such as Aten SN018C0, can be purchased for EUR 2640 for the 8-port model [20]. Perle's IOLAN SCS48 DAC price is around EUR 3917 and allows for the administration of different computers using the console, Telnet, and SSH connections [21]. Dominion® SX from Raritan is similar to the previous one and provides access, monitoring, and control through the serial port [22]. Opengear IM7200 is a device focused on console servers that offers a more significant number of functions such as firewall, DHCP server, VPN, etc., with Telnet/SSH management over serial port [23]. Some solutions that try to emulate this hardware have been created to reduce costs. This is the case for Raspisco, a console server based on Raspberry Pi, 3G modem, SerialToUSB adapter, and console cable for the administration of devices via USB hub and adapters [24]. Also, but focused on the educational environment, Raspberry Pi is used as a low-cost device that emulates a console server by software [25]. The DIGI Connect EZ 8 is a versatile console that supports RS-232/422/485 communication, making it suitable for a variety of industrial applications. The device is easy to configure through an intuitive web interface and provides robust security features such as SSL/TLS encryption and SSH for secure data transmission [26].

Table 2 compares the capabilities and features of the previous solutions concerning the DRACSC-based one. It is worth mentioning that Dominion® SX II can interface with different manufacturers such as HP, Dell, Cisco, or IBM. Noteworthy are the portability and disaster recovery capabilities of the DRACSC solution beyond simple configuration restore. Another notable feature of DRACSC is the use of space to store data from switches and routers whose file exchange protocol is TFTP (RFC1350) and FTP (RFC1123). In addition, DRACSC allows for the extension of online functionalities through a cloud repository, thus increasing the number of actions to be executed on managed devices without updating the firmware.

Table 2. Characteristics of network device management hardware solutions.

Name	Parallel Device Management	Manufacturers Admitted	Portability	Level of Functionality	Control via Console	Control via Telnet/SSH	System Recovery Function	Cloud Repository	Cost
Aten SN0116CO	Several	Several	No	Medium	✔	✔	No	No	High
IOLAN SCS48 DAC	Several	Several	No	Medium	✔	✔	No	No	High
Dominion® SX II	Several	Several	No	High	✔	No	No	No	High
Opengear IM7200	Several	Several	No	High	✔	✔	No	No	High
Raspisco	One	Several	✔	Low	✔	No	No	No	Free
Raspberry Pi-based console server	Several	Several	No	Low	✔	No	No	No	Free
Digi Connect EZ® 8	Several	Several	No	High	✔	✔	No	No	High
DRACSC	Several	Several	✔	Medium	✔	✔	✔	✔	Low

3. Materials and Methods

The system presented in this paper comprises two parts (Figure 1). On the one hand, DRACSC is the primary device that interacts with the managed equipment (i.e., switches or routers to configure or recover). On the other hand, the cloud repository hosts the web platform where several DRACSCs can work with MACRO functions in a centralized way. These functions are the interface between DRACSC and the managed equipment that will be explained in this section.

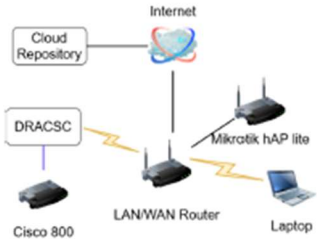


Figure 1. Interaction between DRACSC, cloud repository, and managed devices.

3.1. DRACSC System

DRACSC is a device that interacts with the cloud repository and uses MACRO functions to act on the managed network equipment. The usual operating scenario of DRACSC is within a LAN/WLAN using TCP/IP, allowing it to work in most WAN scenarios, regardless of network size or topology. However, since it has a web application server to which the user can connect via a browser, it could be extended to a WAN via NAT. The system also provides the ability to interact with diverse managed device profiles, thereby showcasing its scalability and adaptability across different manufacturers (e.g., Cisco, Mikrotik, Raspberry Pi), operating systems (e.g., IOS, RouterOS, Raspbian), and models (e.g., routers, switches, computers), extendable to others. There are two ways of working with DRACSC depending on the state of the equipment to be managed (i.e., with or without Telnet/SSH configuration). In the first case, both DRACSC and the equipment to be managed are connected to the same LAN/WLAN. This setup eliminates the need to carry the DRACSC device to extend the autonomy of the internal battery. In the second case, the switch or router cannot communicate with the DRACSC device over the TCP/IP network. Therefore, there is no other way to work with it than to connect directly through the console port. For specific actions—such as loading an IOS image when the equipment is in ROMMON mode—it would be necessary to configure an additional network.

3.1.1. Hardware and Software Implementation

To achieve a prototype based on low-cost components widely available on the market, the DRACSC system has been implemented with Raspberry Pi, a device commonly used in other research projects with acceptable performance in similar applications [27–30]. Another factor influencing its choice was the broad community behind it, which has a significant impact on both development time and peripheral support. Specifically, the hardware used was Raspberry Pi 4 Model B with a 1.5GHz Broadcom BCM2711 Quadcore Cortex-A72 (ARM v8) 64-bit SoC and 4GB of LPDDR4-3200 SDRAM. The functionality of the DRACSC device has been extended with a 5-inch resistive TFT display (800 × 480 pixels); a designed and 3D-printed housing with a tough hard plastic filament (i.e., ABS); a 32GB Kingston micro SD memory (class 10, 45 MB/s) whose function is to store images of the OS of the managed devices; a serial to RJ45 console cable to manage the devices through the console port; a USBtoRS232 adapter; and a 4000 mAh Contact LXBA4000U battery model that allows for the autonomy of ~4 h with a worst-case consumption of ~1000 mAh. In total, the cost of the hardware system was EUR 132 (Figure 2).

Figure 2. DRACSC device.

Regarding its software, Raspberry Pi OS 5.10 was selected as a version of Debian optimized for Raspberry. The system applications have been developed using Ruby 2.3.2, an interpreted programming language distributed under a free software license. Accordingly, we used Bundler [31] to provide a consistent environment for Ruby projects that allows for tracking and installing the libraries needed—called gems—to deploy the system's TFTP and FTP servers (i.e., 'atftpd' and 'ftpd').

The web server selected for DRACSC is Phusion Passenger, an open-source service compatible with our system developed through the 'Ruby on Rails' (RoR) environment. This is a Nginx-based server that can handle HTTP requests, processes, and resources with less load on the system than other web servers (e.g., GlassFish). It also allows for monitoring and diagnosing problems [32]. The client side of this web application has been programmed with HTML, JavaScript, and CSS using the Bootstrap library. This allows for the interface view to be dynamically adapted according to the user device used, thus letting us implement responsive interface versions for desktop and mobile screens.

Moreover, the SQLite database management system (DBMS) was used because it supports a relational data structure that consumes few resources and controls the data atomicity, consistency, isolation, and durability (ACID) in a library written in C language with only ~500 KB. As an additional advantage, SQLite is linked to the main program and becomes an integral part of it, thus reducing access latency in contrast to other databases such as PostgreSQL, MySQL, or MariaDB [33].

Regarding the connection terminals, the Ser2net service has been used to encapsulate the console connection through the serial port in Telnet.

3.1.2. Local Web Application

The web environment has been developed following the model–view–controller (MVC) paradigm. This separates the application's data, business logic, and persistence mechanisms from the interaction and representation of the data in the interface. The managed equipment stands for the view, whilst DRACSC contains the web service (i.e., controller). Based on this scheme, the web client displays the user interface, and its function is to represent the data exchanged between DRACSC and the managed equipment. On the one hand, DRACSC sends information to the user in a specific format that is practical for use. On the other hand, it is ultimately responsible for initiating, maintaining, and terminating the session with the managed equipment. Therefore, the role of the controller is to receive the commands sent from the view to the equipment and update the session information from the equipment to the view.

The communication can occur via different networking means depending on whether it is conducted via console, Telnet, or SSH. However, functionally, it produces the same result for the user. Thus, when the user requests, the controller creates a console, Telnet, or SSH client through a thread that initializes a procedure on the managed equipment. Finally, the model (i.e., the system layer responsible for implementing the business logic) is

made through a relational object mapping system called 'Active Directory'. Its advantages are mainly independence from the database, faster development, and increased security. For this, query strings were defined through annotations, which cannot be altered by the application at runtime and prevent security problems.

The web interface has been designed to allow two different user roles, administrators and basic users, with restricted functions. These are available after accessing a login page with a username and password. Then, a menu of options shows the information and actions to be performed on the managed equipment, the editing of own actions (i.e., MACRO), the console configuration and interfaces of the DRACSC system, and the administration of user roles, as well as general help regarding the web application. These functions correspond to the options 'login', 'devices', 'system', 'users', and 'help', respectively.

An example set of MACRO functions preconfigured in DRACSC is displayed on the TFT screen (Figure 2). This view is also generated in the web application and established by running the browser in «kiosk» mode, which shows the entire screen without a configuration or address bar.

3.2. MACRO Function

In the configuration of communication equipment and computer systems, it is prevalent to use scripts to automate—to a greater or lesser extent—a sequence of commands or instructions. Scripts also avoid errors in the execution of processes. Typically, this task is accomplished by the following methods: (i) A manual method in which an original script is given; instructions that need some changes are visually identified and then modified to achieve the desired operation. This method generates a customized script with the advantage of being able to deploy the same configuration on different devices. (ii) A programmatic method that also starts with the original script and—through a scripting language—modifies the instructions using data and control structures. Through the inputs provided by the user, either as parameters when executing the script or interactively through a command interpreter, it is possible to adapt the original script of managed equipment to a series of specific problems. The programmatic method is a powerful and versatile feature that suits almost any situation. As a disadvantage, the effort to achieve said versatility usually increases as the need to adapt to different problems grows. On the other hand, if a specific solution needs to be stored, it must be programmed in such a way that its code can be stored in an appropriate file.

As a consensus solution to the previous methods of script reuse, this paper proposes using the MACRO function concept. Its main objective is the creation of high-level functions that allow users a greater degree of abstraction in such a way that they can focus on the functionality of the managed equipment instead of having to memorize the syntax of the commands. MACRO functions are designed in DRACSC through a graphical interface for that reason; no programming skills are required to create generalized scripts. They are supported by an XML structure that includes metadata used to work with them, the script, and the optional parameters of the managed equipment (Figure 3). The MACRO functions can be executed on a device through the touch screen (Figure 2) or through the DRACSC web interface (Figure 4). The main advantage of using MACRO functions is to facilitate the quick configuration of the managed equipment without requiring knowledge of specific commands. This methodology adapts to the experience and way of working of each user, allowing professionals a wider range of the configuration and optimization of tasks.

```
<Macro>
<!-- Base64-encoded comment -->
  <name> MACRO Name </name>
  <version> MACRO version </version>
  <via> Method of execution for the type of preconfigured </via>
  <icon> Icon to represent the preconfigured MACRO </icon>
  <!-- The "device" tag defines the device, implying whether or not it can be applied to one device or another --!>
  <device>
    <manufacturer> OS/Manufacturer </manufacturer>
    <serie> Device series </serie>
    <model> Device model </model>
    <osversion> OS version </osversion>
  </device>
  <!-- Within the "inputs" label are the different fields that can be variable using a MACRO --!>
  <inputs>
    <input> </input>
    ...
    <input> </input>
  </inputs>
  <!-- The "commands" tag includes the commands issued to the device for configuration --!>
  <commands>
    <command> command>
    ...
    <command> </command>
  </commands>
</Macro>
```

Figure 3. XML structure of a MACRO function.

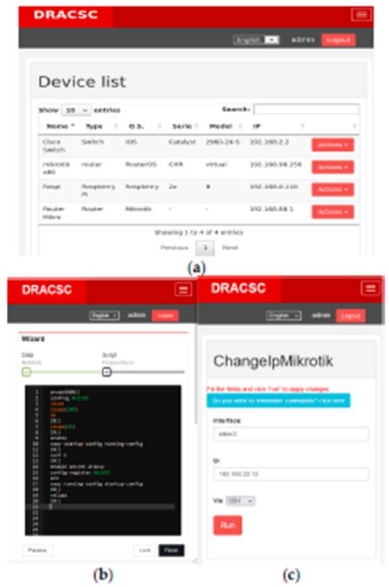


Figure 4. View of the web interface of the DRACSC device: (**a**) list of managed equipment, (**b**) execution of a MACRO function, and (**c**) editing of preset MACRO function.

Another problem that the MACRO functions solve is related to the configuration of variables, the number of which is related to the size of the script. In most configurations, the number of variable parameters is associated with the size of the script. Therefore, the advantage of using MACRO functions is the greater the number, the larger the script, as can be seen in Table 3. To improve the user experience, a command interpreter is included in DRACSC to reduce interaction when manual steps are required in certain specific configurations (Table 4). For example, the Cisco ROM monitor (ROMMON) is a state of operation in which a switch or router is booted directly to perform various actions on the IOS image. This state is required when the IOS image has been deleted or damaged, and the device must start from the ROM, the content of the NVRAM must be ignored to reset the password in 'enable' mode, and the console speed must be set or the diagnostic messages must be enabled, among others [34]. To access the ROMMON mode of a router, a 'control + break' command must be sent through the console or Telnet session during the first 60 s of boot-up. In the case of a switch, it is required to manually turn off the equipment and press the MODE button for 5 s. After restarting, it is essential to know the value of the configuration register, which is responsible for setting the device's boot. The hexadecimal value set by default at the factory is usually 0x2102 for routers. This means that an image from the Flash memory must be loaded, breaks must be ignored (except for the first 60 s), a 9600 baud rate must be set for the console, and the IOS must be loaded from ROM if the boot fails. Password recovery functions require between 9 and 12 commands plus a user parameter, while recovering a device in ROMMON mode requires an IOS image backed up in an external location. As can be seen, this task is certainly complex and, therefore, is reserved for truly experienced technicians. As an example, the password recovery procedure for a Cisco router using a MACRO function with DRACSC can be seen in Figure 4b.

Table 3. Statements and parameters for configuring network services on Cisco devices via command line.

Service Configured in the Equipment	Commands	ASCII Characters	Variables to Define
Backup to FTP server	2	37	2
Restore backup from FTP server	2	44	2
IP interface	4	50	3
Telnet interface	4	58	3
DNS	5	93	2
DHCP	6	119	6
SSH	10	235	7
Restore ROMMON	8	142	1
Reset password (router)	9	129	1
Reset password (switch)	12	209	1

Table 4. Reserved words in the MACRO function interpreter together with their description.

Reserved Word	Description
Break9600()	To produce a stop in the start sequence, the break signal is sent. When using a USB converter for the console cable, it is preferable to go down to 1200 baud and send spaces to simulate the signal.
CR()	Carriage return.
IP()	Prints the IP address assigned to the device.
userTelnet()	Prints the user for "telnet" assigned to the device.
userSSH()	Prints the user for "SSH" assigned to the device.
passwordEnable()	Prints out the "enable" password assigned to the device.
passwordTelnet()	Prints out the "telnet" password assigned to the device.
passwordSSH()	Prints out the password for "SSH" assigned to the device.

Two modalities have been created to achieve greater flexibility when managing equipment through MACRO functions. These are called «standard MACRO» and «preconfigured MACRO», as detailed below.

3.2.1. Standard MACRO

This function is the closest case to the programmatic method, for which flexibility is more important than immediacy. When creating a MACRO function from the graphical interface, the user must first complete the metadata fields that will be used later for filtering purposes. Afterwards, the variant elements in the code will be identified and selected. In some instances—at the user's convenience—modifications to the initial script could be made from the graphical editor itself. This results in each line of code being entered as a command and each variable as an entry in a field structure, as shown in Figure 4c.

3.2.2. Preconfigured MACRO

This modality is used when immediacy prevails over flexibility. This type of function differs from the «standard MACRO» in that there are no inputs. In other words, the commands are not modified when the script is launched on the managed device. This can generate various MACRO functions that are very similar, which is why the "icon" tag is included in the metadata to help identify them using a set icon.

3.3. *Cloud Repository*

This service is conceived as a complete tool developed to facilitate equipment management and share MACRO functions among users from a centralized place on the Internet. The web repository provides an intuitive way of working where it is possible to comment on functions posted by other users, rate them through a scoring system, download functions as XML files, or communicate with different DRACSC devices through a REST API (i.e., an interface for communicating applications via HTTP protocol as between different DRACSC devices and the cloud repository).

3.3.1. Software Implementation

The repository was developed using Laravel, an open-source framework for developing web applications and services with PHP 7.4.0. Like Ruby, it is an interpreted language distributed under a free software license. The repository uses a dependency management system called Composer, which facilitates the administration of libraries and dependencies like 'passport' for authentication, 'purifier' for input sanitization, and 'eloquent' as object-relational mapping. The web repository client was programmed with HTML, JavaScript, and CSS using 'bootstrap' to allow for an adaptive design, as well as different libraries such as 'highlight.js' to mark the code syntax in different programming languages, 'sweetalert2' for modals, or 'Ace' to provide a code editor for the browser, among others.

It also has a REST API. To communicate, it is only necessary to know the URL of the repository and the access key (i.e., API-KEY). The repository was hosted in Hosting Cloud Linux. Specifically, a Debian distribution with scalable performance, 2.5 GB of RAM, SSL certificate, DDoS protection, and geo-redundancy—to avoid availability problems—was used. The DBMS used for the repository was MariaDB, a GPL-licensed MySQL derivative compatible with most cloud hosting sites. It was also necessary to have a web server compatible with PHP, such as Apache or Nginx.

3.3.2. Online Web Client

The cloud repository also follows the MVC paradigm. Figure 5a shows an example of the latest MACRO functions uploaded by users with the rating system in the form of stars, number of views, and date of upload. When using the search tool at the top, the filter can order MACRO functions by OS, the creation date, score, and visualizations. By clicking on 'view details' for a MACRO function, the user can see the script that it contains (e.g., inputs marked with different colors), download an XML file, and view comments. A user with

an active session could make comments and evaluations, as well as create new MACRO functions and edit and import them from an XML file.

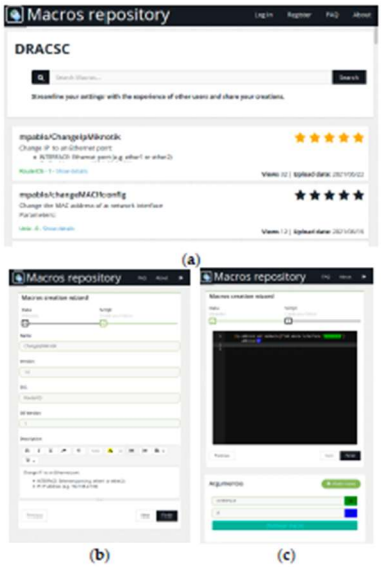


Figure 5. View of the repository web interface: (a) list of MACRO functions, (b) creation of metadata with "standard MACRO", and (c) creation of script with "standard MACRO".

4. Experimentation

This section aims to show the advantages of the DRACSC system in diverse work scenarios through performance evaluation and user assessment.

4.1. DRACSC Performance Test

On the one hand, a case study was conducted using a Cisco 800 router to evaluate the performance of the DRACSC system, comparing the time required to manage the device through both manual and automated procedures. This test involved some basic tasks such as configuring IP interfaces or network services (i.e., DNS, Telnet, and DHCP), and more advanced configuration procedures such as creating a backup, restoring the Flash memory on a remote server, or recovering the administration password. To this end, the topology consisted of a switch using a direct connection via console terminal and a remote connection using Telnet (see Section 5.1 for results on performance metrics).

4.2. Training and Experimentation with Use Cases

On the other hand, several validation tests were carried out with 53 students during the same academic year (18.54 $\pm$ 2.8 years old, 96.22% male and 3.78% female, 67.92% in Vocational Education and Training -VET, and 32.08% in a higher level of VET); 6 teachers (43. 83 $\pm$ 5.15 years old, 100% male); and 30 ICT professionals (35.1 $\pm$ 6.30 years old, 93.33% male and 6.67% female, 20% with VET studies, 53.33% with university studies, and 26.67% with post-university studies). The size and diversity of this sample not only allows for generalizing the findings, but also satisfies the criteria for sound statistical analysis, which is commonly conducted in social studies research [35]. According to the Central Limit Theorem, the sampling distribution of the mean approaches normality when the sample

size exceeds 30. This allows for applying parametric statistical tests under the assumption of normality with confidence.

The network model used in the validation tests consisted of a D-Link DSR1000N router connected to the Internet via WAN and to the DRACSC via WLAN (Figure 1). This testing bench was connected to the managed equipment on which the users had to carry out the experimentation (i.e., Cisco 800 router and Mikrotik hAP lite router). All study groups followed the same methodology during the testing, as indicated in the procedure described below.

The first step involved introducing the DRACSC device, the cloud repository, and the concept of MACRO function to users through a brief presentation. For each of them, examples of different applicable use cases were described. The training time took 15 min. Afterwards, four tasks to be carried out in the experimentation were defined and a brief explanation was given about the objective of each one of them.

4.2.1. Task 1: Interactive Connection

This activity consisted of learning how to initiate an interactive connection from DRACSC's console terminal to the managed equipment. To do this, it was explained how to access the web server hosted on the DRACSC device via WLAN. Then, the user connected to the Cisco 800 router through DRACSC's console port, logged in, and established contact. The activity ended when the user found that the password of the managed equipment was unknown to them, whose recovery was part of the following task. The complete sequence diagram of Task 1 is shown in Figure 6 (red lines). The preliminary explanation took 5 min, whilst the time to perform the task typically required 2 min.

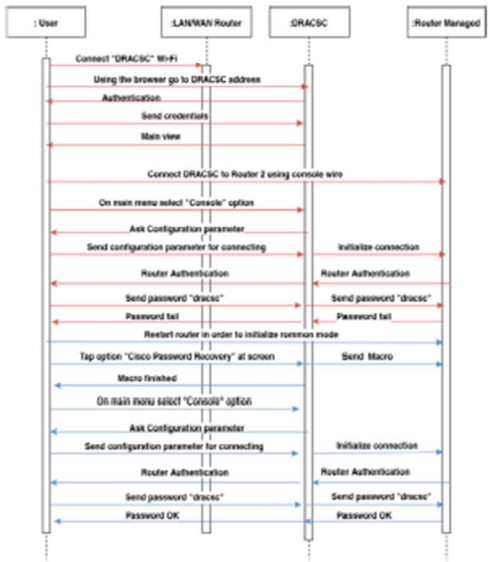


Figure 6. Sequence diagram of Task 1 (red) and Task 2 (blue).

4.2.2. Task 2: Using Preconfigured MACRO

This activity was a continuation of the previous one and consisted of resetting the password to regain access to the Cisco 800 router. The goal of the task was to learn how to execute a «preconfigured MACRO» function from the DRACSC touch screen. Specifically,

the steps involved selecting the "Recover Cisco Router Password" function on the DRACSC, rebooting the managed equipment to enter ROMMON mode, and launching the MACRO function. Once the password was restored, Task 1 had to be repeated to verify the new password set. On average, the explanation took 8 min, while running and verifying the managed equipment took 4 min. The complete sequence of steps is shown in Figure 6 (blue lines).

4.2.3. Task 3: Creation of Standard MACRO

This activity involved learning how to create a «standard MACRO» function from the web repository to change the network configuration of an interface in a managed device (i.e., Mikrotik). To do this, users pointed to the repository's web domain, used access credentials, and navigated to the list of MACRO functions (Figure 5a). They clicked on the "import" option in the horizontal menu and selected an already provided MACRO function to speed up the task. Then, the MACRO function creation/import wizard started, where users defined the metadata (Figure 5b), command set (Figure 5c), and its variables. The explanation took 5 min, whilst the execution of the task lasted 2 min.

4.2.4. Task 4: Using a Cloud Repository

The objective of this activity was for users to deepen the communication between the DRACSC device and the repository. For this, the «standard MACRO» function created in the previous task was used and executed through the web service of the DRACSC.

The steps consisted of accessing the DRACSC web service, opening the 'MACRO' menu, selecting the 'import' option, accessing the list of MACRO functions available in the remote repository, and downloading it to the DRACSC. Once imported, users went to the 'devices' section (Figure 4a) and selected 'execute MACRO' in the submenu. This action started a wizard to set the parameters of the MACRO function of the device and execute it via SSH (Figure 4b). For this, the managed device (i.e., Mikrotik) had to be previously connected to the network with the SSH server operational. This activity required 5 min of explanation and 3 min of execution. The complete steps are shown in the diagram in Figure 7.

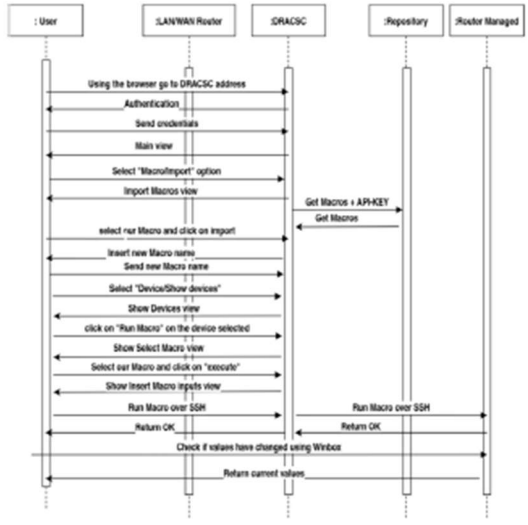


Figure 7. Sequence diagram of Task 4.

5. Results

This section contains an analysis to obtain actual performance metrics in terms of time and the number of keystrokes required to achieve the expected result, as well as a comprehensive analysis of user feedback on the DRACSC system, grouped into four main categories related to its design and evaluation, particularly in the field of technology and education. To this end, a survey among students, teachers, and professionals was conducted using a 5-level Likert scale. Moreover, the responses were analyzed using a two-sample Welch's t-test. The qualitative and quantitative validation of the results indicated strong correlations in perceptions among the different user groups and highlighted consistent trends across various institutes.

5.1. Performance Metrics

This study was completed in a real scenario, where a technician had the network equipment powered on and connected. To this end, a Cisco 800 router was used to evaluate the performance of the DRACSC system in relation to the tasks described in Table 3. Figure 8 highlights the significant reduction in keystrokes across all tasks when using DRACSC, including both basic and more advanced operations conducted on the network equipment. The most notable difference was seen in the "Reset Password" task, which normally requires entering 12 commands and 209 ASCII characters manually versus one variable to be defined in DRACSC. Similarly, Figure 9 shows that DRACSC substantially reduces the time needed to complete these tasks. These results demonstrate the clear efficiency gains of DRACSC in minimizing both user input and time performance.

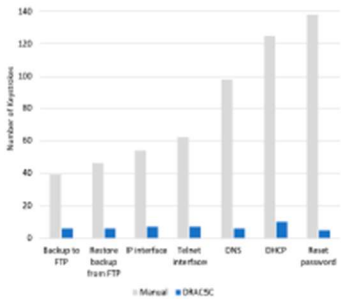


Figure 8. Keystroke comparison between manual and DRACSC settings.

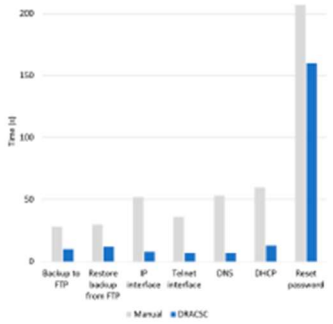


Figure 9. Time comparison between setting up manually vs DRACSC.

5.2. Opinion Survey

Table 5 shows the survey designed to evaluate the user opinions about the DRACSC system. The survey addresses different educational aspects divided into four thematic areas: "knowledge/learning" to identify the users' level of knowledge in the ICT area and the usefulness of the system for teaching (Q1–Q5); "interest/motivation" in order to know the incentive the system has for learning (Q6–Q10); "usability/practicality" to know the performance on the work carried out with the system (Q11–Q16); and "results/feasibility" to measure the general opinion of the users about the system (Q17–Q22).

Table 5. DRACSC evaluation questionnaire.

Question	Knowledge/Learning	Students	Teachers	Professionals
Q1	Level of prior knowledge on data networks	3.20 ± 1.03	3.66 ± 0.52	3.77 ± 0.69
Q2	The system allows them to work with already known theoretical concepts	3.69 ± 0.84	3.83 ± 0.41	4.07 ± 0.48
Q3	The system ensures that new theoretical knowledge is acquired	4.07 ± 0.71	4.00 ± 0.63	4.11 ± 0.89
Q4	Theoretical concepts are learnt by studying and the use of the system makes learning difficult	2.84 ± 1.17	2.00 ± 1.73	1.76 ± 1.09
Q5	My confidence level in working with data networks improved after using this tool	3.55 ± 0.82	3.83 ± 0.98	4.11 ± 0.70
	Interest/Motivation			
Q6	The use of the tool promotes motivation and interest in networking	4.31 ± 0.75	4.50 ± 0.55	4.74 ± 0.44
Q7	Creating my own MACRO solutions influences motivation and interest	4.04 ± 0.96	4.50 ± 0.55	4.70 ± 0.54
Q8	The use of a web repository promotes motivation and interest in the tool	3.98 ± 0.79	4.67 ± 0.52	4.74 ± 0.52
Q9	The independence of the solution from device manufacturers promotes motivation and interest in the tool	3.85 ± 1.02	4.67 ± 0.82	4.52 ± 0.64
Q10	I will create a personal project in the future using low-cost components	3.48 ± 1.20	3.12 ± 1.60	3.07 ± 1.30
	Usability/Practicality			
Q11	The information provided for the use of the tool is sufficient	3.89 ± 0.72	3.67 ± 1.21	4.26 ± 0.81
Q12	The level of interactivity of the system is appropriate	3.98 ± 0.92	4.33 ± 0.52	4.67 ± 0.48
Q13	The appearance of the system GUI is attractive	3.81 ± 0.96	4.33 ± 0.52	4.23 ± 0.76
Q14	The tool's menus have a clear and intuitive structure	4.17 ± 0.69	4.33 ± 0.52	4.33 ± 0.48
Q15	The tool has not presented any technical problems during the development of the use cases	3.52 ± 1.11	3.20 ± 1.33	4.07 ± 0.48
Q16	Portability is an important feature	4.39 ± 0.86	5.00 ± 0.00	4.89 ± 0.32
	Results/Feasibility			
Q17	The functionalities enabled by the system are useful	4.30 ± 0.86	4.83 ± 0.41	4.89 ± 0.42
Q18	The MACRO concept is useful for the configuration of devices	4.31 ± 0.84	4.83 ± 0.41	4.89 ± 0.32
Q19	The system saves time and effort in the tasks performed	4.33 ± 0.73	5.00 ± 0.00	5.00 ± 0.00
Q20	The tool is feasible for implementation in the educational context	4.09 ± 0.85	3.83 ± 0.75	4.44 ± 0.75
Q21	I would use this tool in a professional environment	4.22 ± 0.96	4.20 ± 0.98	4.55 ± 0.66
Q22	My overall assessment about the tool is positive	4.41 ± 0.81	4.33 ± 0.52	4.70 ± 0.47

The questionnaire was assessed using a 5-level Likert scale (1 = strongly disagree, 5 = strongly agree). The responses were statistically analyzed using a two-sample Welch's t-test, which is broadly used in social science studies [36]. Furthermore, this method is recommended when the groups analyzed have substantially different standard deviations, the sample sizes are unequal, or the sample size is less than or equal to 10 values [37]. Additionally, the Likert R package was utilized to easily present data, simplifying the process of converting raw data from the survey into meaningful visual representations [38]. This package was used to show the distribution of the responses related to different topics and user groups in Figures 10–13.

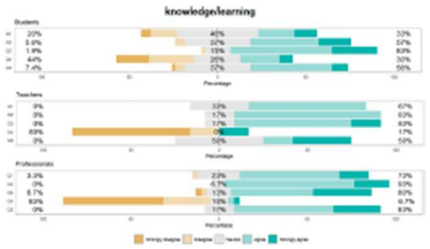


Figure 10. Comparative results between groups for knowledge/learning topics.

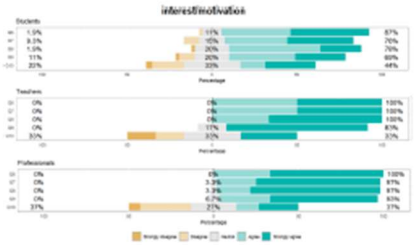


Figure 11. Comparative results between groups for interest/motivation topics.

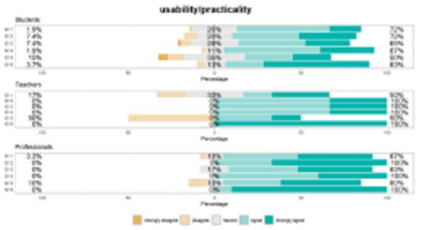


Figure 12. Comparative results between groups for usability/practicality topics.

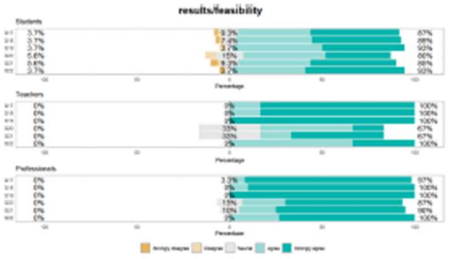


Figure 13. Comparative results between groups for results/feasibility topics.

5.2.1. Knowledge/Learning

Regarding the set of questions on "knowledge/learning", it is observed in Q1 that students rated themselves with a lower level of proficiency than teachers, and teachers,

in turn, with a lower level than experts. The largest significant difference was found between the responses of professionals vs. students ($p << 0.05$). This could be explained by the greater knowledge that—in effect—ICT experts have. In Q2, we observed the same proportion of scores as in Q1, which could indicate that professionals in the ICT sector find the DRACSC functionalities appropriate to apply the theoretical knowledge acquired. This is reinforced by finding the largest significant difference between the responses of professionals vs. students ($p << 0.05$), which is in line with the given interpretation of Q1.

On the other hand, it was valued very positively in Q3—similarly by all users—that the DRACSC system allowed them to acquire new theoretical knowledge ($p >> 0.05$ for all groups). In this regard, the absence of significant differences between groups could indicate that there is no bias between users. Q4 is a control question that reinforces the idea that theoretical concepts are not learned only by studying and that using the proposed system does not hinder learning. Observing the scores, it can be drawn as a conclusion that ICT professionals believe more categorically that learning is achieved in a more experimental than theoretical way. This difference is revealed in the statistical analysis, which found significant differences between students vs. professionals ($p << 0.05$). It was assessed positively in Q5 that the confidence to work with ICT improved after using the DRACSC system. The statistical analysis found the most significant difference between students and experts ($p << 0.05$), which could mean that this is due again to the greater knowledge on the subject of professionals compared to the other user groups. We can refer to the comparative histogram, depicted in Figure 10, which presents the data distribution for this set of questions.

5.2.2. Interest/Motivation

Regarding the set of questions on "interest/motivation", the ratings received in Q6 show that the proposed system can be a helpful tool for training in computer networks. The professionals perceived more potential than the teachers, and the teachers perceived more potential than the students. The statistical analysis found significant differences comparing students vs. professionals ($p << 0.05$), which would mean that they perceive motivation differently due to their role and knowledge of the subject. In Q7, one of the most representative functions of the proposed system was asked (i.e., MACRO functions), receiving very positive ratings from all user groups. The statistical analysis found significant differences between students and professionals ($p << 0.05$). This is interpreted by the fact that professionals are more aware of the potential that MACRO functions can offer by being in contact with the job world. In Q8, the trend continues for another feature of the DRACSC system consulted (i.e., cloud repository). Looking at the ratings, we found that students perceived the advantage of using the web repository less than the other users, who better know or usually work with services like this. This is confirmed in the statistical analysis, for which we found significant differences between students vs. teachers and professionals ($p << 0.05$). In Q9, teachers highly rated the independence of the DRACSC system from manufacturers. This could mean that this group of users values the work in the classroom with equipment from different providers, being aware of the difficulty of performance that the protocols and commands imply in educational environments. Following the trend, the statistical analysis found significant differences between students and the rest of the groups ($p << 0.5$), which could mean that students value this aspect less than the rest of the users motivated by their role. On the contrary, in Q10, we found that students are the group of people with the highest personal expectations to face projects like this in the future. The statistical analysis found no significant differences between the user groups, which could mean no bias in these answers ($p >> 0.05$). Considering the age of the users could explain the lower scores obtained by the students. In this respect, the older age of teachers and professionals could be related to more responsibility at work and less time to tackle personal projects. Alternatively, this could also be related to greater student enthusiasm due to the learning environment in which they find themselves. The histogram in Figure 11 presents the data distribution for this category of questions.

5.2.3. Usability/Practicality

Regarding the set of questions on "usability/practicality", we observed in Q11 that the teachers thought that perhaps more information should be provided to use the DRACSC system effectively, in agreement with the students. The statistical analysis found no significant differences between the two groups ($p >> 0.05$). On the contrary, the professionals considered the information sufficient to work with the tool. The statistical analysis found significant differences between experts and students, suggesting that one's own knowledge may influence the opinion on the adequacy of the information ($p << 0.05$). In Q12, we found that the feeling of interactivity with the DRACSC system was good, as this feature was perceived less by the group of students than by the rest of the users. Likewise, the statistical study found significant differences between students and experts in line with the results obtained ($p << 0.05$). In the case of Q13, students were the group that least valued the appearance of the interface, although positively. However, professionals—and teachers, to a greater extent—perceived the interface as easier to use. The analysis showed significant differences in students vs professionals ($p << 0.05$). This may be due to the higher contact that students have with very attractive websites (e.g., social networks and e-commerce). In Q14, all user groups similarly agreed that the menus were clear and intuitive, not leading to errors or confusion that could result from poor design. The statistical analysis found no significant differences, and no user bias was found in the system ($p >> 0.05$). Looking at Q15, we found that teachers perceived more technical problems during the use of the prototype. Their students supported this idea, not finding significant differences between the two groups ($p >> 0.05$). In contrast, industry professionals perceived fewer faults. Statistical analysis found significant differences between the opinions of professionals and students ($p << 0.05$). One explanation could be due to the fact that professionals were more aware of the difference between handling a prototype and a finished commercial product. Finally, the portability of the proposed system was rated highly in Q16 by all groups surveyed. The statistical study found significant differences between students and teachers, as well as students and practitioners ($p << 0.05$). One explanation could be the greater exposure students have to portable electronic devices (e.g., smartphones or game consoles), which would be a less remarkable aspect for them. We can refer to the comparative histogram, depicted in Figure 12, which presents the data distribution for this group of questions.

5.2.4. Results/Feasibility

About the set of questions on "results/feasibility", all groups agreed positively in Q17 that the functionalities of the DRACSC system were useful. The statistical analysis found significant differences between students and professionals ($p << 0.05$). This difference in opinion could be explained by considering the lower level of prior knowledge of the students detected in Q1, which could influence how they perceived the usefulness of the DRACSC system functions for carrying out tasks in data networks. Similarly, the MACRO functions would be a clear advantage of the DRACSC system according to Q18, although somewhat less for students than for teachers and professionals. The statistical study again found significant differences in the students, having a bias with respect to the other groups in accordance with the previous questions ($p << 0.05$). In Q19, we observed that teachers and professionals rated with the highest score that the system allowed for saving time and effort in the tasks carried out. The statistical analysis confirmed the significant differences between students and the rest of the groups, which could indicate that the former are less aware of the advantages due to their lesser knowledge of the area. In Q20, we observed that teachers were slightly more conservative about the viability of the proposed system in the educational context, although positively. This could be motivated by being the group with more founded criteria on education in the classroom. Likewise, we found significant differences with respect to the students, which could mean a bias due to their role ($p << 0.05$). The three groups had a fairly positive opinion in Q21, not finding significant differences between them, except between students and professionals ($p << 0.05$). In line with the previous interpretations, this significant difference could be due to the

limited exposure of the students to the professional world. Finally, the overall assessment of the DRACSC system was quite positive, with the professionals being the highest of all the groups (Q22). The statistical analysis found the same significant differences between the group of students and experts ($p << 0.05$), possibly due to a more formed opinion of the professional sector over the rest. In this sense, user feedback on the DRACSC system highlighted two main features. On the one hand, portability is a distinctive feature. On the other hand, the potential of the proposed system can be applied both in the educational and professional fields. A comparative histogram, depicted in Figure 13, presents the data distribution for the set of questions.

5.3. Qualitative and Quantitative Validation of Conclusions

Figure 14 shows the responses given by students, teachers, and professionals. Comparing the lines, it can be assumed that they have certain similarities in trend, which is why linear regression has been used to estimate the strength of the relationship between pairs through the r^2 value. As a result, an acceptable correlation was obtained both between students and teachers ($r^2 = 0.72$) and between professionals and teachers ($r^2 = 0.74$), as well as a good correlation between students and professionals ($r^2 = 0.84$). This result implies a significant relationship between the trends of the three groups of users [39].

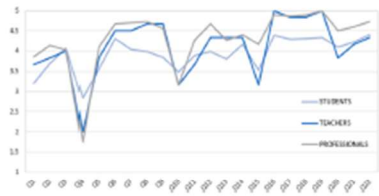


Figure 14. Comparative results between students, teachers, and professionals.

It can be observed in Figure 15 that, after carrying out an analysis of the trend line between the responses of the students from the different institutes, all of them except "Martín Rivero" follow a similar trend. This behavior may be due to the fact that the students at that institute were the only ones who attended a cycle oriented to software development, unlike the other centers, so hardware-oriented tasks could have influenced their perception.

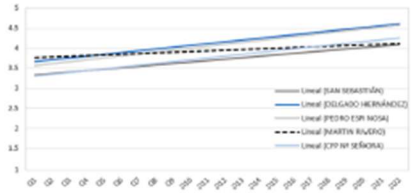


Figure 15. Trend line analysis among students from different high schools.

If the questions are analyzed according to the thematic field (i.e., learning, motivation, usability, and viability), it can be observed that the score of the professionals was always the highest (Figure 16). On the contrary, it can be observed that students were the group of users that gave the lowest score in the surveys in general, except in the block on "feasibility", where the teachers gave a slightly lower score. It is worth mentioning that Q1 and Q4 have not been included in the bar on "learning" as they are independent questions of the analyzed DRACSC system.

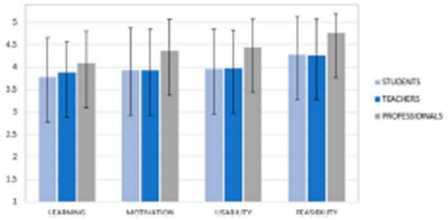


Figure 16. Questions grouped by learning topic.

Figure 17 shows the average time in seconds taken by professionals and students to carry out the tasks proposed in the experimentation. The mean time of the professionals is shown in red, while blue is used to identify the fastest students and striped blue to identify the students who took the longest time. In general, the professionals were faster in the execution of the activities as in Task 1. It should be noted that times in Task 2 are virtually constant in all cases, as it requires pressing a button on the DRACSC device screen and waiting for the response. It is also worth mentioning that, in Task 3, where users work with the cloud repository, the fastest students took less time than the professionals. This could be explained by the greater ease that students tend to have to handle web applications due to their age and familiarity with small electronic devices.

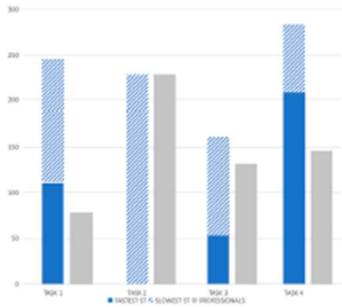


Figure 17. Time taken to complete use cases.

To complete the study, Figure 18 shows a factorial correspondence analysis (FCA) obtained with the Pandas and the Scikit-learn libraries for Python. It delineates significant relationships between various questions (Q1–Q22) and participant groups (teachers, professionals, and students). The foremost principal component (F1), standing for 91.67% of the variance, implies the most substantial fluctuation in the data, while the subsequent principal component (F2) captures the subsequent noteworthy variation. The graph portrays distinct question groups as colored points (blue for Q1–Q5, green for Q11–Q16, purple for Q6–Q10, and red for Q17–Q22), whilst the participant groups are denoted by black points. Regarding knowledge/learning (Q1–Q5), participants highly value the questions. In (Q1), participants with higher prior knowledge found the tool particularly beneficial. The ability to work with already-known theoretical concepts (Q2) is generally well received across all groups. However, acquiring new theoretical knowledge (Q3) appears to be more challenging, especially for students, suggesting a need for additional support or resources. Theoretical concepts are learned effectively (Q4), and participants report an increased confidence level in working with data networks after using the tool (Q5), with teachers and professionals particularly noting this improvement. Questions on motivation/interest

(Q6–Q10) show varied responses. The use of the system to encourage motivation and interest in networking (Q6) is closer to students, indicating a potential area for improvement in engaging this group. Moreover, creating MACRO solutions (Q7) positively influences motivation across all participants. Using a web repository (Q8) and independence from device manufacturers (Q9) are highly motivating features that teachers and professionals appreciate. The potential to create personal projects using low-cost components (Q10) is a strong motivating aspect, suggesting a high incentive for participants to apply the solution in future projects. Features on usability/practicality (Q11–Q16) receive mixed but mostly positive feedback. Participants agree that the information provided (Q11) is sufficient and the system's interactivity (Q12) is appropriate, with teachers and professionals finding these aspects particularly beneficial. The appearance of the system GUI (Q13) is attractive, and the tool's menus are considered clear and intuitive (Q14). Most users confirm the device's technical reliability (Q15), indicating no significant technical issues. However, portability (Q16) is highlighted as an essential feature, especially by students. In the last group, the analysis of results/feasibility (Q17–Q22), the functionalities enabled by the system (Q17), and the usefulness of the MACRO concept for device configuration (Q18) are acknowledged positively. However, the system's ability to save time and effort in tasks performed (Q19) and its feasibility for implementation in educational contexts (Q20) received mixed reviews, particularly from professionals who may be more skeptical about timesaving. The intention to use the tool in a professional environment (Q21) is moderately positive, but the overall assessment (Q22) reveals some reservations, particularly from students.

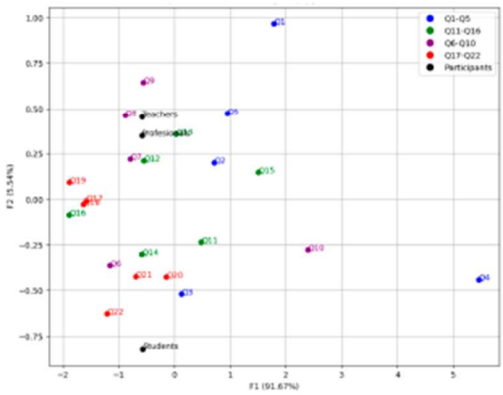


Figure 18. Symmetric plot of the PCA about the evaluation questionnaire.

6. Conclusions, Limitations, and Future Work

The management of communication equipment is a complex task—usually relegated to technical experts in ICT—that demands strong analytical and troubleshooting skills to deliver the best network performance and service. The traditional methodology based on manual equipment management is ineffective due to time and cost. This is especially true when managing larger infrastructures. For this reason, centralized solutions for managing network equipment and services must provide a high degree of configuration and functionality to be truly useful.

With this objective, this paper presents a portable system that automates the management of equipment configurations and OS. The prototype—called DRACSC—was implemented with a low-cost hardware/software solution based on the Raspberry Pi 4 Model B. In addition, this paper presented a centralized cloud repository developed with the aim of facilitating administration tasks through MACRO functions. This system was

tested on managed equipment, including Cisco and MikroTik routers, and is extendible to others. This was possible because the MACRO function concept provides model and manufacturer independence, as well as serves as the basis for developing a procedural language for the harmonized management of network devices. This solution is of particular interest in environments with a significant number of network elements, as it is able to act without having to remember which access credential corresponds to each device or without requiring expert knowledge to perform several different tasks on the devices.

Regarding experimentation, very favorable results were obtained that encourage continuing this research for teaching and professional environments, which are the scenarios for which the DRACSC system is designed. The average score obtained with students, teachers, and ICT professionals out of 5 points was 3.78 ± 0.88, 3.88 ± 0.67, and 4.09 ± 0.71 in the "knowledge/learning" area, which indicates its usefulness as a teaching tool. In the block on "interest/motivation", a rating of 3.93 ± 0.94, 3.92 ± 0.93, and 4.37 ± 0.69 was obtained, indicating the great potential of this system to help work with managed devices and make tasks more pleasant. Regarding "usability/practicality", a score of 3.96 ± 0.88, 3.97 ± 0.86, and 4.4 ± 0.63 was obtained, where all groups agreed on the possible use of the DRACSC system in various fields, significantly saving time in making different configurations and centralizing them in a single system. In "results/feasibility", a rating of 4.28 ± 0.84, 4.27 ± 0.82, and 4.77 ± 0.43 was obtained, which means that all groups approve that this tool has potential in the educational and professional context.

Regarding the execution time of the tasks, it was observed that the professionals performed better in general. This may be because these people have more experience and handling when setting up equipment. However, we also found that some students achieved good times, which could be related to the ease of the DRACSC system in making certain configurations.

Regarding the study limitations, we consider that it would be necessary to include a greater number of teachers to obtain a more representative sample. Nevertheless, although this group is small compared to the others, it provides an acceptable benchmark. This is supported by the fact that trends between groups—and between different institutes—follow similar patterns. To cover a more significant number of teachers, it would be necessary to access a larger number of institutes since, typically, there is one teacher for each group of students. On the other hand, it is also possible to observe the great difference between the number of men and women in the study. In the ICT sector, it is difficult to reach a parity number due to the great gender difference in engineering careers.

Regarding future work, both the gender issue and the number of teachers will be addressed. Moreover, we plan to conduct new experiments using an alternative setup to explore additional use cases for DRACSC that were not covered in this article. For example, we will use new network devices (e.g., switches, access points, and intrusion detection systems), which will increase the complexity of the test network. These new data will enable us to analyze any potential deviations from the current study. On the other hand, current developments are focused on obtaining the greater dissemination and acceptance of the DRACSC system by users. For this, updates will be addressed to improve usability through the graphical interface and manage relays through the GPIO of Raspberry Pi to turn on/off equipment remotely. Also, we will develop a cloud platform that will communicate with DRACSC devices to allow additional actions that require greater resource consumption and that, due to hardware limitations, could not be carried out in the solution provided in this work. These include the administration of multiple networks, the maintenance of a database with configuration history, error recovery, and the automatic repair of network elements using rules defined through expert knowledge, in addition to the development of a new module to preserve the integrity and non-repudiation of user log files using Blockchain technology.

Author Contributions: Conceptualization, J.D.M.R. and Ť.J.M.S.; methodology, Ť.J.M.S.; software, J.D.M.R.; validation, Ť.J.M.S.; formal analysis, Ť.J.M.S.; investigation, J.D.M.R. and Ť.J.M.S.; resources, J.D.M.R. and Ť.J.M.S.; data curation, J.D.M.R. and Ť.J.M.S.; writing—original draft preparation,

J.D.M.R.; writing—review and editing, T.J.M.S.; visualization, J.D.M.R. and T.J.M.S.; supervision, T.J.M.S.; project administration, T.J.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: We are grateful to the Dep. of Electronic Engineering, Computer Systems and Automatics at the UHU for their collaboration, to the educational centers IES San Sebastián (Huelva), IES Delgado Hernández (Bollullos Par del Condado), IES Pedro Espinosa (Antequera), IES Martín Rivero (Ronda), and CFP Nuestra Señora De Las Mercedes (Bollullos Par del Condado), and to all the professionals who participated in the experimentation.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Clarke, G.E. *CompTIA Network+ Certification Study Guide*; McGraw Hill Professional: New York, NY, USA, 2018.
2. Alani, M.M. *Guide to Cisco Routers Configuration: Becoming a Router Geek*; Springer: Cham, Switzerland, 2017.
3. Ariganello, E. *REDES CISCO. Guía de estudio para la certificación CCNA Routing y Switching*; Ra-Ma: Paracuellos de Jarama, Spain, 2017.
4. Mateo Sanguino, T.J.; Fernández de Viana González, I.J.; Fernández, J.E.; Domínguez, A.G. Using Identity Provider and Automatic Resource Management to Improve a Remote Networking Lab. *IEEE Lat. Am. Trans.* **2018**, *16*, 1547–1556. [CrossRef]
5. Morillo Reina, J.D.; Mateo Sanguino, T.J. Portable Device for Easy Management and Automatic Recovery of Networking Systems. *IEEE Lat. Am. Trans.* **2019**, *17*, 401–408. [CrossRef]
6. Mateo Sanguino, T.J.; Fernández de Viana, I.; López García, D.A.; Cortés Ancos, E. OpenGnSys: A Novel System toward Centralized Deployment and Management of Computer Laboratories. *Comput. Educ.* **2014**, *75*, 30–43. [CrossRef]
7. Pardo Garrido, D.; Mateo Sanguino, T.J.; López García, D.A.; Cortés Ancos, E.; Fernández de Viana González, I.J. OpenGnSys: Un Sistema de Gestión Centralizada y Despliegue de Sistemas Operativos en el Aula. In Proceedings of the 2013 8th Iberian Conference on Information Systems and Technologies (CISTI), Lisboa, Portugal, 19–22 June 2013; pp. 295–301.
8. Mateo Sanguino, T.J.; López García, D.A.; Cortes Ancos, E. The Role of Telematic Practices in Computer Engineering: A Low-cost Remote Power Control in a Netwok Lab. *Intern. J. Online Eng.* **2012**, *8*, 15–22. [CrossRef]
9. Andújar Márquez, J.M.; Mateo Sanguino, T.J. Design of Virtual and/or Remote Laboratories. A Practical Case. *Rev. Iberoam. Autom. Inform. Ind.* **2010**, *7*, 64–72. [CrossRef]
10. Mateo Sanguino, T.J.; Fernández de Viana González, I.J.; Cortés Ancos, E.; Espejo Fernández, J. Exploring Strengths and Weaknesses: A Case Study After Developing a Remote Network Lab. *Comput. Appl. Eng. Educ.* **2018**, *26*, 1422–1434. [CrossRef]
11. Mateo Sanguino, T.J.; Morillo Reina, J.D. Dispositivo y Sistema para la Recuperación de Equipos de Comunicación. ES Patent ES 2569414, 14 October 2014.
12. Mateo Sanguino, T.J.; Morillo Reina, J.D. Device and System for the Recovery of Communication Equipment. WO Patent WO/2016/055682, 2016.
13. Docter, Q. *CompTIA IT Fundamentals (ITF+) Study Guide with Online Labs: Exam FC0-U61*; Sybex: Singapore, 2020.
14. Elmansor, K. Towards Automated Network Configuration Management. Master's Thesis, College of Computing and Digital Media, Chicago, IL, USA, 2013.
15. Cisco Networking Academy. *Scaling Networks Companion Guide*; Cisco Press: Hoboken, NJ, USA, 2014.
16. IT Digital Media Group. *Las Claves de una Adecuada WLAN en la Empresa*; IT-User Tech & Business: Madrid, Spain, 2016; pp. 33–38.
17. Elangovan, K.R. Security Framework for Supply-Chain Management. In *Research Anthology on Business Aspects of Cybersecurity*; Information Resources Management Association (USA): Hershey, PA, USA, 2022; pp. 587–610.
18. EMA. Essential IT Monitoring: Seven Priorities for Network Management. Technical Report. 2013. Available online: https://content.solarwinds.com/creative/pdf/whitepapers/ema-solarwinds_netmonitoring-0913-wp.pdf (accessed on 23 October 2024).
19. Siregar, M.H. Network Monitoring Sistem Menggunakan Whatsup Gold Pada Pt. Pembangunan Jaya Ancol, Tbk. *J-SAKTI (J. Sains Komput. Inform.)* **2021**, *5*, 197–208.
20. ATEN Altusem. SN0100CO/SN0100COD/SN9100CO Series Serial Console Server User Manual. Technical Report. 2021. Available online: https://assets.aten.com/product/manual/serial-console-server-user-manual-w.pdf (accessed on 23 October 2024).
21. Perle Systems Ltd. IOLAN SDS/SCS/STS User's Guide. Technical Report. 2018. Available online: https://www.perle.com (accessed on 23 October 2024).
22. Raritan, Inc. Dominion SX II User Guide 2.0.0. Technical Report. 2015. Available online: http://support.raritan.com/sx-ii/v2.0.0/DSX2-v2.0.0-E.pdf (accessed on 23 October 2024).
23. Raritan, Inc. Opengear User Manual 4.12.0. Technical Report. 2021. Available online: https://ftp.opengear.com/download/documentation/manual/previous%20versions/Opengear%20User%20Manual4.12X.pdf (accessed on 23 October 2024).

24. Belunix. Raspisco—Remote access to Cisco through Raspberry Pi. Technical Report. 2013. Available online: http://developers-club.com/posts/192188/ (accessed on 23 October 2024).
25. Kyuchukova, D.; Hristov, G.; Zahariev, P.; Borisov, S. A study on the possibility to use Raspberry Pi as a console server for remote access to devices in virtual learning environments. In Proceedings of the IEEE 2015 International Conference on Information Technology Based Higher Education and Training (ITHET), Lisbon, Portugal, 11–13 June 2015.
26. Nelson, S. The Internet of Getting Things Done. *New Electron.* **2023**, *51*, 16–17.
27. Haro, L.F.D.; Cordoba, R.; Rojo Rivero, J.I.; Diez de la Fuente, J.; Avendano Peces, D.; Bermudo Mera, J.M. Low-Cost Speaker and Language Recognition Systems Running on a Raspberry Pi. *IEEE Lat. Am. Trans.* **2014**, *12*, 755–763. [CrossRef]
28. Moreno, E.; Anjo Lima, F.; Azevedo Dias, W.R. Performance Analysis of a Low Cost Cluster with Parallel Applications and ARM Processors. *IEEE Lat. Am. Trans.* **2016**, *14*, 4591–4596.
29. Makopa, J.; Christopher, A.; Shah, R.; Mandela, N. Internet of Things (IoT) Network Forensic Analysis Using the Raspberry Pi 4 Model B and Open-Source Tools. In Proceedings of the 2023 International Conference on Quantum Technologies, Communications, Computing, Hardware and Embedded Systems Security (iQ-CCHESS), Kottayam, India, 15–16 September 2023; pp. 1–7.
30. Maulana, H.; Al-Khowarizmi. Analyze and Designing Low-Cost Network Monitoring System Using Icinga and Raspberry Pi. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *704*, 012038. [CrossRef]
31. Cooper, P. *Beginning Ruby: From Novice to Professional*; Apress: Berkeley, CA, USA, 2016.
32. Alshanetsky, I. Nginx Tricks for PHP Developers. In Proceedings of the International PHP Conference, Munich, Germany, 25–29 October 2015.
33. Forde, M. Using the Raspberry Pi for Data Collection, Display and Dissemination. In Proceedings of the Technical Conference Meteorological and Environmental Instruments and Methods of Observation, Madrid, Spain, 27–30 September 2016.
34. Larsson, M. Sanitization of Embedded Network Devices. Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.
35. Delacre, M.; Lakensand, D.; Leys, C. Why Psychologists Should by Default Use Welch's t-test Instead of Student's t-test. *Int. Rev. Soc. Psychol.* **2017**, *30*, 92–101. [CrossRef]
36. Gravetter, F.J.; Wallnau, L.B. *Statistics for the Behavioral Sciences*, 10th ed.; Cengage Learning: Boston, MA, USA, 2017; p. 200.
37. McDonald, J.H. *Handbook of Biological Statistics*, 3rd ed.; Sparky House Publishing: Baltimore, MD, USA, 2014; pp. 127–131.
38. Bryer, J. Analysis and Visualization of Likert Based Items. 2013. Available online: https://github.com/jbryer/likert (accessed on 23 October 2024).
39. Moore, D.S.; Notz, W.I.; Flinger, M.A. *The Basic Practice of Statistics*, 6th ed.; W. H. Freeman and Company: New York, NY, USA, 2013; pp. 125–157.

4.3 Artículo 3

Decentralized and Secure Blockchain Solution for Tamper-proof Logging Events

Morillo Reina, J.D., Mateo Sanguino, T.J.

Published in:

Journal: Future Internet
Editorial: MDPI
Editor-in-Chief: Gianluigi Ferrari
Reference: Volume 17, Issue 3, Article ID 108
Year: 2025
ISSN: 1999-5903
DOI: 10.3390/fi17030108

Categoría	**Posición/ Total**	**Cuartil**
Computer Science, Information Systems	110/252	Q2
Impact Factor (2023)	2.8	

future internet

MDPI

Article

Decentralized and Secure Blockchain Solution for Tamper-Proof Logging Events

J. D. Morillo Reina and T. J. Mateo Sanguino *

Department of Electronics, Computer Systems, and Automation Engineering, University of Huelva, Avda. de las Artes S/N, 21007 Huelva, Spain; juandiego.morillo@alu.uhu.es
* Correspondence: tomas.mateo@diesia.uhu.es

Abstract: Log files are essential assets for IT engineers engaged in the security of server and computer systems. They provide crucial information for identifying malicious events, conducting cybersecurity incident analyses, performing audits, system maintenance, and ensuring compliance with security regulations. Nevertheless, there is still the possibility of deliberate data manipulation by own personnel, especially with regard to system access and configuration changes, where error tracking or debugging traces are vital. To address tampering of log files, this work proposes a solution to ensure data integrity, immutability, and non-repudiation through different blockchain-based public registry systems. This approach offers an additional layer of security through a decentralized, tamper-resistant ledger. To this end, this manuscript aims to provide a solid guideline for creating secure log storage systems. For this purpose, methodologies and experiments using two different blockchains are presented to demonstrate their effectiveness in various contexts, such as transactions with and without metadata. The findings suggest that Solana's response times make it well suited for environments with moderately critical records requiring certification. In contrast, Cardano shows higher response times, thus making it suitable for less frequent events with metadata that requires legitimacy.

Keywords: blockchain; Cardano; Solana; logs; Raspberry Pi; networks

check for updates

Academic Editor: Paolo Bellavista

Received: 9 January 2025
Revised: 20 February 2025
Accepted: 27 February 2025
Published: 1 March 2025

Citation: Morillo Reina, J.D.; Mateo Sanguino, T.J. Decentralized and Secure Blockchain Solution for Tamper-Proof Logging Events. *Future Internet* **2025**, *17*, 108. https://doi.org/10.3390/fi17030108

1. Introduction

Log files are used as a control mechanism to detect security incidents, policy violations, regulatory compliance, and operational problems in computer applications. These logs contain information about specific events on an organization's networks or systems. They provide both the current states of the systems and the traceability of the actions performed on the network devices, providing a detailed description of their behavior. Logs are therefore essential for forensic investigation and security auditing, such as compliance with regulations and standards like the Health Insurance Portability and Accountability Act (HIPAA) in the United States or the General Data Protection Regulation (GDPR) in the European Union [1].

Different organizations are defining diverse standards for managing log events that lack global uniformity. This is the case for the regulations to which a sector is subject in a specific country (e.g., Law 25/2007 in Spain) or the international standards such as PCI DSS (Payment Card Industry Data Security Standard). Either way, different types of logs include audit logs, whose primary purpose is to track changes made to systems or their behavior during use, and application logs, written by applications indicating what happens at runtime [2].

One of the classic challenges in log management is preserving the integrity of stored information to ensure its availability and reliability. Data can be subject to threats from third parties, such as encryption, rendering it unusable through ransomware infection, alteration, or content deletion to cover the trace of cybercriminals, as well as manipulation of information by insiders seeking to take advantage of a situation. Therefore, it is recommended to follow a proactive Zero Trust approach that ensures trust in log records about events occurring in our network rather than automatically relying on the trust of a user, device, or network [3]. Recently, blockchain technology has become popular as it provides secure, decentralized data storage. However, other inherent characteristics of this technology, such as immutability or complete traceability of the stored data, have also generated interest in its potential use. Consequently, researchers have conducted several studies on utilizing blockchain for storing certain types of data, such as access, configurations, or payments [4].

Given these precedents, the present hypothesis posits the potential use of a public blockchain to certify the integrity, immutability, and non-repudiation of the system's activity log messages. By leveraging this technology, it may be possible to establish a reliable and secure system for recording and verifying data, thereby enhancing the trust and transparency of the overall system. Therefore, this solution has the potential to be universally adopted as an openly auditable service by organizations looking to reinforce their existing logging infrastructures while ensuring critical audit events remain tamper-proof.

In the related work, several blockchain-based approaches have been proposed to ensure the integrity and immutability of log events. Some solutions, such as Exonum and Hyperledger, employ private blockchains to offer improved control; however, they encounter issues related to reduced transparency and centralized trust. Alternatively, solutions based on public blockchain, like Bitcoin, provide a high level of security but are limited by high transaction costs and scalability challenges. The proposed system addresses these limitations by implementing a decentralized, open-source log storage mechanism that utilizes public blockchains optimized for cost efficiency and transaction speed. Unlike previous methods, the proposed solution can effectively manage a substantial rate of log events on public blockchains, thereby enhancing trust, transparency, and compliance with industry regulations while ensuring practical deployment feasibility.

This work arises as a new feature of the DRACSC system, the acronym in Spanish for Automatic Recovery and Configuration Device for Communication Systems, whose objective is to facilitate and centralize typical management tasks of network equipment (e.g., applying new configurations, installing firmware, creating, or restoring backups). To illustrate the relevance of secure logging, consider a real-world use case involving a network administrator managing router configurations (Figure 1). This administrator is authorized to execute sensitive actions, including adding routes to the routing table, creating user accounts, modifying firewall rules, and enabling promiscuous mode. Each of these actions carries inherent security risks; unauthorized modifications could potentially disrupt network traffic, grant unintentional access, expose systems to threats, or facilitate the interception of sensitive data [5].

This solution is part of a larger educational project, which was introduced in routers and switches in a network laboratory at the University of Huelva (Spain) and validated by students, teachers, and IT professionals [6–8].

In summary, the contributions of this work concerning the previous ones are the following: (i) to present a novel solution that allows immutable storage and recording of log events generated by network equipment; (ii) to solve the limitations of the current state of the art in terms of visibility and transparency of log records through public blockchains; (iii) to generalize the applicability of the system in other contexts through use cases that

use transactions with and without metadata; and (iv) to provide the source code to the community in an open way to achieve transparency and move towards the establishment of a standard for log storage on the blockchain.

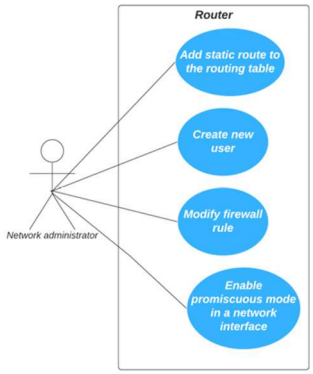


Figure 1. Use case on secure access in router management.

In order to study the feasibility of this solution, two public blockchains (i.e., Cardano and Solana) have been selected, both with a level of maturity suitable for use in a production environment. The reason for choosing these blockchains is that they allow low-cost transactions with a high number of events, such as those that commonly occur in a communication network (i.e., our test scenario). Therefore, the authors have analyzed the response times through comprehensive experimentation with different rates of events per second.

To this end, a complete software system has been developed to handle everything from data acquisition to persistence storage in public blockchains. Additionally, it includes a server for reading and processing the information effectively. This solution has been made available as open-source software, which enhances its transparency and facilitates its adoption by secure IT professionals, allowing them to establish customized functionalities and collaborate in its continuous development and improvement [9].

The article is structured as follows. Section 2 presents state-of-the-art solutions using blockchain for log storage. Section 3 describes the complete architecture of the system by separating it into three subsections with each of the different implementations. Section 4 presents the experiment and analyzes the impact of the solution in different scenarios. Section 5 provides the outcomes and corresponding discussions. Finally, Section 6 provides the conclusions, limitations, and future work.

2. Related Work

There are various studies in the state of the art that focus on the use of blockchain technology for the registration of information. In the context of this work, we highlight those solutions focused on the persistence of activity records. Log files generated by systems cover a wide range of possibilities, including user access, security incidents, and events generated by both applications and networks, among others [10]. The auditing processes also pursue the persistence of log files, which may seek to know the degree of compliance an entity has with a specific regulation [1].

Outstanding research proposes an auditable system using Exonum [11], a private blockchain with several salient features. These include using the Byzantine Fault Tolerance

(BFT) consensus algorithm, the ability to execute smart contracts, and the use of public blockchains to additionally establish periodic trustworthiness via Bitcoin. This system integrates with a Security Information and Event Management (SIEM) solution to acquire events via the Syslog standard. Its hybrid architecture guarantees the immutability and integrity of logs by storing them both on-chain, using hashes, and off-chain in a local cluster. At the same time, it allows efficient access and proper management of security events. Nevertheless, the level of transparency and reliability of the system is compromised by storing a part of the data outside the public blockchain. In addition, the use of smart contracts complicates its implementation.

A different study [12] uses Hyperledger, a private blockchain implemented to store log files encrypted by users presumably before they are sent. Storing log files directly on the blockchain raises concerns about ensuring data integrity as it becomes impossible to verify that the data have not been tampered before being included in the blockchain. Additionally, similar to the previous scenario, implementing a private blockchain reduces the transparency and reliability of the system.

In [13], a system is proposed to help auditors verify compliance with regulations using Bitcoin, a public blockchain that uses the proof of work (PoW) consensus mechanism. Data are stored both on-chain and off-chain, but the system is not designed to store events generated by network equipment. In addition, neither the price of fees nor the number of transactions per second of the Bitcoin network are scalable for use in this context.

Several authors propose BlockAudit [14], a secure and transparent system for auditing logs in charge of storing the actions executed in the system. It records on-chain data through the private Hyperledger blockchain, so it lacks transparency and reliability compared to public blockchains.

An outstanding work [15] has proposed an autonomous log storage management system for IoT devices that uses both public and private blockchains using Ethereum and Hyperledger, respectively. The system stores the content of off-chain logs and their signatures on the private blockchain, whose transactions are managed through smart contracts. A summary of these transactions is signed and stored in the public blockchain, providing greater transparency and reliability. However, there is still a risk of tampering with the blocks on the private blockchain before they are sent to the public blockchain. Additionally, the system is subject to the transaction price volatility imposed by Ethereum.

Moreover, [16] introduced the blockchain-based secure log management system for cloud computing (BCALS) using Multichain, a private blockchain that stores log messages on-chain. The data contained in the event logs are sent to Elasticsearch for further exploitation, an open-source project used to search, analyze, and visualize large amounts of information in real-time. As in previous cases, the use of a private blockchain implies a compromise in the level of transparency and reliability of the system.

Another work [17] proposes using Hyperledger as a private blockchain, where smart contracts manage the transactions and the information associated with events is stored on-chain. This open-source project is specifically designed for IoT devices and focuses on their security. However, the transparency and reliability of a private network differ significantly from those of a public blockchain.

Furthermore, [18] proposes a blockchain-based service used in supply chain and logistics management, also known as Logchain Logistics as a Service (LCaaS). This service is divided into two levels of hierarchy to achieve scalability. The first one stores signatures on a blockchain that can be either public (i.e., Ethereum) or private (i.e., IBM blockchain), while the data corresponding to the signatures is stored both off-chain and on-chain. However, the service makes it impossible to certify that said data have not been modified before inclusion when the log files are stored on the chosen blockchain.

Another study has presented a Blockchain-Enabled Scalable Network Log Management System [19]. This system utilizes Multichain to store hashes, timestamps, and ownership data while employing the InterPlanetary File System (IPFS) to store complete logs. It also features a query mechanism that enables efficient retrieval and verification of logs. This effectively addresses the limitations often encountered in other blockchain-based log management systems, which typically lack structured search capabilities. However, similar to previous examples, relying on a private blockchain may diminish the overall transparency and reliability of the system.

In [20], a decentralized data storage network is introduced, utilizing a custom-built blockchain written in Go to prevent data manipulation. This system employs a PoW consensus mechanism to maintain data integrity and uses a Gossip Protocol for automatic block repair, enabling nodes to detect inconsistencies and retrieve valid copies of the blockchain from their peers. The data are stored on-chain to ensure immutability and an API Gateway facilitates log transactions through RESTful API endpoints. Nevertheless, being a private and custom blockchain raises concerns regarding transparency and reliability.

Finally, a blockchain-based audit log system has been proposed to ensure the data integrity of log files [21]. The system assumes that both loggers and auditors may be untrustworthy and mitigates these risks by utilizing smart contracts within Hyperledger Fabric. This method presents a way to generate on-chain integrity proofs using Non-Fungible Tokens (NFT), with log files stored off-chain in the IPFS. While this system enhances scalability and security, it is limited to a permissioned blockchain, which may decrease transparency compared to public blockchain solutions.

Table 1 presents a comparison of the previous solutions compared to the approach presented in this paper. It should be noted that only our solution and the one presented in [11] use the Syslog standard as a data source to ensure compatibility and interoperability with numerous tools and applications designed to work with this standard. It is also worth noting that most of the proposed works use private networks due to the cost of storing data in public blockchains, implying less transparency in the process. On the other hand, only [11,15,19] use some additional tools to visualize the stored information in a user-friendly way, as in the proposed solution. Finally, it is worth highlighting the open-source nature of the proposed system together with the proposal of [18], which allows transparency and facilitates project release for reproducibility within the scientific community, aiming to establish a standard for storing registries in the blockchain.

Table 1. Related work.

Reference	Data Storage	Kind of Blockchain	Consensus Algorithm	Transfer	Specific for Logging	Data Visualization	Multi Blockchain	Open Source	Year
[11]	Off-chain/On-chain	Private	BFT	Direct	✔	Web app	No	No	2019
[12]	On-chain	Private	BFT	Direct	✔	No	No	No	2018
[13]	Off-chain/On-chain	Public	PoW	Direct	✔	No	No	No	2017
[14]	On-chain	Private	BFT	Direct	✔	No	No	No	2018
[15]	Off-chain/On-chain	Both	PoW	Smart Contracts	✔	Web app	No	No	2020
[16]	On-chain	Private	PBFT	Direct	✔	Elasticsearch	No	No	2021
[17]	On-chain	Private	PBFT	Smart Contracts	No	No	No	No	2022
[18]	Off-chain/On-chain	Private	BFT	Direct	✔	No	✔	✔	2018
[19]	Off-chain/On-chain	Private	Permission-based mining	Direct	✔	Web app	No	No	2022
[20]	On-chain	Private	PoW	Direct	✔	No	No		2024
[21]	Off-chain/On-chain	Private	PBFT	Smart Contracts	✔	No	No	No	2024
Authors	Off-chain/On-chain	Public	Various *	Direct	✔	Web app	✔	✔	2024

* PoH + PoS for Solana and PoS for Cardano.

Finally, a comparative overview of blockchain applications is provided in Table 2 to emphasize the key aspects discussed in this section. Public blockchains, including Bitcoin, Ethereum, Cardano, and Solana, are characterized by open participation and decentralized governance. For instance, Bitcoin employs a PoW mechanism, allowing for approximately seven transactions per second (TPS). However, it experiences high and unpredictable fees due to market-driven congestion and offers limited smart contract functionality. Ethereum has transitioned to a PoS consensus model, achieving around 15 TPS. Despite this improvement, transaction fees remain high and variable because of a dynamic gas market, although it supports comprehensive smart contracts. In contrast, Cardano utilizes a deterministic fee model within its PoS system, reaching approximately 250 TPS while maintaining low fees and full smart contract capabilities. Solana enhances performance with a hybrid PoH and PoS mechanism, enabling the processing of up to 65,000 TPS at low and variable fees. On the private blockchain side, platforms such as Multichain, Exonum, Hyperledger Fabric, and IBM Blockchain are designed for enterprise applications. These systems generally feature configurable consensus mechanisms and administratively controlled fee structures, which can result in deterministic fees, sometimes as low as zero. However, standard TPS figures are not reported for these platforms, as they depend on specific configurations and the number of nodes utilized [22].

Table 2. Comparison between the several blockchains studied in the related work.

Blockchain	Kind of Blockchain	Consensus Algorithm	TPS	Transaction Fees	Deterministic Fees	Smart Contract Support	Year Launched
Bitcoin	Public	PoW	7	High	No	Limited	2009
Ethereum	Public	PoS	15	High	No	✔	2015
Multichain	Private	Several	*	*	✔	Limited	2015
Exonum	Private	BFT	*	*	✔	✔	2016
Hyperledger Fabric	Private	Several	*	*	✔	✔	2016
Cardano	Public	PoS	250	Low	✔	✔	2017
IBM Blockchain	Private	Several	*	*	✔	✔	2017
Solana	Public	PoH + PoS	65,000	Low	No	✔	2020

* These features on private blockchain depend on the configuration.

3. Materials and Methods

This section outlines the system components, types of transactions used for certifying log events, and blockchains deployed.

3.1. Components of the System

This section describes the components used in the system and their relationship, which are illustrated in Figure 2. The process is broken down into four steps from left to right: (1) the managed equipment (e.g., router, switch or PC) sends a log event to the Syslog server located in the DRACSC system; (2) the events are then queued in the message broker to be further processed by the log module; (3) depending on the blockchain used, the log module will perform specific tasks such as obtaining the hash of the event information, connecting to a blockchain node and sending the corresponding transaction to the network; and (4) when the hash obtained in the previous step is available and signed, it is sent to the log management server for storage and further exploitation.

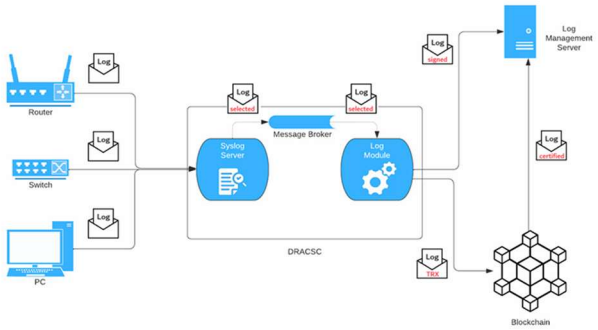


Figure 2. System components.

3.1.1. Managed Equipment

Network devices such as routers, switches, computers, servers or APs continuously log events and send corresponding log messages to the Syslog server.

3.1.2. Syslog Server

This component is responsible for receiving, filtering, transforming, storing, or redirecting log records generated by the managed network equipment. It is relevant to highlight that, to operate, the network equipment must activate a Syslog event notification service based on RFC 5424 [23], a widely accepted standard. Typically, the configuration process is straightforward. In our case, the Syslog server was installed in the DRACSC system. An open-source version of Syslog-ng has been chosen for this work, specifically version 4.1.1. This software can easily filter the event logs by utilizing attributes such as the severity level, IP address, or payload. This possibility of filtering optimizes log delivery, enhances profitability, and enables the adaptation of our solution to public blockchain performance. Eventually, the data received by the Syslog server is forwarded to the message broker via the advanced message queuing protocol (AMQP) for further processing in subsequent stages.

3.1.3. Message Broker

The purpose of this component is to temporarily store event messages while the log module is handling them. The message broker was implemented in the DRACSC system using RabbitMQ, specifically version 3.10.5. It is based on AMQP, which brings the advantage of scalability and asynchrony [24]. Additionally, the software offers other benefits, such as decoupling with other system services or assuring that events will not be lost due to the public blockchain being highly congested at a specific moment or failures in the log module service that the broker feeds. Other factors influencing their choice include a lightweight design, which is crucial because of hardware resource limitations and ease of use for prototyping.

3.1.4. Log Module

This service resides on the DRACSC system and receives the log records filtered through the message broker. Once received, it processes them to adapt them for both the blockchain format and the log management server. The programming language used is Node.js, specifically version 18.21. The choice was motivated by its high performance and asynchrony, as it is event-driven [25].

3.1.5. Blockchain

A blockchain is a distributed ledger technology (DLT). It is a database that can be shared by many entities in a peer-to-peer manner and allows information to be stored in an immutable and ordered way. In general, once a transaction has occurred, it cannot be altered or deleted. One of the fundamental elements is the consensus algorithm used, which is responsible for validating the information added to the blockchain between all the nodes that make it up and ensuring that all transactions are correct. There are a multitude of consensus algorithms, such as the two related to this work (i.e., Proof of History and Proof of Stake), which are described in the section on implemented blockchains. It is important to note that once the transaction is sent to the blockchain, it persists after the nodes that make up the network validate the transaction and add its information to a block after a few transactions established by the network.

3.1.6. Log Management Server

This module is a web service designed to receive Syslog messages and blockchain transaction data through a RESTful API, which are then stored for later exploitation. The server is also responsible for communicating with the blockchain to retrieve the timestamp of the generated block that permanently records the transaction (i.e., the certification of a log event). Moreover, this server offers an intuitive interface for easy access to the stored information as seen in Figure 3. To this end, Angular, an open-source JavaScript framework for the web interface, has been used in the development, specifically version 14. On the server side, the service uses version 18.21 of Node.js.

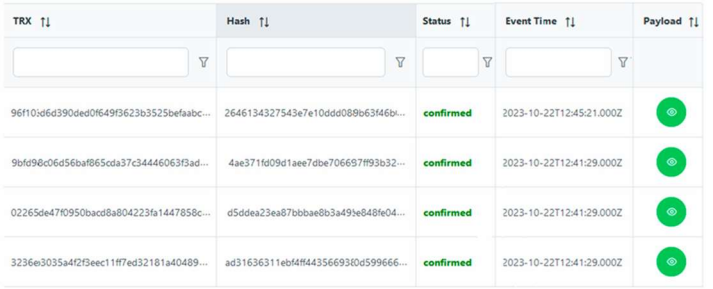


Figure 3. Main view of the log management server interface.

3.2. Types of Transactions

This section details two different implementations aimed at providing a universal, blockchain-independent solution for certifying registration events. One approach involves using a blockchain that can append metadata to transactions, thereby including the hash of the event generated by the managed equipment. The second approach provides a solution using a blockchain that avoids attaching metadata to transactions.

When an event from a managed network device arrives at the Syslog server, it is automatically forwarded to the message broker. Once there, the event will wait for the log module service to process it for subsequent transfer to the blockchain. When the blockchain selected does not allow metadata, the log module will perform a transaction with the blockchain that will require the payment of a certain number of tokens. The transaction ID generated in response will be used as the key, and the content of the log event as the message, thus letting us calculate a SHA2-based hash message authentication key (HMAC) [26]. These three elements (i.e., log event, transaction ID and resulting hash)

are then sent to the log management server for safekeeping. Finally, the log module will also send confirmation to the message broker indicating that the log event has been registered and is waiting for further events. Figure 4 shows the interaction process, where the managed device (e.g., a router) that has the Syslog protocol configured generates the event with a size of 256 bytes (e.g., incorrect login access).

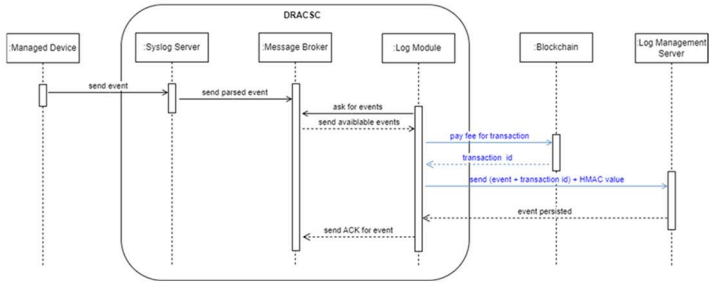


Figure 4. Storage of a log event using a transaction without metadata.

When the blockchain allows metadata, the log module receives a log event and calculates a hash of the event's content. This hash is then included in the metadata field before the corresponding transaction is initiated, and the result will persist in the blockchain once the block is completed. As in the previous case, a transaction ID will be received and sent to the log management server along with this event's hash and content. Figure 5 illustrates the interaction process, which differs from the previous method after the log module. Specifically, the log module will generate a SHA2 hash of the event and perform the transaction in the blockchain, adding the hash to the transaction's metadata and obtaining a transaction ID. Once the transaction ID is received, the log module will send the log event, transaction ID, and resulting hash as in the previous case.

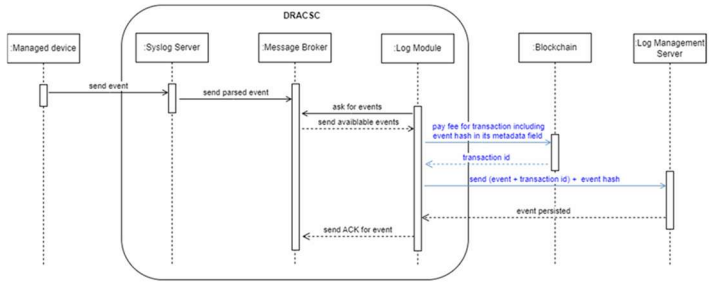


Figure 5. Storage of a log event using a transaction with metadata.

3.3. Case Studies

This section provides a concise depiction of the two blockchains examined in this manuscript, with a comparative analysis delineated in Table 3.

Table 3. Comparison of deployed blockchains.

Name	Token	Metadata	TPS	Consensus Algorithm	Deterministic Fee	Average Transaction Cost	Average Transaction Price (€) *	Release Year
Solana	SOL	No	65.000	PoH, PoS	No	5.001×10^{-6} SOL	0.019	2020
Cardano	ADA	✔	250	PoS	✔	0.2 ADA	0.064	2017

* Average price during the study in 2024.

3.3.1. Solana

This is an open source blockchain platform launched in 2020 [27]. As usual, among blockchains that do not include metadata in transactions, Solana offers a high processing capacity measured in transactions per second (TPS), exceeding 60,000 operations at maximum for all users [28]. It uses a combination of two consensus algorithms. Proof of History (PoH) generates chained timestamps to cryptographically certify the time and order of events on the blockchain. Additionally, it uses Proof of Stake (PoS) to decide which blocks should be added to the network.

The blockchain's native token is SOL, but transactions are measured in Lamports. The smallest unit of measurement in the Solana network is Lamport, equivalent to 10^{-9} SOL. It is important to note that the transaction fee cost depends on network congestion, so it is not deterministic.

To enable programmatic interaction with this blockchain, the official library 'solana-web3.js' has been used to communicate with Solana's Remote Procedure Call (RPC) JavaScript Object Notation (JSON) API. This same library was used by both the log module and the log management server. One of the key features of this library is that it establishes the connection with the blockchain completely, simplifying the process by avoiding using other additional software packages (e.g., installation of a blockchain node or wallet management software).

Several functions have been implemented for the operation of the log module: the 'initSolana' function is in charge of establishing the connection with the blockchain (i.e., the testnet in our case); the 'createWallet' function creates a new wallet if it does not already exist; the 'writeLog' function sends a transaction to the blockchain to store a log event; and the 'getBalance' function checks the amount of SOL tokens available in the wallet. Moreover, the log management server also uses the 'solana-web3.js' library to check if the block containing the log record persists correctly on the blockchain.

3.3.2. Cardano

This project, launched in 2017 [29], is also open source. Its blockchain allows the inclusion of metadata in transactions, which implies a lower transaction rate per second due to the higher complexity. This results in 250 TPS at maximum for all users [30]. Cardano uses the Ouroboros consensus algorithm based on PoS, where validator nodes are selected based on the number of tokens they hold. The native token used in this network is ADA, and the smallest unit of measurement is known as Lovelace, which is equivalent to 10^{-6} ADA. This blockchain uses the Unspent Transaction Output (UTxO) scheme, which means that the transaction rate will be deterministic, allowing for more accurate planning.

The official library 'cardano-wallet-js' has been used to carry out the interaction between the log module and the log management server with the block network. This library communicates with a blockchain node via the Cardano Wallet software (v2023-07-18). This node and the Cardano Wallet run in the cloud in conjunction with the log management server, so it plays an active role in the Cardano network by maintaining a copy of the blockchain, participating in validation, and propagating both transactions and blocks. Furthermore, the Cardano Wallet allows secure management of transactions and wallets.

As a major disadvantage, using these three elements implies greater implementation complexity than the Solana network.

The log module has various functions for interacting with the Cardano network. The 'initCardano' function connects to the Preprod network (i.e., testnet), while the 'getBalance' function checks the available ADA amount. Additionally, the 'createWallet' and 'writeLog' functions have been implemented to support backward compatibility with the previous implementation. The log management server uses these three elements to verify if the information has persisted in a block on the blockchain.

4. Experimentation

To assess the feasibility of the secure log storage, this section presents both the procedures implemented in the stress tests conducted and presents the results derived from the analysis of the test data acquired.

The hardware and software components were implemented within a local network to minimize latency in the scenario used, except for the public blockchains. On the one hand, there is a Raspberry Pi 4 Model, which uses a Broadcom BCM2711 Quad core Cortex-A72 (ARM v8) 64-bit SoC 1.5 GHz processor and 4 GB of LPDDR4-3200 SDRAM running the Raspberry Pi OS 5.10. This OS version is a Debian variant optimized for Raspberry Pi. On the other hand, a server with an Intel Core i7-10870H 2.20 GHz processor and 16 GB of DDR4 RAM running the Ubuntu 22.0.4.3 LTS operating system was used to install the Syslog-ng server, the RabbitMQ Message Broker, and the log module. This server supported the log server management, as well as the node and wallet in the Cardano scenario.

It is important to note that in the log module, which communicates directly with the message broker, several tests have been performed depending on the prefetch (i.e., the number of messages in the queue that are consumed in a request). For instance, in case it is set to 1, it will wait until all the processing is finished before consuming the second message. Given the asynchronous nature of NodeJS, when blocking operations (e.g., HTTP requests, disk accesses, etc.) are encountered in the processing of each message, this language could continue with the actions of the second message until the previous one is unblocked, which is a significant improvement in efficiency, as can be seen in Figure 6, an example execution with a prefetch of 3 events. As shown in the example, after retrieving the events from the message broker, the potential requests that could cause blockages are sent to the blockchain and the log management server. However, certain factors beyond our control, such as high congestion in the blockchain or temporary network outages, may impact these requests. In these cases, NodeJS will continue executing other instructions regardless of the blocked flow. The optimal prefetch value will depend on the specific scenario, considering factors such as the number of logs per second being processed or log module instances. Achieving an equilibrium between these variables will provide optimal system performance.

A comprehensive experiment was conducted using Loggen, a high-performance log generation tool designed for stress testing syslog servers. This tool enables benchmarking and load testing by generating and transmitting syslog messages at configurable rates. The UDP protocol was utilized during the tests, and the maximum average message size parameter was set to 8192 bytes. A single instance of the log module was used, with prefetch values of 1, 10, 20, and 50 tested for loads of 1 event/s, 10 events/s, 20 events/s, and 50 events/s on both blockchains. We also evaluated several processing times, such as the processing time within the log module, the time from when the event is generated until it is certified in the blockchain, the time from when the event is generated until it has persisted in the log management server, and the total execution time of each test bench. We performed 10 iterations for each case to average values and used the NodeJS performance API with millisecond level resolution to measure time.

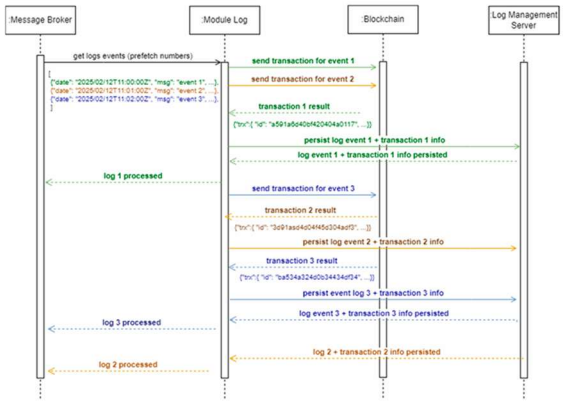


Figure 6. Example of asynchronous execution of messages in the experimentation carried out.

5. Results and Discussion

This section analyzes test results on Solana and Cardano blockchains, comparing average processing times at various event rates. Subsequently, the section examines enterprise log data management, blockchain certification costs, and suggests advanced filtering to reduce expenses by focusing on crucial logs.

5.1. DRACSC Performance Test

The results of the tests on the Solana blockchain indicate that the average time for a request per second is over 12.82 s for 1 event/s, 17.94 s for 10 events/s, 22 s for 20 logs/s, and 31.77 s for 50 events/s, as shown in Figure 7. Table 4 illustrates the average outcomes for 10 iterations and their standard deviation. The "Log Module" column represents the event's duration in the log module service. The "From event to blockchain" column measures the interval from an event's arrival at the Syslog server until its recording on the blockchain. The last column, "From event to Log Management Server" indicates the time taken from the event's arrival until its persistence in the log management system.

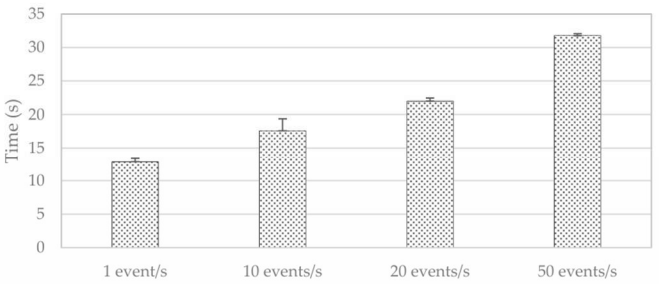


Figure 7. Execution times with the Solana blockchain.

Table 4. Average time for a request in Solana.

Events/Second	Log Module (ms)	From Event to Blockchain (ms)	From Event to Log Management Server (ms)
1	5.45 ± 4.69	12,836.45 ± 293.33	12,824.20 ± 287.71
10	5.99 ± 3.32	1835.59 ± 1192.15	14,810.23 ± 1056.99
20	5.26 ± 4.08	3866.75 ± 2430.11	15,532.33 ± 1234.68
50	5.17 ± 4.41	7546.99 ± 5501.71	18,623.97 ± 10,797.62

The experimental results on Solana indicate that the average processing time for individual requests remains relatively stable across different event rates. Nonetheless, as the event rate increases, there is a noticeable increase in time, especially in the interaction of the event with the blockchain. This suggests that while Solana demonstrates consistency in processing individual requests, the system's scalability and responsiveness may be challenged under higher loads of concurrent events. On the other hand, the results suggest that the system appears well suited for environments that require certification for a moderate number of critical logs; however, further optimization strategies should be considered for scenarios involving increased event volumes. Complementarily, higher hardware resources would be required to support the proposed system if the number of module log instances exceeds this limit due to the limitations of the Raspberry Pi.

The results found with the Cardano blockchain that supports metadata achieved 4.35 s for 1 event/s, 626.55 s for 10 events/s, 1098.45 s for 20 events/s, and 3071.65 s for 50 events/s, as shown in Figure 8. The average results for the 10 iterations and their standard deviation can be seen in Table 5. The experimentation reveals that the log module processing times fluctuate with increasing event rates. The times from event to blockchain and to the log management server experience significant increases, particularly under higher event rates. This suggests that Cardano exhibits longer processing times for individual requests than Solana, and scalability challenges become more evident in scenarios with increased concurrent events. Consequently, Cardano appears viable only in scenarios where certification events occur very infrequently. This is due to two factors: on the one hand, there is a restriction on the number of transactions per second imposed by the Cardano network. On the other hand, Cardano Mempool's deliberate design prevents certain actors from monopolizing the entire blockchain bandwidth by processing transactions in a FIFO queue. In conclusion, optimization measures may need to be implemented to improve overall system responsiveness.

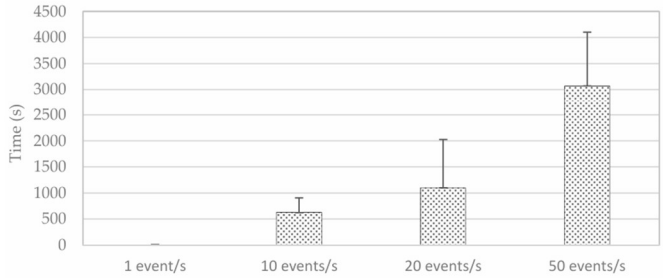


Figure 8. Execution times with the Cardano blockchain.

Table 5. Average time for a request in Cardano.

Events/Second	Log Module (ms)	From Event to Blockchain (ms)	From Event to Log Management Server (ms)
1	16.57 ± 6.48	22,171.70 ± 30,389.99	4357.8 ± 6855.37
10	7.59 ± 4.43	240,995.88 ± 164,862.90	219,455.22 ± 167,407.68
20	7.61 ± 4.58	313,123.15 ± 302,980.38	305,814.52 ± 304,370.36
50	4.95 ± 2.96	298,423.43 ± 246,633.25	274,762.13 ± 245,587.09

5.2. Feasibility in an Enterprise System

The proposed solution is designed to effectively manage the large volume of log data generated within an enterprise system. However, the cost of sending transactions to the blockchain for certification remains a concern. In the case of Cardano, transaction costs are influenced by two components as shown in Figure 9: the transaction fee and the UTxO storage cost [31]. In our study, only one SHA2 hash was stored for each transaction, resulting in a fixed size of 32 KB with a corresponding cost of 200,000 lovelaces. In contrast, transaction costs on the Solana blockchain vary based on network congestion; therefore, they are non-deterministic [32]. These costs are calculated as the sum of a base fee and a priority fee as shown in Figure 10. Costs increase when more compute units (CU) are consumed, reflecting the computational resources used by the transaction on the network. In our study, the average cost was 5001 lamports per transaction.

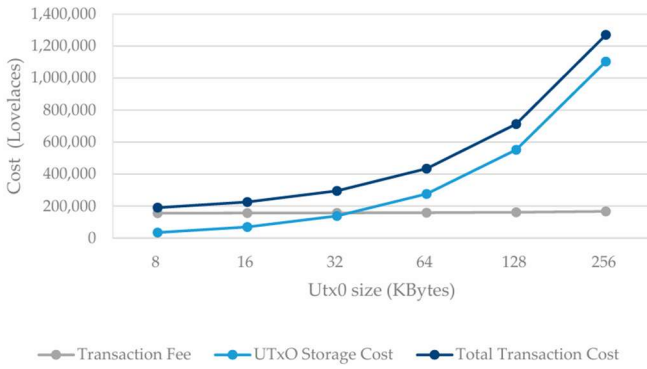


Figure 9. Transaction cost breakdown in Cardano.

The rate of logs generated per second varies for each company, but for the purpose of this discussion, we will consider a rate of one log per second, which implies around 86,400 logs daily [11]. Approximately 10% of these logs are critical, which equates to 8640 logs per day. Through experimentation, we have found that sending all critical logs is not affordable due to cost constraints, specifically 164.16 € per day in Solana and 552.96 € per day in Cardano. To minimize costs, it is advisable to operate during periods of low congestion or reduce the transaction priority, which allows taking advantage of the lowest fees available. To further mitigate this issue, we can employ the advanced filtering functions of Syslog-ng to selectively filter the most crucial logs, thereby reducing costs while ensuring that the most important events are certified.

Although the costs may be substantial, they are justifiable given the critical importance of data integrity, legal compliance, and security, particularly in industries such as finance

and healthcare, where non-repudiation and long-term verifiability are paramount. In the finance sector, immutable log storage plays a crucial role in preventing fraud and adhering to regulatory standards like PCI DSS [33]. The investment in this technology can be rationalized by selectively applying blockchain to high-risk transactions. Likewise, in healthcare, it is essential to uphold the integrity of electronic health records (EHR) and manage access to sensitive medical information in order to comply with regulations like HIPAA [34].

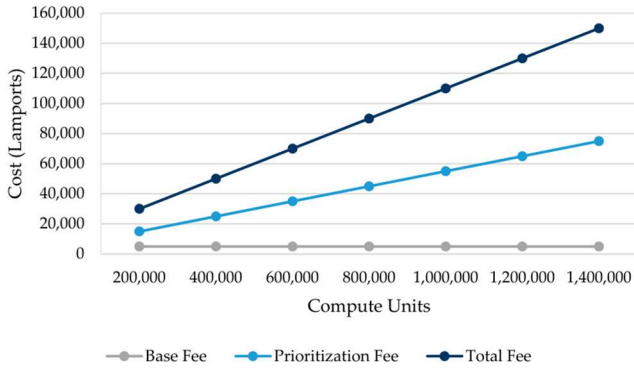


Figure 10. Transaction cost breakdown in Solana.

6. Conclusions

The secure storage of log events is a complex task, becoming increasingly important due to the complete digitalization of production processes. This is related to different factors, such as the growing complexity of IT systems, society digitization, or the use of technology in signature processes. Traditionally, these events produced by computer systems have been used internally as a reference for detecting erroneous behavior or tracing operations. Therefore, it could be challenging to use in a judicial process due to the doubts raised by these data that do not comply with the principles of integrity, immutability, or non-repudiation.

The DRACSC system introduced a new blockchain-based approach to overcome the limitations of traditional log methods used for communication networks. This solution is designed to address common problems in managing communication equipment, such as error recovery, device configuration, and certification of critical log events generated by company network components.

The research produced mixed outcomes during the experimentation phase. On the one hand, the methodology followed fulfilled its intended function of preserving coherence, persistence and non-repudiation of the data. However, considering the inherent characteristics of blockchains, the choice of implementation becomes a crucial factor in achieving satisfactory performance. Solana demonstrated exceptional performance, establishing itself as a viable choice for implementing this solution with favorable times for a moderate number of events and lower transaction costs. Conversely, Cardano's feasibility is somewhat limited, as it exhibits higher response times for this specific use case. This limitation significantly reduces the number of events, resulting in higher transaction costs. However, it is worth noting that Cardano's capability to include metadata in transactions adds substantial value for certifying log events. Therefore, each option caters to opposing scenarios, and this duality should be regarded as an advantageous aspect

of the study. The main limitation of the proposed approach lies in the bandwidth and congestion issues associated with public blockchains. Despite this, the selection of a new network with optimal performance could be compromised in the medium term due to the widespread adoption of blockchain technology. This work also contributes by making the research code accessible as open source, promoting transparency and collaboration within the scientific community. The code will be publicly accessible through the specified repository at https://github.com/jdmorei/seclogs (accessed on 26 February 2025), encouraging collaboration and advancements in the field.

Future work will focus on enhancing security, cost efficiency, and scalability. Regarding security, one of the next developments will be creating a blockchain migration tool due to advancements in classical and quantum computing, which poses risks to existing blockchain protocols. This tool will facilitate secure data and state transfers between blockchains and act as a contingency plan for transitioning to more secure platforms if vulnerabilities arise or if quantum computing threatens current cryptographic methods, thereby ensuring continuity and data integrity.

In parallel with these security enhancements, another goal is to increase the number of messages per second and reduce costs. One potential option would be the use of a layer-2 blockchain such as Cardano Hydra. This network improves Cardano's scalability and performance, reaching 10^6 TPS and introducing the concept of microtransactions at a low cost, making it well-suited for high-frequency payment systems. Hydra achieves this efficiency through State Channels, a technique used in blockchain networks to reduce the number of transactions required for each interaction on the main chain. State Channels are smart contracts that apply predefined rules for managing transactions between parties. Said channels originate from the main blockchain and are merged back into them, verifying their validity through cryptographic methods. The previous features and the ability to add metadata to transactions, thanks to the UTxO schema, are the reasons we will research Cardano Hydra. In addition, other technologies are also being explored, such as DAG (Directed Acyclic Graphs), a mathematical and computational construct predating the blockchain concept with many functional similarities. In our case, we will study the use of IOTA 2.0, which uses the Nakamoko Consensus on a DAG model. This concept brings three main features: parallel writing, where blocks can be validated and added in parallel; on tangle voting, where each validating node has voting power thanks to staking and validation; and approval weight where, in cases of double-spend, the transaction with higher weight will win. These features create a seamless payment network, making it particularly attractive for micropayments and IoT-based transactions. Given the current extensible architecture of the software created for this work, integrating these technologies will require only the development of new connectors. This flexibility ensures that the system remains adaptable to emerging blockchain advancements while maintaining compatibility with existing infrastructures. Therefore, integrating Cardano Hydra and IOTA 2.0 will overcome these limitations by lowering transaction costs and supporting high-frequency transactions with minimal latency.

Finally, we will explore advanced filtering techniques that integrate machine learning (ML) to prioritize critical logs and reduce unnecessary blockchain transactions. Techniques such as anomaly detection and risk-based classification, together with the Syslog-ng filters, could dynamically determine which logs require blockchain certification, optimizing both security and cost-effectiveness.

Author Contributions: Conceptualization, J.D.M.R. and T.J.M.S.; methodology, T.J.M.S.; software, J.D.M.R.; validation, T.J.M.S.; formal analysis, T.J.M.S.; investigation, J.D.M.R. and T.J.M.S.; resources, J.D.M.R. and T.J.M.S.; data curation, J.D.M.R. and T.J.M.S.; writing—original draft preparation, J.D.M.R.; writing—review and editing, T.J.M.S.; visualization, J.D.M.R. and T.J.M.S.; supervision,

T.J.M.S.; project administration, T.J.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
AP	Access Point
BCALS	Blockchain-based Secure Log Management System for Cloud Computing
BFT	Byzantine Fault Tolerance
CU	Compute Unit
DAG	Directed Acyclic Graph
DLT	Distributed Ledger Technology
EHR	Electronic Health Record
FIFO	First In, First Out
GB	Gigabyte
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
IPFS	InterPlanetary File System
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
KB	Kilobyte
LCaaS	Logchain Logistics as a Service
LTS	Long-Term Support
NFT	Non-Fungible Token
ML	Machine Learning
OS	Operating System
PBFT	Practical Byzantine Fault Tolerance
PC	Personal Computer
PCI DSS	Payment Card Industry Data Security Standard
PoH	Proof of History
PoS	Proof of Stake
PoW	Proof of Work
RAM	Random-Access Memory
RFC	Request for Comments
RPC	Remote Procedure Call
SDRAM	Synchronous Dynamic Random-Access Memory
SIEM	Security Information and Event Management
SoC	System on a Chip
TPS	Transactions Per Second
UTxO	Unspent Transaction Output

References

1. Vazão, A.P.; Santos, L.; Costa, R.L.d.C.; Rabadão, C. Implementing and evaluating a GDPR-compliant open-source SIEM solution. *J. Inf. Secur. Appl.* **2023**, *75*, 103509. [CrossRef]
2. Khan, S.; Gani, A.; Wahab, A.W.A.; Bagiwa, M.A.; Shiraz, M.; Khan, S.U.; Buyya, R.; Zomaya, A.Y. Cloud Log Forensics: Foundations, State of the Art, and Future Directions. *ACM Comput. Surv.* **2017**, *49*, 6. [CrossRef]
3. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *NIST SP 800-207: Zero Trust Architecture*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020; p. 10.
4. Silva, R.; Inácio, H.; Marques, R.P. Blockchain implications for auditing: A systematic literature review and bibliometric analysis. *Int. J. Digit. Account. Res.* **2022**, *22*, 163–192. [CrossRef] [PubMed]
5. Bellovin, S.M.; Bush, R. Configuration management and security. *IEEE J. Sel. Areas Commun.* **2009**, *27*, 268–274. [CrossRef]
6. Sanguino, T.M.; González, I.F.V.; Fernández, J.E.; Domínguez, A.G. Using Identity Provider and Automatic Resource Management to Improve a Remote Networking Lab. *IEEE Lat. Am. Trans.* **2018**, *16*, 1547–1556. [CrossRef]
7. Morillo Reina, J.D.; Mateo Sanguino, T.J. Portable Device for Easy Management and Automatic Recovery of Networking Systems. *IEEE Lat. Am. Trans.* **2019**, *17*, 401–408. [CrossRef]
8. Morillo Reina, J.D.; Mateo Sanguino, T.J. Cloud-Based Automatic Configuration and Disaster Recovery of Communication Systems Applied in Engineering Training. *Electronics* **2024**, *13*, 4203. [CrossRef]
9. Russo, D. Benefits of Open Source Software in Defense Environments. In Proceedings of the 4th International Conference in Software Engineering for Defence Applications, Rome, Italy, May 2016; Advances in Intelligent Systems and Computing. 2016; Volume 422.
10. Landauer, M.; Skopik, F.; Wurzenberger, M.; Rauber, A. System log clustering approaches for cyber security applications: A survey. *Comput. Secur.* **2020**, *92*, 101739. [CrossRef]
11. Putz, B.; Menges, F.; Pernul, G. A secure and auditable logging infrastructure based on a permissioned blockchain. *Comput. Secur.* **2019**, *87*, 101602. [CrossRef]
12. Landauer, M.; Skopik, F.; Wurzenberger, M.; Rauber, A. EngraveChain: A Blockchain-Based Tamper-Proof Distributed Log System. *Future Internet* **2021**, *13*, 143. [CrossRef]
13. Sutton, A.; Samavi, R. Blockchain Enabled Privacy Audit Logs. In *The Semantic Web—ISWC 2017*; Springer: Cham, Switzerland, 2017; Volume 10587, pp. 645–660.
14. Ahmad, A. Towards Blockchain-Driven, Secure and Transparent Audit Logs. In Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18), New York, NY, USA, 5–7 November 2018; pp. 443–448.
15. Hsu, C.L.; Chen, W.X.; Le, T.V. An Autonomous Log Storage Management Protocol with Blockchain Mechanism and Access Control for the Internet of Things. *Sensors* **2020**, *20*, 6471. [CrossRef] [PubMed]
16. Ali, A.; Khan, A.; Ahmed, M.; Jeon, G. BCALS: Blockchain-based secure log management system for cloud computing. *Trans. Emerg. Telecommun. Technol.* **2021**, *33*, e4272. [CrossRef]
17. Na, D.; Park, S. IoT-Chain and Monitoring-Chain Using Multilevel Blockchain for IoT Security. *Sensors* **2022**, *22*, 8271. [CrossRef] [PubMed]
18. Pourmajidi, W.; Miransky, A. Logchain: Blockchain-Assisted Log Storage. In Proceedings of the IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018; pp. 978–982.
19. Rakib, M.H.; Hossain, S.; Jahan, M.; Kabir, U. A Blockchain-Enabled Scalable Network Log Management System. *J. Comput. Sci.* **2022**, *18*, 496–508. [CrossRef]
20. Putra, R.A.; Ardiansyah, R.; Pusadan, M.Y.; Kasim, A.A.; Joefrie, Y.Y. Developing Decentralized Data Storage Network Using Blockchain Technology to Prevent Data Alteration. *Adv. Sustain. Sci. Eng. Technol.* **2024**, *6*, 02401017. [CrossRef]
21. Liu, Z.; Zhang, X.; Li, G.; Cui, H.; Wang, J.; Xiao, B. A Secure and Reliable Blockchain-Based Audit Log System. In Proceedings of the IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024; pp. 2010–2015.
22. Dinh, T.T.A.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.L. BLOCKBENCH: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17), Chicago, IL, USA, 14–19 May 2017; pp. 1085–1100.
23. Ćatović, A.; Buzađija, N.; Lemes, S. Microservice development using RabbitMQ message broker. *Sci. Eng. Technol.* **2022**, *2*, 30–37. [CrossRef]
24. Luo, J.; Zhou, B.; Zheng, Y.; Pan, W. *Research on High Performance Web Service Construction Method Based on JavaScript Asynchronous Programming Technique*; Sciendo: Warsaw, Poland, 2024. [CrossRef]
25. Sahni, N. A review on cryptographic hashing algorithms for message authentication. *Int. J. Comput. Appl.* **2015**, *120*, 29–32. [CrossRef]
26. Gerhards, R. *RFC 5424: The Syslog Protocol*; RFC Editor: Marina del Rey, CA, USA, 2009.

27. Hebooks. *The Complete Solana Guide: All You Need to Know About SOL Crypto Before Investing*; Amazon Digital Services LLC—Kdp: Washington, DC, USA, 2023; p. 10.
28. Li, X.; Wang, X.; Kong, T.; Zheng, J.; Luo, M. From Bitcoin to Solana—Innovating Blockchain Towards Enterprise Applications. In *Blockchain—ICBC 2021*; Springer: Cham, Switzerland, 2022; Volume 12991.
29. Houben, R.; Snyers, A. *Cryptocurrencies and Blockchain: Legal Context and Implications for Financial Crime, Money Laundering and Tax Evasion*; European Parliament: Strasbourg, France, 2018; p. 40.
30. Kiayias, A.; Russell, A.; David, B.; Oliynykov, R. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In Proceedings of the 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2017; Volume 10401.
31. Chegenizadeh, M.; Larionov, N.; Niya, S.R.; Yanovich, Y.; Tessone, C.J. Cardano Shared Send Transactions Untangling in Numbers. *Blockchain Res. Appl.* **2025**, *2025*, 100269. [CrossRef]
32. Ashraf, M.; Heavy, C. A prototype of supply chain traceability using Solana as blockchain and IoT. *Procedia Comput. Sci.* **2023**, *217*, 948–959. [CrossRef]
33. King, J.T.; Williams, L.A. Secure Logging and Auditing in Electronic Health Records Systems: What Can We Learn from the Payment Card Industry. In Proceedings of the 3rd USENIX Conference on Health Security and Privacy, Berkeley, CA, USA, 6–7 August 2012.
34. Gaynor, M.; Bass, C.; Duepner, B. A tale of two standards: Strengthening HIPAA security regulations using the PCI-DSS. *Health Syst.* **2015**, *4*, 111–123. [CrossRef]

Capítulo 5. Conclusiones Generales

5.1 Introducción

Este último capítulo tiene como finalidad exponer las principales conclusiones derivadas de la investigación llevada a cabo en la tesis doctoral. Además, se identifican las limitaciones encontradas durante el proceso de desarrollo, las cuales ofrecen un marco para entender los desafíos y restricciones afrontados. Finalmente, se proponen líneas de trabajo futuro con el objetivo de ampliar y mejorar los resultados obtenidos, abriendo nuevas oportunidades para investigaciones adicionales y mejoras en la implementación de las soluciones desarrolladas.

5.2 Conclusiones generales

La administración de equipos de comunicación es una tarea compleja —generalmente relegada a técnicos expertos en redes e Internet— que demanda grandes dotes para el análisis y resolución de problemas con el objetivo de ofrecer el mejor rendimiento y servicio. Mientras la metodología tradicional basada en la administración manual de equipos es poco efectiva debido al tiempo y el coste, mayor cuanto más grande es la infraestructura que gestionar, las soluciones centralizadas para administrar equipos y servicios de red deben proporcionar un alto grado de configuración y funcionalidad para ser realmente útiles.

Con este objetivo, se presentó en el primer artículo un dispositivo portable que automatiza la gestión de configuraciones y SO, así como la recuperación de errores en los equipos que forman una red de comunicaciones. El prototipo —denominado DRACSC— fue implementado mediante una solución de tipo hardware/software mediante Raspberry Pi 2, el cual ha sido probado sobre sistemas embebidos incluyendo conmutadores y enrutadores de los fabricantes Cisco y MikroTik, aunque extensible a otros.

Esta solución es de particular interés en entornos con un número significativo de elementos de red, siendo capaz de actuar sin tener que recordar qué credenciales de acceso corresponden a cada dispositivo o sin requerir conocimientos expertos para realizar varias tareas diferentes en los equipos. Además, aporta otra contribución novedosa como es la creación/edición de funciones de alto nivel basadas en plantillas estructuradas —denominadas MACROS— que permiten clonar configuraciones y desplegarlas en una infraestructura de red. En este sentido, las pruebas de rendimiento han mostrado que cuanto más compleja sea la tarea de administración de enrutadores y conmutadores —contabilizada por el número de sentencias y caracteres a teclear— mayor es la ventaja de usar las funciones

automatizadas del sistema DRACSC, permitiendo clonar configuraciones y desplegarlas en una infraestructura de red en tan solo unos pasos. Así, en el caso extremo de desplegar equipos con configuraciones desde cero, el ahorro de tiempo es considerable. Esto puede apreciarse en los resultados de la experimentación, donde el número de pulsaciones promedio se redujo en un 92,17 ± 4,24%. Ello resultó en un ahorro de tiempo promedio de 68,72 ± 19,58% con valores mínimos de 7s (configuración de Telnet/DNS) y máximos de 160s (recuperación de errores de la imagen del IOS).

En el segundo artículo se presentó una actualización del hardware con Raspberry Pi 4 Model B. Además, se procedió al diseño y creación de una carcasa personalizada para alojar tanto el dispositivo como la batería. En cuanto al software de DRACSC, se han optimizado diversos procesos para mejorar la eficiencia. Además, se ha añadido un analizador sintáctico en la ejecución de las funciones MACROS que permite, mediante palabras claves, la ejecución de comandos del sistema y de acciones determinas. Por ejemplo, este analizador permite ejecutar una señal de retorno de carro en el equipo gestionado o sobrescribir —en tiempo de ejecución en la función MACRO— la palabra reservada de obtención del nombre o contraseña del usuario por los definidos en el DRACSC para ese dispositivo. En consecuencia, este analizador sintáctico representa un nuevo paso en la dirección que posibilite la creación de un lenguaje procedural para la gestión armonizada de equipos de red.

Por otra parte, se ha creado un repositorio de funciones MACRO en la nube con una doble finalidad. Por un lado, permite la actualización del DRACSC con estas funciones a través de la red en sentido bidireccional. Por otro, provee una nueva interfaz de usuario para la creación/edición de funciones más completa e intuitiva, incluyendo la posibilidad de interacción entre usuarios mediante el uso de comentarios y un sistema de puntuación. El principal objetivo de estas funciones es superar las limitaciones encontradas en el prototipo inicial. Esta nueva versión, fue probada sobre sistemas embebidos incluyendo conmutadores y enrutadores de los fabricantes Cisco y MikroTik en distintos escenarios controlados que representan posibles situaciones reales.

En cuanto a la experimentación y evaluación, se han obtenido resultados muy favorables que animan a continuar esta investigación en otros escenarios docentes y profesionales. Para recopilar estos datos, se realizó un cuestionario utilizando una escala Likert de 5 niveles (1 = totalmente en desacuerdo, 5 = totalmente de acuerdo), donde la puntuación media obtenida con estudiantes, profesores y profesionales TIC fue de 3,78 ± 0,88; 3,88 ± 0,67 y 4,09 ± 0,71 en el área de conocimiento/aprendizaje,

lo que indica su utilidad como herramienta educativa. En el bloque de interés / motivación, se obtuvo una calificación de 3,93 ± 0,94; 3,92 ± 0,93 y 4,37 ± 0,69, lo que indica el gran potencial del sistema para ayudar a trabajar con dispositivos gestionados y hacer las tareas más agradables. En cuanto a usabilidad/practicidad, se obtuvo una puntuación de 3,96 ± 0,88; 3,97 ± 0,86 y 4,4 ± 0,63, donde todos los grupos coincidieron en el posible uso del sistema DRACSC en varios campos, ahorrando significativamente tiempo en la realización de diferentes configuraciones y centralizándolas en un único sistema. En el área de resultados/viabilidad, se obtuvo una calificación de 4,28 ± 0,84; 4,27 ± 0,82 y 4,77 ± 0,43, lo que significa que todos los grupos aprueban que esta herramienta tiene potencial en el contexto educativo y profesional.

Otro punto que destacar se encuentra en la ejecución de las tareas, donde se observó que los profesionales obtuvieron mejores resultados en general. Esto puede deberse a que estas personas tienen más experiencia y manejo al configurar equipos, aprovechando mejor esta circunstancia del sistema DRACSC. Sin embargo, también se encontró que algunos estudiantes lograron buenos tiempos, lo que podría estar relacionado con la facilidad del sistema DRACSC para que personas menos experimentadas realicen ciertas configuraciones.

En el tercer artículo, se aborda la problemática del almacenamiento seguro del registro de eventos, una cuestión que adquiere una importancia cada vez mayor debido a la completa digitalización de los procesos de producción. Esto está relacionado con diferentes factores, como la creciente complejidad de los sistemas informáticos, la digitalización de la sociedad o el uso de la tecnología en los procesos de firma. Tradicionalmente, estos eventos producidos por los sistemas informáticos se han utilizado internamente como referencia para detectar comportamientos erróneos o rastrear operaciones. Por lo tanto, podría ser un desafío utilizarlos en un proceso judicial debido a las dudas que plantean estos datos que no cumplen con los principios de integridad, inmutabilidad o no repudio.

Para superar las limitaciones de los métodos tradicionales de registro utilizados en las redes de comunicación, el sistema DRACSC introdujo un nuevo enfoque basado en cadena de bloques. Esta solución está diseñada para abordar problemas comunes en la gestión de equipos de comunicación, como la recuperación de errores, la configuración de dispositivos y la certificación de eventos críticos de registro generados por los componentes de la red en una empresa.

Como parte de los objetivos de esta tesis doctoral, se ha liberado una importante parte del código desarrollado durante la investigación a través de dos repositorios en GitHub. Esta iniciativa no solo contribuye a la transparencia y reproducibilidad de los resultados, sino que también fomenta la colaboración dentro

de la comunidad científica. Al poner el software y los métodos empleados a disposición de cualquier interesado, se facilita la validación de los hallazgos, así como el desarrollo de nuevas propuestas a través de un ecosistema. Esta estrategia de divulgación se alinea con el compromiso de fomentar la ciencia abierta y la transferencia de conocimiento, pilares esenciales para la generación de futuros avances en la disciplina. Dichos repositorios se encuentran accesibles en las URL siguientes: https://github.com/jdmorei/wadc y https://github.com/jdmorei/seclogs. El primero de los enlaces incluye una versión del DRACSC que ha sido publicada como software libre. El segundo enlace contiene el código de la solución empleada en la cadena de bloques descentralizada y segura que registra eventos a prueba de manipulaciones.

5.3 Limitaciones de la tesis doctoral

En cuanto a las limitaciones de la investigación del primer artículo, estas están principalmente relacionadas con la experiencia del usuario respecto al sistema DRACSC. Debido a ello, se estableció la necesidad de crear una nueva interfaz gráfica que permitiera crear, editar o compartir funciones MACRO de forma sencilla e intuitiva. También, la posibilidad de incluir comandos del sistema como palabras reservadas que puedan ser de interés para las tareas de configuración y recuperación. Por último, se aumentaron los recursos del hardware utilizado.

En el segundo artículo se consideró necesario incluir un mayor número de profesores para obtener una muestra representativa. Aunque este grupo es pequeño en comparación con los otros (53 estudiantes, 6 profesores y 30 profesionales), proporciona un punto de referencia para futuras investigaciones. Esto se apoya en el hecho de que las tendencias entre los grupos —y entre diferentes institutos— siguen patrones similares. Para cubrir un mayor número de profesores, sería necesario acceder a un mayor número de institutos, ya que típicamente hay un profesor para cada grupo de estudiantes. En nuestra región, hay una proporción de 1 profesor por cada 20 estudiantes, habiendo disminuido a 1 profesor por cada 10 estudiantes debido a las limitaciones de espacio físico durante la pandemia por COVID-19. Por otro lado, también es posible observar la gran diferencia entre el número de hombres y mujeres en el estudio. En el sector TIC es difícil alcanzar un número paritario debido a la gran diferencia de género que existe en el campo de la ingeniería.

En relación con el tercer artículo, la investigación arrojó resultados mixtos durante la fase de experimentación. Por un lado, la metodología seguida cumplió su función prevista de preservar la coherencia, persistencia y no repudio de los datos. Sin embargo, considerando las características inherentes de las cadenas de datos, la elección de implementación se convierte en un factor crucial para lograr un

rendimiento satisfactorio. Solana demostró un desempeño excepcional, estableciéndose como una opción viable para implementar esta solución con tiempos favorables para un número moderado de eventos y menores costos de transacción. Por el contrario, la viabilidad de Cardano es algo limitada, ya que exhibe tiempos de respuesta más altos para este caso de uso específico. Esta limitación proporciona una reducción significativa en el número de eventos, resultando en mayores costos de transacción. Sin embargo, cabe destacar que la capacidad de Cardano para incluir metadatos en las transacciones añade un valor sustancial para la certificación de eventos de registro. Por lo tanto, cada opción atiende a escenarios opuestos y, esta dualidad, debe considerarse como un aspecto ventajoso del estudio. La principal limitación del enfoque propuesto radica en los problemas de ancho de banda y congestión asociados con las cadenas de bloque públicas. Con Solana, los tiempos de procesamiento aumentan de 12,82 segundos para 1 evento/segundo a 31,77 segundos para 50 eventos/segundo. Con Cardano, el incremento fue aún más drástico, pasando de 4,35 segundos a 3.071,65 segundos para las mismas tasas de eventos.

5.4 Trabajos futuros

Con respecto al futuro, la investigación actual está siendo centrada en la ampliación del repositorio público de MACROS, consiguiendo una comunidad que apoye y posibilite el crecimiento del proyecto. En este sentido —gracias a la estructura software flexible del dispositivo— los esfuerzos están siendo dirigidos a concebir DRACSC como una herramienta universal con opciones mejoradas para la configuración y administración de redes, extendiéndose a diferentes fabricantes y entornos como la nube. Ello incluye utilizar el DRACSC en la WAN, lo que requiere tener un fichero de configuración de red que establezca automáticamente todos los parámetros e implementar medidas de seguridad para gestionar equipos remotos mediante servicios confinados de manera que el entorno quede limitado a ese servicio, evitando que escale accesos.

Conforme al estudio de viabilidad, se abordarán tanto la cuestión de género como el número de profesores para analizar posibles desviaciones del estudio. Por otro lado, los desarrollos actuales se centran en obtener una mayor difusión y aceptación del sistema DRACSC por parte de los usuarios. Para ello, se abordarán actualizaciones para mejorar la usabilidad a través de la interfaz gráfica y gestionar relés a través del GPIO de Raspberry Pi para encender/apagar equipos de forma remota. También desarrollaremos una plataforma en la nube que se comunicará con dispositivos DRACSC para permitir acciones adicionales que requieran un mayor

consumo de recursos y que, debido a las limitaciones de hardware, no podrían llevarse a cabo en la solución proporcionada en esta tesis doctoral. Estas incluyen la administración de múltiples redes, el mantenimiento de una base de datos con historial de configuraciones, la recuperación de errores y la reparación automática de elementos de red mediante reglas definidas a través de conocimientos expertos.

Con respecto al almacenamiento seguro, los esfuerzos se enfocarán en reforzar la seguridad, la reducción de costes operativos y mejorar la escalabilidad. En cuanto a la seguridad, uno de los próximos desarrollos trata sobre la creación de una herramienta de migración entre distintas cadenas de bloques. El objetivo es facilitar la transferencia segura de datos en caso de contingencia hacia plataformas más seguras cuando existan vulnerabilidades o la computación cuántica amenace los métodos criptográficos actuales. Ello permitirá garantizar la continuidad e integridad de los datos.

En paralelo con las mejoras en seguridad, otro objetivo es aumentar el número de mensajes por segundo y reducir los costes. Una opción potencial sería el uso de una cadena de bloques de capa 2 como Cardano Hydra. Esta red mejora la escalabilidad y el rendimiento de Cardano, alcanzando 10^6 TPS, además de introducir el concepto de microtransacciones a bajo coste. Hydra logra esta eficiencia a través de State Channels, una técnica utilizada en redes de cadenas de bloques para reducir el número de transacciones requeridas para cada interacción en la cadena principal. Los canales de estado son contratos inteligentes que aplican reglas predefinidas para gestionar transacciones entre las partes. Dichos canales se originan en la cadena de bloques principal y se fusionan de nuevo en ella, verificando su validez mediante métodos criptográficos.

Por otro lado, se están explorando otras tecnologías como DAG (Grafos Acíclicos Dirigidos), una construcción matemática y computacional que precede al concepto de cadena de bloques con muchas similitudes funcionales. Para ello, se estudiará el uso de IOTA 2.0, que utiliza el Consenso de Nakamoto en un modelo DAG. Este concepto aporta tres características principales: *i*) escritura paralela, donde los bloques pueden validarse y añadirse en paralelo; *ii*) votación en la red de nodos o tangle, donde cada nodo validador tiene poder de voto gracias a la participación y validación; y *iii*) peso de aprobación que, en caso de doble gasto, la transacción con mayor peso ganará. Otra característica interesante es que permite el uso de UTxO como en el caso de Cardano, concepto que se refiere a una transacción de salida no gastada.

De forma adicional para el apartado de seguridad y rentabilidad, se explorarán de técnicas avanzadas de filtrado que integren aprendizaje automático (ML) para priorizar registros críticos y reducir transacciones innecesarias en la cadena de bloques. Técnicas como la detección de anomalías y la clasificación basada en riesgos, junto con filtros Syslog-ng, podrían determinar dinámicamente qué registros requieren certificación en blockchain.

Por último, otro trabajo futuro consiste en integrar un sistema de RAG (Generación Aumentada por Recuperación) acoplado a un LLM (Modelo de Lenguaje de Gran Tamaño), habilitando la comunicación con el DRACSC mediante lenguaje natural con el fin de obtener información exhaustiva de la red. Este sistema estaría alimentado por los datos de configuración de los dispositivos gestionados por el DRACSC, así como por información recopilada a través del protocolo SNMP (Simple Network Management Protocol). La implementación de esta solución mediante IA permitiría a los administradores interactuar de manera más intuitiva con la infraestructura de red, sin necesidad de realizar comandos específicos o usar lenguajes de programación. Ello facilitaría aún más la gestión de los sistemas en entornos de red complejos y dinámicos.

LISTA DE ACRÓNIMOS

ABS	Acrylonitrile Butadiene Styrene
ACID	Atomicity, Consistency, Isolation, Durability
AJAX	Asynchronous JavaScript and XML
AMQP	Advanced Message Queuing Protocol
AP	Access Point
API	Application Programming Interface
ARM	Advanced RISC Machines
BFT	Byzantine Fault Tolerance
BSON	Binary JSON
CLI	Command-Line Interface
CPD	Data Center
CPU	Central Processing Unit
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DAG	Directed Acyclic Graph
DHCP	Dynamic Host Configuration Protocol
DLT	Distributed Ledger Technology
DOI	Digital Object Identifier
DOM	Document Object Model
DRACSC	Dispositivo de Recuperación Automática y Configuración de Sistemas de Comunicación
DRAM	Dynamic Random Access Memory
FTP	File Transfer Protocol
GB	Gigabyte
GPIO	General Purpose Input/Output
GPL	General Public License
HIPAA	Health Insurance Portability and Accountability Act
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IEE	Institute of Electrical and Electronics Engineers

IoT	Internet of Things
IPFS	InterPlanetary File System
JCR	Journal Content Repository
JSON	JavaScript Object Notation
LED	Light Emitting Diode
LLM	Large Language Model
MB	Megabyte
ML	Machine Learning
MVC	Model-View-Controller
NPM	Node Package Manager
ORM	Object-Relational Mapping
PBFT	Practical Byzantine Fault Tolerance
PCI	Payment Card Industry
PCT	Patent Cooperation Treaty
PHP	Hypertext Preprocessor
PoW	Proof of Work
RAG	Retrieval Augmented Generation
RFC	Request for Comments
RoR	Ruby on Rails
SASS	Syntactically Awesome Stylesheets
SD	Secure Digital
SDRAM	Synchronous Dynamic Random Access Memory
SIEM	Security Information and Event Management
SNMP	Simple Network Management Protocol
SO	Sistema Operativo
SOHO	Small Office/Home Office
SOX	Sarbarnes-Oxley Act
SSH	Secure Shell
SSL	Secure Sockets Layer
TFT	Thin-Film Transistor
TFTP	Trivial File Transfer Protocol
TIC	Tecnologías de la Información y la Comunicación
UTx0	Unspent Transaction Output
VPN	Virtual Private Network
WAN	Wide Area Network
XML	eXtensible Markup Language

REFERENCIAS

[1] Morillo Reina J.D., Mateo Sanguino T.J., “Portable Device for Easy Management and Automatic Recovery of Networking Systems”, IEEE Latin America Transactions, vol. 17(03), pp. 401-408, 2019.

[2] Morillo Reina J.D., Mateo Sanguino T.J., “Cloud-Based Automatic Configuration and Disaster Recovery of Communication Systems Applied in Engineering Training”, Electronics 13, no. 21: 4203, 2024.

[3] Morillo Reina J. D., Mateo Sanguino T. J., “Decentralized and Secure Blockchain Solution for Tamper-Proof Logging Events”, Future Internet, vol. 17, no. 3, p. 108, 2025.

[4] K. Elmansor, “Towards Automated Network Configuration Management”, Master’s Thesis, College of Computing and Digital Media, 2013.

[5] Cisco Networking Academy, Scaling Networks Companion Guide, Cisco Press, 2014.

[6] It Digital Media Group, “Las Claves de una Adecuada WLAN en la Empresa”, It - User Tech & Business, pp. 33-38, 2016.

[7] K.R. Elangovan, “Security Framework for Supply-Chain Management”. Information Resources Management Association (USA), pp. 604, 2021

[8] EMA, “Essential IT Monitoring: Seven Priorities for Network Management”, Tech. Rep., 2013. [Online] http://content.solarwinds.com/

[9] ATEN Altusem, “SN0100CO/SN0100COD/SN9100CO Series Serial Console Server User Manual”, Tech. Rep., 2021. [Online] https://assets.aten.com

[10] M.H. Siregar, “Network Monitoring Sistem Menggunakan Whatsup Gold Pada Pt”, Pembangunan Jaya Ancol, Tbk, 2021.

[11] Perle Systems Ltd., “IOLAN SDS/SCS/STS User’s Guide”, Tech. Rep., 2018. [Online] https://www.perle.com

[12] Raritan, Inc., “Dominion SX II User Guide 2.0.0”, Tech. Rep., 2015. [Online] http://support.raritan.com

[13] Raritan, Inc., “Opengear User Manual 4.12.0”, Tech. Rep., 2021. [Online] http://ftp.opengear.com/download/manual/current/Opengear%20User%20Manual.pdf

[14] Belunix, “Raspisco — remote access to Cisco through Raspberry Pi”, Tech. Rep., 2013. [Online] http://developers-club.com/posts/192188/

[15] D. Kyuchukova, G. Hristov, P. Zahariev and S. Borisov, “A study on the possibility to use Raspberry Pi as a console server for remote access to devices in virtual learning environments”, IEEE 2015 International Conference on Information Technology Based Higher Education and Training (ITHET), 2015.
[16] S. Nelson, “The Internet of Getting Things Done”. New Electronics, vol. 51, no. 13, 2023.
[17] Vazão A.P. et al., “Implementing and evaluating a GDPR-compliant open-source SIEM solution”, Journal of Information Security and Applications, vol. 75, 2023.
[18] Putz B., et al., “A secure and auditable logging infrastructure based on a permissioned blockchain”, Computers & Security, vol. 87, 2019.
[19] Shekhtman, L. and Waisbard, E. “EngraveChain: A Blockchain-Based Tamper-Proof Distributed Log System”, Future Internet, vol. 13, pp.143, 2021
[20] Sutton, A., Samavi, R, “Blockchain Enabled Privacy Audit Logs”, The Semantic Web – ISWC 2017, vol 10587, pp. 645-660, 2017.
[21] Ahmad A. et al., “Towards Blockchain-Driven, Secure and Transparent Audit Logs”, 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18), pp. 443-448, 2018.
[22] Hsu C.-L., et al., “An Autonomous Log Storage Management Protocol with Blockchain Mechanism and Access Control for the Internet of Things”, Sensors, vol. 20, no. 22, p. 6471, 2020.
[23] Ali A., et al., “BCALS: Blockchain-based secure log management system for cloud computing”, Transactions on Emerging Telecommunications Technologies, vol. 33, 2021.
[24] Na D. and Park S., “IoT-Chain and Monitoring-Chain Using Multilevel Blockchain for IoT Security”, Sensors, vol. 22, no. 21, p. 8271, 2022.
[25] Pourmajidi W. and Miransky A., “Logchain: Blockchain-Assisted Log Storage”, IEEE 11th International Conference on Cloud Computing (CLOUD), pp. 978-982, 2018.
[26] Rakib M. H., et al., “A Blockchain-Enabled Scalable Network Log Management System”, Journal of Computer Science, vol. 18, no. 6, pp. 496-508, 2022.
[27] Putra R. A., et al., “Developing Decentralized Data Storage Network Using Blockchain Technology to Prevent Data Alteration,” Advance Sustainable Science, Engineering and Technology, vol. 6, no. 1, 02401017, 2024.

[28] Liu Z., et al., “A Secure and Reliable Blockchain-Based Audit Log System”, IEEE International Conference on Communications, pp. 2010-2015, 2024.