

## IMPLEMENTACIÓN DE LA HERRAMIENTA DE GESTIÓN DE REDES OPMANAGER EN CONTENEDORES DOCKER

Alexei Victores Hernández<sup>1</sup>, Leslye Villar Ledo<sup>2</sup>, Joel Paz León<sup>3</sup>

<sup>1</sup>Dirección de Tecnologías y Sistemas, Playa, La Habana <sup>2</sup>ETECSA, Edificio Beijing 5to piso 3ra Ave e/ 76 y 78, Miramar, Habana, <sup>3</sup>Dirección de Tecnologías y Sistemas, Playa, La Habana.

<sup>1</sup>e-mail: alexei.victores@nauta.cu

<sup>2</sup>e-mail: leslye.villar@etecsa.cu

### RESUMEN

Las máquinas virtuales durante el proceso de virtualización de los servicios que despliegan hacen un uso ineficiente de los recursos de hardware que le son asignados previamente. En el ámbito de la virtualización, la plataforma de contenedores virtuales Docker, permite mayor eficiencia. Gracias al requerimiento de menos hardware y recursos humanos para el despliegue y mantenimiento de los servicios los costos son inferiores. En este proyecto se realiza un estudio de la tecnología de máquinas y contenedores virtuales. Posterior a ello se analizan varias plataformas útiles para el despliegue de contenedores virtuales. El mismo finaliza con la implementación de la herramienta de gestión y operación de redes de datos de telecomunicaciones OpManager en un contenedor Docker. Esta investigación sienta las bases teóricas para optimizar los recursos de hardware empleados por las máquinas virtuales durante el proceso de virtualización, esto se logra al migrar un gran volumen de servicios a la infraestructura de contenedores virtuales. Docker es una plataforma que permite crear aplicaciones en contenedores de software que son ligeros y portátiles, las estadísticas indican que Docker es el siguiente paso en las tecnologías de virtualización.

**PALABRAS CLAVES:** Máquinas virtuales, contenedores virtuales, Docker.

## IMPLEMENTATION OF THE OPMANAGER NETWORK MANAGEMENT TOOL IN DOCKER CONTAINERS

### ABSTRACT

Virtual machines, during the virtualization process of the services they deploy, make inefficient use of the hardware resources that are previously assigned to them. In virtualization, the Docker virtual container platform allows greater efficiency; Thanks to the requirement of less hardware and human resources for the deployment and maintenance of the services, the costs are lower. In this project, a study of the virtual machine and container technology is carried out. Subsequently, several valuable platforms for the deployment of virtual containers are analyzed. It ends with implementing the OpManager telecommunications data network management and operation tool in a Docker container. This research lays the theoretical foundations to optimize the hardware resources used by virtual machines during the virtualization process; this is achieved by migrating a large volume of services to the virtual container infrastructure. Docker is a platform that allows you to create applications in software containers that are lightweight and portable; statistics indicate that Docker is the next step in virtualization technologies.

**INDEX TERMS:** Virtual machines, virtual containers, Docker.

### 1. INTRODUCCIÓN

*ManageEngine OpManager* es un software de monitoreo de red asequible y fácil de usar. Monitorea dispositivos como routers, conmutadores, cortafuegos, balanceadores de carga y controladores de Red de Área Local Inalámbrica (WLAN por las siglas del término en inglés *Wireless Local Area Network*). La herramienta OpManager también permite monitorear servidores, máquinas virtuales, impresoras, dispositivos de almacenamiento y otros dispositivos. El monitoreo de equipos de red es posible siempre que tengan una dirección del Protocolo de Internet (IP por las siglas del término en inglés *Internet Protocol*) y estén conectado a la red.

OpManager es una herramienta utilizada para gestionar las redes de datos en las instituciones que ofrecen servicios de telecomunicaciones, usualmente es desplegada en máquinas virtuales. El despliegue de la aplicación OpManager empleando como soporte de virtualización las máquinas virtuales genera un uso irracional de los recursos de hardware, debido a que gran parte de ellos son dedicados solo al funcionamiento de la aplicación en la plataforma virtualizada. Las máquinas virtuales simulan el proceso de arranque de una computadora real. Al iniciar una máquina virtual, la interfaz comparable con el Sistema Básico de Entrada Salidas (BIOS por las siglas del término en inglés *Basic Input-Output System*) pone en funcionamiento el gestor de arranque, el sistema operativo se inicializa, se le asigna a la máquina virtual las tarjetas de red, unidades de disco y otros periféricos útiles para su funcionamiento. Estos dispositivos son emulados y usualmente no todos los recursos asignados son empleados durante el proceso de virtualización de los servicios [1].

Los recursos de hardware empleados para realizar el despliegue de los servicios y la gestión de las redes de datos de telecomunicaciones a menudo se encuentran limitados en muchas instituciones. El problema planteado anteriormente se hace complejo dado que en muchas ocasiones no es posible incrementar en los servidores de red dedicados el desarrollo de servicios, la Memoria de Acceso Aleatorio (RAM por las siglas del término en inglés *Random-Access Memory*), la Unidad Central de Procesamiento (CPU las siglas del término en inglés *Central Processing Unit*) y el almacenamiento disponible para el despliegue de los servicios, aún si contarán con recursos de hardware externo.

En la actualidad para las empresas que brindan servicios de telecomunicaciones ha aumentado la frecuencia de lanzamiento requerida para la entrega de programas, aplicaciones y servicios. Este desarrollo ha impulsado a las organizaciones a explorar formas más efectivas de diseñar sus lanzamientos. Una tecnología que ha ganado un atractivo significativo para muchas organizaciones de desarrollo es la tecnología de contenedores virtuales [2].

Para dar solución al problema planteado surge como objetivo general: Implementar la herramienta OpManager en un contenedor virtual para optimizar los recursos de hardware empleados en el proceso de virtualización. Para desarrollar la solución será importante tener en cuenta que la imagen de la aplicación OpManager no existe actualmente en los repositorios oficiales de contenedores virtuales.

## 2. CONTENEDORES VIRTUALES

La tecnología de contenedores virtuales representa una alternativa frente a las máquinas virtuales, dado que permiten virtualizar sistemas operativos que empaquetan aplicaciones junto con las dependencias necesarias y las despliega en una instancia del sistema operativo anfitrión. Esto hace posible que las aplicaciones se ejecuten de manera independiente y consuman solamente los recursos necesarios, así permiten obtener el mejor rendimiento posible. Un contenedor virtual comparte el kernel del sistema operativo con el host nativo y no tiene que recrear un sistema operativo completo. Gracias a esto el tamaño de los contenedores es mucho menor que el de las máquinas virtuales, haciéndolos eficientes, más fáciles de migrar, iniciar y recuperar [2]. Los contenedores son tecnologías mucho más portables y menos exigentes a nivel de recursos de cómputo que las máquinas virtuales [3].

### Máquinas y contenedores virtuales

Las máquinas virtuales surgieron en la década de los años 60 y han cobrado fuerza en los últimos 20 años a medida que las empresas adoptaron la virtualización de servidores para utilizar la potencia de computación de sus servidores físicos de manera más eficiente. Una máquina virtual dispone de los elementos de un equipo de cómputo, estos elementos en vez de ser físicos son virtuales. Que sus componentes sean virtuales no quiere decir que no existan, pues la máquina virtual obtiene los recursos del equipo donde se esté ejecutando [4]. Se entiende por máquina virtual a un programa de cómputo que simula a una computadora, en la cual se puede instalar y usar otros sistemas operativos de forma simultánea como si fuese una computadora real sobre el sistema operativo huésped [5].

La virtualización convencional añade capas software al sistema que ralentizan considerablemente el rendimiento de las aplicaciones. Para algunos usuarios ha quedado descartada por no cumplir los requisitos de rendimiento y por el agotamiento de recursos que supone. Los contenedores a diferencia de las técnicas de virtualización convencionales, usan el *kernel* del sistema anfitrión en lugar de uno propio. Se ejecutan como un proceso normal en el anfitrión, lo cual permite acceder al hardware sin sobrecostos añadidos. Ofrecen al sistema informático virtual acceso directo al hardware del anfitrión, mientras que las máquinas virtuales requieren de una capa software [6]. La

Fig. 1 y la Fig. 2 muestran algunas diferencias en la arquitectura de la máquina anfitriona durante la virtualización completa y la virtualización a nivel de sistema operativo.

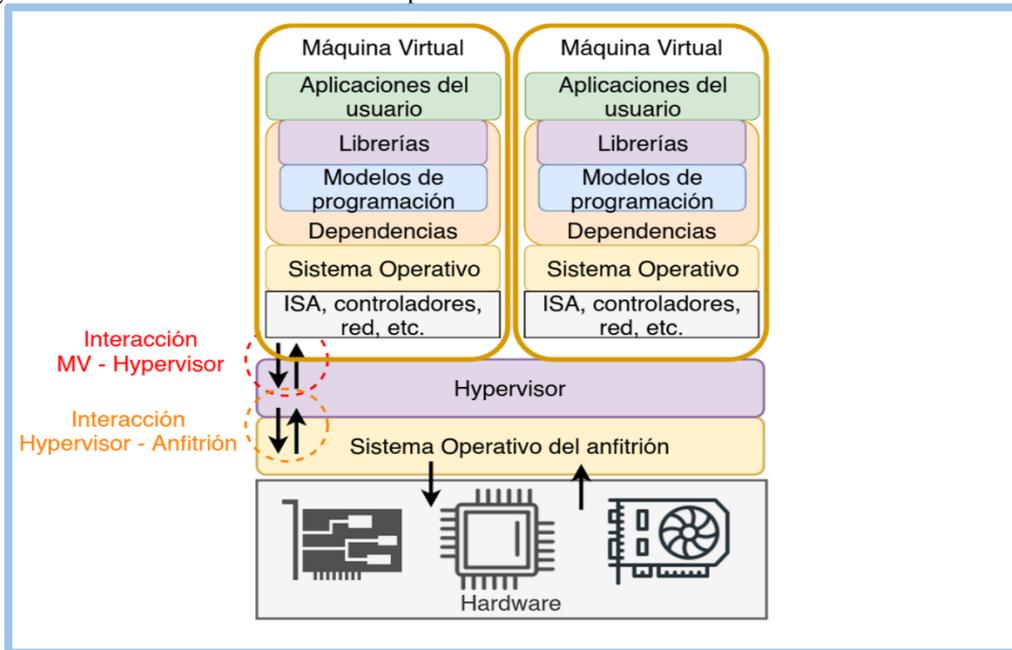


Figura 1: Virtualización completa con el empleo de máquinas virtuales [6].

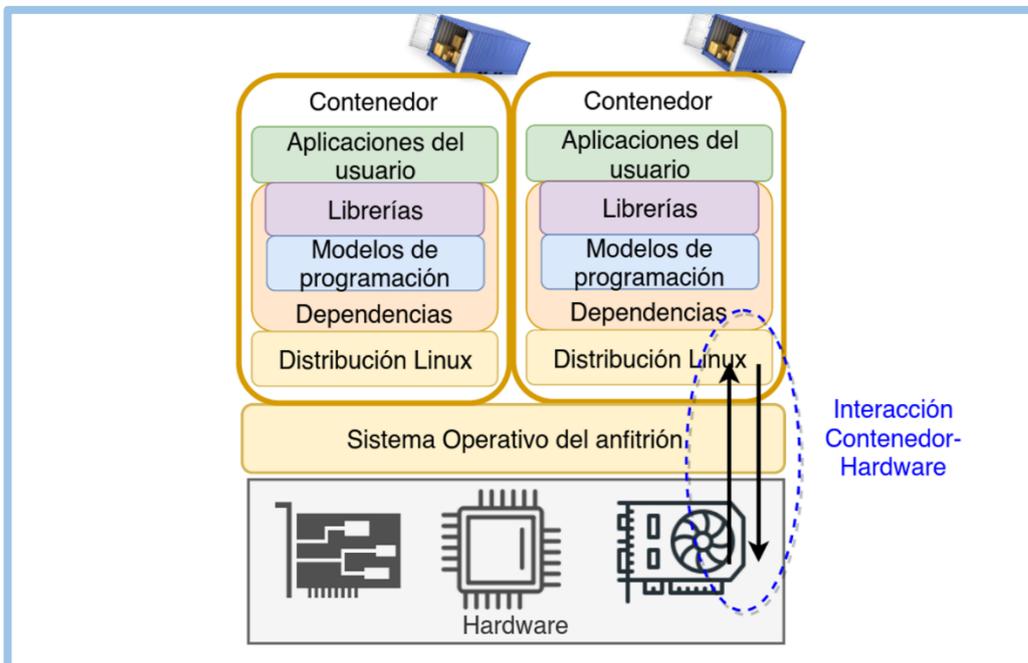


Figura 2: Virtualización a nivel de sistema operativo con el empleo de contenedores virtuales [6].

Las máquinas virtuales agregan gran complejidad al sistema en tiempo de ejecución, esto tiene como efecto la ralentización del sistema, el programa no alcanzará la misma velocidad de ejecución que si se instalase directamente en el sistema operativo anfitrión o directamente sobre la plataforma de hardware. Para usar una máquina virtual en condiciones favorables, se necesita un equipo de cómputo potente [4].

En un contenedor la aplicación junto con sus dependencias puede ser virtualizada con muchos menos gastos. Mientras algunos estudiosos del tema opinan que el desarrollo de contenedores puede anular las máquinas virtuales, hay beneficios de estas que mantienen la tecnología en movimiento. Aun son útiles cuando se ejecutan múltiples

aplicaciones juntas o cuando se ejecutan aplicaciones heredadas de sistemas operativos antiguos. Otros estudiosos del tema opinan que los contenedores son menos seguros que los hipervisores de las máquinas virtuales porque tienen un solo sistema operativo que las aplicaciones comparten, mientras que las máquinas virtuales pueden aislar la aplicación y el sistema operativo [4].

Hoy en día la técnica de virtualización a nivel de sistema operativo conocida como contenedores, emplea variadas herramientas para erradicar la mayoría de los problemas de seguridad y elimina así este inconveniente [7]. En los últimos años la tecnología de contenedores ha crecido en popularidad como un posible reemplazo para los hipervisores ya que pueden colocar más aplicaciones en un solo servidor físico [8].

Ambas tecnologías se pueden utilizar para fines distintos. Los contenedores no vienen a sustituir a las máquinas virtuales, pero constituyen una alternativa a su despliegue para mejorar el rendimiento de los equipos de cómputo. Ambas tecnologías son totalmente compatibles y se pueden utilizar juntas para generar una infraestructura más potente, segura, flexible y eficiente.

### Motivación a un salto tecnológico

La necesidad de un cambio tecnológico para muchas empresas es inminente producto de las limitaciones de recursos existentes y la necesidad de evitar costos por adquisición de nuevos servidores. De ahí la necesidad de ver la virtualización a paso más allá de la virtualización tradicional con máquinas virtuales y poner la visión en la virtualización a nivel de sistema operativo. Los contenedores permiten la aplicación de automatización para agilizar los procesos de despliegue, su implementación hace posible utilizar mejor los recursos de una computadora, ya que funcionan como procesos en el sistema operativo y generan menor sobrecarga [9]. Algunos de los factores que influyen en la decisión de implementar contenedores son los siguientes, por su orden de mayor a menor importancia:

- Eficiencia mejorada respecto a máquinas virtuales en cuanto tiempo de ejecución y ahorro de recursos.
- Oportunidad de lanzar funcionalidades rápidamente, lo que se traduce en una rápida y fácil escalabilidad.
- Manejo simplificado de administración y despliegue de aplicaciones.
- Interacción rápida, las máquinas virtuales emplean mucho tiempo en la constante grabación de paquetes de datos al disco virtual.
- Mantenimiento simplificado respecto a máquinas virtuales.
- Reducción de tiempos de configuración y simplicidad de creación.

### 3. DEFINICIÓN Y FUNCIONAMIENTO DE LA PLATAFORMA DOCKER

Docker es una tecnología de código libre que permite automatizar el despliegue de aplicaciones dentro de contenedores virtuales de procesos. Puede encapsular las aplicaciones dentro de un contenedor junto con los archivos necesarios para que la aplicación se pueda utilizar y posteriormente ejecutarlos sobre cualquier equipo que tenga instalada la plataforma Docker [10]. Con Docker los procesos, las redes, los sistemas de datos y usuarios se encuentran separados entre sí, lo que permite que su inicio sea rápido sin agregar mucha sobrecarga, para ello utiliza una configuración individualizada de los recursos del sistema operativo. En Docker cada contenedor tiene su propio adaptador de red virtualizado y sistema de archivos como grupos de control y espacios de nombre que segregan los procesos para su independiente ejecución [11].

#### Justificación de la elección tecnológica

Docker es la plataforma líder dentro de los softwares para contenedores virtuales. Consiguió rescatar la virtualización basada en contenedores y convertirla en una opción más ligera que la virtualización por medio de máquinas virtuales [6].

La tecnología Rocket conocido como rkt en su entorno de ejecución inicia contenedores como máquinas virtuales. Estas máquinas virtuales están desarrolladas de una Máquina Virtual Basada en el Núcleo (KVM por las siglas del término en inglés *Kernel-Based Virtual Machine*) o de la tecnología *Clear Container* de Intel para garantizar el aislamiento entre contenedores. Docker con el uso de los espacios de nombre del *kernel* de Linux realiza el aislamiento entre contenedores de manera eficiente sin necesidad de herramientas externas [12]-[13]. El proyecto

rkt aunque aún es empleado por la comunidad, fue discontinuado en febrero del año 2020, por lo que puede ser una tecnología que en pocos años no tenga soporte para un mejor desarrollo [14].

La tecnología Contenedores Linux (LXC por las siglas del término en inglés *Linux Containers*), suele iniciar una distribución completa en un entorno virtual. Inicia de una imagen del sistema operativo y los usuarios interactúan con ella de forma parecida a como lo harían con una máquina virtual. LXC es un contenedor de sistemas operativos. En LXC al igual que en una máquina virtual las aplicaciones que se ejecutan disponen de los recursos que se hayan asignado previamente a este entorno, emular un proceso semejante no es la solución a lo que se necesita. Es válido destacar que su despliegue se realiza sin la necesidad de utilizar un hipervisor que emule el hardware [13].

Las tecnologías LXC, El Demonio de Contenedor de Linux (LXD por las siglas del término en inglés *Linux Container Daemon*) y OpenVZ virtualizan sistemas operativos completos, mientras que Docker se centra en virtualizar y desplegar aplicaciones, convirtiéndose en un entorno más ligero de ejecución, de esta forma maximiza la portabilidad. Desarrollar un software basado en LXC en un sistema de prueba no garantiza el funcionamiento sin errores del contenedor en un sistema en producción. Docker abstrae las aplicaciones de un modo más efectivo del sistema subyacente, de forma que un contenedor Docker puede funcionar en cualquier anfitrión que tenga Docker instalado sin depender de la configuración de hardware del equipo [13]. Otra tecnología que se podría utilizar como soporte para los contenedores es *Linux VServer*, pero para realizar la virtualización a nivel de sistema operativo necesita parches para el *kernel* de Linux, convirtiéndose esto es una gran desventaja [12]. OpenVZ 7 / Virtuozzo con respecto a Docker no son plataformas alternativas pues no realizan de forma eficiente el aislamiento de los procesos además su empleo solo es posible con las distribuciones Linux RHEL7 y Virtuozzo Linux respectivamente [13].

Existen otras plataformas viables como Podman, que no necesita un gran servicio demonio para funcionar, pero Docker es una herramienta con infinidad de ventajas y opciones. Es capaz de manejar todos los aspectos referidos a los contenedores, mientras más contenedores se usen, más grande y complejo se hace el servicio de Docker, pero para ello cuenta con su propio orquestador para contenedores, diferencia destacable que sitúa a Docker por encima de PodMan. Docker tiene más aceptación que Podman, además posee sus propias herramientas entre las que destacan Docker Swarm y Docker Compose [15].

Singularity y Shifter aunque revelan puntos significativos no demuestran superioridad respecto a Docker. Shifter es un contenedor difícil de instalar, su pobre documentación junto con la gran cantidad de dependencias hace improbable lograr una instalación funcional. Además parece no estar completamente implementado, pues algunas funcionalidades básicas como borrar imágenes descargadas de su base de datos, no existen [6].

Docker y Singularity son contenedores lo suficientemente desarrollados para poder ejecutarse en prácticamente cualquier sistema. Además, poseen pocas dependencias con otro software, lo que facilita su mantenimiento. Dado que Docker utiliza un proceso demonio para funcionar requiere un poco más de configuración que Singularity, pero es ahí donde radica las ventajas que puede llegar a ofrecer el paquete completo de Docker en producción, además Singularity limita las funcionalidades del contenedor y su portabilidad [6].

Se decide emplear la plataforma Docker por la superioridad que ha demostrado respecto a las plataformas abordadas. Docker es el estándar de facto de este tipo de tecnologías, gran parte de los motores de orquestación de contenedores son compatibles con Docker. Es una tecnología que cuenta con un registro de imágenes propio, tanto públicas como privadas en Docker Hub. Docker posee un orquestador de contenedores virtuales por lo que no necesita herramientas externas para su funcionamiento. Cuenta además con la herramienta para despliegue y gestión de multi-servicios Docker Compose que permite automatizar los procesos y agrega gran granularidad al despliegue de los contenedores.

## IMPLEMENTACIÓN DE LA APLICACIÓN OPMANAGER EN UN CONTENEDOR DOCKER

La implementación de la arquitectura se realiza en una máquina virtual creada con el software de virtualización VMware Workstation 16 Pro. El ambiente de simulación está basado en el sistema operativo Linux y es empleada la distribución Debian 11. Es utilizado el protocolo de Cubierto Segura (SSH por sus siglas del término en inglés Secure SHell) para garantizar la conexión remota con la máquina virtual creada. SSH es un protocolo de red cifrado

que garantiza la conexión remota a un equipo desde cualquier puesto de trabajo. Las conexiones que se realizan mediante el protocolo SSH son seguras dado que utilizan algoritmos de cifrado actuales y seguros [16].

## Instalación de Docker

Se accede a la documentación oficial de Docker en la página <https://docs.docker.com>. Como paso inicial se procede a seleccionar el sistema operativo sobre el cual se va a instalar la plataforma, luego se escoge su distribución. Seguidamente se elige el método de instalación. Para actualizar el índice de paquetes del directorio /apt se procede a ejecutar como superusuario el comando `apt-get update`. Para permitir que la Herramienta Avanzada de Administración de Paquetes (APT por las siglas del término en inglés *Advance Packaging Tool*) use el repositorio Docker a través del Protocolo Seguro de Transferencia de Hipertexto (HTTPS por las siglas del término en inglés *Secure Hypertext Transfer Protocol*), se ejecuta como superusuario el comando `apt-get install`.

Desde Cuba se puede acceder a la documentación, pero no es posible el acceso a la página de Docker por limitaciones en cuanto al tema del bloqueo. Con el objetivo de resolver esta problemática, se hace uso de la página clonada y segura del repositorio Docker <sup>1</sup>, para añadir la llave cifrada Guardia de Seguridad de Linux (GPG acrónimo del término en inglés *GNU Privacy Guard*) para actualizar los repositorios propios de la plataforma Docker. Es empleado el comando `wget` para descargar la llave GPG a través de la consola y así poder establecer la conexión cifrada con el repositorio de paquetes.

El acceso al repositorio oficial de Docker está bloqueado para Cuba, sin embargo, existen varios repositorios que clonan esa información. Diversas páginas cubanas contienen estos archivos <sup>2</sup>. Los repositorios adquiridos son ubicados en el directorio `etc/apt/source.list`. Luego al ejecutar como superusuario el comando `apt-get install docker-ce docker-ce-cli containerd.io` son instaladas las herramientas necesarias para el funcionamiento de la plataforma Docker en el sistema. Posteriormente se procede a iniciar el motor de ejecución de Docker con los repositorios incorporados.

Desde Cuba no es posible acceder al repositorio oficial de imágenes de Docker. Por ello, en el fichero `/etc/docker` que apunta a Docker Hub, ubicado en el directorio de configuración de Docker se crea el fichero `daemon.json`. En él se añaden varios nombres de dominios de páginas clonadas del repositorio oficial de Docker. Luego, cada vez que Docker tenga que acceder a su repositorio lo hará hacia estos. Los repositorios utilizados fueron proporcionados por la Comunidad Docker de Cuba.

Luego se reinicia el demonio de Docker para que actualice los repositorios añadidos y se restablece el servicio de Docker al ejecutar como superusuario los comandos; `systemctl daemon-reload` y `systemctl restart docker.service` respectivamente. Para comprobar el correcto funcionamiento de los sitios añadidos se realiza la descarga de la imagen Hello World.

## Proceso de empaquetado de la aplicación OpManager

Como paso inicial se instala la versión 12 de OpManager por consola porque se necesitan privilegios de superusuario para iniciar esta acción. En el transcurso de este proceso es establecido el directorio `/opt/ManageEngine/OpManager`, en el cual serán ubicados los ficheros de la aplicación. Se introduce el puerto 80 para el servicio web y el 9996 para la conexión de OpManager con su base de datos y así poder analizar los datos de flujos que recopila para suministrar la visibilidad sobre el volumen de tráfico.

En el proceso de instalación se otorgaron los permisos y se definió la configuración de la aplicación. Posterior a ello fue establecida la licencia del producto para evitar problemas al realizar el proceso de empaquetado de la aplicación.

## Construcción del contenedor

Es creado el directorio `OpManager-Image` en la cual posteriormente se creará el archivo `Dockerfile` para crear la imagen del contenedor. Luego hacia el interior del archivo `OpManager-Image` son copiados los siguientes archivos:

- El archivo `passwd`: se encuentra en el directorio `/etc/passwd`, este fichero contiene registradas las cuentas de usuarios, así como las claves de accesos y privilegios. El contenido del fichero `/passwd` determina que

usuario puede acceder al sistema y que acciones puede realizar de manera legítima una vez dentro. Este fichero es la primera línea de defensa del sistema contra accesos no deseados. Debe mantenerse libre de errores y fallos de seguridad.

- El archivo shadow: fichero que permite mejorar el sistema de contraseñas y ofrece ventajas sobre el estándar previo de almacenamiento de las contraseñas en Linux. Al igual que passwd se encuentra en el directorio /etc. Ambos ficheros son de gran importancia, si se corrompen su efecto puede tener un impacto catastrófico ya que contienen la raíz básica de los permisos que se usan en el sistema.
- También es agregado el directorio ManageEngine creado producto de la instalación de OpManager. En él se encuentran todas las dependencias de funcionamiento de la aplicación OpManager

OpManager contiene una base de datos que utiliza para su correcto funcionamiento, está incluida en el paquete de instalación de la aplicación, es una base de datos de PostgreSQL que al instalarse crea una línea de configuración en los archivos shadow y passwd referida a esta base de datos. Sino son copiados estos archivos juntos con la línea de configuración, la base de datos no funcionará dentro del contenedor.

## Creación del archivo Dockerfile

Dockerfile es un archivo de configuración que permite crear imágenes con instrucciones y comandos individualizados previamente establecidos [11]. En la Tabla 1 se puede apreciar de forma breve las instrucciones empleadas en el archivo Dockerfile para construir la imagen del contenedor.

Tabla 1: Instrucciones del archivo Dockerfile.

<b>FROM</b>	Especifica la imagen base utilizada para crear la nueva imagen
<b>WORKDIR</b>	Establece la ruta del directorio en el cual será ejecutada la próxima instrucción una vez descargada la imagen base
<b>COPY</b>	Instrucción que permite copiar el archivo indicado previamente desde fuera del contenedor en proceso de creación hacia su interior
<b>EXPOSE</b>	Expone los puertos que serán empleados por la aplicación para su funcionamiento
<b>CMD</b>	Método de ejecución que despliega los comandos que le son indicados previamente

La Fig. 3 muestra la secuencia de instrucciones asignadas en el archivo Dockerfile. La secuencia de creación de la imagen comienza con la instrucción FROM que especifica la imagen a partir de la cual se va a iniciar el archivo Dockerfile, en este caso la última versión de la distribución Debian. Le sigue la instrucción WORKDIR que establece la ruta del directorio de trabajo dentro de la imagen base una vez descargada, en este caso el directorio /opt. La instrucción COPY indica la copia del directorio ManageEngine ubicado en el interior del directorio OpManager-Image hacia el directorio ManageEngine dentro del directorio de trabajo /opt.

Posterior a ello se declara el directorio de trabajo /etc y se indica la realización del proceso semejante, pero ahora con la instrucción que indica la copia del fichero passwd que se encuentra en el directorio OpManager-Image hacia el fichero passwd en el interior del directorio /etc y de forma similar se realiza este procedimiento con el fichero shadow. La instrucción EXPOSE indica exponer los puertos que serán utilizados por la aplicación para su funcionamiento, estos serán; el puerto 22 para establecer la conexión remota con el empleo del protocolo SSH y el puerto 80 para el servicio web de la aplicación OpManager, los puertos 9996 y 13306 son empleados por la aplicación OpManager para la intercomunicación con su base de datos.

```
FROM debian:latest
WORKDIR /opt
COPY ManageEngine/ ManageEngine/
WORKDIR /etc
COPY passwd passwd
COPY shadow shadow
EXPOSE 22 80 9996 13306
WORKDIR /opt/ManageEngine/OpManager/bin
CMD /opt/ManageEngine/OpManager/bin/initPgsql.sh ;
/opt/ManageEngine/OpManager/bin/run.sh ; /bin/bash
```

Figura 3: Parámetros de descripción del archivo Dockerfile.

La siguiente instrucción se dirige al directorio `opt/ManageEngine/OpManager/bin`. Este directorio aparecerá luego que se copie el directorio `ManageEngine` en el interior del directorio de trabajo `/opt`. Una vez dentro de él, el motor de ejecución durante el proceso de compilación de las instrucciones establecidas, se dirige al directorio `bin` para desplegar el método de ejecución `CMD`. El `CMD` de forma predeterminada despliega los comandos básicos que le son indicados inicialmente. Primero ejecuta la base de datos y autoseguido ejecuta el contenedor.

Con el comando `run.sh` se inicia el servicio del contenedor. Las instrucciones establecidas previamente son ejecutadas con el intérprete de comandos de Linux `bin/bash`. Los ficheros `shadow`, `passwd` y el directorio `ManageEngine` fueron situados en el interior del directorio `OpManager-Image`, se hace necesario dado que todos los archivos que se requieren para conformar la imagen de la aplicación `OpManager` se deben situar en el mismo directorio del archivo `Dockerfile`.

### Creación de la imagen del contenedor

Se procede a crear, nombrar y etiquetar la imagen haciendo uso del comando `docker build -t opmanager: v1`. Durante la creación de la imagen del contenedor cada vez que se ejecuta una instrucción para conformar la imagen, se genera una capa sobre la imagen base. Este proceso permite visualizar la ejecución eficiente durante la compilación de las nueve instrucciones establecidas en el archivo `Dockerfile`.

### Validación de la implementación

El despliegue del contenedor para comprobar su funcionamiento se hará con el empleo de la herramienta `Docker Compose`, pues con un solo comando permite poner el contenedor en marcha. `Docker Compose` trabaja instrucciones en el lenguaje de programación `yaml`. La programación de las instrucciones en el fichero se hace de forma declarativa y por el orden que tengan los servicios como se puede apreciar en la Fig. 4.

Primero se declara la versión de `Docker Compose` que se va a utilizar, se prevé siempre que labore con la versión instalada de `Docker`. Luego se programan los servicios, en el caso que ocupa es la aplicación `OpManager`, a continuación se declara el nombre de la imagen y el nombre del contenedor. Para continuar con la programación en el fichero `Docker Compose` se especifican los puertos análogos que utilizará la aplicación en el dispositivo anfitrión durante su ejecución. Entre los puertos que utiliza la aplicación `OpManager` se encuentra el `80`, este puerto se traduce al puerto `8080` del equipo anfitrión y de igual forma sucede con los restantes puertos declarados en el fichero `Docker Compose`. Luego de especificar los puertos se asigna un volumen al contenedor, en este caso `data`. Los volúmenes en `Docker` permiten incorporar datos que estarán disponibles para el contenedor. Estos datos serán persistentes después de un reinicio del contenedor.

En este ambiente de trabajo no es empleado el orquestador Docker Swarm, herramienta que gestiona frecuentemente los contenedores. El lanzamiento de la aplicación con Docker Compose requiere especificar la política de restablecimiento ante caídas del contenedor. Por ello, en el código introducido en el fichero Docker Compose se definió la instrucción restablecer siempre, para garantizar el restablecimiento automático de los servicios ante algún fallo.

```
version: "3.9"

services:
  opmanager:
    image: opmanager:v1
    container_name: opmanager
    ports:
      - 222:22
      - 8080:80
      - 9996:9996
      - 13306:13306
    volumes:
      - data:/opt/ManageEngine/OpManager/pgsql/data
    restart: always

volumes:
  data:
```

Figura 4: Creación del archivo Docker Compose.yaml.

Una vez realizada la programación del fichero Docker Compose se inicia el proceso de compilación de las instrucciones establecidas, para ello se emplea el comando docker-compose up. En la Fig. 5 se puede apreciar el resultado de la implementación realizada. El empaquetado de la aplicación OpManager en el contenedor Docker ha sido satisfactorio. Se ha comprobado al acceder por el navegador al puerto 8080 del sistema anfitrión que es el análogo del puerto 80 de OpManager en el contenedor.

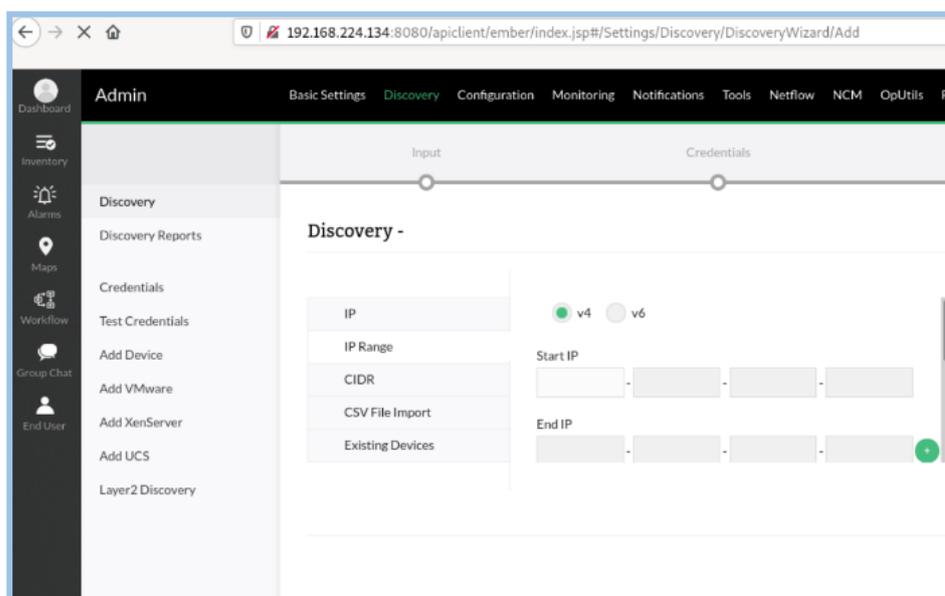


Figura 5: Verificación del funcionamiento de la aplicación OpManager encapsulada en el contenedor Docker.

Con la creación del contenedor basado en la imagen creada denominada opmanager: v1 se brinda una solución a la problemática abordada inicialmente. El empaquetado de la herramienta de gestión y operación de redes de datos de telecomunicaciones OpManager ha sido satisfactorio. Con la implementación realizada las empresas pasan a estar en condiciones de reemplazar la estructura tecnológica que han utilizado hasta el momento como soporte para el despliegue de la aplicación OpManager. La implementación realizada cumple los requisitos de funcionamientos necesarios para la aplicación, pues OpManager labora de forma eficiente sobre la plataforma de despliegue de

contenedores Docker. De esta forma ofrece a las empresas solvencia de recursos y automatización en su funcionamiento.

#### 4. ANÁLISIS DE RESULTADOS

El modelo de calidad de la ISO/IEC 25010 establece el sistema para evaluar la calidad de un producto software. La característica eficiencia de desempeño permite valorar el desempeño relativo de los recursos utilizados bajo determinadas condiciones, esta característica se divide en las subcaracterísticas: comportamiento temporal, utilización de recursos y capacidad [17].

##### Comportamiento temporal

En Docker los contenedores comparten los recursos disponibles y los procesos solo hacen uso de los mismos cuando están en ejecución a diferencia de una máquina virtual donde los recursos son asignados de forma dedicada para la ejecución de las aplicaciones, lo que implica el uso irracional de los recursos de hardware. Docker permite una mayor frecuencia de lanzamiento de las aplicaciones frente a las máquinas virtuales.

##### Utilización de recursos

A continuación se muestran los resultados obtenidos durante el lanzamiento de la aplicación OpManager en la arquitectura Docker y en la arquitectura tradicional de máquinas virtuales. Para la medición de los recursos utilizados se empleó la herramienta de gestión de procesos de Linux Htop en ambas arquitecturas. Htop es una de las herramientas más precisas y fiables que ofrece Linux para monitorear el sistema en tiempo real. La Fig. 6 muestra el uso de la CPU durante el lanzamiento de la aplicación Opmanager en una máquina virtual y en la plataforma de contenedores Docker durante un tiempo de prueba de cinco minutos.

Luego de apreciar la Fig. 6 se puede concluir que Docker optimiza el uso de la CPU durante el despliegue de la aplicación OpManager respecto al despliegue de la aplicación en una máquina virtual.

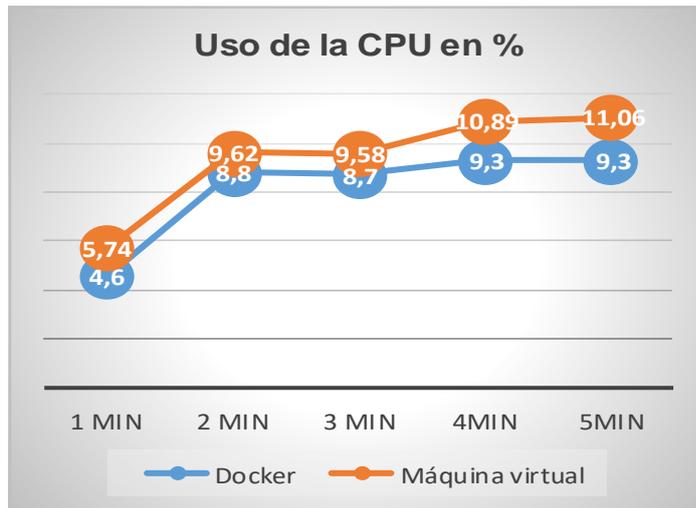


Figura 6: Uso de la CPU durante el lanzamiento de la aplicación OpManager en la plataforma Docker y en una máquina virtual.

Para valorar el comportamiento de la RAM se realizó un análisis del uso de la misma en un rango de cinco minutos durante el despliegue de la aplicación OpManager encapsulada en un contenedor en la plataforma Docker y también en una máquina virtual. Docker fue desplegado en un sistema anfitrión que igual a la máquina virtual donde se desplegó OpManager le fueron asignados 2GB de RAM. En la Fig. 7 se reflejan las estadísticas de las mediciones realizadas durante el lanzamiento de la aplicación.

El despliegue de la aplicación OpManager en un contenedor Docker arroja un resultado de un uso de la RAM inferior respecto al despliegue en una máquina virtual. Docker permite un lanzamiento óptimo con el menor consumo de recursos posibles. Durante la medición del uso de la RAM ambas plataformas permitieron desplegar la aplicación de forma correcta, pero Docker evidencia sus ventajas al mostrar un menor índice de consumo de RAM.

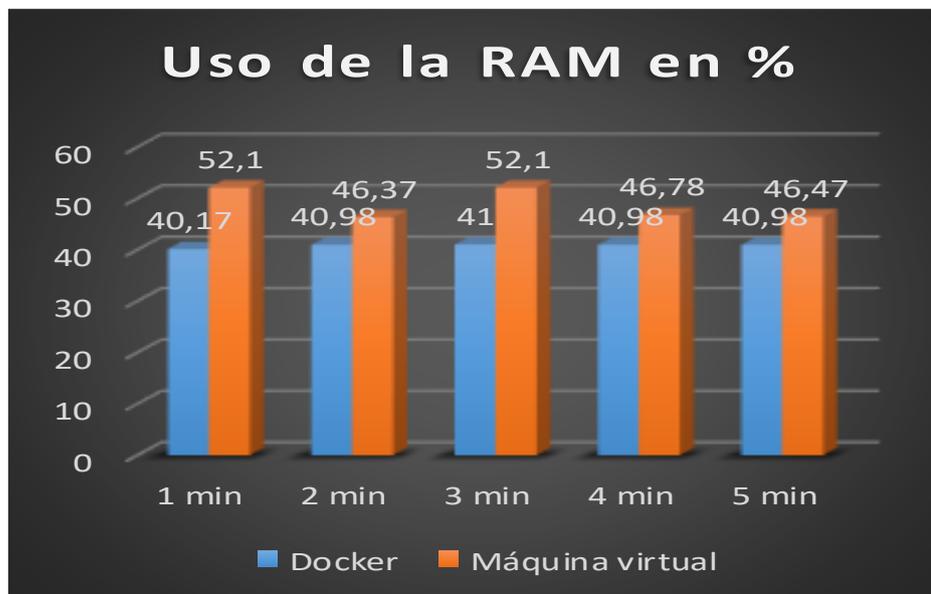


Figura 7: Uso de la RAM durante el lanzamiento de la aplicación OpManager en la plataforma Docker y en la máquina virtual.

## Capacidad

Docker es una plataforma de contenedores creada para el despliegue de múltiples contenedores. En esta arista marca una gran diferencia de rendimiento con respecto a otras tecnologías, por tanto presenta soporte para el despliegue de múltiples aplicaciones. Las máquinas virtuales por el contrario son plataformas para el despliegue de aplicaciones monolíticas y usualmente son usadas para ejecutar una aplicación en específico [6].

## 5. ANÁLISIS DE IMPACTO

Con la presente investigación se brindan herramientas para llevar a cabo la automatización y monitoreo de tiempos, costos y recursos al implementar aplicaciones en contenedores Docker. Para realizar el análisis se establecieron los tres tipos de impactos que se detallan a continuación.

### Impacto tecnológico

El uso de la tecnología Docker permite mejorar habilidades y conocimientos para desarrollar, probar, corregir errores y lanzar a producción sistemas óptimos para los usuarios. Docker permite crear contenedores ligeros y portables, enfocándose en el uso mínimo de recursos y tiempos de despliegue de una aplicación. Su facilidad de iniciar, detener, borrar y migrar contenedores es lo que lo diferencia de las máquinas virtuales.

### Impacto ambiental

Docker permite la creación de varios contenedores en una misma máquina anfitriona, con ello se aprovecha al máximo los recursos que esta provee, sin necesidad de adquirir una nueva infraestructura. Con ello se minimiza la contaminación ambiental, pues los equipos electrónicos cuando son desechados se descomponen y lanzan al medio ambiente todas las sustancias tóxicas empleadas en su fabricación. Docker además permite desplegar múltiples

servicios virtuales empleando la misma energía del sistema anfitrión, por tanto no es necesario generar más electricidad.

## Impacto económico

Con el empleo de la plataforma Docker no es necesario crear varias máquinas virtuales que alojan diferentes aplicaciones. Al desplegar Docker se usan los recursos de una sola máquina, al ejecutar múltiples contenedores el coste de inversión en infraestructura disminuye con la reducción del empleo de máquinas como servidor y se obtienen los mismos resultados con mayor eficiencia, rendimiento y menores costos.

## 6. CONCLUSIONES

En el proceso investigativo del funcionamiento de las máquinas y contenedores virtuales se hallaron aspectos interesantes sobre estas tecnologías. Estos hallazgos fueron de gran importancia para entender ampliamente la problemática existente en las instituciones que hacen uso de entornos virtuales y buscar la vía idónea para darle solución. La implementación de la herramienta de gestión y operación de redes de datos de telecomunicaciones OpManager en el contenedor Docker permite optimizar los recursos de hardware empleados por la empresa durante el proceso de virtualización de los servicios. Por lo que se cumple con resultado satisfactorio el objetivo central del presente trabajo.

Ha sido comprobado que Docker mejora el rendimiento de los recursos de hardware en comparación con las máquinas virtuales durante el lanzamiento de la aplicación OpManager. Los resultados alcanzados en cuanto al uso de los recursos de hardware con el empleo de contenedores Docker, permiten concluir que es más rentable para las empresas desplegar en Docker las aplicaciones que utilizan para brindar los servicios. Si una empresa que utiliza una máquina virtual para desplegar cada una de las aplicaciones, despliega en Docker, mínimo dos aplicaciones por máquina virtual, libera un porcentaje significativo de los recursos utilizados. Así mejora la escalabilidad, pues al liberar recursos presenta solvencia para desplegar otros servicios y optimizar los que ya brinda.

## REFERENCIAS

- [1] Contenedores: Un poco de historia. SG Buzz. Gunnar Wolf. Oct, 12, 2016. <https://sg.com.mx/revista/52/contenedores-un-poco-historia>. (Accedido abr, 2021)
- [2] F. Ortiz. Diseño, desarrollo y despliegue de plataforma web para gestión de actividades deportivas usando contenedores y computación en la nube. Trabajo Fin de Máster. Universidad de Sevilla, 2017.
- [3] F. J. Pacheco. Edge Computing on the IoT Gateway. Máster en Internet de las cosas. Universidad Complutense de Madrid, 2019.
- [4] A. Carrillo & K. I. González Máquinas virtuales. Facultad de Ciencias, UNAM, 2021.
- [5] A. Cajica. Máquinas virtuales sobre un clúster para laboratorio virtual. Universidad Autónoma de Puebla, 2020.
- [6] O. Rudy. Viabilidad del uso de contenedores en entornos HPC. Universidad Politécnica de Cataluña (UPC), 2019.
- [7] S. Vega, J. M. de la Fe, Y. Rodríguez, L. R. García, & A. A. Garófalo, Desarrollo de un driver LXC para Open Nebula. Ingeniería, investigación y tecnología, vol. 19, n° 1, enero-marzo 2018 63-76, DOI: <http://dx.doi.org/10.22201/ii.25940732e.2018.19n1.006>. Versión impresa ISSN 1405-7743.
- [8] L. E. Tabasco, Análisis de la virtualización de conmutadores virtuales en redes empresariales. Universidad Central "Marta Abreu" de Las Villas, 2019.
- [9] F. N. Funes, Análisis de solución eficiente para escalabilidad de sistemas online mediante orquestación de contenedores. Universidad Siglo 21, 2018.
- [10] F. Balderas, Método para la supervisión de sistemas de contenedores virtuales basado en multi-modelado. Centro de Investigación y de Estudios Avanzados del IPN Unidad Tamaulipas, 2020.
- [11] Y. D. Vizcaino, Optimización de aplicaciones Java Enterprise monolíticas mediante el uso de contenedores Docker. Universidad técnica del norte. Ibarra – Ecuador, 2021.
- [12] C. C. Ibancos, Contenedores: Análisis e implantación de ejemplos prácticos. Universidad de Jaén, 2019.
- [13] IONOS. Virtualización con contenedores Docker: alternativas. Julio, 7, 2019). <https://www.ionos.es/digitalguide/servidores/know-how/alternativas-a-los-contenedores-en-docker/>. (Accedido sept. 2021)

- [14] Herramientas de contenedor, alternativas de Docker en 2021. Marzo, 7, 2021. <https://compilar.es/herramientas-de-contenedor-alternativas-de-docker-en-2021/>. (Accedido sept. 2021).
- [15] R. Almejeiras. ¿Qué es Podman?, ¿Podrá con el mismísimo rey Docker? Julio, 15, 2021. <https://pandorafms.com/blog/es/que-es-podman/> (Accedido oct. 2021).
- [16] F. Rey. Guía básica sobre SSH: Qué es, cómo instalarlo y cómo utilizarlo correctamente. Febrero, 26, 2020. <https://raiolanetworks.es/blog/ssh/>. (Accedido nov. 2021).
- [17] ISO/IEC 25010. <https://iso25000.com/index.php/normas-iso-25000/iso-25010?limit=3&start=2>. (Accedido nov. 2021)

## SOBRE LOS AUTORES

**Alexei Victores Hernández:** Graduado de Ingeniería en Telecomunicaciones y Electrónica en la Universidad Tecnológica de la Habana, José Antonio Echeverría, CUJAE. Mercedor del reconocimiento de Destacado a nivel provincial en la jornada científica celebrada en 2014. Actualmente se desempeña como Especialista en Telecomunicaciones en la Dirección de Tecnologías y Sistemas. Delegado y ponente en el evento X SIMPOSIO INTERNACIONAL DE TELECOMUNICACIONES, ha realizado los cursos de posgrados Introducción a las redes sociales en Internet y Gestión de redes y servicios telemáticos en la Universidad de Ciencias Informáticas (UCI), ha realizado cursos de superación de la asociación LACNIC con base en las telecomunicaciones. Identificador ORCID: <https://orcid.org/0000-0001-5335-4476>.

**Leslye Villar Ledo:** Graduado de Ingeniero en de Telecomunicaciones y Electrónica en la CUJAE (2003). Graduado de Master en Sistemas de Radiocomunicaciones en la CUJAE (2009). Trabaja en la Empresa de Telecomunicaciones de Cuba S.A (ETECSA), se desempeña como Jefe de Grupo del Centro de Estudio de Telecomunicaciones e Informática (CETI) donde dirige y coordina todas las actividades de formación e Investigación que se realizan de forma conjunta entre ETECSA y la CUJAE. Mantiene su condición de Profesor Asistente impartiendo asignaturas de la disciplina de Sistemas de Radiocomunicaciones y Redes de Telecomunicaciones, ha tutorado más de 20 trabajos de diploma, ha publicado tres artículos en revistas y ha presentado 8 trabajos en eventos científicos internacionales, ha cursado tres posgrados de la Universidad Tecnológica de Múnich (TUM) y obtuvo en el año 2018 la certificación HCNA-WLAN de la empresa HUAWEI. Identificador ORCID: <https://orcid.org/0000-0002-5813-7793>.

**Joel Paz León:** Graduado de Ingeniería en Telecomunicaciones y Electrónica en la Universidad Tecnológica de la Habana, José Antonio Echeverría, CUJAE. Actualmente se desempeña como Especialista en Telecomunicaciones en la Dirección de Tecnologías y Sistemas. Identificador ORCID: <https://orcid.org/0000-0001-9635-3027>.

## CONFLICTO DE INTERESES

El presente proyecto es resultado de una exhaustiva investigación surgida de la necesidad de varias empresas de liberar y hacer rentable a largo plazo los recursos de hardware que poseen. Los autores de conjunto con las instituciones que representan, están de acuerdo a socializar la solución realizada con el objetivo de promover el uso racional de los recursos de hardware durante la virtualización de los servicios en las entidades que lo requieran.

## CONTRIBUCIONES DE LOS AUTORES

- **Autor 1:** Conceptualización, preparación, creación y desarrollo del proyecto investigativo.
- **Autor 2:** Contribución a la idea y organización del artículo, sugerencias acertadas para la conformación de la versión final.
  - **Autor 3:** Revisión crítica de cada una de las versiones del borrador del artículo y aprobación de la versión final a publicar.

El trabajo en equipo ha prevalecido en la realización del presente artículo. Todos los autores han estado inmersos en el proceso investigativo tanto como en el desarrollo metodológico y científico del artículo, siendo cada uno capaz de aportar y reflejar en el mismo sus conocimientos con el fin de lograr un excelente proyecto.

