



# PROYECTO SEGUNDO CORTE

## Desarrollo de un Entorno Híbrido de Pruebas para Análisis de Sistemas Operativos y

### Contenedores

Presentado a: Ing. Diego Alejandro Barragan Vargas  
Juan Diego Báez Guerrero, Cód.: 2336781.

**Resumen—** Este documento presenta el desarrollo de un entorno híbrido de pruebas para sistemas operativos y arquitecturas de contenedores. Se analiza la instalación virtualizada de Rocky Linux, Manjaro Linux y Arch Linux mediante QEMU, así como contenedores Docker con imágenes de Garuda, Alpine y Debian. Un contenedor central con Fedora proporciona monitoreo de red y análisis de rendimiento del sistema mediante Grafana, Prometheus y Zabbix. El estudio evalúa la gestión de recursos, el comportamiento de la red y el monitoreo de servicios en diferentes arquitecturas de sistemas operativos.

**Abstract—** This paper presents the development of a hybrid testing environment for operating systems and container-based infrastructures. The analysis includes virtualized installations of Rocky Linux, Manjaro Linux, and Arch Linux using QEMU, as well as Docker containers with Garuda, Alpine, and Debian images. A central container system running Fedora provides network monitoring and system performance analysis using Grafana, Prometheus, and Zabbix. The study evaluates resource management, network behavior, and service monitoring in different operating system architectures.

## I. Introducción

En el ámbito de la administración de infraestructuras virtualizadas, los sistemas operativos y los contenedores han desempeñado un papel fundamental en la evolución de entornos computacionales. La virtualización permite el despliegue de múltiples sistemas sobre un único hardware físico, lo que optimiza recursos y facilita la gestión de entornos distribuidos [8]. En este estudio, se desarrollará un entorno híbrido de pruebas que integra máquinas virtuales y contenedores para evaluar el rendimiento y el monitoreo de sistemas operativos, empleando herramientas avanzadas de análisis.

Para la infraestructura de máquinas virtuales, se han seleccionado Rocky Linux, Manjaro Linux y Arch Linux, ejecutados mediante QEMU. Cada distribución tiene características particulares en la gestión de procesos y servicios, lo que permite examinar diferentes enfoques de administración [9]. La virtualización con QEMU ofrece una emulación eficiente del hardware, asegurando una separación efectiva de los entornos de prueba sin comprometer la estabilidad del sistema anfitrión [11]. Rocky Linux, Manjaro y Arch Linux cuentan con una comunidad activa y constantes mejoras, lo que los convierte en opciones ideales para entornos virtualizados [16], [17], [18].

Por otro lado, los contenedores han revolucionado la forma en que se ejecutan aplicaciones, proporcionando un aislamiento eficiente sin la sobrecarga de una máquina virtual

completa [10]. Docker ha permitido la creación y gestión sencilla de entornos de ejecución de software, reduciendo problemas de compatibilidad entre sistemas [2]. En este estudio, se analizarán tres contenedores con imágenes de Garuda, Alpine y Debian, explorando sus capacidades de gestión y rendimiento [7].

El monitoreo de estos sistemas es clave para comprender su eficiencia en la administración de recursos [4]. Grafana y Prometheus han demostrado ser herramientas versátiles para la supervisión en tiempo real de métricas de CPU, red y almacenamiento [11], mientras que Zabbix permite gestionar alertas y eventos en entornos complejos [12]. La implementación de estas herramientas en el contenedor central basado en Fedora permitirá visualizar datos y detectar anomalías en los sistemas analizados.

En los sistemas Linux seleccionados, el control de servicios y procesos se gestiona principalmente con ‘systemd’ y ‘journalctl’, proporcionando herramientas eficaces para el registro y administración del estado del sistema [3]. En Rocky Linux, Manjaro y Arch Linux, estos comandos permiten optimizar la respuesta del sistema ante fallos y mejorar la administración de recursos [13].

El análisis de redes también desempeña un papel crucial en el estudio, permitiendo evaluar el tráfico generado entre los nodos del entorno híbrido [6]. Aplicaciones como Wireshark, ‘iftop’ y ‘nethogs’ ayudarán a visualizar y comprender las interacciones de los sistemas [14]. Además, herramientas como ‘nmap’ y ‘netstat’ contribuirán a la identificación de servicios activos y puertos abiertos en las máquinas virtuales y contenedores [15].

El almacenamiento y la administración de archivos son otro aspecto clave del estudio, ya que permiten una gestión eficiente de los datos utilizados en los sistemas analizados [13]. La combinación de herramientas como ‘ncdu’ y ‘baobab’ facilitará la identificación de patrones de uso de disco, mientras que ‘rsync’ optimizará la sincronización de archivos y la transferencia de datos entre nodos [1].

Finalmente, este informe documentará el procedimiento de instalación, configuración y análisis de cada componente, destacando el impacto de la virtualización y contenedorización en la gestión de infraestructura distribuida. La implementación de un contenedor central para el monitoreo



## FACULTAD DE INGENIERÍA ELECTRÓNICA

permitirá una comparación detallada entre los distintos sistemas, proporcionando un análisis integral de la eficiencia y estabilidad de cada tecnología empleada.

### II. Marco Teórico

#### A. Virtualización y Máquinas Virtuales

La virtualización ha revolucionado la infraestructura de TI al permitir la ejecución de múltiples sistemas en un mismo hardware físico [8]. Herramientas como QEMU [9] ofrecen emulación de hardware y administración eficiente de máquinas virtuales. Además, sistemas operativos como Rocky Linux, Manjaro y Arch Linux son ideales para analizar gestión de recursos en entornos virtualizados [1]. Cada distribución tiene su propia filosofía y enfoque en la administración del sistema, lo que las hace atractivas para pruebas comparativas [16], [17], [18].

#### B. Contenedores y Docker

Los contenedores proporcionan una alternativa ligera a la virtualización, aislando aplicaciones sin necesidad de hipervisores [10]. Docker ha sido fundamental en este avance, facilitando despliegues rápidos y portabilidad [2]. Los contenedores con sistemas como Garuda, Alpine y Debian permiten gestionar software sin impacto significativo en el rendimiento del host [7].

#### C. Gestión de Recursos en Sistemas Operativos

Cada sistema operativo administra sus recursos de manera distinta, afectando el rendimiento en entornos virtualizados y contenedorizados [1]. Herramientas como 'glances', 'bpytop' y 'htop' permiten analizar el consumo de CPU, memoria y procesos activos en tiempo real [11].

#### D. Monitoreo de Sistemas

El monitoreo es crucial para la gestión de rendimiento y seguridad en infraestructuras virtualizadas [4]. Herramientas como Grafana y Prometheus recopilan métricas de CPU, red y almacenamiento [11], mientras que Zabbix se especializa en supervisión de servicios y alertas en entornos distribuidos [12].

#### E. Gestión de Servicios en Linux

La administración de procesos y servicios en sistemas Linux se basa en 'systemd' y 'journalctl', herramientas clave para la supervisión de eventos del sistema [3]. En Rocky Linux, Manjaro y Arch Linux, estos comandos permiten identificar fallas y mejorar la eficiencia del sistema [13].

#### F. Análisis de Redes y Seguridad

El monitoreo de tráfico es fundamental en entornos distribuidos [6]. Aplicaciones como Wireshark, 'iftop' y 'nethogs'

permiten analizar el flujo de datos en la red [14]. Además, comandos como 'nmap' y 'netstat' facilitan la identificación de puertos abiertos y servicios activos en las máquinas virtuales y contenedores [15].

#### G. Administración de Almacenamiento y Archivos

La gestión eficiente del almacenamiento es esencial en entornos virtualizados [13]. Herramientas como 'ncdu' y 'baobab' permiten visualizar el uso del disco, mientras que 'rsync' optimiza la sincronización de archivos [1].

#### H. Automatización y Gestión de Infraestructura

La administración automatizada de infraestructuras ha cobrado importancia con la adopción de herramientas como Ansible y Terraform [7]. Estas permiten configurar servidores y contenedores de manera repetible y eficiente.

#### I. Integración de Monitoreo con Contenedor Central

El contenedor central basado en Fedora cumple la función de nodo de supervisión en esta infraestructura híbrida [5]. Gracias a la combinación de Grafana, Prometheus y Zabbix, es posible recolectar métricas de rendimiento y analizar tendencias en el uso de recursos [11].

### III. PROCEDIMIENTO

Explica los métodos y materiales utilizados en el estudio. Debe ser lo suficientemente detallado para que otro investigador pueda replicar el experimento o análisis.

### IV. RESULTADOS

Presenta los datos obtenidos en la investigación o experimentación, sin interpretarlos. Se pueden incluir tablas, gráficos y figuras para representar los hallazgos.

### V. ANALISIS DE RESULTADOS

Se interpretan los resultados obtenidos, comparándolos con la teoría y/o estudios previos. Aquí se discuten los posibles errores, limitaciones y se extraen conclusiones parciales.

### VI. CONCLUSIONES

Resume los hallazgos principales del estudio, responde a los objetivos planteados y sugiere posibles mejoras o investigaciones futuras.

### REFERENCIAS

- [1] A. S. Tanenbaum, *Modern Operating Systems*. Pearson, 2015.
- [2] D. Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment," *Linux Journal*, vol. 2014, no. 239, 2014.
- [3] M. Martin, "System Logging and Monitoring with journalctl," *SysAdmin Magazine*, 2017.



---

**FACULTAD DE INGENIERÍA ELECTRÓNICA**

---

- [4] J. Fonseca, *Monitoring IT Infrastructures with Netdata*. Packt Publishing, 2018.
- [5] T. Chacon, "Grafana: Real-time Monitoring and Visualization," *IEEE IT Journal*, 2020.
- [6] G. Stewart, "Cloud Monitoring with Prometheus and Grafana," *ACM SIGOPS*, vol. 19, no. 2, 2019.
- [7] R. Sharma, *Containerization and Kubernetes in Modern IT Deployments*. Springer, 2021.
- [8] P. Mavridis, "Trends in Virtualization Technologies," *Journal of IT Research*, 2019.
- [9] H. Luo, "QEMU and Virtualization Performance," *IEEE Transactions on Computing*, vol. 18, no. 1, 2018.
- [10] D. Bernstein, "Containers and the Future of Cloud Computing," *ACM Cloud Computing*, vol. 32, no. 7, 2014.
- [11] S. Khan, *Mastering Prometheus and Grafana*. O'Reilly Media, 2020.
- [12] A. Gorodov, "Advanced System Monitoring with Zabbix," *IT Operations Journal*, 2013.
- [13] M. Kerrisk, *The Linux Programming Interface*. No Starch Press, 2020.
- [14] K. Ishida, "Deep Packet Analysis with Wireshark," *Network Security Journal*, 2016.
- [15] B. Douglass, "Hybrid IT Architectures and
- [16] Rocky Linux, "Descargas de Rocky Linux," [En línea]. Disponible en: <https://rockylinux.org/es-ES/download>. [Accedido: 02-may-2025].
- [17] Manjaro Linux, "Descargas de Manjaro Linux," [En línea]. Disponible en: <https://manjaro.org/products/download/x86>. [Accedido: 02-may-2025].
- [18] Arch Linux, "Última versión de Arch Linux," [En línea]. Disponible en: <https://archlinux.org/releng/releases/2025.04.01/>. [Accedido: 02-may-2025].