

Carrera: Tecnología en Desarrollo de Software

Nivel: III Semestre

Nombre del Estudiante: Segura Pesantes Edson Gabriel

Asignatura: SISTEMAS OPERATIVOS

Docente: ING. ERNESTO ROBLES

2025

2026

Investigación sobre Escalamiento Vertical y Horizontal, Máquinas Virtuales y Contenedores

1. ¿Qué es crecimiento vertical (scale-up) y crecimiento horizontal (scale-out)?

El crecimiento vertical, también conocido como *scale-up*, es una estrategia que consiste en ampliar la capacidad de un mismo servidor. Esto se logra al agregar más memoria RAM, aumentar la cantidad de procesadores o ampliar el almacenamiento disponible en la máquina que ya se tiene. Es una forma rápida y directa de mejorar el rendimiento de un sistema, aunque tiene un límite físico, ya que llega un punto en que no se pueden añadir más recursos.

El crecimiento horizontal, o *scale-out*, implica sumar más servidores o instancias que trabajen en conjunto para dividir la carga de trabajo. En lugar de depender de un solo equipo más potente, se agregan varios que cooperan para atender a los usuarios. Este modelo permite crecer prácticamente de manera ilimitada y mejora la disponibilidad, ya que, si una máquina falla, las demás siguen funcionando.

2. Ejemplos de servicios en la nube que implementan ambos enfoques

- **Escalamiento vertical (scale-up):**
 - En **Amazon EC2 (AWS)**, un administrador puede pasar de una instancia básica como t2.micro a otra más robusta como m5.large, obteniendo más memoria RAM, CPU y capacidad de procesamiento sin necesidad de cambiar de infraestructura.
 - En **Google Cloud Platform (GCP)**, es posible ajustar las características de una máquina virtual aumentando núcleos de CPU o asignando más gigabytes de RAM para mejorar el rendimiento de una aplicación.
- **Escalamiento horizontal (scale-out):**
 - En Microsoft Azure, una aplicación web puede desplegarse en múltiples instancias balanceadas con un *load balancer*, lo que garantiza que el tráfico se distribuya de manera equitativa y el servicio no se interrumpa en caso de sobrecarga.

- En Kubernetes, el *Horizontal Pod Autoscaler* crea más contenedores cuando la demanda aumenta, garantizando que las aplicaciones mantengan un buen nivel de rendimiento incluso con picos de tráfico.

3. Diferencia entre máquina virtual y contenedor

Una máquina virtual (VM) es un entorno que emula un hardware completo, lo que permite instalar un sistema operativo propio (llamado *guest OS*) y ejecutar aplicaciones sobre él. Para funcionar, requiere de un software llamado *hipervisor*, que se encarga de gestionar las VMs sobre el sistema físico. Las VMs ofrecen gran aislamiento y compatibilidad con diferentes aplicaciones, pero su principal desventaja es que consumen muchos recursos y tardan más tiempo en iniciar.

En cambio, un contenedor es un entorno mucho más ligero que comparte el mismo kernel del sistema operativo del host, pero mantiene aisladas las aplicaciones y todas sus dependencias. Los contenedores se inician en cuestión de segundos, consumen pocos recursos y son muy portables, ya que pueden ejecutarse en cualquier infraestructura donde esté instalada una herramienta como Docker. Esto los hace ideales para entornos de desarrollo y despliegue continuo.

Ejemplos de empresas usando máquinas virtuales y contenedores

1. Máquinas virtuales (VMs):

- **Bancos y entidades financieras:** muchos bancos, como Banco Santander o BBVA, utilizan máquinas virtuales para ejecutar sus aplicaciones críticas. Esto se debe a que requieren un aislamiento fuerte, alta seguridad y compatibilidad con software antiguo que no funciona en contenedores.
- **Gobierno y organismos públicos:** sistemas de administración de datos, registros civiles y plataformas de servicios en línea también suelen usar VMs, porque necesitan entornos estables y controlados.

2. Contenedores:

- **Netflix:** utiliza contenedores para sus microservicios, lo que les permite desplegar rápidamente nuevas funciones y escalar según la demanda de usuarios que ven contenido en streaming.

- **Spotify:** sus servicios de música en streaming se ejecutan en contenedores, facilitando que los equipos de desarrollo desplieguen cambios constantes sin afectar el servicio global.
- **Airbnb:** usa contenedores para ejecutar su plataforma web y móvil, permitiendo un desarrollo ágil y escalabilidad automática cuando aumenta el tráfico de usuarios.

4. Tres aspectos clave para elegir entre VMs y contenedores

1. Consumo de recursos:

- Las máquinas virtuales requieren más memoria y CPU porque cada una incluye su propio sistema operativo. Esto hace que en un mismo servidor se puedan ejecutar menos VMs.
- Los contenedores son mucho más ligeros, permiten ejecutar una mayor cantidad en el mismo hardware y aprovechar mejor los recursos disponibles.

2. Portabilidad y rapidez de despliegue:

- Una máquina virtual puede tardar varios minutos en arrancar y no siempre resulta sencilla de mover entre diferentes plataformas de nube.
- Un contenedor arranca en segundos y puede ejecutarse prácticamente igual en entornos de desarrollo, pruebas o producción, facilitando la portabilidad y la consistencia del software.

3. Escalabilidad y administración:

- Las VMs son recomendables en aplicaciones que necesitan aislamiento total o compatibilidad con sistemas antiguos, pero su administración puede ser más complicada y lenta.
- Los contenedores son ideales para arquitecturas modernas basadas en microservicios. Se integran fácilmente con orquestadores como Kubernetes, que permiten escalar automáticamente el número de contenedores según la demanda del sistema.