



Universidad Nacional Autónoma de México



Facultad de Ingeniería

Ingeniería en Computación

# Sistemas Operativos

**Profesor: Ing. Gunnar Eyal Wolf Isazaevich**

Sistemas operativos en la nube

Realizó:

Cano Nieto Carlos Arturo

Cortés Bolaños Angel David

Semestre 2025-2

Fecha de entrega: 18 de abril de 2025

# Sistemas Operativos en la Nube: Caso de Estudio OSv

---

## Introducción

En la era de la computación en la nube, uno de los pilares ha sido la virtualización de sistemas operativos completos (Linux, Windows) sobre hipervisores (Xen, KVM, VMware). Esto permitió la consolidación de servidores, migración en vivo y aislamiento multi-tenant. Sin embargo, montar un sistema operativo genérico dentro de una máquina virtual añade capas de abstracción poco necesarias para muchas aplicaciones:

- Soporte de protocolos físicos antiguos (IDE, SCSI).
- Optimizaciones históricas irrelevantes (algoritmos de “elevador de disco” en SSD).
- Interfaces de compatibilidad (POSIX, Win32).
- Espacios de usuario con procesos e hilos que casi siempre ejecutan una sola tarea final.
- Runtimes gestionados (Java, .NET).

Todas estas capas conviven bajo cada VM (Virtual Machine), generando sobrecarga de memoria, arranque lento y mayor superficie de ataque .

Los unikernels o sistemas operativos mínimos (a veces llamados *library OS*) nacen de la idea de reducir el sistema operativo hasta lo esencial, los cuales están optimizados para ejecutar una única aplicación. A diferencia de los sistemas operativos tradicionales como Linux o Windows, los sistemas mínimos eliminan capas innecesarias, resultando en menor consumo de recursos y mayor rendimiento. En este contexto, OSv se presenta como un caso destacado, al ser un sistema operativo diseñado desde cero para la nube, con características únicas que permiten ejecutar aplicaciones con gran eficiencia y simplicidad.

## ¿Qué es OSv?

OSv es un sistema operativo minimalista orientado a la nube. Está diseñado para ejecutar una sola aplicación por máquina virtual, maximizando la eficiencia y eliminando la sobrecarga típica de los sistemas tradicionales. OSv permite correr aplicaciones Linux sin modificar su código fuente, siempre que estén compiladas de forma dinámica y como ejecutables tipo PIE (Position Independent Executable). Opera en un único espacio de direcciones, lo que significa que no existe distinción entre el espacio del kernel y el de la aplicación.

## Ventajas de OSv

Entre las principales ventajas de OSv destaca su alto rendimiento. Al eliminar la separación entre kernel y usuario, las llamadas al sistema se sustituyen por simples llamadas a funciones, reduciendo la latencia y el consumo de CPU. Además, el sistema no necesita manejar múltiples usuarios o procesos, lo que simplifica enormemente su arquitectura. La eficiencia en el uso de memoria también es notable, con el uso de páginas grandes (2MB) que reducen el overhead de la paginación. OSv también reduce la superficie de ataque, al incluir solo el código necesario para la aplicación, mejorando la seguridad.

## Desventajas de OSv

Pese a sus beneficios, OSv tiene limitaciones importantes. Su compatibilidad es reducida en comparación con sistemas tradicionales. No soporta ejecutables estáticos convencionales ni múltiples procesos, lo cual restringe el tipo de aplicaciones que pueden correr en él. Tampoco tiene drivers para hardware físico, ya que depende completamente del entorno virtualizado. Estas restricciones hacen que no sea adecuado para todos los escenarios, especialmente aquellos que requieren alta flexibilidad o soporte de dispositivos diversos.

## **Beneficios en rendimiento y eficiencia**

Uno de los principales logros de OSv es la optimización del rendimiento. Gracias a su arquitectura, los cambios de contexto entre hilos son de 3 a 10 veces más rápidos que en Linux. Las llamadas a funciones del sistema no requieren interrupciones ni cambios de modo, reduciendo significativamente el tiempo de ejecución. La gestión de memoria también está optimizada, utilizando grandes bloques contiguos y soporte eficiente de mmap y malloc, lo que mejora el desempeño general de las aplicaciones.

## **Impacto en tecnologías modernas: serverless y contenedores**

OSv se alinea muy bien con tecnologías modernas como el cómputo sin servidor (serverless) y los contenedores ligeros. En entornos serverless, donde las funciones deben iniciar y ejecutarse rápidamente, OSv proporciona un tiempo de arranque casi instantáneo y un consumo mínimo de recursos. En el caso de los contenedores, OSv puede verse como una alternativa más eficiente al usar un sistema operativo mínimo embebido con la aplicación, eliminando la necesidad de una capa intermedia como Docker más Linux.

## **Reducción de la carga administrativa**

Otro beneficio importante de OSv es la simplificación en la gestión del sistema. Al no tener que preocuparse por múltiples usuarios, paquetes del sistema, o configuraciones complejas, los administradores pueden enfocarse en desplegar y mantener únicamente la aplicación. Esto reduce errores, necesidades de actualización, y costos operativos. OSv puede ser gestionado mediante una interfaz web o API REST que permite monitorear y controlar el sistema de forma remota.

Un ejemplo práctico de la aplicación de OSv es su uso en empresas que ejecutan aplicaciones Java, como OpenJDK. Gracias a su diseño, OSv ofrece un entorno eficiente y controlado para correr aplicaciones Java sin necesidad

de todo el ecosistema Linux. Esto mejora el rendimiento y la confiabilidad, especialmente en ambientes donde se requiere alta escalabilidad.

## Conclusión

OSv representa una innovación importante en el diseño de sistemas operativos adaptados a la nube. Al enfocarse en una sola aplicación y eliminar las capas innecesarias, ofrece ventajas significativas en cuanto a rendimiento, simplicidad y eficiencia. Sin embargo, sus limitaciones en compatibilidad y flexibilidad hacen que no sea adecuado para todos los casos de uso. Para aplicaciones en la nube, especialmente en entornos serverless y microservicios, OSv puede ser una excelente opción que reduce la complejidad operativa y mejora los tiempos de respuesta. Como parte de una arquitectura moderna, OSv ejemplifica cómo repensar el sistema operativo puede traer grandes beneficios en la computación en la nube.

## Referencias

Kivity, A., Laor, D., Gud, M., & Bellard, F. (2014). *OSv—Optimizing the Operating System for Virtual Machines*. USENIX Annual Technical Conference. <https://www.usenix.org/conference/atc14/technical-sessions/presentation/kivity>

Avi Kivity, Yaniv Kamay y col. (2007). «KVM: The Linux Virtual Machine Monitor». En: Proceedings of the Linux Symposium. Qumranet / IBM. URL: <http://kernel.org/doc/ols/2007/ols2007v1-pages-225-230.pdf>

Madhavapeddy, A., & Scott, D. J. (2013). Unikernels: Rise of the Virtual Library Operating System. *Queue*, 11(11), 30–44. <https://doi.org/10.1145/2557963.2566628>

OSv.io. (2015). *OSv: The operating system designed for the cloud*. Recuperado de: <https://github.com/cloudius-systems/osv>.