

Marco de migración de aplicaciones monolíticas hacia arquitecturas serverless en la nube

Roberto Orozco Montenegro

Maestría Profesional en Ingeniería del Software con énfasis en Arquitectura y Diseño de Software
Universidad CENFOTEC

Correo electrónico: rorozcom@ucenfotec.ac.cr | roboromon29@gmail.com

Tutor académico: MSc. Miguel Pérez Montero | mperez@ucenfotec.ac.cr

Resumen

Las organizaciones que mantienen aplicaciones monolíticas en producción enfrentan limitaciones relacionadas con escalabilidad, mantenibilidad, resiliencia y eficiencia operativa. Aunque la computación en la nube ha impulsado la adopción de arquitecturas modernas como *serverless*, la migración desde sistemas monolíticos continúa siendo un proceso complejo y potencialmente riesgoso cuando no se apoya en criterios estructurados de evaluación. Aun así, su adopción no garantiza, por sí sola, mejores resultados técnicos, económicos u organizacionales.

En este artículo se presenta un marco de migración de aplicaciones monolíticas hacia arquitecturas *serverless* en la nube, sin una fase intermedia basada en microservicios. El marco fue desarrollado a partir de una revisión sistemática de literatura y un análisis conceptual de atributos de calidad, patrones de migración y factores técnicos y organizacionales que influyen en este tipo de transformaciones. Más que explicar en detalle cómo funciona *serverless*, el artículo se enfoca en analizar cuándo esta arquitectura puede aportar valor, cuándo requiere condiciones previas y cuándo no debería recomendarse.

El resultado principal es un marco por fases acompañado de un instrumento cuantitativo en Excel que permite enfocarse en valorar la viabilidad de la migración mediante criterios ponderados. El aporte del estudio es ordenar una decisión que muchas veces depende del criterio experto, para que pueda documentarse, discutirse y justificarse con evidencia ante equipos técnicos, áreas de negocio y responsables de la toma de decisiones.

Abstract

Organizations that operate monolithic applications in production face increasing limitations related to scalability, maintainability, resilience, and operational efficiency. Although cloud computing has promoted the adoption of modern architectural paradigms such as *serverless*, migrating from monolithic systems remains a complex and potentially risky endeavor when not supported by

structured evaluation criteria. Even so, adopting *serverless* does not automatically ensure better technical, economic, or organizational outcomes.

This article presents a migration framework for the direct migration of monolithic applications to *serverless* architectures in the cloud, without an intermediate transition through microservices. The framework was developed through a systematic literature review and a conceptual analysis of architectural quality attributes, migration patterns, and technical and organizational factors that influence such transformations. Rather than explaining *serverless* technology in depth, the article focuses on analyzing when this architecture can create value, when it requires prior conditions, and when migration should not be recommended.

The main result is a phase-based framework supported by a quantitative Excel instrument that evaluates migration feasibility through weighted criteria. The contribution of this study is to organize a decision that often depends on expert judgment, making it easier to document, discuss, and justify with evidence for technical teams, business areas, and decision makers.

Palabras clave

Arquitectura monolítica; arquitectura *serverless*; migración de software; modernización de aplicaciones; toma de decisiones arquitectónicas.

Keywords

Monolithic architecture; serverless architecture; software migration; application modernization; architectural decision-making.

1. Introducción

La modernización de aplicaciones monolíticas se ha vuelto una preocupación frecuente para organizaciones que necesitan responder con mayor rapidez a cambios del negocio, variaciones de demanda y presión por reducir costos operativos. El problema no es que todo monolito sea una mala arquitectura. De hecho, muchos monolitos siguen siendo útiles porque simplifican el despliegue, facilitan la consistencia transaccional y evitan cierta complejidad distribuida. La dificultad aparece cuando el sistema crece hasta el punto de acumular acoplamiento, despliegues lentos, escalamiento poco selectivo y mayor costo de mantenimiento.

Serverless surge como una alternativa de modernización porque permite trasladar parte de la administración de infraestructura al proveedor de nube y escalar componentes de manera más granular. Sin embargo, el valor de esta arquitectura no está en sus características técnicas vistas

de forma aislada, sino en qué tan bien responde al problema concreto que tiene la organización. Una función que escala automáticamente puede ser muy útil en cargas variables, pero poco relevante en un sistema con demanda estable. De la misma forma, el pago por uso puede reducir costos en ciertos escenarios, pero no reemplaza un análisis económico completo.

Por eso, la pregunta central no debería ser si *serverless* es técnicamente posible, sino si es conveniente para el sistema, la organización y el modelo operativo analizado. Esta distinción es clave, porque muchas iniciativas de modernización se justifican desde expectativas generales como “es más moderno”, “escala solo” o “será más barato”, sin evaluar dependencias, estado, criticidad, capacidades del equipo, observabilidad o riesgos de proveedor. Este enfoque coincide con estudios recientes sobre migración *serverless*, donde se señala que la decisión involucra tanto aspectos técnicos como organizacionales y que no puede reducirse a una simple sustitución tecnológica (Hamza et al., 2023; Hamza et al., 2025).

Además, resulta especialmente relevante para organizaciones pequeñas o medianas con baja madurez operativa, donde una fase intermedia basada en microservicios puede exigir capacidades DevOps, gobierno de contenedores y observabilidad distribuida que aún no están consolidadas.

Con base en lo anterior, este artículo propone un marco de migración de aplicaciones monolíticas hacia arquitecturas *serverless* en la nube. El propósito no es presentar *serverless* como la salida correcta para todos los monolitos, sino ofrecer un mecanismo ordenado para decidir con evidencia. El marco permite clasificar los escenarios en tres resultados: migración recomendada, migración condicionada a mitigaciones o pilotos, y migración no recomendada. Esta última salida es importante, porque en arquitectura de software decidir no migrar también puede ser una decisión técnicamente responsable cuando evita riesgos desproporcionados.

2. Método

2.1 Enfoque y tipo de investigación

Este trabajo se desarrolla bajo un enfoque cualitativo y aplicado, orientado a construir un marco de migración utilizable por arquitectos de software. A diferencia de una investigación experimental, el propósito aquí no es medir el rendimiento de una implementación particular en un entorno controlado, sino integrar evidencia y recomendaciones existentes en una estructura que facilite la toma de decisiones en escenarios reales.

2.2 Revisión sistemática de la literatura

Para sustentar el marco, se ejecutó una revisión sistemática que permitió identificar patrones, ventajas, riesgos y criterios recurrentes en migraciones hacia *serverless*. El objetivo de esta etapa fue consolidar hallazgos dispersos y traducirlos a dimensiones medibles dentro del instrumento cuantitativo. Por lo que permitió identificar criterios recurrentes asociados con escalabilidad, costos, complejidad estructural, dependencia tecnológica y modernización de sistemas monolíticos.

2.3 Síntesis conceptual y construcción del marco

Posteriormente, los hallazgos obtenidos fueron organizados en dimensiones de evaluación aplicables a sistemas monolíticos en producción. Estas dimensiones se utilizaron para construir un marco de migración y un instrumento cuantitativo en Excel, diseñado para convertir criterios cualitativos en una recomendación ponderada.

Para evitar fragmentación metodológica, el artículo presenta el proceso de manera integrada: la revisión aporta la base conceptual, la síntesis organiza los criterios de decisión y el instrumento permite aplicarlos en un contexto práctico de evaluación arquitectónica. El artículo no incorpora datos empíricos nuevos fuera de los desarrollados en la investigación base, sino una reorganización e interpretación de los hallazgos obtenidos.

3. Resultados

Los resultados se presentan con una intención analítica. En lugar de enumerar únicamente fases o características técnicas, esta sección explica cómo cada parte del marco ayuda a responder la pregunta: bajo qué condiciones conviene migrar un monolito hacia *serverless*.

3.1 Criterios que determinan la conveniencia de migrar

La revisión permitió identificar que la viabilidad de una migración a *serverless* no depende de un solo factor. Un sistema puede ser técnicamente migrable y, aun así, no ser económicamente conveniente. También puede mostrar potencial económico, pero carecer de la madurez organizacional necesaria para operar un modelo distribuido. Por eso, el marco agrupa los criterios en dimensiones que permiten mirar el problema desde distintos ángulos.

La Tabla 1 resume las dimensiones de evaluación y su implicación práctica.

Tabla 1. Dimensiones de evaluación e implicación para la decisión de migración

Dimensión	Qué evalúa	Implicación para la decisión
Viabilidad técnica	Afinidad del sistema con ejecución desacoplada, manejo de estado externo, latencia tolerable, límites de ejecución y dependencias controlables.	Permite valorar si el monolito puede separarse sin degradar la operación ni introducir una complejidad técnica mayor a la que se busca resolver.
Viabilidad económica	Relación entre patrón de consumo, costos actuales, costos proyectados, esfuerzo de migración y punto de equilibrio.	Ayuda a distinguir una modernización rentable de una migración que solo mueve los costos hacia otro modelo operativo.
Viabilidad organizacional	Madurez del equipo, prácticas de CI/CD, observabilidad, gobierno cloud, seguridad y capacidad de operación.	Indica si la organización está preparada para sostener el cambio después de la migración, no solo para construirlo.
Complejidad del monolito	Acoplamiento, tamaño, criticidad, deuda técnica, dependencias internas y claridad de límites funcionales.	Orienta si conviene una migración incremental, un piloto acotado o una modernización previa del monolito.
Riesgos de diseño y operación	<i>Vendor lock-in</i> , seguridad, trazabilidad, eventos, consistencia eventual y recuperación ante fallos.	Ayuda a anticipar riesgos que pueden afectar la estabilidad del sistema y la gobernanza futura.

Fuente: Elaboración propia a partir del documento de investigación y del instrumento de evaluación.

El hallazgo relevante es que *serverless* tiende a ser más atractivo cuando existe variabilidad de demanda, componentes funcionales delimitables, necesidad de escalamiento selectivo y capacidad organizacional para operar con métricas, trazas y automatización. En cambio, pierde fuerza cuando el sistema requiere ejecución prolongada, baja latencia constante, fuerte

comunicación interna, alto estado transaccional o cuando la organización no cuenta con prácticas mínimas de operación cloud. Esta lectura es consistente con la evidencia revisada en la tesis, donde se destaca que *serverless* responde mejor a picos inesperados, mientras que el monolito puede mantener mayor estabilidad en cargas regulares (Manousian, 2022).

3.2 Marco por fases orientado a decisiones

El marco se organiza en fases, pero su valor no está en seguir una secuencia rígida. Su utilidad está en marcar puntos de decisión que evitan avanzar hacia una migración sin evidencia suficiente. Cada fase funciona como un filtro, si una fase muestra riesgos no controlados, la organización puede detenerse, ajustar o replantear la estrategia antes de comprometer una transformación más grande.

Tabla 2. Fases del marco y valor analítico dentro de la decisión

Fase	Propósito	Valor para la toma de decisiones
1. Diagnóstico del monolito	Comprender arquitectura, acoplamiento, estado, dependencias y criticidad.	Evita decidir desde percepciones generales y establece una línea base real del sistema.
2. Evaluación de viabilidad	Aplicar criterios técnicos, económicos y organizacionales.	Determina si la migración es recomendable, condicionada o no recomendable. Esta fase será la que estaremos desarrollando en profundidad y explicando la herramienta de evaluación creada.
3. Selección de estrategia	Definir ruta de transición y patrones de migración.	Alinea el enfoque de migración con el nivel de riesgo, criticidad de la aplicación y nivel de madurez identificado.
4. Planificación	Priorizar capacidades, pilotos, <i>backlog</i> , controles y riesgos.	Convierte la recomendación en una ruta ejecutable, medible y gradual.
5. Ejecución incremental	Migrar componentes o capacidades de forma controlada.	Reduce exposición operativa y permite aprender antes de ampliar la transformación.

6. Validación y operación	Verificar métricas, costos, SLO, seguridad y gobernanza.	Confirma si los beneficios esperados se sostienen en operación real.
----------------------------------	--	--

Fuente: Elaboración propia a partir del documento de investigación y del instrumento de evaluación.

La fase más sensible es la evaluación de viabilidad y es la fase que se estará desarrollando en profundidad. Si se omite, la migración queda expuesta a decisiones guiadas por presión tecnológica o por supuestos no comprobados. En cambio, cuando la evaluación se documenta, el resultado permite justificar tanto la migración como la decisión de no realizarla. Esto es especialmente útil cuando la recomendación debe explicarse ante áreas técnicas, financieras y directivas.

3.3 Instrumento cuantitativo de evaluación

El marco se complementa con una plantilla en Excel, documentada como anexo 1 de la investigación, que convierte criterios de análisis en una puntuación ponderada. La plantilla no sustituye el juicio del arquitecto; más bien ordena la conversación y obliga a relacionar cada puntuación con evidencia mínima del sistema evaluado.

El instrumento utiliza cinco dimensiones con pesos diferenciados. La ponderación da mayor relevancia a la viabilidad técnica y a la complejidad del monolito, sin dejar fuera los factores económicos, organizacionales y de riesgo, sin embargo, estos valores podrán ir mutando en cada organización.

Tabla 3. Dimensiones y pesos del instrumento de evaluación

Dimensión	Peso aproximado	Razón del peso dentro del marco
Viabilidad técnica	30 %	Es la base para determinar si los componentes candidatos pueden operar bajo restricciones <i>serverless</i> sin afectar atributos críticos.
Viabilidad económica	20 %	Permite revisar si el patrón de consumo y el esfuerzo de migración justifican el cambio.
Viabilidad organizacional	15 %	Evalúa si la organización tiene prácticas y capacidades para operar el nuevo modelo.
Complejidad del monolito	20 %	Mide el esfuerzo de separación, dependencia y riesgo estructural de la aplicación actual.

Riesgos y diseño	15 %	Considera impactos de seguridad, observabilidad, proveedor, eventos y operación distribuida.
-------------------------	------	--

Fuente: Elaboración propia a partir del documento de investigación y del instrumento de evaluación.

La escala final se expresa de 0 a 100. Un resultado mayor o igual a 80 indica migración recomendada; un resultado entre 40 y 79 indica migración condicionada; y un resultado menor a 40 indica que no se recomienda migrar. La categoría intermedia es especialmente útil, porque evita una lectura binaria. En muchos casos, la conclusión responsable no es migrar todo ni descartar *serverless*, sino ejecutar un piloto, resolver bloqueos o seleccionar capacidades de menor riesgo.

Tabla 4. Interpretación estratégica del puntaje obtenido

Rango	Resultado	Lectura estratégica	Acción sugerida
80-100	Migración recomendada	El sistema muestra afinidad suficiente con <i>serverless</i> y los riesgos parecen gestionables.	Planificar una migración incremental con componentes candidatos priorizados.
40-79	Migración condicionada	Existe potencial, pero hay brechas técnicas, económicas u organizacionales que deben atenderse antes de ampliar el alcance.	Ejecutar piloto, mitigar riesgos, fortalecer observabilidad, revisar costos o reducir acoplamiento.
0-39	No se recomienda migrar	Los riesgos o costos superan los beneficios esperados en el contexto evaluado.	Documentar la decisión y considerar alternativas como optimización del monolito, mejora de CI/CD, modularización interna, microservicios o modernización parcial.

Fuente: Elaboración propia a partir del documento de investigación y del instrumento de evaluación.

Una vez obtenido el resultado del instrumento, el marco no concluye únicamente con una puntuación. La calificación integra cinco dimensiones de evaluación: viabilidad técnica, viabilidad

económica, viabilidad organizacional, complejidad del monolito y riesgos de diseño. A partir de ese resultado, las secciones siguientes cumplen una función complementaria: primero, interpretar los impactos esperados de *serverless* en el caso evaluado; y luego, relacionar la clasificación obtenida con una estrategia de migración adecuada. De esta forma, el instrumento indica el nivel de viabilidad, mientras que el análisis posterior ayuda a definir qué significa ese resultado y qué acción debería tomarse.

3.4 Comparación de impactos esperados

Después de obtener la clasificación del instrumento, el siguiente paso consiste en interpretar los impactos esperados de la migración. Esta interpretación no sustituye las cinco dimensiones evaluadas en el Excel: viabilidad técnica, viabilidad económica, viabilidad organizacional, complejidad del monolito y riesgos de diseño. Más bien, ayuda a comprender cómo los elementos observados de *serverless* pueden afectar el caso analizado desde una perspectiva técnica, económica y organizacional.

Por esta razón, la Tabla 5 no presenta los beneficios de *serverless* como afirmaciones generales, sino como una comparación entre cada elemento observado y su impacto técnico, económico y organizacional. La finalidad de esta comparación es identificar si cada elemento aporta valor real al monolito evaluado o si introduce restricciones que deben analizarse antes de recomendar la migración.

Tabla 5. Ventajas y limitaciones de *serverless* según impacto en escenarios de migración

Elemento observado	Impacto técnico	Impacto económico	Impacto organizacional	Conclusión aplicada
Escalamiento automático	Alto cuando hay picos de demanda o componentes con consumo variable.	Alto si evita mantener capacidad ociosa.	Medio: exige monitoreo de límites y comportamiento en producción.	Conviene cuando la variabilidad de carga es real y medible, no asumida.
Pago por uso	Medio: no mejora por sí mismo la arquitectura.	Alto en cargas intermitentes; bajo o negativo en cargas constantes.	Medio: requiere seguimiento disciplinado de costos.	Debe validarse con estimaciones y punto de equilibrio antes de decidir.

Reducción de operación de servidores	Medio: simplifica infraestructura, pero no elimina diseño ni operación.	Medio: puede reducir administración, aunque introduce costos de servicios gestionados.	Alto: cambia responsabilidades del equipo hacia observabilidad y gobierno cloud.	Aporta valor si la organización desea reducir administración de infraestructura y puede asumir operación distribuida.
Event-driven y desacoplamiento	Alto si el dominio permite asincronía y límites claros.	Medio: puede mejorar eficiencia, pero añade complejidad si se usa sin criterio.	Alto: exige coordinación entre equipos, contratos y trazabilidad.	Es útil para capacidades específicas; no debe imponerse a todo el monolito.
Cold start, latencia y límites de ejecución	Puede ser crítico en flujos interactivos o de baja latencia.	Puede aumentar costos si se requieren mitigaciones permanentes.	Medio: obliga a definir SLO y monitoreo más preciso.	Debe evaluarse como restricción de diseño, no como detalle técnico secundario.

Fuente: Elaboración propia a partir del documento de investigación y del instrumento de evaluación.

La lectura de la tabla permite observar que ningún elemento debe evaluarse de forma aislada. Por ejemplo, el escalamiento automático puede tener un impacto técnico y económico alto cuando existe variabilidad real de carga, pero también introduce un impacto organizacional porque requiere monitoreo y control de límites. De manera similar, el pago por uso puede ser atractivo desde el impacto económico, pero si la carga es constante o si se requieren mitigaciones adicionales, la conclusión aplicada puede cambiar.

En ese sentido, la conclusión aplicada de cada fila funciona como un criterio de decisión. No basta con identificar que *serverless* ofrece escalamiento automático, pago por uso o reducción de operación de servidores; el marco exige valorar si ese elemento observado produce un beneficio técnico, económico u organizacional en el contexto específico del monolito analizado. Si el

beneficio aparece en una dimensión, pero genera riesgos relevantes en otra, la migración debe condicionarse, ajustarse o incluso descartarse.

Por tanto, esta comparación refuerza la lógica central del marco: *serverless* no se evalúa como una tecnología aislada, sino como una alternativa que debe demostrar valor frente al sistema existente. La decisión de migrar depende de la relación entre cada elemento observado, su impacto técnico, su impacto económico, su impacto organizacional y la conclusión aplicada para el caso concreto.

3.5 Estrategias de migración según el resultado de la evaluación

Una vez interpretados los impactos esperados, el marco utiliza el resultado de la evaluación para orientar la estrategia de migración. En este punto, la decisión ya no se limita a determinar si *serverless* es viable o no, sino a definir cómo proceder según el nivel de riesgo identificado, la criticidad del módulo evaluado, el acoplamiento del sistema y el nivel de confianza obtenido en la evaluación. Por ello, la ejecución no debería plantearse como una sustitución completa e inmediata del monolito, sino como un proceso incremental que permita reducir riesgos, validar supuestos y facilitar el aprendizaje organizacional.

En la siguiente tabla se presenta una síntesis de las estrategias de migración analizadas y los escenarios donde se recomienda su utilización. Sin embargo, su aplicación puede variar según la criticidad del módulo analizado, aspecto que se identifica en la Fase 1, “Diagnóstico del monolito”, del marco de migración.

Tabla 6. Estrategias recomendadas según escenario de viabilidad

Escenario	Estrategia más adecuada	Justificación
Migración recomendada con bajo riesgo	Strangler Fig o migración por capacidades puntuales.	Permite trasladar funcionalidades de forma gradual, validar resultados y mantener rollback disponible.
Migración recomendada con alto valor en eventos	Migración orientada a eventos.	Aprovecha la afinidad de <i>serverless</i> con procesamiento asincrónico y desacoplamiento.
Migración condicionada	Piloto controlado o shadow traffic.	Reduce incertidumbre antes de exponer usuarios o procesos críticos.

Alta criticidad funcional	Ejecución paralela, shadow traffic y comparación de resultados.	Permite comprobar equivalencia funcional antes de mover tráfico real.
No recomendada	No ejecutar migración; priorizar mejoras internas.	Evita introducir complejidad distribuida cuando los beneficios no compensan los riesgos.

Fuente: Elaboración propia a partir del documento de investigación y del instrumento de evaluación.

El uso de patrones como Strangler Fig, Bounded Context y Branch by Abstraction se entiende dentro del marco como una forma de controlar riesgo, no como una lista obligatoria de técnicas. Su valor está en permitir que la organización migre solo aquello que demuestra valor, manteniendo continuidad operativa y posibilidad de reversión.

4. Discusión

Los resultados refuerzan una idea central la cual indica que *serverless* no debe analizarse solo como una tecnología de reemplazo, sino como una decisión de modernización. Esta diferencia cambia la forma de evaluar el problema. Una explicación técnica puede mostrar que una función en la nube escala automáticamente, pero la pregunta relevante es ¿qué significa ese escalamiento para el negocio, los costos, el equipo y la estabilidad del sistema?.

Desde una perspectiva técnica, el marco ayuda a identificar cuándo la arquitectura monolítica se ha convertido en un obstáculo para evolucionar el producto. No todos los monolitos deben migrarse. Algunos pueden seguir siendo adecuados si el dominio es estable, la carga es predecible y la complejidad distribuida no aporta beneficios claros. Pero cuando existen componentes con demanda variable, límites funcionales claros y necesidad de despliegues más independientes, *serverless* puede ser una alternativa razonable. La decisión, por tanto, no depende de la edad del sistema, sino de la relación entre sus restricciones actuales y los beneficios esperados.

Desde una perspectiva económica, la implicación principal es que el pago por uso debe evaluarse con datos. *Serverless* puede reducir costos cuando evita capacidad ociosa, pero también puede generar gastos difíciles de anticipar si el volumen es constante, si se agregan varios servicios gestionados o si se requieren mitigaciones para latencia y disponibilidad. Por eso, la plantilla obliga a revisar carga, punto de equilibrio y esfuerzo de migración. Así, la conversación

económica no se limita al costo de infraestructura, sino que incluye mantenimiento, operación, riesgo y velocidad de entrega.

Desde una perspectiva organizacional, el cambio más importante es que la migración a *serverless* redistribuye responsabilidades. La organización puede administrar menos servidores, pero necesita mayor disciplina en observabilidad, automatización, seguridad, gobierno cloud y seguimiento de costos. En otras palabras, *serverless* no elimina la operación; cambia su naturaleza. Por eso, la migración debe entenderse como una decisión sociotécnica, donde la madurez del equipo, la gobernanza y la alineación de actores son tan importantes como la solución técnica seleccionada (Hamza et al., 2025). Esta idea es crítica para la toma de decisiones, porque un equipo sin madurez operativa puede terminar con un sistema más fragmentado y más difícil de controlar que el monolito original.

El marco también ayuda a justificar cuándo es mejor no migrar. En muchas organizaciones, modernizar se asocia con avanzar siempre hacia nuevas arquitecturas. Sin embargo, una evaluación rigurosa puede concluir que no conviene migrar en ese momento. Esa conclusión no debería verse como una debilidad. Al contrario, puede proteger a la organización de invertir en una transformación con bajo retorno o alto riesgo. En ese sentido, el instrumento no solo ayuda a iniciar migraciones; también ayuda a detenerlas cuando no están justificadas.

Finalmente, el valor aplicado del marco está en conectar evidencia técnica con lenguaje de decisión. Los arquitectos necesitan explicar sus recomendaciones a públicos distintos: desarrollo, operaciones, seguridad, finanzas y dirección. Al separar dimensiones y exigir evidencia por criterio, el marco facilita una conversación común. La migración deja de depender solo de una preferencia técnica y pasa a ser una decisión documentada, comparable y revisable.

5. Conclusiones

Este artículo presentó un marco de migración de aplicaciones monolíticas hacia arquitecturas *serverless* en la nube, orientado a apoyar decisiones de modernización desde una perspectiva aplicada. La mejora principal del marco es que no asume *serverless* como una solución universal, sino como una alternativa que debe evaluarse según condiciones técnicas, económicas y organizacionales.

Los resultados muestran que la viabilidad de migrar no depende solamente de que el sistema pueda ejecutarse en funciones o servicios gestionados. Depende de la relación entre el patrón de carga, el manejo de estado, el acoplamiento del monolito, el costo esperado, la madurez del

equipo y la capacidad de operar una arquitectura distribuida. Esta mirada multidimensional ayuda a evitar decisiones simplificadas y facilita una ruta de modernización más responsable.

El instrumento cuantitativo en Excel constituye el aporte práctico más relevante, porque convierte criterios de arquitectura en una evaluación trazable. Sus rangos de decisión permiten distinguir entre migraciones recomendadas, condicionadas y no recomendadas. Esta clasificación es útil porque reconoce que muchas organizaciones no necesitan una migración total e inmediata, sino una estrategia gradual, un piloto controlado o incluso una decisión explícita de no migrar.

En términos estratégicos, el marco ayuda a alinear modernización tecnológica con valor organizacional. Permite que las decisiones no se justifiquen solo por características de cloud, sino por su impacto real en costos, escalabilidad, mantenibilidad, resiliencia y capacidad operativa. Como línea futura, se recomienda aplicar el marco en estudios de caso reales para contrastar las puntuaciones del instrumento con resultados observados en producción y ajustar pesos o criterios según evidencia empírica.

Referencias

- Hamza, M. A. (2023). The journey to serverless migration: An empirical analysis of intentions, strategies, and challenges. <https://arxiv.org/pdf/2311.13249>
- Hamza, M. A. (2025). Navigating decision-making in serverless migration: A socio-technical grounded theory approach. <https://papers.ssrn.com/sol3/Delivery.cfm/5161731.pdf?abstractid=5161731&mirid=1>
- Manousian, J. (2022). Investigating differences in performance between monolithic and serverless based architectures. <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1692921&dswid=4207>
- Orozco Montenegro, R. (2026). Plantilla_Evaluacion_Migracion. Zenodo. <https://doi.org/10.5281/zenodo.18843253>

Anexo A. Instrumento de evaluación de viabilidad

Instrumento de evaluación de viabilidad (Excel)

Se adjunta la plantilla **Plantilla_Evaluacion_Migracion.xlsx**, compuesta por dos hojas:

1. **Evaluación Serverless**: instrumento principal (solo se llena columna D, 0–5).

Figura 1. Evaluación Serverless

Plantilla de Evaluación Cuantitativa – Migración Monolito → Serverless						
Instrucciones: Complete únicamente la columna D (Puntuación 0-5). La nota global y recomendación se calculan automáticamente.						
Sección B – Viabilidad técnica (30%)						
ID	Criterio	Descripción corta	Puntuación (0-5)	Peso (%)	Puntaje ponderado	Notas / Evidencia
B1	Variabilidad del workload	Picos y ociosidad vs carga estable		6	0	
B2	Latencia P99	Rendimiento extremo (peor caso típico)		5	0	
B3	Cold start promedio	Penalidad de arranque en frío		5	0	
B4	Externalización de estado	Facilidad para volver stateless		5	0	
B5	Compatibilidad técnica	Lenguajes, runtimes, dependencias		5	0	
B6	Tamaño del artefacto	Paquete vs límites comunes		4	0	
				TOTAL sección	0	
Sección C – Viabilidad económica (20%)						
ID	Criterio	Descripción corta	Puntuación (0-5)	Peso (%)	Puntaje ponderado	Notas / Evidencia
C1	Costo por millón de invocaciones	Proyección base del costo serverless		6	0	
C2	Duración promedio de ejecución	Driver económico por invocación		5	0	
C3	Punto de equilibrio (break-even)	Volumen donde deja de convenir		5	0	
C4	Riesgo por sobreuso	Picos inesperados y costos		4	0	
				TOTAL sección	0	
Sección D – Viabilidad organizacional (15%)						
ID	Criterio	Descripción corta	Puntuación (0-5)	Peso (%)	Puntaje ponderado	Notas / Evidencia
D1	Madurez cloud del equipo	Experiencia en cloud y distribuido		5	0	
D2	Automatización CI/CD	Pipelines, pruebas, despliegue		4	0	
D3	Capacidad event-driven	Eventos, colas, orquestación		3	0	
D4	Documentación de flujos críticos	Procesos, integraciones, dominio		3	0	
				TOTAL sección	0	
Sección E – Complejidad del monolito (20%)						
ID	Criterio	Descripción corta	Puntuación (0-5)	Peso (%)	Puntaje ponderado	Notas / Evidencia
E1	Acoplamiento interno	Dependencias entre módulos		7	0	
E2	Dependencias internas/externas	Servicios, librerías, integraciones		5	0	
E3	% módulos con estado interno	Sesión/estado en memoria		4	0	
E4	% endpoints a refactorizar	Reescritura/ajuste de contratos		4	0	
				TOTAL sección	0	
Sección F – Riesgos y diseño (15%)						
ID	Criterio	Descripción corta	Puntuación (0-5)	Peso (%)	Puntaje ponderado	Notas / Evidencia
F1	% funciones fuera de límites	Duración/timeout incompatibles		5	0	
F2	Observabilidad	Métricas, trazas, logs correlados		4	0	
F3	Riesgo operativo	Rollback, fallos, criticidad		3	0	
F4	Claridad del diseño objetivo	Arquitectura target definida		3	0	
				TOTAL sección	0	
NOTA GLOBAL			0			
RECOMENDACIÓN: No se recomienda migrar						

Fuente: Elaboración propia.

2. **Métricas avanzadas:** registro de mediciones y guía para traducir valores a puntuaciones.

Figura 2. Métricas avanzadas.

Métricas avanzadas (registro) – valores medidos y evidencias						
Use esta hoja para documentar mediciones (ms, %, MB, etc.) y luego traducirlas a puntuación 0-5 en la hoja principal.						
Métrica	Unidad	Valor medido	Cómo medir (resumen)	Valores de referencia (si aplica)	Herramientas sugeridas	
Cold start promedio	ms		Primera invocación tras inactividad	<150 favorable; 150-500 aceptable; >500 riesgoso (interactivo)	OpenTelemetry, logs, k6	
Warm start promedio	ms		Invocaciones repetidas sin inactividad	Mientras más bajo, mejor	OpenTelemetry, logs	
Ratio cold/warm	ratio		ColdStart / WarmStart	<3 estable; ≥5 variable	Grafana, hojas de cálculo	
Latencia P99	ms		Percentil 99 en pruebas/producción	<200 favorable; 200-500 aceptable; >500 riesgoso	k6, JMeter, Grafana	
Concurrency sostenible	req simultáneas		Prueba de carga hasta degradación	Definir SLO propio (ej. 100 req) y validar	k6, JMeter	
Variabilidad workload (pico/prom)	ratio		Max req/h / Prom req/h	>5 favorable; <2 baja afinidad	Prometheus, Grafana	
Tamaño del artefacto	MB		Paquete desplegable comprimido	≤50 recomendado; 50-100 riesgoso; >100 no recomendado	Build tools	
Duración promedio de ejecución	ms		Promedio de tiempo de ejecución	<200 favorable; >1000 alto impacto	APM, logs	
Costo por millón de invocaciones	moneda		Modelo de costo vs uso	≤ costo actual favorable; ≥+20% desfavorable	Excel/Calc	
Break-even	req/mes		Comparar costos acumulados	Break-even > tráfico esperado favorable	Excel/Calc	
% módulos con estado	%		Inventario y análisis de sesión/estado	<40% favorable; >60% alto esfuerzo	Revisión arquitectura	
Acoplamiento interno	0-10		Dependencias entre módulos (métrica o juicio guiado)	≤3 bajo; ≥7 alto	SonarQube, CodeQL	

Fuente: Elaboración propia.

Link de descarga: Orozco Montenegro, R. (2026). Plantilla_Evaluacion_Migracion.

Zenodo. <https://doi.org/10.5281/zenodo.18843253>