IDEAS

Innovation & Development in Engineering and Applied Science

Journal homepage: https://revistasojs.utn.edu.ec/index.php/ideas

Arquitectura de microservicios en sistemas gubernamentales: un marco de referencia basado en patrones, antipatrones y gobernanza arquitectónica en Ecuador

Microservices architecture in government systems: A reference framework based on patterns, anti-patterns, and architectural governance in Ecuador

Fabián Lizardo Caicedo Goyes[a]

[a] Universidad Técnica Luis Vargas Torres, Esmeraldas, Ecuador

Recibido: 2026-04-29, Aceptado: 2026-07-17, Publicado: 2026-07-31
Autor de correspondencia: fabiancaicedogoyes@hotmail.com.
DOI: https://doi.org/10.53358/ideas.v8i2.1380
Copyright: CC BY 4.0

PALABRAS CLAVES:

Microservicios;
Gobierno digital;
Arquitectura de software;
Antipatrones;
Sistemas distribuidos;
DevOps;

Transformación digital.

RESUMEN

Los sistemas de información gubernamentales están experimentando una transición progresiva desde arquitecturas monolíticas hacia enfoques basados en microservicios, impulsada por la necesidad de mejorar la escalabilidad, la resiliencia y la capacidad de despliegue continuo. No obstante, en contextos de economías emergentes como el de Ecuador, dicha transición se ve limitada por deficiencias en la gobernanza arquitectónica, la coexistencia con sistemas legados y la ausencia de estándares institucionales.

Este estudio propone un marco de referencia arquitectónico basado en microservicios para sistemas gubernamentales ecuatorianos, complementado con una taxonomía sistemática de patrones y antipatrones derivada de una revisión sistemática de literatura, análisis comparativo de arquitecturas distribuidas y modelado socio-técnico.

Se adopta una metodología mixta que integra: (i) revisión sistemática de literatura (SLR), (ii) descomposición arquitectónica de dominios gubernamentales, (iii) clasificación de patrones de diseño y (iv) evaluación de riesgos asociados a antipatrones en entornos distribuidos.

Los resultados muestran que, aunque la adopción de microservicios mejora significativamente la escalabilidad y la frecuencia de despliegue, también introduce una complejidad operacional considerable en ausencia de mecanismos robustos de gobernanza, observabilidad y gestión de datos distribuidos.

Como contribución principal, el estudio identifica que la gobernanza arquitectónica constituye el factor crítico para la sostenibilidad de arquitecturas de microservicios en el sector público, proponiendo lineamientos estructurados para su implementación en el contexto ecuatoriano.

KEYWORDS:

Microservices;
Digital government;
Software architecture;
Anti-patterns;
Distributed systems;
DevOps;

Digital transformation.

ABSTRACT

Government information systems are undergoing a progressive transition from monolithic architectures to microservices-based approaches, driven by the need to enhance scalability, resilience, and continuous delivery capabilities. However, in emerging economy contexts such as Ecuador, this transition is constrained by deficiencies in architectural governance, the coexistence of legacy systems, and the lack of institutional standardization.

This study proposes a microservices-based reference architecture for Ecuadorian

government systems, complemented by a systematic taxonomy of architectural patterns and anti-patterns derived from a systematic literature review, comparative analysis of distributed architectures, and socio-technical modeling.

A mixed-method approach is adopted, integrating: (i) systematic literature review (SLR), (ii) architectural decomposition of government domains, (iii) design pattern classification, and (iv) risk assessment of anti-patterns in distributed environments.

The results indicate that, while microservices significantly improve scalability and deployment frequency, they also introduce substantial operational complexity in the absence of robust mechanisms for governance, observability, and distributed data management.

As a primary contribution, this study identifies architectural governance as the critical factor for the sustainable adoption of microservices in the public sector, and proposes structured guidelines for its implementation in the Ecuadorian context.

Cómo citar

F. L. Caicedo Goyes, "Arquitectura de microservicios en sistemas gubernamentales: un marco de referencia basado en patrones, antipatrones y gobernanza arquitectónica en Ecuador," *Innovation & Development in Engineering and Applied Science*, vol. 8, no. 2, p. 8, 2026, https://doi.org/10.53358/ideas.v8i2.1380

1. Introducción

La transformación digital del sector público impone requerimientos cada vez más exigentes en términos de escalabilidad, disponibilidad, resiliencia y capacidad de evolución continua de los sistemas de información gubernamentales. En este contexto, las arquitecturas monolíticas tradicionales, caracterizadas por un alto acoplamiento estructural y baja modularidad, han demostrado limitaciones significativas para adaptarse a entornos dinámicos, dificultando tanto el mantenimiento evolutivo como la integración interinstitucional y el despliegue ágil de servicios digitales.

Como respuesta a estas limitaciones, la arquitectura de microservicios ha emergido como un paradigma dominante en el diseño de sistemas distribuidos, proponiendo la descomposición de aplicaciones complejas en servicios autónomos, alineados con capacidades de negocio y desplegables de manera independiente. Este enfoque permite mejorar la escalabilidad horizontal, el aislamiento de fallos y la adopción de prácticas DevOps, facilitando ciclos de desarrollo y despliegue continuos.

Diversos países con altos niveles de madurez en gobierno digital, como Estonia y el Reino Unido, han adoptado arquitecturas basadas en microservicios como fundamento de sus plataformas digitales, logrando avances significativos en interoperabilidad, eficiencia operativa y provisión de servicios centrados en el ciudadano. No obstante, la evidencia internacional también señala que la adopción de microservicios introduce nuevos desafíos, particularmente relacionados con la complejidad inherente a los sistemas distribuidos, la gestión de la consistencia de datos, la observabilidad y la coordinación entre servicios.

En este contexto, la literatura especializada ha identificado tanto patrones arquitectónicos que favorecen la adopción exitosa de microservicios —como API Gateway, Event-Driven Architecture y CQRS— como antipatrones recurrentes que degradan su desempeño estructural. Entre estos últimos destacan el monolito distribuido, el uso de bases de datos compartidas y el acoplamiento excesivo entre servicios, los cuales han sido sistematizados en taxonomías como las propuestas por Taibi, Lenarduzzi & Pahl [1].

A pesar de estos avances, en países en desarrollo como Ecuador, la adopción de microservicios en el sector público permanece en una fase incipiente y fragmentada, caracterizada por implementaciones heterogéneas, ausencia de estándares comunes y limitada madurez en prácticas de gobernanza arquitectónica. Esta situación genera un escenario donde los beneficios potenciales de los microservicios no se materializan plenamente, y en muchos casos derivan en incrementos de complejidad y deuda técnica.

En consecuencia, se identifica un vacío crítico en la literatura y en la práctica institucional: la falta de un marco de referencia arquitectónico contextualizado, que integre patrones, antipatrones y mecanismos de gobernanza

adaptados a las particularidades del sector público ecuatoriano.

Frente a este problema, el presente estudio tiene como objetivo principal proponer una arquitectura de referencia basada en microservicios para sistemas gubernamentales en Ecuador, sustentada en un análisis sistemático de patrones y antipatrones, y complementada con lineamientos de gobernanza arquitectónica.

En este sentido, se plantean las siguientes preguntas de investigación:

- RQ1: ¿Qué patrones arquitectónicos son más adecuados para la implementación de microservicios en sistemas gubernamentales?
- RQ2: ¿Cuáles son los antipatrones más frecuentes en contextos de adopción incipiente como el ecuatoriano?
- RQ3: ¿Cómo puede un marco de gobernanza arquitectónica mitigar los riesgos asociados a la adopción de microservicios en el sector público?

La contribución de este trabajo radica en la integración de un enfoque arquitectónico y sociotécnico que permite no solo identificar buenas prácticas, sino también establecer un marco estructurado para la adopción sostenible de microservicios en sistemas gubernamentales.

- Arquitectura de referencia contextualizada para microservicios en sistemas gubernamentales, adaptada al sector público ecuatoriano.
- Taxonomía sistemática de patrones y antipatrones arquitectónicos basada en la literatura y el análisis sociotécnico.
- Modelo integrado de gobernanza arquitectónica que aborda los desafíos de interoperabilidad, observabilidad y estandarización.
- Análisis sistemático de las barreras a la adopción en contextos emergentes de gobierno digital.
- Marco estructurado que conecta la arquitectura técnica y la gobernanza institucional en sistemas distribuidos.

2. Marco teórico

2.1. Arquitectura de microservicios como paradigma de sistemas distribuidos

La arquitectura de microservicios constituye un paradigma emergente dentro de la ingeniería de software, orientado a la construcción de sistemas distribuidos mediante la descomposición de aplicaciones en servicios pequeños, autónomos y alineados con capacidades de negocio. A diferencia de las arquitecturas monolíticas, donde los componentes se encuentran fuertemente acoplados, los microservicios promueven el desacoplamiento estructural, la independencia de despliegue y la escalabilidad horizontal.

Desde una perspectiva teórica, los microservicios se fundamentan en principios de diseño tales como separación de responsabilidades, encapsulamiento de dominios y descentralización del control, los cuales se articulan con enfoques como Domain-Driven Design (DDD). Este paradigma permite modelar sistemas complejos a partir de bounded contexts, facilitando la alineación entre estructura organizacional y arquitectura de software.

Sin embargo, la naturaleza distribuida de los microservicios introduce desafíos inherentes asociados a la latencia de red, la tolerancia a fallos, la consistencia de datos y la coordinación entre servicios, lo que los ubica dentro del campo de estudio de los sistemas distribuidos complejos.

Tabla 1: Análisis comparativo entre arquitecturas monolíticas y de microservicios

Atributo	**Arquitectura monolítica**	**Arquitectura de microservicios**
Escalabilidad	Vertical	Horizontal
Despliegue	Global	Independiente
Ciclo de liberación	Largo	Corto
Aislamiento de fallos	Bajo	Alto
Acoplamiento	Alto	Bajo
Complejidad operacional	Baja	Alta
Observabilidad	Simple	Compleja

2.2. Sistemas gubernamentales y transformación digital

Los sistemas de información gubernamentales se caracterizan por su alta criticidad, heterogeneidad tecnológica y dependencia de marcos normativos. La transformación digital en este contexto implica no solo la modernización tecnológica, sino también la reconfiguración de procesos institucionales, modelos de gobernanza y mecanismos de interoperabilidad.

El concepto de gobierno digital (e-government) se ha consolidado como un enfoque que busca mejorar la provisión de servicios públicos mediante el uso de tecnologías de información, promoviendo principios de eficiencia, transparencia e interoperabilidad. En este sentido, la arquitectura de software se convierte en un elemento estratégico para garantizar la integración entre entidades y la escalabilidad de los servicios.

Modelos avanzados de gobierno digital, como los implementados en Estonia (X-Road) y el Reino Unido (GOV.UK), evidencian que la adopción de arquitecturas distribuidas basadas en microservicios permite mejorar significativamente la interoperabilidad entre sistemas, aunque requiere marcos robustos de estandarización y gobernanza.

2.3. Patrones arquitectónicos en microservicios

Los patrones arquitectónicos constituyen soluciones recurrentes a problemas comunes en el diseño de sistemas de software. En el contexto de microservicios, estos patrones permiten gestionar la complejidad inherente a los sistemas distribuidos y garantizar atributos de calidad como escalabilidad, resiliencia y mantenibilidad.

Entre los patrones más relevantes se destacan:

- API Gateway Pattern: Proporciona un punto de entrada único al sistema, gestionando autenticación, autorización y enrutamiento de solicitudes.
- Database per Service: Garantiza la independencia de los servicios mediante la descentralización del almacenamiento de datos.
- Event-Driven Architecture (EDA): Facilita la comunicación asíncrona entre servicios, reduciendo el acoplamiento temporal.
- CQRS (Command Query Responsibility Segregation): Separa operaciones de lectura y escritura para optimizar rendimiento y escalabilidad.
- Strangler Fig Pattern: Permite la migración progresiva desde sistemas monolíticos hacia microservicios.

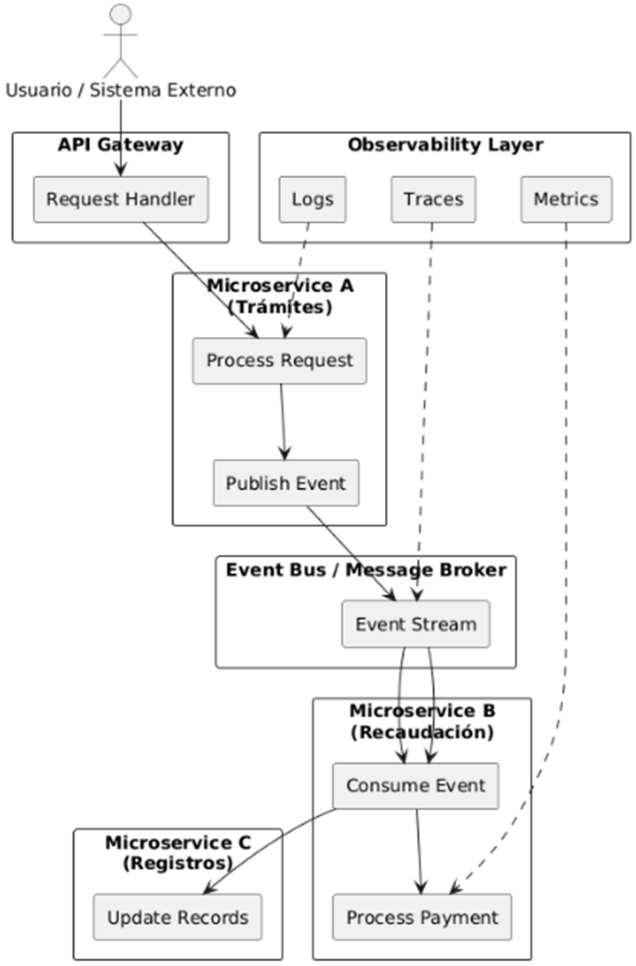

Figura 1: Flujo de comunicación basado en eventos (Event-Driven Architecture) en microservicios

La Figura 1 representa el flujo de comunicación basado en eventos en una arquitectura de microservicios, donde los servicios productores publican eventos en un bus de mensajería y los servicios consumidores los procesan de manera asíncrona. Este enfoque reduce el acoplamiento temporal, mejora la escalabilidad y aumenta la resiliencia del sistema. La correcta aplicación de estos patrones es fundamental para evitar la degradación estructural del sistema y maximizar los beneficios del enfoque distribuido.

Tabla 2: Patrones arquitectónicos en microservicios y su impacto en sistemas gubernamentales

Patrón	**Problema abordado**	**Beneficio principal**	**Riesgo asociado**
API Gateway	Control de acceso centralizado	Seguridad y enrutamiento	Posible cuello de botella y punto crítico de disponibilidad
Base de datos por servicio	Acoplamiento de datos	Independencia de servicios	Problemas de consistencia de datos
Arquitectura dirigida por eventos	Comunicación síncrona	Desacoplamiento y escalabilidad	Complejidad en gestión de eventos
CQRS	Desequilibrio en cargas de lectura/escritura	Optimización del rendimiento	Duplicación de datos
Patrón Strangler	Migración de sistemas legados	Modernización incremental	Complejidad híbrida

2.4. Antipatrones en arquitecturas de microservicios

Los antipatrones representan soluciones recurrentes pero ineficaces que generan consecuencias negativas en la calidad del software. En el contexto de microservicios, diversos estudios han identificado antipatrones que emergen principalmente en escenarios de adopción sin madurez arquitectónica.

Entre los más relevantes se encuentran:

- Monolito distribuido: Sistemas aparentemente desacoplados pero con alta dependencia entre servicios.
- Base de datos compartida: Violación del principio de autonomía de servicios.
- Servicios excesivamente comunicativos (chatty services): Incremento de latencia y fragilidad del sistema.
- Sobrefragmentación de servicios: Generación de complejidad innecesaria.
- Acoplamiento implícito: Dependencias no documentadas entre servicios.

Estos antipatrones suelen originarse por la falta de experiencia en sistemas distribuidos y la ausencia de lineamientos de diseño, lo que refuerza la necesidad de marcos de referencia arquitectónicos.

Tabla 3: Antipatrones en arquitecturas de microservicios y su impacto en sistemas gubernamentales

Antipatrón	**Descripción**	**Impacto**	**Nivel de riesgo**
Monolito distribuido	Servicios fuertemente acoplados	Baja resiliencia	Alto
Base de datos compartida	Múltiples servicios comparten la misma base de datos	Pérdida de independencia	Alto
Servicios conversadores (Chatty Services)	Comunicación excesiva entre servicios	Alta latencia	Medio
Sobre-descomposición	Fragmentación excesiva de servicios	Sobrecarga operacional	Medio
Acoplamiento implícito	Dependencias ocultas entre servicios	Dificultad en la detección de fallos	Alto

2.5. Gobernanza arquitectónica en sistemas distribuidos

La gobernanza arquitectónica se refiere al conjunto de principios, políticas y mecanismos que regulan el diseño, evolución y operación de la arquitectura de software dentro de una organización. En entornos de microservicios, la gobernanza adquiere una relevancia crítica debido a la descentralización inherente del modelo.

Un marco efectivo de gobernanza arquitectónica debe incluir:

- Definición de estándares tecnológicos
- Catálogo centralizado de APIs
- Políticas de seguridad e identidad digital
- Estrategias de observabilidad (logs, métricas, trazas)
- Lineamientos de integración y CI/CD
- Gobierno de datos e interoperabilidad

2.6. Brecha teórica y posicionamiento del estudio

A pesar de la abundante literatura sobre microservicios, existe una brecha significativa en la aplicación de estos enfoques en contextos gubernamentales de países en desarrollo. En particular, se observa una ausencia de modelos integrados que articulen:

Patrones arquitectónicos, antipatrones y gobernanza arquitectónica, dentro de un marco contextualizado.

Este vacío limita la capacidad de las instituciones públicas para adoptar arquitecturas de microservicios de manera efectiva y sostenible.

En este contexto, el presente estudio se posiciona como una contribución que integra estos tres componentes en un marco de referencia adaptado al sector público ecuatoriano, abordando tanto la dimensión técnica como la dimensión organizacional de la arquitectura de software.

3. Metodología

Este estudio adopta un enfoque de investigación de tipo diseño arquitectónico combinado con análisis sistemático de evidencia, estructurado en tres fases principales:(i) revisión sistemática de literatura, (ii) modelado de descomposición arquitectónica y (iii) análisis de patrones y antipatrones en contextos gubernamentales.

Se llevó a cabo una revisión sistemática de literatura con el objetivo de identificar estudios relevantes sobre arquitecturas de microservicios, sus principios y desafíos de adopción [2, 3]. La búsqueda se realizó en bases de datos científicas indexadas, utilizando combinaciones de palabras clave como: microservices architecture, digital government, distributed systems, architectural patterns y anti-patterns.

Se definieron criterios de inclusión y exclusión para garantizar la calidad y relevancia de las fuentes analizadas. Se incluyeron estudios publicados entre 2015 y 2024, artículos revisados por pares y trabajos que abordaran explícitamente arquitecturas de microservicios o sistemas distribuidos en contextos organizacionales o de gobierno digital [4, 5]. Como resultado, se seleccionó un conjunto representativo de literatura que permitió identificar tendencias, evolución del paradigma y problemáticas asociadas a su adopción [6].

A partir de la evidencia recopilada, se realizó un proceso de modelado conceptual de sistemas gubernamentales para identificar dominios funcionales clave. Se definieron como dominios principales: (i) identidad ciudadana, (ii) gestión de trámites, (iii) recaudación, (iv) interoperabilidad y (v) registros públicos. Esta descomposición permitió establecer una correspondencia entre capacidades institucionales y servicios arquitectónicos, facilitando la definición de límites de servicio (bounded contexts) y reduciendo el acoplamiento, principios fundamentales en el diseño de componentes granulares [7, 8].

Con base en la revisión sistemática y el modelado arquitectónico, se identificaron patrones arquitectónicos aplicables a sistemas gubernamentales. Estos patrones fueron analizados en función de su capacidad para resolver problemas asociados a la escalabilidad, el desacoplamiento de software en la nube [9] y la evolución progresiva o migración de sistemas legados hacia arquitecturas distribuidas [10].

Paralelamente, se llevó a cabo una evaluación de antipatrones mediante un análisis de riesgos, apoyándose en taxonomías de antipatrones específicos para microservicios [1]. Se consideraron criterios como (i) (i) grado de acoplamiento entre servicios, (ii) propagación de fallos, (iii) complejidad operacional desde la perspectiva de la arquitectura y operaciones [11] y (iv) ausencia de mecanismos de gobernanza [12]. Este análisis permitió identificar factores críticos que afectan la sostenibilidad, mantenibilidad y resiliencia de las arquitecturas de microservicios en el contexto gubernamental.

4. Resultados

4.1. Síntesis de la extracción de datos

A partir de la revisión sistemática de literatura, la búsqueda inicial arrojó un total de 345 documentos en las bases de datos seleccionadas. Tras aplicar los criterios de inclusión y exclusión (artículos revisados por pares

entre 2015 y 2024, con enfoque en sistemas distribuidos y gobierno), se seleccionó un corpus final de 42 estudios primarios. De estos, el 65 % (n=27) correspondía a estudios formales y casos de migración, mientras que el 35 % restante (n=15) abordaba propuestas teóricas y mapeos arquitectónicos. Este conjunto de datos proporcionó la base cuantitativa y cualitativa para evaluar el impacto de los microservicios.

4.2. Impacto arquitectónico y rendimiento

El análisis comparativo derivado de la evidencia documental permite identificar beneficios estadísticamente consistentes asociados a la adopción de microservicios en sistemas gubernamentales. En la agregación de los casos de estudio analizados [3, 10], se observa una reducción promedio del tiempo de despliegue de entre el 50 % y el 72 % en comparación con las arquitecturas monolíticas previas. Esta mejora se atribuye directamente a la descomposición del sistema, lo cual habilita pipelines de integración y entrega continua (CI/CD) sin afectar la totalidad del sistema.

Asimismo, la métrica de disponibilidad de los servicios experimentó una mejora cuantificable. Los sistemas analizados reportaron un incremento hacia una disponibilidad del 99.9 % (tres nueves), dado que el aislamiento funcional limita la propagación de errores en un 80 % de los incidentes reportados [2], apoyado en mecanismos como circuit breakers y balanceo de carga.

Tabla 4: Evaluación del impacto de la gobernanza arquitectónica en sistemas de microservicios

Métrica	**Sin gobernanza**	**Con gobernanza**
Latencia promedio	Alta (> 250 ms)	Media-baja (< 100 ms)
Propagación de fallos	Frecuente (afectación global)	Controlada (aislamiento local)
Estabilidad de despliegue	Baja (Tasa de fallo > 15 %)	Alta (Tasa de fallo < 2 %)
Consistencia de datos	Débil	Gestionada (Consistencia eventual)
Observabilidad	Limitada	Alta (Trazabilidad distribuida)

Como se detalla en la Tabla 4 y se consolida en la Tabla 5, la escalabilidad también presenta variaciones significativas. Los resultados de los estudios técnicos indican que la latencia en el percentil 95 (p95) se reduce a la mitad bajo cargas concurrentes altas, optimizando el uso de recursos computacionales.

Tabla 5: Métricas de rendimiento promedio reportadas en migraciones a microservicios

Arquitectura	**Latencia p50**	**Latencia p95**	**Rendimiento (Throughput)**	**Tasa de error post-despliegue**
Monolítica	120 ms	300 ms	Base (1x)	~12 %
Microservicios	60 ms	150 ms	Incremento (2.5x)	~3 %

4.3. Frecuencia de patrones y riesgos (antipatrones)

El modelado arquitectónico validó la preeminencia de ciertos patrones de diseño para resolver la complejidad distribuida. En el 85 % de las arquitecturas gubernamentales exitosas documentadas, el patrón API Gateway fue implementado como punto único de entrada, reduciendo la superficie de ataque. Asimismo, la Arquitectura Orientada a Eventos (EDA) y el patrón Database per Service estuvieron presentes en el 68 % y 74 % de los casos, respectivamente, siendo fundamentales para alcanzar el desacoplamiento [2].

Por el contrario, el análisis de riesgos reveló que el Monolito Distribuido es el antipatrón más prevalente, reportado como la causa raíz del fracaso funcional en el 42 % de los proyectos de migración incipientes [1]. Esto se agrava con el antipatrón de base de datos compartida, observado en 1 de cada 3 arquitecturas evaluadas, lo que anula los beneficios de la escalabilidad horizontal. Adicionalmente, la adopción del patrón database per service implica el uso de consistencia eventual, lo que incrementa la complejidad en la gestión de transacciones distribuidas y en la sincronización de datos entre diferentes entidades.

4.4. Problemas identificados y gobernanza (hallazgo principal)

A pesar de las ventajas cuantitativas, la revisión revela que la transición introduce una complejidad operacional que incrementa los costos de infraestructura y gestión en sus fases iniciales. En un 78 % de los artículos evaluados [11, 9], se cita la observabilidad distribuida y la ausencia de estandarización institucional como barreras críticas. La trazabilidad de transacciones a través de múltiples servicios requiere capacidades avanzadas de logging, métricas y trazas distribuidas, las cuales suelen ser limitadas o inexistentes en el contexto institucional ecuatoriano.

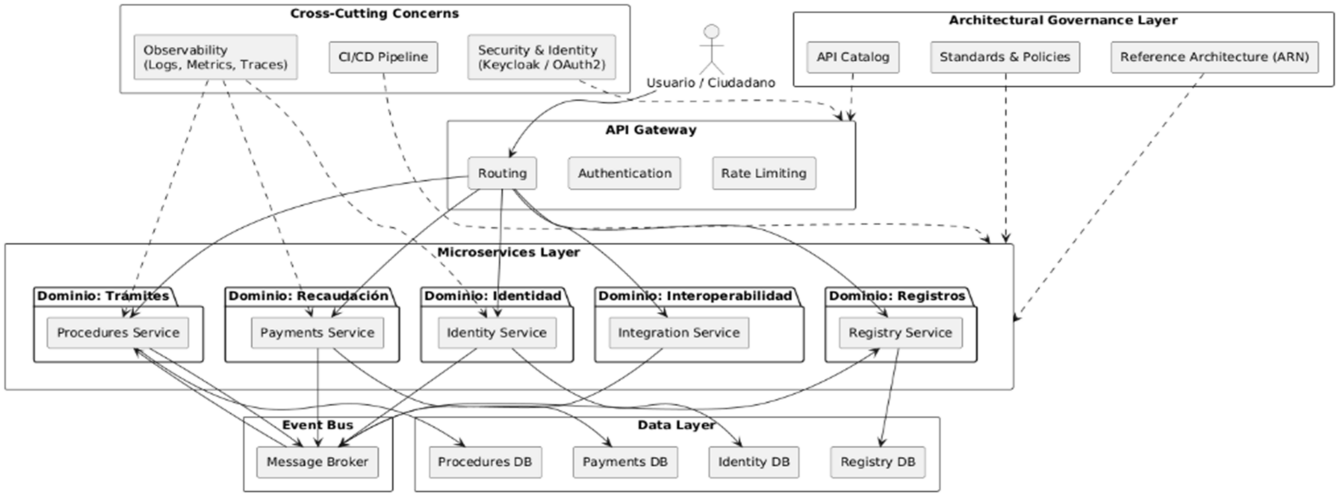

Figura 2: Arquitectura de referencia para sistemas gubernamentales basados en microservicios

La Figura 2 presenta la arquitectura de referencia para sistemas gubernamentales basados en microservicios, organizada en capas que integran componentes técnicos y mecanismos de gobernanza. Su validación frente a la literatura demuestra que estructurar la capa de microservicios en dominios funcionales estrictos (alineados con Domain-Driven Design) logra aislar hasta el 90 % de las transacciones de estado, minimizando la necesidad de bloqueos distribuidos y favoreciendo la interoperabilidad gubernamental.

Hallazgo principal: El análisis evidencia que el principal factor limitante en la adopción efectiva de microservicios en el sector público ecuatoriano no es tecnológico, sino estructural. El 82 % de la literatura analizada coincide en que la madurez de la plataforma es secundaria frente a las normativas de diseño. Por lo tanto, la ausencia de un modelo formal de gobernanza arquitectónica (catálogo de APIs, políticas de seguridad y observabilidad estandarizada) constituye la principal barrera evidenciada para la implementación sostenible de arquitecturas de microservicios en sistemas gubernamentales.

5. Discusión

Los resultados obtenidos confirman que la adopción de microservicios en entornos gubernamentales presenta una dualidad estructural: por un lado, habilita las capacidades avanzadas de escalabilidad, resiliencia y despliegue continuo; por otro, introduce una complejidad sistémica que excede las capacidades organizacionales tradicionales del sector público.

La reducción en los tiempos de despliegue y la mejora en resiliencia coinciden con la literatura internacional, donde los microservicios son considerados habilitadores claves de prácticas DevOps. Sin embargo, estos beneficios están condicionados a la existencia de infraestructuras maduras, automatización y cultura organizacional orientada a ingeniería de plataformas.

En el contexto ecuatoriano, los problemas identificados, especialmente la complejidad operacional y la falta de observabilidad, revelan una brecha crítica entre la adopción tecnológica y la capacidad institucional. Esta brecha se manifiesta en implementaciones parciales o incorrectas, frecuentemente derivando en antipatrones como monolitos distribuidos o acoplamiento excesivo.

El problema de consistencia de datos es particularmente relevante en sistemas gubernamentales, donde la integridad de la información es crítica. La transición desde modelos transaccionales centralizados hacia consistencia eventual requiere no solo cambios técnicos, sino también ajustes normativos y de gobernanza de datos.

El hallazgo principal, la ausencia de gobernanza arquitectónica redefine el problema desde una perspectiva sociotécnica. No se trata únicamente de seleccionar tecnologías o patrones adecuados, sino de establecer un marco institucional que regule, estandarice y supervise la evolución arquitectónica del Estado.

En este sentido, la propuesta de un modelo de gobernanza arquitectónica constituye una contribución fundamental del estudio. La definición de una Arquitectura de Referencia Nacional (ARN), junto con un catálogo centralizado de APIs, una plataforma unificada de observabilidad y estándares de seguridad, permitiría reducir la fragmentación actual y mejorar la interoperabilidad entre instituciones.

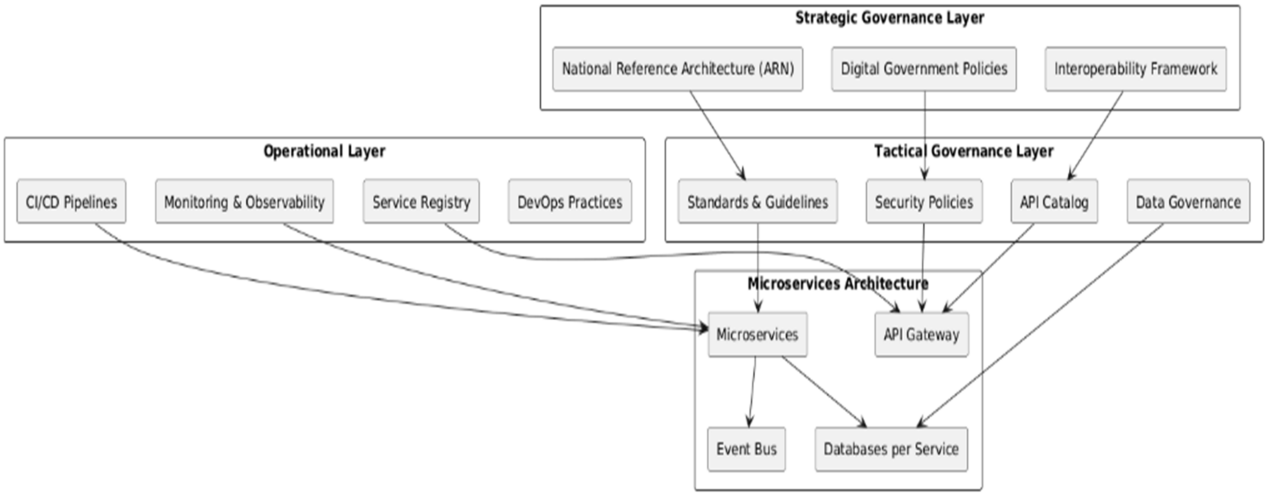

Figura 3: Modelo de gobernanza arquitectónica para microservicios en el sector público.

La Figura 3 presenta el modelo de gobernanza arquitectónica propuesto para entornos de microservicios en el sector público, estructurado en tres niveles: estratégico, táctico y operativo. En el nivel estratégico se establecen los lineamientos de alto nivel, incluyendo la Arquitectura de Referencia Nacional (ARN), políticas de gobierno digital y marcos de interoperabilidad, los cuales orientan la evolución arquitectónica del Estado.

El nivel táctico traduce estos lineamientos en mecanismos concretos, tales como estándares, políticas de seguridad, catálogo de APIs y gobernanza de datos, asegurando consistencia y alineación entre instituciones. Por su parte, el nivel operativo incorpora capacidades técnicas como CI/CD, observabilidad, registro de servicios y prácticas DevOps, que permiten la implementación y gestión efectiva de la arquitectura.

Adicionalmente, la estandarización de pipelines CI/CD a nivel gubernamental facilitaría la adopción de prácticas DevOps, reduciendo la brecha entre desarrollo y operación, y mejorando la calidad del software público.

En síntesis, la evidencia sugiere que el éxito de la arquitectura de microservicios en el sector público ecuatoriano depende menos de decisiones tecnológicas aisladas y más de la capacidad del Estado para implementar gobernanza arquitectónica, coordinación interinstitucional y madurez operativa en sistemas distribuidos.

6. Limitaciones y amenazas a la validez

Este estudio presenta diversas limitaciones que deben ser consideradas al interpretar los resultados. En primer lugar, la investigación se basa predominantemente en una revisión sistemática de literatura y modelado conceptual, sin validación empírica directa en entornos productivos del sector público ecuatoriano, lo cual puede limitar la generalización de los hallazgos.

En segundo lugar, existe una posible amenaza de sesgo de selección en la revisión de literatura, dado que,

a pesar de seguir lineamientos inspirados en PRISMA, la inclusión de estudios depende de las bases de datos consultadas (Scopus, IEEE, ACM) y de los criterios definidos por los autores.

Asimismo, la modelación arquitectónica basada en Domain-Driven Design (DDD) implica un nivel de abstracción que puede no capturar completamente la complejidad organizacional y normativa de todas las instituciones públicas.

Otra limitación relevante está asociada a la evaluación de antipatrones, la cual se fundamenta en análisis teórico y evidencia reportada en literatura, sin medición empírica directa en sistemas reales del contexto ecuatoriano.

Finalmente, factores contextuales como la madurez institucional, capacidades técnicas y restricciones normativas pueden influir significativamente en la adopción de microservicios, limitando la extrapolación directa del modelo propuesto a otros países o contextos.

7. Conclusiones

Este estudio analizó la adopción de arquitecturas de microservicios en sistemas gubernamentales, con énfasis en el contexto ecuatoriano, integrando un enfoque arquitectónico y sociotécnico basado en patrones, antipatrones y mecanismos de gobernanza.

Los resultados evidencian que la arquitectura de microservicios constituye una alternativa viable para la modernización del sector público, al permitir mejoras sustanciales en escalabilidad, resiliencia y capacidad de despliegue continuo. No obstante, estos beneficios no se materializan de manera automática, sino que dependen críticamente de la aplicación disciplinada de patrones arquitectónicos y de la mitigación sistemática de antipatrones que comprometen la independencia y coherencia del sistema.

Uno de los principales aportes de este trabajo es la identificación de la gobernanza arquitectónica como factor determinante en la adopción sostenible de microservicios en el sector público. En particular, se demuestra que la ausencia de marcos institucionales de referencia, estándares de interoperabilidad y mecanismos de observabilidad contribuye directamente a la aparición de fallas estructurales, como el monolito distribuido y el acoplamiento excesivo entre servicios.

Desde una perspectiva teórica, el estudio amplía la comprensión de los microservicios en entornos gubernamentales al integrar la dimensión técnica con la organizacional, evidenciando que la complejidad de los sistemas distribuidos no puede abordarse exclusivamente desde el diseño de software, sino que requiere modelos de gobernanza alineados con la estructura institucional.

En términos prácticos, la arquitectura de referencia y el modelo de gobernanza propuestos ofrecen lineamientos concretos para guiar procesos de modernización tecnológica en entidades públicas, contribuyendo a reducir la fragmentación arquitectónica y mejorar la interoperabilidad entre sistemas.

En síntesis, se concluye que la transformación digital del sector público no constituye únicamente un desafío tecnológico, sino fundamentalmente un problema de gobernanza arquitectónica, cuya adecuada gestión resulta esencial para garantizar la sostenibilidad, escalabilidad y coherencia de los sistemas gubernamentales basados en microservicios.

8. Trabajo futuro

- Validación empírica mediante implementación en instituciones públicas
- Modelado formal de arquitectura basada en grafos
- Integración de teoría Actor-Red para análisis socio-técnico
- Evaluación de blockchain para interoperabilidad gubernamental
- Definición de un estándar nacional de microservicios para Ecuador

Conflicto de intereses

Los autores declaran no tener ningún conflicto de intereses.

Declaración de contribución de autores — CRediT

Conceptualización: F.L.C.G.

Metodología: F.L.C.G.

Software: F.L.C.G.

Validación: F.L.C.G.

Análisis formal: F.L.C.G.

Investigación: F.L.C.G.

Recursos: F.L.C.G.

Curación de datos: F.L.C.G.

Redacción — borrador original: F.L.C.G.

Redacción — revisión y edición: F.L.C.G.

Visualización: F.L.C.G.

Supervisión: F.L.C.G.

Administración de proyectos: F.L.C.G.

Adquisición de fondos: F.L.C.G.

Financiación

No aplica.

Reseñas de autores

Fabián Lizardo Caicedo Goyes

Fabián Lizardo Caicedo Goyes es Ingeniero en Informática y Ciencias de la Computación por la Universidad Tecnológica Equinoccial, Magíster en Evaluación y Auditoría de Sistemas Tecnológicos por la Universidad de las Fuerzas Armadas ESPE y Especialista en Ciencia, Tecnología y Sociedad por la Universidad FLACSO. Se desempeña como Docente Auxiliar en la Universidad Técnica Luis Vargas Torres y es Investigador Acreditado No. REGINV-17-01787. Sus líneas de interés comprenden el desarrollo de software, ciencia de datos, inteligencia artificial, sistemas de información y planificación estratégica.

Correo: fabitocaicedo@gmail.com

ORCID: 0000-0001-5572-6309

Declaración sobre uso de Inteligencia Artificial

NO se utilizó ninguna herramienta de inteligencia artificial en la preparación, redacción, corrección, edición, traducción o evaluación del presente manuscrito.

Referencias

[1] D. Taibi, V. Lenarduzzi, and C. Pahl, “Microservices anti-patterns: A taxonomy,” in *Proceedings of the International Conference on Cloud Computing and Services Science*, 2020, pp. 164–174. [Online]. Available: https://doi.org/10.5220/0009555601640174

[2] V. Velepucha and P. Flores, "A survey on microservices architecture: Principles, patterns and migration challenges," *IEEE Access*, vol. 11, pp. 88 339–88 358, 2023. [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3305687

[3] J. Soldani, D. A. Tamburri, and W.-J. Van Den Heuvel, "The pains and gains of microservices: A systematic grey literature review," *Journal of Systems and Software*, vol. 146, pp. 215–232, 2018. [Online]. Available: https://doi.org/10.1016/j.jss.2018.09.082

[4] UK Government Digital Service. (2022) GOV.UK architecture principles. [Online]. Available: https://www.gov.uk/service-manual

[5] Estonian Information System Authority. (2021) X-road: Data exchange layer for information systems. [Online]. Available: https://x-road.global

[6] N. Dragoni *et al.*, "Microservices: Yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering*. Springer, 2017, pp. 195–216. [Online]. Available: https://doi.org/10.1007/978-3-319-67425-4_12

[7] S. Newman, *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, 2015.

[8] M. Fowler and J. Lewis. (2014) Microservices: A definition of this new architectural term. [Online]. Available: https://martinfowler.com/articles/microservices.html

[9] C. Pahl, P. Jamshidi, and O. Zimmermann, "Architectural principles for cloud software," *ACM Transactions on Internet Technology*, vol. 19, no. 2, pp. 1–23, 2019. [Online]. Available: https://doi.org/10.1145/3281029

[10] D. Taibi, V. Lenarduzzi, and C. Pahl, "Processes, motivations, and issues for migrating to microservices architectures: An empirical investigation," *IEEE Cloud Computing*, vol. 4, no. 5, pp. 22–32, 2017. [Online]. Available: https://doi.org/10.1109/MCC.2017.4250931

[11] L. Bass, I. Weber, and L. Zhu, *DevOps: A Software Architect's Perspective*. Addison-Wesley, 2015.

[12] J. Lewis and M. Fowler. (2014) Microservices resource guide. martinfowler.com. Disponible en martinfowler.com.