



**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**  
**LICENCIATURA EN INGENIERÍA DE SOFTWARE**  
**TÓPICOS ESPECIALES**

**INVESTIGACION # 1**  
**ARQUITECTURA MONOLÍTICA VS MICROSERVICIO**

**ESTUDIANTE:**  
**PÉREZ, ELIECER 8-1003-1720**

**PROFESOR:**  
**IBRAIN LIN**

**5 DE AGOSTO DEL 2025**

## Índice

Introducción.....	2
Arquitectura monolítica vs. microservicios.....	3
Arquitectura monolítica.....	3
Arquitectura de microservicios.....	3
Beneficios de la arquitectura monolítica .....	4
Beneficios de la arquitectura de microservicios .....	5
Migrar a la arquitectura de microservicios .....	6
Conclusión .....	7
Bibliografía.....	8

## **Introducción**

La evolución de la arquitectura de software ha sido fundamental para responder a las crecientes demandas de eficiencia, escalabilidad y mantenimiento de las aplicaciones modernas. Entre los modelos más relevantes se encuentran la arquitectura monolítica y la de microservicios, cada una con características, ventajas y limitaciones particulares. Mientras que la arquitectura monolítica representa un enfoque tradicional, centralizado y cohesivo, los microservicios proponen una estructura modular, flexible y distribuida que permite a las organizaciones adaptarse rápidamente a cambios en los requisitos del negocio y en el entorno tecnológico. Esta investigación tiene como objetivo analizar y comparar ambos enfoques, destacando sus beneficios, desafíos y la relevancia actual de la adopción de microservicios, así como las estrategias para migrar de un sistema monolítico a uno basado en microservicios, con el fin de proporcionar un panorama integral que apoye la toma de decisiones en proyectos de software contemporáneos.

## **Arquitectura monolítica vs. microservicios**

Para realizar una comparación entre estas dos arquitecturas, que presentan diferentes formas de organización, primero es necesario entender los conceptos por separado.

### **Arquitectura monolítica**

Las arquitecturas monolíticas se basan en un modelo centralizado donde todas las funciones, servicios y componentes de una aplicación desde la interfaz de usuario hasta la lógica de negocio y el acceso a datos se integran dentro de una misma base de código. Este enfoque, característico del desarrollo tradicional de software, ofrece una estructura unificada que facilita la implementación y el mantenimiento inicial, ya que todo el sistema se despliega como una única entidad en un servidor principal que gestiona todas las solicitudes y respuestas.

La comunicación interna dentro de este tipo de arquitectura se realiza mediante llamadas directas a funciones o a través de objetos compartidos, lo cual permite un control claro del flujo de ejecución. Sin embargo, esta interdependencia genera un acoplamiento estrecho entre los componentes, dificultando la aplicación de principios como los SOLID, que promueven la baja dependencia y la alta cohesión para lograr sistemas escalables y sostenibles.

Aunque el diseño monolítico simplifica la gestión en etapas tempranas y ofrece eficiencia operativa, presenta limitaciones significativas en términos de escalabilidad y flexibilidad. Cada modificación o actualización requiere desplegar nuevamente toda la aplicación, lo que puede ser ineficiente y propenso a errores. Además, su escalabilidad suele ser vertical, es decir, se mejora el rendimiento agregando más recursos al mismo servidor, lo cual tiene límites físicos y económicos. La escalabilidad horizontal, que consiste en añadir más instancias del sistema, resulta más compleja de implementar en este tipo de arquitectura.

### **Arquitectura de microservicios**

La arquitectura de microservicios surge como un enfoque moderno y flexible que contrasta con el modelo monolítico tradicional. Su principio fundamental consiste en dividir una aplicación en múltiples servicios independientes, donde cada uno cumple una función específica y posee su propia lógica de negocio y base de datos. Esta segmentación modular

favorece la escalabilidad, el mantenimiento y la implementación, ya que cada microservicio puede desarrollarse, probarse y desplegarse de manera autónoma.

En este modelo, los servicios se comunican entre sí mediante interfaces bien definidas o APIs, comúnmente basadas en protocolos como HTTP o SOAP, lo que facilita la interoperabilidad sin generar acoplamiento excesivo. Esta independencia permite que los equipos de desarrollo trabajen de forma paralela, actualizando o mejorando servicios sin afectar el funcionamiento del resto del sistema.

Una de las principales ventajas de esta arquitectura es su capacidad para escalar horizontalmente, asignando recursos únicamente a los servicios que lo requieran según la demanda. De esta forma, se optimiza el uso de los recursos del sistema, a diferencia del modelo monolítico que obliga a escalar toda la aplicación incluso si solo una parte necesita mayor rendimiento. Además, la modularidad de los microservicios impulsa la implementación continua, facilitando la incorporación rápida de nuevas funcionalidades y mejorando la capacidad de respuesta ante los cambios del negocio o del entorno operativo.

Otro beneficio importante es la resiliencia. Si un microservicio falla, los demás pueden continuar operando de manera independiente, evitando que un error aislado afecte a toda la aplicación. Asimismo, cada servicio puede incluir sus propios mecanismos de manejo de errores y recuperación, fortaleciendo la robustez general del sistema.

No obstante, este modelo también implica nuevos desafíos. La fragmentación del sistema en múltiples servicios incrementa la complejidad operativa, exigiendo una gestión cuidadosa de la comunicación entre componentes y un diseño claro de las interfaces. Además, el desarrollo basado en microservicios suele requerir más tiempo y recursos, ya que cada servicio debe diseñarse, probarse y desplegarse por separado. Esto demanda equipos técnicos experimentados y una infraestructura adecuada para coordinar los distintos elementos del sistema, lo que puede aumentar la dificultad de mantenimiento y la probabilidad de errores si no se gestiona correctamente.

### **Beneficios de la arquitectura monolítica**

La arquitectura monolítica, a pesar de sus limitaciones, presenta varias ventajas significativas, especialmente en proyectos de menor escala o con requisitos estables. Al

concentrar todos los componentes de la aplicación en un único bloque de código, el desarrollo inicial resulta más sencillo y comprensible, facilitando la depuración y la coordinación entre los diferentes elementos del sistema. El despliegue se realiza de manera centralizada, ya que toda la aplicación se implementa como una sola entidad en un servidor, lo que simplifica la gestión, la configuración y el seguimiento del sistema. Esta concentración en un servidor único permite una eficiencia operativa notable, dado que las solicitudes y respuestas se manejan de forma centralizada, reduciendo la sobrecarga de comunicación interna entre componentes. Además, la menor complejidad técnica, al no requerir la gestión de múltiples servicios, bases de datos separadas o interfaces de comunicación complejas, facilita la curva de aprendizaje y la organización de los equipos de desarrollo. Por último, esta simplicidad inicial también se traduce en un costo reducido en las primeras etapas del proyecto, al requerir menos infraestructura y herramientas de orquestación. En síntesis, la arquitectura monolítica resulta adecuada para sistemas pequeños o con funcionalidades limitadas, donde la eficiencia operativa, la simplicidad en el desarrollo y la facilidad de despliegue predominan sobre la necesidad de escalabilidad o flexibilidad avanzada.

### **Beneficios de la arquitectura de microservicios**

La adopción de la arquitectura de microservicios proporciona soluciones efectivas a las limitaciones de las aplicaciones monolíticas. Al dividir la aplicación en servicios independientes con límites claros, se facilita tanto el desarrollo como el mantenimiento, ya que los equipos de trabajo pueden operar de manera autónoma sin depender de otras áreas del sistema. Esta modularidad permite incluso utilizar diferentes lenguajes de programación o tecnologías en distintos microservicios, optimizando la aplicación en términos de rendimiento, funcionalidad y mantenibilidad.

Otro beneficio significativo es la implementación independiente de cada servicio, lo que mejora la rapidez de despliegue y permite escalar únicamente los componentes que lo requieren, aumentando la eficiencia y la capacidad de respuesta del sistema. Además, esta arquitectura favorece la modernización gradual de aplicaciones heredadas y la reutilización de servicios empresariales, facilitando la integración de nuevas funcionalidades y la adaptación a cambios en los requisitos del negocio.

En síntesis, los microservicios ofrecen mayor escalabilidad, agilidad y velocidad de entrega, constituyendo una alternativa flexible y eficiente frente a las restricciones de los sistemas monolíticos tradicionales.

### **Migrar a la arquitectura de microservicios**

En la actualidad, la arquitectura de microservicios se considera la opción más eficiente y flexible para el desarrollo de aplicaciones complejas, debido a su capacidad de escalabilidad, resiliencia y despliegue independiente de cada servicio. Este enfoque permite a las organizaciones adaptar sus sistemas rápidamente a las necesidades del negocio, optimizando recursos y acelerando la entrega de nuevas funcionalidades, a diferencia de las arquitecturas monolíticas, que presentan limitaciones en términos de flexibilidad y escalabilidad. La migración desde un sistema monolítico hacia microservicios suele realizarse de manera gradual, comenzando por identificar los componentes críticos que pueden convertirse en servicios independientes, definiendo límites claros de responsabilidad y estableciendo interfaces de comunicación mediante APIs. Posteriormente, se implementan mecanismos de orquestación y gestión de servicios, asegurando que cada microservicio pueda operar de manera autónoma sin afectar la integridad del sistema completo. Esta transición escalonada no solo permite minimizar riesgos durante la migración, sino que también facilita la modernización progresiva de aplicaciones heredadas, promoviendo un entorno más ágil, modular y resistente a fallos, en línea con las tendencias actuales del mercado global de microservicios en la nube.

## **Conclusión**

La comparación entre la arquitectura monolítica y la de microservicios evidencia que, aunque los sistemas monolíticos pueden resultar adecuados para proyectos pequeños o con requerimientos estables, presentan limitaciones significativas en términos de escalabilidad, flexibilidad y mantenimiento a largo plazo. Por su parte, la arquitectura de microservicios ofrece un enfoque más moderno y eficiente, permitiendo dividir la aplicación en servicios independientes, escalables y resilientes, con despliegue y desarrollo autónomo. La adopción de microservicios no solo mejora la capacidad de adaptación a cambios del negocio, sino que también facilita la modernización de sistemas heredados y la reutilización de componentes. No obstante, esta arquitectura requiere de una planificación cuidadosa, herramientas adecuadas y equipos experimentados para gestionar la complejidad operativa y de desarrollo. En definitiva, los microservicios representan la opción más viable para el desarrollo de aplicaciones complejas en la actualidad, contribuyendo a entornos de software más ágiles, modulares y sostenibles, alineados con las tendencias globales del mercado tecnológico.



## **Bibliografía**

Torassa Colombero, V., Estelles, J. P., Gallegos, L., & López, P. (2024). Monolitos vs. Microservicios en Arquitectura de Software: Perspectivas para un desarrollo eficiente. In *Concurso de Trabajos Estudiantiles (EST 2024)-JAIIO 53 (Universidad Nacional del Sur, 12 al 16 de agosto de 2024)*.

Gonzalez, R. A., Giménez, S. A., Molina Pualuk, N. R., & Zalazar, R. (2024). ¿ Son los microservicios la mejor opción?. In *Concurso de Trabajos Estudiantiles (EST 2024)-JAIIO 53 (Universidad Nacional del Sur, 12 al 16 de agosto de 2024)*.

Peña Huérfano, L. D. Metodología para la migración de aplicaciones monolíticas a sistemas basados en microservicios.