

# **Diseño de una arquitectura de software escalable basada en computación en la nube para la integración de RedssApp con proveedores y clientes**

**Jesús Madariaga Escobar<sup>1</sup>, Edna Esteban Regino<sup>1</sup>, Margarita Cogollo Esteban<sup>2</sup>**

**<sup>1</sup>Universidad Cooperativa de Colombia. Montería, Colombia**

**<sup>2</sup>Universidad de Córdoba. Montería, Colombia**

## **Resumen**

El sector social y solidario en Colombia enfrenta dificultades debido a la falta de información centralizada y accesible sobre las organizaciones de Economía Solidaria, lo que limita su visibilidad y crecimiento y a su vez la optimización de sus cadenas de suministro, lo que hace necesario incorporar herramientas digitales que permitan mapear y validar experiencias de asociatividad solidaria, como también la gestión efectiva de la Cadena de Suministro 4.0. Para resolver la necesidad, se planteó como objetivo del estudio diseñar una arquitectura de software basada en computación en la nube, que contribuyera al desarrollo de una aplicación móvil que facilitara la integración de proveedores y clientes, permitiendo la interoperabilidad y garantizando un flujo de información seguro, eficiente y en tiempo real.

Para ello, se aplicó una metodología basada en la propuesta del estudio “Sistema IoT para el monitoreo de variables climatológicas en cultivos de agricultura urbana”, adaptada para la construcción de una arquitectura de software en computación en la nube. Esta metodología está estructurada en cuatro fases, pero en este estudio se implementaron dos fases, que comprendieron la selección de herramientas y tecnologías y el diseño de la arquitectura del sistema. Este enfoque permitió desarrollar una solución escalable y adaptable a las necesidades del sector solidario.

En la Fase 1, se identificaron herramientas de software y hardware que aseguraron la interoperabilidad del sistema en la nube, priorizando plataformas de código abierto y estándares de comunicación eficientes para la gestión y procesamiento de datos en tiempo real. En la Fase 2, se diseñó una arquitectura modular en la nube, integrada con sistemas empresariales existentes y organizada en tres capas: almacenamiento, para la gestión segura y accesible de datos; análisis, para la optimización de la toma de decisiones en la Cadena de Suministro 4.0; y visualización, para el acceso intuitivo y en tiempo real a la información, facilitando la interacción entre los actores

involucrados. Además, se seleccionaron herramientas de Google Cloud para garantizar la interoperabilidad, escalabilidad y seguridad del sistema. La arquitectura incluyó Google Firebase, Google Analytics, Maps SDK e Identity Toolkit API, optimizando el almacenamiento, procesamiento y análisis de datos en tiempo real. Firebase ha permitido gestionar datos de forma estructurada y segura, Google Analytics facilita el análisis del comportamiento de los usuarios, Maps SDK mejora la visualización de ubicaciones y Identity Toolkit API garantiza la autenticación y seguridad en el acceso.

Se cuenta con una arquitectura en la nube con las características necesarias para iniciar su implementación y continuar con pruebas que validen su funcionalidad. Esto permitirá considerar a RedssApp como una solución efectiva para la gestión de la Cadena de Suministro 4.0, facilitando la conexión entre proveedores y clientes del sector solidario. A través de este enfoque, se ha logrado desarrollar una plataforma escalable y segura, capaz de adaptarse a los desafíos del entorno digital y fortalecer la integración y colaboración entre los actores involucrados.

**Palabras clave:** cadena de suministro 4.0; proveedores y clientes; computación en la nube; arquitectura de software

### **Abstract**

*The social and solidarity sector in Colombia faces difficulties due to the lack of centralized and accessible information on solidarity economy organizations, which limits their visibility and growth, and in turn, the optimization of their supply chains, which makes it necessary to incorporate digital tools that allow mapping and validation of experiences of solidarity associativity, as well as the effective management of the Supply Chain 4.0. To solve this need, the objective of the study was to design a software architecture based on cloud computing, which would contribute to the development of a mobile application that would facilitate the integration of suppliers and customers, allowing interoperability and ensuring a secure, efficient and real-time flow of information.*

*For this purpose, a methodology based on the proposal of the study "IoT system for monitoring climatological variables in urban agricultural crops" was applied, adapted to the construction of a software architecture in cloud computing. This methodology is structured in four phases, but in this study two phases were implemented, including the selection of tools and technologies and the design of the system architecture. This approach allowed the development of a scalable and adaptable solution to the needs of the solidarity sector.*

*Phase 1 identified software and hardware tools to ensure interoperability of the cloud system, prioritizing open source platforms and efficient communication standards for real-time data management and processing. In Phase 2, a modular cloud architecture was designed that integrates with existing enterprise systems and is organized into three tiers: storage, for secure and accessible data management; analytics, for optimizing Supply Chain 4.0 decision making; and visualization, for intuitive, real-time access to information that facilitates stakeholder interaction. In addition, Google Cloud tools were selected to ensure interoperability, scalability, and security of the system. The architecture included Google Firebase, Google Analytics, Maps SDK and Identity Toolkit API to*

*optimize data storage, processing and analysis in real-time. Firebase has enabled structured and secure data management, Google Analytics facilitates the analysis of user behavior, Maps SDK improves the visualization of locations, and Identity Toolkit API ensures authentication and access security.*

*We have an architecture in the cloud with the necessary characteristics to start its implementation and continue with tests to validate its functionality. This allows us to consider RedssApp as an effective solution for the management of Supply Chain 4.0, facilitating the connection between suppliers and customers in the solidarity sector. Through this approach, it has been possible to develop a scalable and secure platform, capable of adapting to the challenges of the digital environment and strengthening the integration and collaboration between the actors involved.*

**Keywords:** *supply chain 4.0; providers and costumers; cloud computing; software architecture*

## 1. Introducción

El sector social y solidario en Colombia enfrenta la carencia de herramientas digitales que permitan mapear y validar experiencias de asociatividad, así como gestionar con eficacia la cadena de suministro 4.0. La asociatividad solidaria comprende las prácticas colaborativas entre organizaciones y comunidades para potenciar habilidades mutuas, mientras que la cadena de suministro 4.0 incorpora servicios en la nube, inteligencia artificial y análisis de datos en tiempo real para mejorar la trazabilidad, reducir fallos y agilizar procesos (Guo et al., 2023; Thomas, 2020). La dispersión de la información entre proveedores, entidades y usuarios limita la coordinación, reduce la visibilidad de los actores y dificulta la innovación (Tiwari et al., 2024).

RedssApp surge como una plataforma basada en la nube que centraliza la información y facilita la interacción entre proveedores y clientes del sector solidario. La solución integra almacenamiento en tiempo real, autenticación segura, análisis de comportamiento y geolocalización para garantizar disponibilidad continua, protección de los datos y respuestas ágiles (Khawas & Shah, 2018; Koutsoukos et al., 2021; Krämer & Senner, 2015).

Este estudio presenta el diseño de una arquitectura de software escalable y robusta basada en servicios en la nube, orientada a integrar RedssApp con proveedores y clientes. Se describen las fases de selección tecnológica y la implementación distribuida en los niveles de presentación, lógica y datos, junto con los beneficios esperados en términos de interoperabilidad, desempeño y crecimiento sostenible del ecosistema solidario colombiano (Mesa et al., 2020; Hannousse & Yahiouche, 2021).

## 2. Antecedentes

El desarrollo de una arquitectura de software apoyada en computación en la nube para conectar RedssApp con proveedores y clientes responde al objetivo de aumentar la eficiencia operativa en el sector (Shee et al., 2018). El uso de herramientas como Google Cloud, Firebase y Google



Analytics asegura que la solución ofrezca escalabilidad, protección e interoperabilidad. Así, se anticipa que esta arquitectura fomente la cooperación entre los participantes de la cadena de suministro y eleve la competitividad del sector social y solidario en Colombia (Giannakis et al., 2019).

El estudio de Han et al. (2021) permitió establecer criterios técnicos para digitalizar cadenas de suministro en contextos con baja integración, lo cual resulta pertinente para el entorno solidario. Sus planteamientos sobre organización de datos y servicios de análisis influyeron directamente en la selección de Firebase como sistema de gestión de información y en el uso de Google Analytics para obtener reportes dinámicos sobre el comportamiento de los usuarios. Estos aportes facilitaron el desarrollo de una capa analítica que contribuye a mejorar la toma de decisiones.

Por otro lado, la investigación de Surucu-Balci et al. (2024) proporcionó lineamientos sobre cómo superar barreras como los costos de implementación y la protección de datos. Con base en ese análisis, se adoptaron mecanismos de autenticación y control de acceso mediante Identity Toolkit API, lo cual refuerza la seguridad de la plataforma. Además, su enfoque sobre visibilidad e integración orientó la estructura modular de RedssApp, diseñada para adaptarse a las necesidades de distintos actores dentro del ecosistema solidario.

Finalmente, la metodología propuesta por Chanchí-Golondrino et al. (2022) sirvió como guía para estructurar las fases del proyecto. Aunque dicha metodología fue diseñada para un sistema de monitoreo en agricultura urbana, sus principios orientados por capas y su enfoque práctico fueron adaptados con éxito al contexto de la computación en la nube. Esta base metodológica permitió organizar la arquitectura en niveles funcionales claros, facilitando la integración de servicios y asegurando un flujo de información eficiente entre proveedores y clientes.

### 3. Metodología

En el desarrollo del presente artículo se han definido dos fases metodológicas que se encuentran presentadas en la ilustración 1, esta metodología fue basada en el artículo de (Chanchí-Golondrino et al., 2022). Se llevó a cabo una selección de herramientas tecnológicas junto con el diseño de una arquitectura de sistema, cuyo objetivo fue conectar de manera efectiva a proveedores y clientes a través de la aplicación RedssApp.



Ilustración 1: Estructura metodológica tomada de (Chanchí-Golondrino et al., 2022)

Durante la fase 1 se investigó un amplio conjunto de herramientas y tecnologías de software que permitieron capturar, almacenar y visualizar datos de manera eficiente dentro del sistema.

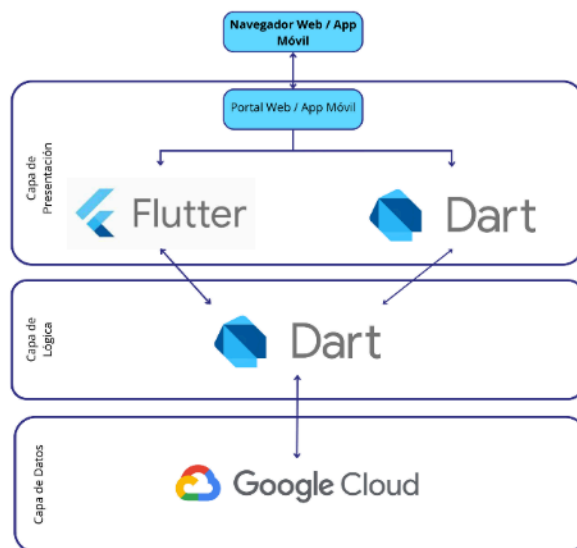


Ilustración 2: Tecnologías seleccionadas para arquitectura de software

En la capa de presentación se emplearon las tecnologías Flutter y Dart para desarrollar interfaces de usuario consistentes y adaptables en diversas plataformas, como aplicaciones móviles y portales web. Flutter permitió una experiencia visual unificada mediante el uso de componentes reutilizables y un mismo lenguaje de diseño, mientras que Dart posibilitó la interacción dinámica y el manejo eficiente de eventos, asegurando así una comunicación fluida y moderna entre el usuario y la aplicación.

En la capa lógica de la aplicación se gestionó mediante Dart, lo cual centralizó la validación de datos, el control de flujos de trabajo y la ejecución de las reglas de negocio. Esta organización permitió coordinar de manera coherente las distintas funcionalidades y módulos internos, asegurando una operación fluida y consistente en el procesamiento de la información.

En la capa de datos se utilizó Google Cloud para establecer una infraestructura robusta y escalable. Esta plataforma posibilitó el almacenamiento seguro de los datos y la gestión eficiente de la información en tiempo real, integrando servicios adicionales como notificaciones, respaldos y mecanismos de alta disponibilidad. De esta forma, se garantizó el rendimiento y la continuidad operativa de la aplicación.

Esta estructura en capas permitió que la solución tecnológica se apoyara en bases modernas y confiables, facilitando el desarrollo, la operación y la escalabilidad de la aplicación.

Durante la fase 2, y a partir de las tecnologías seleccionadas en la fase 1, se especificaron tanto la vista funcional como la de implementación de la arquitectura del sistema RedssApp, diseñada para conectar proveedores con clientes.

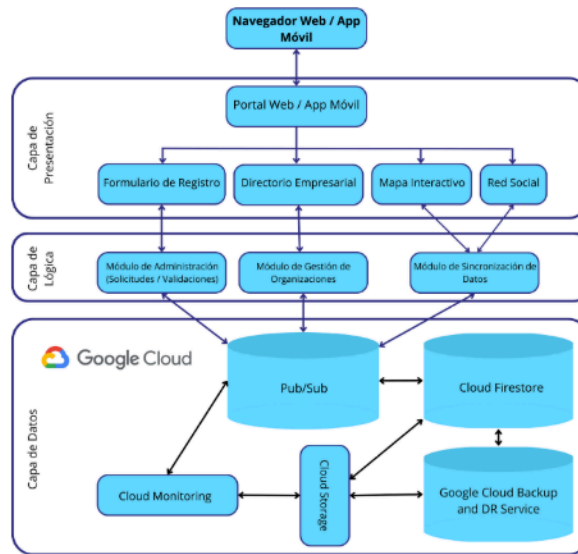


Ilustración 3: Arquitectura del sistema

Esta arquitectura se estructuró en tres capas principales: presentación, lógica y datos, las cuales se articularon de manera coordinada para garantizar el funcionamiento integral de la plataforma.

**Capa de presentación:** En esta capa se definieron los componentes visuales y de interacción con el usuario, como el formulario de registro, el directorio empresarial, el mapa interactivo y la red social, accesibles desde un portal web o aplicación móvil. Estos módulos permitieron a los usuarios explorar la plataforma, registrarse, ubicar empresas y establecer conexiones dentro de una interfaz centralizada.

**Capa lógica:** En esta capa se organizaron los módulos funcionales responsables de gestionar las operaciones del sistema. Se incluyeron componentes como el módulo de administración, que gestionó las solicitudes y validaciones; el módulo de gestión de organizaciones, encargado del manejo de perfiles empresariales; y el módulo de sincronización de datos, que permitió la comunicación efectiva entre los distintos elementos de la plataforma, garantizando la coherencia de la información visualizada.

**Capa de datos:** En esta capa se implementaron servicios en la nube mediante herramientas de Google Cloud. Se utilizaron tecnologías como Pub/Sub para la transmisión de eventos entre servicios, Cloud Firestore para el almacenamiento principal de datos, y Cloud Storage para la gestión de archivos y recursos multimedia. Además, se integraron mecanismos de respaldo y recuperación mediante Google Cloud Backup and DR Service, así como monitoreo continuo del sistema a través de Cloud Monitoring, asegurando así la disponibilidad, seguridad y rendimiento de la plataforma.

La ilustración 3 representa esta arquitectura mediante un diagrama funcional que muestra la interacción entre las distintas capas, evidenciando cómo se coordinan para permitir el registro,

búsqueda, comunicación y gestión de información entre proveedores y clientes a través de RedssApp.

## 4. Resultados

Para validar el diseño se recrearon cargas concurrentes de registro y consulta con picos de demanda para evaluar latencia, coherencia, escalabilidad y uso de recursos. Se verificó la fiabilidad de respaldos y procesos de recuperación, comprobando tiempos de restauración breves, y se activaron alertas automáticas para captar anomalías al instante. En conjunto, estas pruebas confirmaron que la arquitectura ofrece interoperabilidad, seguridad y respuesta inmediata, a la vez que mantiene la plataforma estable y controla los costos operativos.

**Pub/Sub:** En la “ilustración N°4: Métricas de Pub/Sub” se evidencia cómo RedssApp monitoriza el rendimiento de su canal de eventos a través de indicadores clave como `ack_latency` y `backlog_bytes`. A través de dashboards que muestran latencias de confirmación cercanas a 1 ms y un backlog en cero, se comprueba la entrega inmediata de mensajes y la ausencia de acumulación de eventos pendientes. Conforme a los principios de arquitecturas orientadas a eventos, esta visibilidad garantiza un flujo de datos continuo y eficiente, permitiendo a la plataforma escalar dinámicamente y responder en tiempo real a las interacciones de proveedores y clientes.

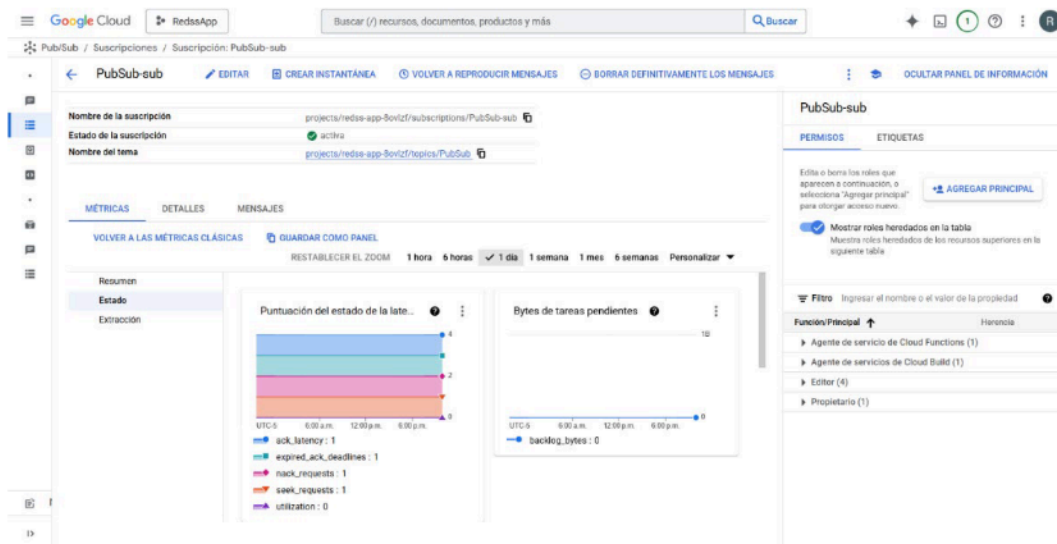


Ilustración 4: Pub/Sub (Captura propia tomada desde la consola de administración de Google Cloud Platform)

**Cloud Firestore:** En la “ilustración N°5: Telemetría Cloud Firestore” se evidencia cómo RedssApp mantiene la sincronización de sus datos en tiempo real mediante conexiones activas y snapshot listeners. A través de los gráficos de “Clientes conectados” y “Objetos de escucha de instantáneas”, se aprecia que cada modificación de perfil, directorio o interacción se réplica al instante en todas las instancias web y móviles. Conforme al teorema CAP, este mecanismo prioriza la consistencia y la tolerancia a particiones, garantizando que la experiencia de usuario sea coherente incluso ante caídas o variaciones en la red.

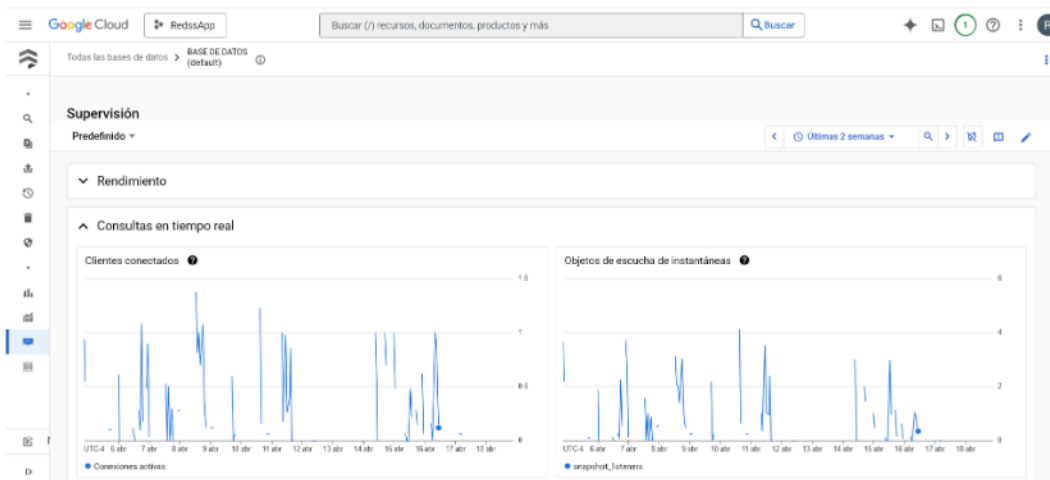


Ilustración 5: Telemetría Cloud Firestore (Captura propia tomada desde la consola de administración de Google Cloud Platform)

**Cloud Storage:** En la “ilustración N°6: Estructura Cloud Storage” se evidencia cómo RedssApp organiza sus recursos multimedia en buckets estructurados por carpetas y aplica políticas de ciclo de vida para migrar automáticamente objetos según su antigüedad y frecuencia de acceso. Mediante la visualización de la jerarquía de directorios y las clases de almacenamiento asignadas, se comprueba la optimización de costos al desplazar contenido poco consultado a niveles de menor precio, sin sacrificar la disponibilidad rápida de los archivos más solicitados.

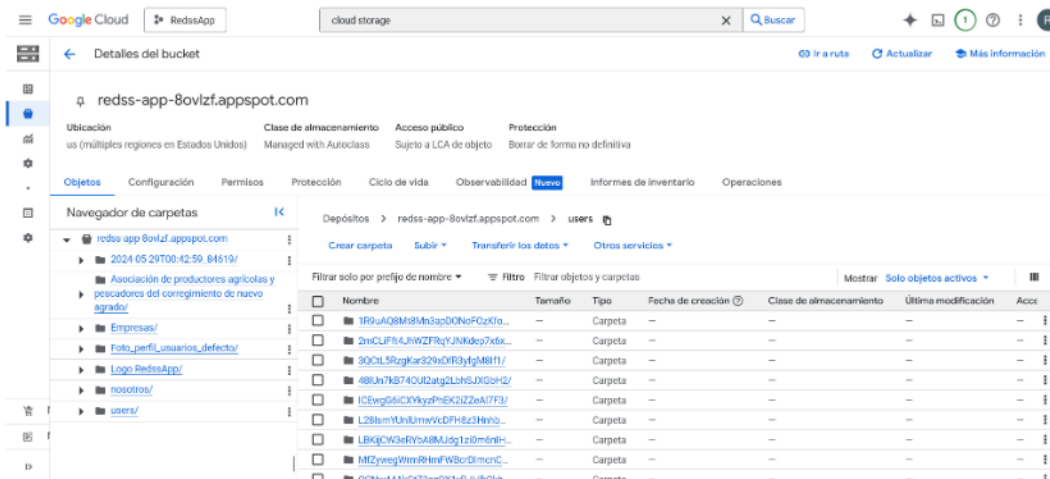
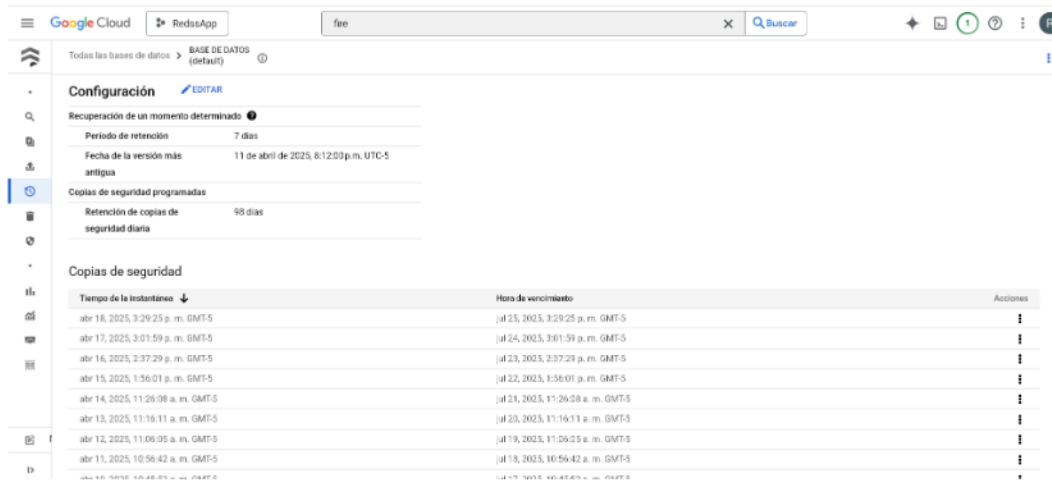


Ilustración 6: Estructura Cloud Storage (Captura propia tomada desde la consola de administración de Google Cloud Platform)

**Google Cloud Backup and DR Service:** En la “ilustración N°7: Google Cloud Backup and DR Service” se evidencia la configuración de las copias de seguridad incrementales y los periodos de retención establecidos para Firestore y Cloud Storage. Mediante la lista de snapshots programados y el “Período de retención de copias de seguridad diaria”, se constata la capacidad de restaurar el sistema en otra región o proyecto en cuestión de minutos. De acuerdo con las



mejores prácticas de continuidad de negocio, esta estrategia minimiza el tiempo de inactividad y asegura la integridad de la información ante cualquier contingencia.



The screenshot shows the Google Cloud Backup and DR Service interface. The left sidebar contains navigation options like 'Configuración', 'Copia de seguridad', and 'Copia de seguridad programada'. The main content area is titled 'Configuración' and shows settings for a backup policy. Below this, there is a table of scheduled backups.

Tempo de la instantánea ↓	Hora de vencimiento	Acciones
abr 18, 2025, 3:29:25 p. m. GMT-5	jul 25, 2025, 3:29:25 p. m. GMT-5	⋮
abr 17, 2025, 3:01:59 p. m. GMT-5	jul 24, 2025, 3:01:59 p. m. GMT-5	⋮
abr 16, 2025, 2:37:29 p. m. GMT-5	jul 23, 2025, 2:37:29 p. m. GMT-5	⋮
abr 15, 2025, 1:56:01 p. m. GMT-5	jul 22, 2025, 1:56:01 p. m. GMT-5	⋮
abr 14, 2025, 11:26:08 a. m. GMT-5	jul 21, 2025, 11:26:08 a. m. GMT-5	⋮
abr 13, 2025, 11:16:11 a. m. GMT-5	jul 20, 2025, 11:16:11 a. m. GMT-5	⋮
abr 12, 2025, 11:06:05 a. m. GMT-5	jul 19, 2025, 11:06:05 a. m. GMT-5	⋮
abr 11, 2025, 10:56:42 a. m. GMT-5	jul 18, 2025, 10:56:42 a. m. GMT-5	⋮

Ilustración 7: Google Cloud Backup and DR Service (Captura propia tomada desde la consola de administración de Google Cloud Platform)

**Cloud Monitoring:** En la “ilustración N°8: Telemetría Cloud Monitoring” se evidencia cómo RedssApp recopila métricas y eventos del sistema para su análisis en tiempo real. A través de dashboards y alertas automatizadas, lo que permite identificar comportamientos inusuales, como aumentos en la latencia. Conforme al enfoque de observabilidad, esta herramienta proporciona una visión completa del estado de los servicios, permitiendo actuar de forma proactiva ante cualquier incidente y asegurar la estabilidad continua de la plataforma.

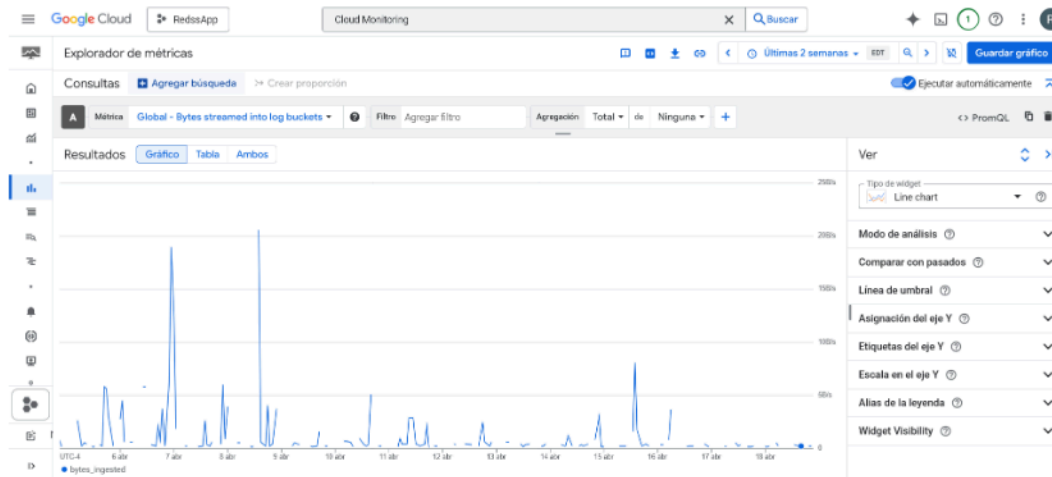


Ilustración 8: Telemetría Cloud Monitoring (Captura propia tomada desde la consola de administración de Google Cloud Platform)

Con estos datos, se concluye que la arquitectura de RedssApp demuestra un rendimiento óptimo en latencia, coherencia de datos y escalabilidad, cumpliendo con los requerimientos de interoperabilidad y respuesta en tiempo real; al mismo tiempo, sus mecanismos de respaldo y

monitoreo garantizan resiliencia operativa y control de costos, reforzando la seguridad de la plataforma.

## 5. Discusión

La arquitectura en la nube de RedssApp consiguió conectar de manera fluida a proveedores y clientes del sector solidario en Colombia, garantizando que distintas aplicaciones pudieran comunicarse, crecer sin problemas y mantener la información protegida. Las pruebas de uso que midieron el tiempo de envío de mensajes, la fiabilidad de los datos y la capacidad de soportar picos de actividad demostraron que el sistema interno envía cada mensaje al instante, que la base de datos en la nube actualiza la información en todas las pantallas al momento y que las alertas avisan de cualquier fallo en tiempo real. Estos hallazgos coinciden con la idea de Shee et al. (2018) sobre cómo la integración digital mejora la eficiencia operativa y con la observación de Giannakis et al. (2019) acerca del impulso competitivo que ofrece la nube. La división del proyecto en niveles para crear la arquitectura de software, basada en la propuesta de Chanchí Golondrino et al. (2022), facilitó la creación de dicha arquitectura y mantuvo un flujo de información claro.

Para llevar a cabo la digitalización, se siguieron los criterios de Han et al. (2021) para entornos con poca conexión entre sistemas, optando por una base de datos organizada y por herramientas de análisis que generan informes en vivo para facilitar la toma de decisiones. Al mismo tiempo, el estudio de Surucu Balci et al. (2024) aportó estrategias para sortear barreras de costos y privacidad, lo que condujo a implementar un sistema simple de inicio de sesión seguro y un diseño flexible que fomenta la colaboración entre todos los participantes. Junto con el almacenamiento inteligente de archivos y las copias de respaldo periódicas, RedssApp se consolida como una solución de Cadena de Suministro 4.0 capaz de adaptarse a futuros desafíos.

## 6. Conclusiones

RedssApp demuestra cómo la adopción de tecnologías de la Industria 4.0, como IoT, inteligencia artificial y computación en la nube, optimiza la gestión de la cadena de suministro en el sector social y solidario colombiano. Mediante la centralización de datos dispersos y la facilitación de la interacción entre proveedores, organizaciones comunitarias y consumidores, garantiza latencias mínimas, coherencia de la información y tiempos de respuesta inmediatos. Las pruebas técnicas validaron la arquitectura en la nube basada en Google Cloud, Firebase y Google Analytics, confirmando su escalabilidad, seguridad e interoperabilidad según estándares globales.

La incorporación de mecanismos de respaldo de datos y monitoreo continuo asegura resiliencia operativa y sostenibilidad a largo plazo. El énfasis en usabilidad y accesibilidad permite que usuarios con limitada experiencia tecnológica participen plenamente en el ecosistema digital, promoviendo inclusión y generación de valor social. Estas características consolidan a RedssApp como un modelo replicable en contextos emergentes y fomentan la innovación sostenible en la cadena de suministro.

## 7. Referencias

- Giannakis, M., Spanaki, K., & Dubey, R. (2019). A cloud-based supply chain management system: effects on supply chain responsiveness. *Journal of Enterprise Information Management*, 32(4). <https://doi.org/10.1108/JEIM-05-2018-0106>
- Guo, C., Ke, Y., & Zhang, J. (2023). Digital transformation along the supply chain. *Pacific Basin Finance Journal*, 80. <https://doi.org/10.1016/j.pacfin.2023.102088>
- Han, L., Hou, H., Bi, Z. M., Yang, J., & Zheng, X. (2021). Functional Requirements and Supply Chain Digitalization in Industry 4.0. *Information Systems Frontiers*. <https://doi.org/10.1007/s10796-021-10173-1>
- Hannousse, A., & Yahiouche, S. (2021). Securing microservices and microservice architectures: A systematic mapping study. In *Computer Science Review* (Vol. 41). <https://doi.org/10.1016/j.cosrev.2021.100415>
- Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A Study. *International Journal of Computer Applications*, 179(46). <https://doi.org/10.5120/ijca2018917200>
- Koutsoukos, D., Müller, I., Marroquín, R., Klimovic, A., & Alonso, G. (2021). Modularis: Modular relational analytics over heterogeneous distributed platforms. *Proceedings of the VLDB Endowment*, 14(13). <https://doi.org/10.14778/3484224.3484229>
- Krämer, M., & Senner, J. (2015). A modular software architecture for processing of big geospatial data in the cloud. *Computers and Graphics* (Pergamon), 49. <https://doi.org/10.1016/j.cag.2015.02.005>
- Mesa, J. A., Esparragoza, I., & Maury, H. (2020). Modular architecture principles–MAPs: a key factor in the development of sustainable open architecture products. *International Journal of Sustainable Engineering*, 13(2). <https://doi.org/10.1080/19397038.2019.1634157>
- Shee, H., Miah, S. J., Fairfield, L., & Pujawan, N. (2018). The impact of cloud-enabled process integration on supply chain performance and firm sustainability: the moderating role of top management. *Supply Chain Management*, 23(6). <https://doi.org/10.1108/SCM-09-2017-0309>
- Surucu-Balci, E., Iris, Ç., & Balci, G. (2024). Digital information in maritime supply chains with blockchain and cloud platforms: Supply chain capabilities, barriers, and research opportunities. *Technological Forecasting and Social Change*, 198. <https://doi.org/10.1016/j.techfore.2023.122978>
- Tiwari, M. K., Bidanda, B., Geunes, J., Fernandes, K., & Dolgui, A. (2024). Supply chain digitisation and management. In *International Journal of Production Research* (Vol. 62, Issue 8). <https://doi.org/10.1080/00207543.2024.2316476>

## Sobre los autores

- **Jesús Madariaga Escobar:** Estudiante de Ingeniería de Sistemas de Universidad Cooperativa de Colombia. [jesus.madariagaes@campusucc.edu.co](mailto:jesus.madariagaes@campusucc.edu.co)
- **Edna Esteban Regino:** Ingeniera de Sistemas, Especialista en Ingeniería de Software, Magíster en Tecnología Educativa, Aspirante a Doctor en Economía y Empresa de UNIR. Profesora Asistente de Universidad Cooperativa de Colombia. [edna.estebanr@campusucc.edu.co](mailto:edna.estebanr@campusucc.edu.co)
- **Margarita Cogollo Esteban:** Estudiante de Licenciatura en Lenguas Extranjeras con énfasis en Inglés de Universidad de Córdoba. [mcogolloesteban93@correo.unicordoba.edu.co](mailto:mcogolloesteban93@correo.unicordoba.edu.co)

Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2025 Asociación Colombiana de Facultades de Ingeniería (ACOFI)