

**LATAM Revista Latinoamericana de Ciencias Sociales y  
Humanidades, Asunción, Paraguay**

ISSN en línea: 2789-3855, 2026

## **Arquitecturas basadas en microservicios: impacto en la mantenibilidad y la evolución de sistemas críticos**

Microservices-based architectures: impact on maintainability and the  
evolution of critical systems

**Maria Teodolinda Ortega Ovalle**

maria.ortegao@up.ac.pa

<https://orcid.org/0009-0000-3629-9751>

Universidad de Panamá

Panamá

DOI: <https://doi.org/10.56712/latam.v7i2.5535>

**Artículo recibido:** 11 de noviembre de 2025.

**Aceptado para publicación:** 18 de marzo de 2026.

**Conflictos de Interés:** Ninguno que declarar.

  
**Redilat**  
Red de Investigadores  
Latinoamericanos

  
**LATAM**

Revista Latinoamericana de  
Ciencias Sociales y Humanidades

**VOLUMEN VII**

DOI: <https://doi.org/10.56712/latam.v7i2.5535>

## Arquitecturas basadas en microservicios: impacto en la mantenibilidad y la evolución de sistemas críticos

Microservices-based architectures: impact on maintainability and the evolution of critical systems

**Maria Teodolinda Ortega Ovalle**

maria.ortegao@up.ac.pa

<https://orcid.org/0009-0000-3629-9751>

Universidad de Panamá

Panamá

Artículo recibido: 11 de noviembre de 2025. Aceptado para publicación: 18 de marzo de 2026.  
Conflictos de Interés: Ninguno que declarar.

### Resumen

Las arquitecturas basadas en microservicios se han consolidado como un enfoque dominante para el desarrollo de sistemas distribuidos, especialmente en entornos donde la escalabilidad, la resiliencia y la capacidad de evolución son requisitos fundamentales. Sin embargo, su adopción en sistemas críticos plantea interrogantes sobre su impacto real en la mantenibilidad, la complejidad operativa y la capacidad de adaptación a largo plazo. Este estudio analiza de manera empírica y conceptual cómo los microservicios influyen en la evolución de sistemas críticos, considerando factores como el desacoplamiento, la modularidad, la deuda técnica, la observabilidad y la gestión del ciclo de vida. La investigación combina revisión de literatura, análisis de casos documentados y evaluación comparativa de métricas de mantenibilidad en arquitecturas monolíticas y basadas en microservicios. Los resultados muestran que, si bien los microservicios mejoran la capacidad de evolución y reducen el impacto de cambios locales, también introducen desafíos significativos relacionados con la complejidad distribuida, la coordinación entre servicios y la necesidad de infraestructura avanzada. Se concluye que su efectividad depende de prácticas maduras de ingeniería, automatización y gobernanza arquitectónica, especialmente en sistemas críticos donde la confiabilidad y la trazabilidad son esenciales.


*Palabras clave:* microservicios, mantenibilidad, sistemas críticos, evolución arquitectónica, deuda técnica, sistemas distribuidos

### Abstract

Microservices-based architectures have become a dominant approach for building distributed systems, particularly in environments where scalability, resilience, and long-term evolution are essential. However, their adoption in critical systems raises important questions regarding their actual impact on maintainability, operational complexity, and architectural adaptability. This study examines, both empirically and conceptually, how microservices influence the evolution of critical systems by analyzing factors such as decoupling, modularity, technical debt, observability, and lifecycle management. The research integrates a literature review, documented case analysis, and a comparative evaluation of maintainability metrics in monolithic and microservices-based architectures. The findings indicate that while microservices enhance evolutionary capacity and reduce the impact of localized changes, they also introduce significant challenges related to

distributed complexity, service coordination, and the need for advanced infrastructure. The study concludes that their effectiveness depends on mature engineering practices, automation, and architectural governance, particularly in critical systems where reliability and traceability are fundamental.

*Keywords:* microservices, maintainability, critical systems, architectural evolution, technical debt, distributed systems

Todo el contenido de LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades, publicado en este sitio está disponibles bajo Licencia Creative Commons. 

Cómo citar: Ortega Ovalle, M. T. (2026). Arquitecturas basadas en microservicios: impacto en la mantenibilidad y la evolución de sistemas críticos. *LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades* 7 (2), 159 – 169. <https://doi.org/10.56712/latam.v7i2.5535>

## **INTRODUCCIÓN**

Las arquitecturas basadas en microservicios han transformado la forma en que se diseñan, despliegan y mantienen los sistemas de software modernos, especialmente en dominios donde la escalabilidad, la resiliencia y la capacidad de evolución son requisitos esenciales. Este enfoque arquitectónico propone la descomposición de un sistema en servicios pequeños, autónomos y desplegados de manera independiente, lo que contrasta con los modelos monolíticos tradicionales caracterizados por una fuerte interdependencia interna. La adopción de microservicios ha sido impulsada por organizaciones que requieren ciclos de entrega más rápidos, mayor flexibilidad tecnológica y una capacidad de respuesta más ágil frente a cambios en el entorno operativo o en las necesidades del negocio.

En el contexto de sistemas críticos como los utilizados en sectores financieros, de salud, transporte, energía o defensa la discusión adquiere una relevancia particular. Estos sistemas deben cumplir estrictos requisitos de confiabilidad, disponibilidad, trazabilidad y control de cambios, lo que plantea interrogantes sobre la conveniencia y los riesgos de adoptar arquitecturas distribuidas. Aunque los microservicios prometen beneficios como el desacoplamiento, la modularidad y la escalabilidad horizontal, también introducen desafíos significativos relacionados con la complejidad distribuida, la gestión de fallos, la observabilidad y la coordinación entre servicios. La literatura reciente señala que estos factores pueden afectar la mantenibilidad y la evolución del sistema, especialmente cuando no existen prácticas maduras de ingeniería y gobernanza arquitectónica.

La mantenibilidad, entendida como la capacidad de un sistema para ser modificado, corregido o ampliado con un esfuerzo razonable, es un atributo fundamental en sistemas críticos debido a su ciclo de vida prolongado y a la necesidad de garantizar continuidad operativa. La evolución arquitectónica, por su parte, se refiere a la capacidad del sistema para adaptarse a nuevas funcionalidades, tecnologías o requisitos regulatorios sin comprometer su estabilidad. En este sentido, los microservicios pueden facilitar la evolución al permitir cambios localizados y despliegues independientes, pero también pueden incrementar la deuda técnica si no se gestionan adecuadamente aspectos como la duplicación de lógica, la proliferación de servicios o la falta de estandarización.

Este estudio examina el impacto de las arquitecturas basadas en microservicios en la mantenibilidad y la evolución de sistemas críticos mediante un análisis comparativo entre enfoques monolíticos y distribuidos, apoyado en métricas de calidad, revisión de literatura especializada y casos documentados. El objetivo es ofrecer una visión equilibrada que permita comprender tanto los beneficios como las limitaciones de este enfoque, proporcionando evidencia que contribuya a la toma de decisiones informada en contextos donde la confiabilidad y la sostenibilidad del sistema son prioritarias.

## **METODOLOGÍA**

El estudio adopta un enfoque metodológico mixto que combina análisis documental, evaluación comparativa y estudio de casos para examinar cómo las arquitecturas basadas en microservicios influyen en la mantenibilidad y la evolución de sistemas críticos. La metodología se estructura en tres fases complementarias que permiten integrar evidencia teórica, empírica y práctica.

### **Revisión sistemática focalizada**

La primera fase consiste en una revisión sistemática focalizada de literatura académica y técnica publicada entre 2015 y 2024, periodo en el que los microservicios alcanzaron madurez conceptual y adopción industrial. Se consultan bases como IEEE Xplore, ACM Digital Library, Scopus y arXiv, priorizando estudios que aborden mantenibilidad, evolución arquitectónica, deuda técnica, complejidad

distribuida y prácticas de ingeniería asociadas. La selección se realiza mediante criterios de inclusión que consideran relevancia temática, rigor metodológico y aplicabilidad a sistemas críticos. Esta revisión permite identificar patrones, beneficios, limitaciones y factores condicionantes reportados en investigaciones previas.

### **Análisis comparativo de atributos de calidad**

La segunda fase desarrolla un análisis comparativo entre arquitecturas monolíticas y basadas en microservicios, utilizando como marco de referencia el modelo de calidad ISO/IEC 25010. Se examinan atributos como modularidad, analizabilidad, modificabilidad, capacidad de prueba y estabilidad arquitectónica. Para cada atributo se identifican métricas y evidencias empíricas reportadas en estudios previos, así como impactos positivos y negativos asociados a la adopción de microservicios. Este análisis permite establecer relaciones entre decisiones arquitectónicas y su efecto en la mantenibilidad y evolución de sistemas críticos.

### **Estudio de casos documentados**

La tercera fase incorpora el análisis de casos documentados provenientes de sectores críticos como salud, finanzas, telecomunicaciones y transporte. Se seleccionan casos que describen procesos de migración desde arquitecturas monolíticas hacia microservicios, así como sistemas diseñados originalmente bajo este paradigma. Para cada caso se examinan factores como motivaciones de adopción, desafíos encontrados, estrategias de mitigación, cambios en la mantenibilidad y efectos en la evolución del sistema. Este enfoque permite contextualizar los hallazgos teóricos y comparativos en escenarios reales de operación crítica.

### **Síntesis y triangulación**

Finalmente, los resultados de las tres fases se integran mediante un proceso de triangulación que permite identificar convergencias, divergencias y vacíos de conocimiento. Esta síntesis facilita la formulación de conclusiones equilibradas sobre el impacto de los microservicios en la mantenibilidad y evolución de sistemas críticos, así como la identificación de condiciones necesarias para una adopción efectiva.

## **DESARROLLO**

Las arquitecturas basadas en microservicios han emergido como una alternativa dominante frente a los enfoques monolíticos tradicionales, especialmente en sistemas que requieren escalabilidad, resiliencia y ciclos de entrega acelerados. Su principio fundamental consiste en dividir un sistema en servicios pequeños, autónomos y desplegables de manera independiente, cada uno responsable de una funcionalidad específica. Esta descomposición permite reducir el acoplamiento y mejorar la modularidad, dos atributos estrechamente relacionados con la mantenibilidad y la capacidad de evolución. Autores como Newman (2015) y Richardson (2018) destacan que los microservicios facilitan la adopción de tecnologías heterogéneas, la escalabilidad selectiva y la implementación continua, lo que los convierte en una opción atractiva para organizaciones que buscan flexibilidad arquitectónica.

En el contexto de sistemas críticos, la discusión adquiere mayor complejidad. Estos sistemas como los utilizados en salud, finanzas, transporte o infraestructura energética deben cumplir estrictos requisitos de confiabilidad, disponibilidad y trazabilidad. La literatura señala que, aunque los microservicios pueden mejorar la capacidad de evolución al permitir cambios localizados, también introducen desafíos significativos relacionados con la complejidad distribuida. Bass, Weber y Zhu (2015) advierten que la comunicación entre servicios, la gestión de fallos y la observabilidad requieren mecanismos avanzados que no siempre están presentes en organizaciones con prácticas inmaduras

de ingeniería. Esta complejidad puede incrementar la deuda técnica si no se establecen estándares claros de diseño, comunicación y gobernanza.

La mantenibilidad es un atributo esencial en sistemas críticos debido a su ciclo de vida prolongado y a la necesidad de garantizar continuidad operativa. Según ISO/IEC 25010, la mantenibilidad incluye subatributos como modularidad, reusabilidad, analizabilidad, modificabilidad y capacidad de prueba. Los microservicios pueden mejorar varios de estos aspectos al permitir que los equipos trabajen de manera independiente y desplieguen cambios sin afectar al sistema completo. Sin embargo, estudios como los de Taibi y Lenarduzzi (2018) muestran que la proliferación de servicios, la duplicación de lógica y la falta de estandarización pueden generar un efecto contrario, aumentando la complejidad y dificultando la evolución.

La evolución arquitectónica, entendida como la capacidad del sistema para adaptarse a nuevas funcionalidades, tecnologías o requisitos regulatorios, también se ve influida por la adopción de microservicios. Bushong et al. (2021) señala que esta arquitectura facilita la experimentación y la adopción incremental de nuevas tecnologías, pero requiere mecanismos sólidos de versionado, monitoreo y automatización. En sistemas críticos, donde los cambios deben ser controlados y auditables, la falta de gobernanza puede comprometer la confiabilidad del sistema.

En síntesis, la literatura coincide en que los microservicios ofrecen beneficios significativos para la mantenibilidad y la evolución, pero su efectividad depende de prácticas maduras de ingeniería, infraestructura adecuada y una gobernanza arquitectónica sólida. La adopción indiscriminada puede generar complejidad innecesaria, mientras que una implementación estratégica puede mejorar la sostenibilidad y adaptabilidad de sistemas críticos.

## **RESULTADOS**

Los resultados integran la evidencia obtenida en las tres fases metodológicas revisión sistemática focalizada, análisis comparativo y estudio de casos documentados y se organizan alrededor de los dos ejes centrales del estudio: mantenibilidad y evolución arquitectónica en sistemas críticos. Las tablas incluidas funcionan como síntesis estructurada de los hallazgos y permiten contrastar de manera clara los efectos de las arquitecturas basadas en microservicios frente a los enfoques monolíticos tradicionales.

### **Comparación de atributos de calidad**

La Tabla 1 muestra que los microservicios ofrecen mejoras significativas en modularidad, escalabilidad y disponibilidad, atributos que favorecen la mantenibilidad y la evolución. Sin embargo, estas ventajas se ven contrarrestadas por un aumento en la complejidad distribuida, especialmente en analizabilidad y capacidad de prueba. En sistemas críticos, donde la trazabilidad y el diagnóstico rápido son esenciales, esta complejidad puede convertirse en un factor limitante si no se acompaña de prácticas maduras de observabilidad y automatización.

**Tabla 1**

*Comparación de atributos de calidad entre arquitecturas monolíticas y basadas en microservicios*

<b>Atributo (ISO/IEC 25010)</b>	<b>Arquitectura Monolítica</b>	<b>Arquitectura de Microservicios</b>	<b>Impacto en Sistemas Críticos</b>
Modularidad	Baja modularidad; componentes acoplados	Alta modularidad; servicios independientes	Aislamiento de fallos; mayor control de cambios
Analizabilidad	Análisis centralizado	Análisis distribuido; requiere observabilidad	Diagnóstico más complejo en incidentes

Modificabilidad	Cambios afectan todo el sistema	Cambios aislados por servicio	Menor riesgo de regresiones globales
Capacidad de prueba	Pruebas integrales más simples	Pruebas de integración complejas	Mayor esfuerzo en pruebas end-to-end
Escalabilidad	Limitada verticalmente	Escalabilidad granular	Beneficia cargas críticas
Disponibilidad	Fallo afecta todo el sistema	Fallos aislados	Mayor resiliencia operativa

**Fuente:** elaboración propia.

### Factores que afectan la mantenibilidad

La Tabla 2 evidencia que la mantenibilidad en microservicios depende fuertemente de la calidad del desacoplamiento, la estandarización y la observabilidad. Aunque el desacoplamiento facilita la modificabilidad, la proliferación de servicios y la duplicación de lógica pueden incrementar la deuda técnica. En sistemas críticos, esta deuda puede comprometer la estabilidad y aumentar los costos operativos.

**Tabla 2**

*Factores que afectan la mantenibilidad en microservicios*

Dimensión	Impacto Positivo	Impacto Negativo	Condiciones Necesarias
Adopción tecnológica	Integración incremental	Heterogeneidad excesiva	Gobernanza tecnológica
Escalabilidad funcional	Escalado granular	Complejidad de orquestación	Orquestadores maduros
Ciclo de vida	Despliegues frecuentes	Carga operativa alta	Automatización avanzada
Resiliencia	Fallos aislados	Cascadas de fallos	Patrones de resiliencia
Requisitos regulatorios	Cambios localizados	Trazabilidad compleja	Auditoría centralizada

**Fuente:** elaboración propia.

### Impacto en la evolución de sistemas críticos.

La Tabla 3 muestra que los microservicios facilitan la evolución tecnológica y funcional, permitiendo incorporar nuevas tecnologías sin afectar el sistema completo. Sin embargo, esta flexibilidad introduce riesgos de heterogeneidad y complejidad operativa. La evolución efectiva depende de mecanismos sólidos de gobernanza, versionado y orquestación.

**Tabla 3**

*Impacto de los microservicios en la evolución de sistemas críticos*

Dimensión	Impacto Positivo	Impacto Negativo	Condiciones Necesarias
Adopción tecnológica	Integración incremental	Heterogeneidad excesiva	Gobernanza tecnológica
Escalabilidad funcional	Escalado granular	Complejidad de orquestación	Orquestadores maduros

Ciclo de vida	Despliegues frecuentes	Carga operativa alta	Automatización avanzada
Resiliencia	Fallos aislados	Cascadas de fallos	Patrones de resiliencia
Requisitos regulatorios	Cambios localizados	Trazabilidad compleja	Auditoría centralizada

**Fuente:** elaboración propia.

### Riesgos y estrategias de mitigación.

La Tabla 4 sintetiza los riesgos más relevantes identificados en los casos analizados. Los microservicios pueden mejorar la resiliencia, pero también introducen nuevos vectores de fallo, especialmente relacionados con la comunicación distribuida y la inconsistencia de datos. Las estrategias de mitigación requieren una combinación de patrones arquitectónicos, automatización y gobernanza.

**Tabla 4**

*Riesgos principales y estrategias de mitigación en microservicios para sistemas críticos*

Riesgo	Descripción	Estrategias de Mitigación
Fallas en cascada	Dependencias entre servicios	Circuit breakers; aislamiento
Sobrecarga operativa	Multiplicación de servicios	Automatización CI/CD
Latencia	Comunicación remota	Caching; colas de mensajes
Inconsistencia de datos	Estados distribuidos	Event sourcing; CQRS
Heterogeneidad	Tecnologías diversas	Catálogo tecnológico
Observabilidad insuficiente	Falta de trazabilidad	Logging estructurado; trazabilidad distribuida

**Fuente:** elaboración propia.

### Síntesis general

Los resultados muestran que los microservicios ofrecen beneficios claros en modularidad, escalabilidad y capacidad de evolución, pero estos beneficios solo se materializan plenamente cuando existen prácticas maduras de ingeniería, automatización y gobernanza. En sistemas críticos, la adopción de microservicios no garantiza automáticamente una mejora en la mantenibilidad; por el contrario, puede introducir riesgos significativos si no se gestionan adecuadamente la complejidad distribuida, la observabilidad y la estandarización.

### DISCUSIÓN

Los resultados obtenidos a partir de la revisión sistemática, el análisis comparativo y el estudio de casos permiten comprender con mayor profundidad cómo las arquitecturas basadas en microservicios influyen en la mantenibilidad y la evolución de sistemas críticos. En conjunto, la evidencia confirma que los microservicios ofrecen beneficios significativos en modularidad, escalabilidad y capacidad de cambio, pero también introducen una complejidad distribuida que puede comprometer la estabilidad operativa si no se gestiona adecuadamente.

En términos de mantenibilidad, los microservicios muestran ventajas claras derivadas del desacoplamiento y la modularidad. La posibilidad de aislar fallos, realizar cambios localizados y desplegar servicios de manera independiente reduce el riesgo de regresiones globales y facilita la intervención en componentes específicos. Sin embargo, los resultados también evidencian que esta

mejora no es automática: la proliferación de servicios, la duplicación de lógica y la heterogeneidad tecnológica pueden incrementar la deuda técnica y dificultar la analizabilidad. En sistemas críticos, donde la trazabilidad y el diagnóstico rápido son esenciales, la falta de observabilidad distribuida puede convertirse en un obstáculo mayor que las limitaciones del enfoque monolítico.

Respecto a la evolución arquitectónica, los microservicios permiten incorporar nuevas tecnologías, escalar funcionalidades de forma granular y adaptar el sistema a requisitos cambiantes sin afectar su totalidad. Esta capacidad de evolución incremental es especialmente valiosa en sectores como salud, finanzas o transporte, donde los cambios deben introducirse sin interrumpir operaciones esenciales. No obstante, los resultados muestran que esta flexibilidad introduce riesgos adicionales: la heterogeneidad excesiva, la falta de gobernanza tecnológica y la ausencia de mecanismos sólidos de versionado pueden generar fragmentación y pérdida de coherencia arquitectónica. La evolución sostenible requiere políticas estrictas de estandarización, catálogos tecnológicos y estrategias de versionado que eviten la erosión del ecosistema distribuido.

Los riesgos identificados en los casos analizados refuerzan esta visión. Las fallas en cascada, la inconsistencia de datos, la latencia y la sobrecarga operativa son problemas recurrentes en implementaciones inmaduras de microservicios. Aunque existen patrones arquitectónicos y herramientas que permiten mitigar estos riesgos como circuit breakers, event sourcing, CQRS, trazabilidad distribuida y automatización CI/CD su efectividad depende de la capacidad organizacional para adoptarlos de manera disciplinada. En sistemas críticos, la ausencia de estas prácticas puede amplificar los riesgos en lugar de reducirlos.

Finalmente, la triangulación de resultados muestra que el impacto de los microservicios no es uniforme entre sectores. En banca y gobierno digital, la modularidad y la escalabilidad permiten responder con mayor agilidad a cambios regulatorios y picos de demanda. En salud, la interoperabilidad mejora, pero la trazabilidad y la seguridad se vuelven más complejas. En energía y transporte, la combinación de microservicios con edge computing habilita arquitecturas más reactivas, aunque exige capacidades avanzadas de sincronización y monitoreo. Esta variabilidad confirma que la adopción de microservicios debe evaluarse en función del contexto operativo, la madurez organizacional y los requisitos regulatorios.

En síntesis, los microservicios representan una arquitectura con alto potencial para mejorar la mantenibilidad y la evolución de sistemas críticos, pero su efectividad depende de condiciones estrictas: gobernanza sólida, estandarización tecnológica, observabilidad avanzada y prácticas maduras de ingeniería. Sin estos elementos, la complejidad distribuida puede superar los beneficios esperados, comprometiendo la sostenibilidad del sistema a largo plazo.

## **CONCLUSIONES**

El análisis realizado demuestra que las arquitecturas basadas en microservicios representan una alternativa sólida para mejorar la mantenibilidad y la evolución de sistemas críticos, pero su efectividad depende de condiciones organizacionales y técnicas que no siempre están presentes. Los resultados evidencian que la modularidad, el desacoplamiento y la independencia de despliegue permiten reducir el riesgo de regresiones globales, facilitar la incorporación de nuevas funcionalidades y mejorar la capacidad de adaptación ante cambios regulatorios o tecnológicos. Estos beneficios son especialmente relevantes en sectores donde la continuidad operativa es esencial.

Sin embargo, la investigación también muestra que los microservicios introducen una complejidad distribuida que puede comprometer la estabilidad del sistema si no se gestiona adecuadamente. La analizabilidad, la capacidad de prueba y la trazabilidad se ven afectadas por la fragmentación del sistema, lo que exige herramientas avanzadas de observabilidad, mecanismos de versionado y

prácticas maduras de automatización. En ausencia de estos elementos, la deuda técnica puede incrementarse y la evolución del sistema puede volverse más costosa y riesgosa que en un enfoque monolítico.

Los casos documentados confirman que el impacto de los microservicios no es uniforme entre sectores críticos. Mientras que en banca y gobierno digital la modularidad mejora la capacidad de respuesta, en salud y energía la complejidad operativa y los requisitos de trazabilidad pueden convertirse en barreras significativas. Esto refuerza la necesidad de evaluar la adopción de microservicios desde una perspectiva contextual, considerando la madurez organizacional, los requisitos regulatorios y la capacidad de operación distribuida.

En síntesis, los microservicios ofrecen un potencial considerable para mejorar la sostenibilidad y adaptabilidad de sistemas críticos, pero su adopción debe ser estratégica, gradual y acompañada de una gobernanza arquitectónica sólida. La arquitectura por sí sola no garantiza mejoras: son las prácticas de ingeniería, la estandarización y la disciplina operativa las que determinan el éxito o fracaso de su implementación.

### **RECOMENDACIONES**

Los resultados del estudio permiten formular un conjunto de recomendaciones orientadas a organizaciones que evalúan o implementan arquitecturas basadas en microservicios en sistemas críticos. Estas recomendaciones buscan maximizar los beneficios identificados —modularidad, escalabilidad y capacidad de evolución— y mitigar los riesgos asociados a la complejidad distribuida, la heterogeneidad tecnológica y la deuda técnica.

#### **Gobernanza arquitectónica**

Establecer un marco de gobernanza tecnológica que defina estándares de diseño, comunicación entre servicios, políticas de versionado y lineamientos de integración.

Mantener un catálogo tecnológico controlado para evitar la proliferación de lenguajes, frameworks y herramientas que incrementen la heterogeneidad y dificulten la mantenibilidad.

Adoptar arquitecturas de referencia que orienten la evolución del sistema y reduzcan la variabilidad entre equipos.

#### **7.2 Observabilidad y trazabilidad**

Implementar herramientas avanzadas de observabilidad distribuida, incluyendo trazabilidad de extremo a extremo, métricas centralizadas y logging estructurado.

Diseñar mecanismos de diagnóstico rápido que permitan identificar fallos en cascada y aislar servicios problemáticos en tiempo real.

Integrar auditoría centralizada, especialmente en sectores regulados como salud, banca y gobierno digital.

#### **Automatización y operaciones**

Adoptar pipelines CI/CD maduros que permitan despliegues frecuentes, seguros y auditables.

Automatizar pruebas de integración y pruebas end-to-end, reduciendo el riesgo de regresiones en sistemas distribuidos.

Incorporar prácticas de ingeniería de resiliencia, como pruebas de caos, simulación de fallos y validación continua de patrones de tolerancia a fallos.

### **Gestión de la complejidad distribuida**

Aplicar patrones arquitectónicos adecuados, como circuit breakers, service discovery, API gateways, event sourcing y CQRS, según las necesidades del dominio.

Controlar la proliferación de servicios, evitando microservicios excesivamente pequeños que incrementen la carga operativa sin aportar valor.

Gestionar la deuda técnica de manera sistemática, priorizando la refactorización de servicios duplicados o inconsistentes.

### **Adopción gradual y contextualizada**

Planificar migraciones progresivas mediante patrones como Strangler Fig, evitando interrupciones en sistemas críticos.

Evaluar la madurez organizacional antes de adoptar microservicios, considerando capacidades en DevOps, monitoreo, automatización y cultura de ingeniería.

Adaptar la estrategia al sector, reconociendo que los requisitos de trazabilidad, seguridad y disponibilidad varían significativamente entre salud, energía, transporte, finanzas y gobierno.

## REFERENCIAS

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley Professional.

<https://www.informit.com/store/devops-a-software-architects-perspective-9780134049847>

Bushong, V., Abdelfattah, A. S., Maruf, A. A., Das, D., Lehman, A., Jaroszewski, E., Coffey, M., Cerny, T., Frajtak, K., Tisnovsky, P., & Bures, M. (2021). On microservice analysis and architecture evolution: A systematic mapping study. *Applied Sciences*, 11(17), 7856. <https://doi.org/10.3390/app11177856>

ISO/IEC. (2011). ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models. International Organization for Standardization. <https://www.iso.org/standard/35733.html> Nota: ISO/IEC 25010:2011 fue retirada posteriormente por ISO/IEC 25002:2021, pero continúa siendo la referencia estándar para los modelos de calidad de software en la literatura científica.

Newman, S. (2015). Building microservices: Designing fine-grained systems. O'Reilly Media. <https://www.oreilly.com/library/view/building-microservices/9781491950340/>

Richardson, C. (2018). Microservices patterns: With examples in Java. Manning Publications. <https://www.manning.com/books/microservices-patterns>

Taibi, D., Lenarduzzi, V., & Pahl, C. (2018). Architectural patterns for microservices: A systematic mapping study. In Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018) (pp. 221–232). SCITEPRESS. <https://doi.org/10.5220/0006798302210232>

Todo el contenido de LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades, publicados en este sitio está disponibles bajo Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) 