



# LABORATORIO DE VIRTUALIZACIÓN UTILIZANDO VIRTUALIZACIÓN ANIDADA **CON HYPER-V**

Ramón Parra Loera<sup>1</sup>, Jansel Edmundo Arreola García<sup>2</sup> y Víctor Manuel Morales Rocha<sup>3</sup>

<sup>1,2,3</sup> Universidad Autónoma de  
Ciudad Juárez  
Av. Del Charro 450 Norte,  
Ciudad Juárez, Chihuahua, México,  
C.P. 32310.



rparra@uacj.mx  
jansel.arreola@hotmail.com,  
victor.morales@uacj.mx.

Recibido: Agosto 28, 2017.

Recibido en forma revisada: Septiembre 29, 2017.

Aceptado: Julio 3, 2018.

**Resumen:** En este trabajo se demuestra la funcionalidad de virtualización anidada en un ambiente práctico, como lo son los laboratorios de cómputo en instituciones educativas. Se describe un caso real de implementación de virtualización anidada en un laboratorio creado con Hyper-V de Microsoft. Se implementaron dos capas de virtualización por encima del nivel físico y se realizaron pruebas de conexión entre las diferentes máquinas virtuales, probando así la correcta configuración de las mismas.

**+ Palabras clave:** *Virtualización anidada, ambiente de prueba, laboratorio de cómputo.*

**Abstract:** This paper demonstrates the functionality of nested virtualization in a practical environment, such as computer laboratories in educational institutions. A real case of nested virtualization implementation in a lab created with Microsoft Hyper-V is described. Two virtualization layers were implemented above the physical layer and communication tests were performed between the different virtual machines, thus proving the correct configuration and implementation of nested virtualization.

**+ Keywords:** *Nested virtualization, testing environment, computing lab.*

## Antecedentes

Hoy en día es muy común encontrar con equipos de cómputo en instituciones educativas destinados para fines didácticos, de manera que los estudiantes de cualquier nivel educativo estén preparados para la utilización de recursos informáticos básicos. En instituciones de nivel superior, usualmente se encuentran salas especializadas para dicho fin, las cuales se conforman de varios equipos de cómputo. Sin embargo, la manera tradicional de desplegar laboratorios de cómputo presenta una serie de inconvenientes, entre los cuales tenemos: adquisición de gran cantidad de equipos de cómputo, instalación y configuración de cada equipo, mantenimiento preventivo y correctivo para cada máquina, licencias costosas de software especializado, espacio ocupado por equipo que pudiera ser destinado para otros propósitos y recursos limitados para el alumnado.

Una de las soluciones que ha tenido más crecimiento en los últimos años son las tecnologías de virtualización, las cuales facilitan la accesibilidad y disponibilidad de los recursos educativos, particularmente en aprendizaje a distancia. Las plataformas educativas virtuales son un claro ejemplo de aplicación de dichas tecnologías, ya que los usuarios pueden acceder a los recursos educativos mediante aulas de clases virtuales (Misevicien, Ambrazien, Tuminauskas y Pazereckas, 2012).

En diversas universidades a lo largo del mundo se han puesto en marcha proyectos innovadores que involucran virtualización. Un claro ejemplo está en la infraestructura educativa implementada en la Universidad de Tecnología en Kaunas, Lituania. Dicha infraestructura les permite acceder a los estudiantes y staff a escritorios virtuales y aplicaciones vía Web dentro y fuera de las instalaciones de la institución. La particularidad de dicha infraestructura es que cuenta con una interfaz de administración donde es posible ver cuáles y en qué cantidad de recursos están siendo utilizados por los usuarios (Misevicien, Ambrazien, Tuminauskas y Pazereckas, 2012). En el caso de implementación de laboratorios de virtualización, se tiene el caso de la plataforma de ética educacional llamada DoS\_VLab, desarrollada en la Universidad de los Emiratos Árabes Unidos. Este laboratorio fue creado para permitir a los estudiantes emular ataques comunes de denegación de servicio (DoS) en un ambiente académico seguro. Dicha plataforma está basada en tecnologías de virtualización y simulación de redes virtuales, la cual ha ayudado a que los estudiantes puedan realizar más prácticas y en menor tiempo que en un laboratorio de cómputo convencional (Al Kaab, Al Kindi Al Fazari y Trabelsi, 2016).

## Virtualización anidada

La virtualización de cómputo permite crear, mediante hipervisores, un ambiente donde se pueden correr múltiples sistemas operativos dentro de un mismo equipo físico, y así aislar dichos sistemas operativos para utilizarlos como equipos de cómputo independientes. En recientes años se han desarrollado tecnologías que permiten virtualizar dentro de un ambiente ya virtualizado, siendo esto lo que se conoce como virtualización anidada. En términos más sencillos, es la práctica de correr un hipervisor dentro de una máquina virtual, lo que proveería una segunda capa de virtualización. Esto puede ser útil para aplicaciones como:

**a) Ambientes de prueba:** Debido a que la compra de hardware en general no es barata, a las empresas se les dificulta encontrar la manera

de emplear un método para entrenar correctamente a su personal de TI en materia de virtualización. La virtualización anidada provee la capacidad a los profesionales de TI de simular ambientes de virtualización enteros, sin tener que lidiar con el gasto de equipo de cómputo adicional. Dichos ambientes de prueba virtualizados son sumamente útiles cuando se emplean para fines didácticos. Las universidades se pueden apoyar con dichas plataformas para que sus estudiantes en Sistemas o carreras afines puedan aplicar de manera práctica sus conocimientos sobre virtualización.

**b) Contenedores:** Los contenedores son un tipo de máquina virtual que se enfoca en proveer un ambiente aislado para una aplicación específica, sin tener que virtualizar y utilizar recursos en un sistema operativo completo (Syrewicze, 2016).

A pesar de las inherentes ventajas que nos presenta la virtualización anidada como lo son el ahorro de costos, flexibilidad de uso e implementación, además de ser una plataforma con gran potencial didáctico, también presenta la gran desventaja de una disminución considerable del rendimiento del equipo donde se corra. La virtualización dentro de virtualización tiene una sobrecarga de recursos hasta de un 15%. Las nuevas tecnologías de hardware, así como las actualizaciones de los hipervisores están mejorando esta situación, ya que, con una mejora continua de los equipos de cómputo y la afluencia de controladores para virtualización anidada, esta sobrecarga podría llegar a reducirse hasta un 5% (Stuart, 2012).

Debido a lo anterior, no se recomienda utilizar la virtualización anidada en ambientes de producción, ya que algunas funcionalidades siguen en fases de prueba y aún no han sido perfeccionadas por los desarrolladores (Syrewicze, 2016). No obstante, es la herramienta perfecta tanto para estudiantes como para profesionales de tecnologías de la información debido a que los ayuda a capacitarse y entender más a detalle los conceptos de virtualización.

## Implementación de virtualización anidada

El primer objetivo de este trabajo ha sido demostrar la funcionalidad de la virtualización anidada. Primeramente, se debió determinar cuál sería el hipervisor principal de la solución. Dicho hipervisor sería aquel que estuviera instalado directamente en un servidor físico y que además contendría toda la estructura de virtualización. Para esto se pusieron a prueba tanto Hyper-V de Microsoft como ESXi de VMware, los cuales son de los hipervisores más populares en el mercado en la actualidad, de manera que se pudieran comparar sus características.

### • Virtualización anidada en Hyper-V

Luego de probar ambos hipervisores, finalmente se optó por hacer el desarrollo a través de Hyper-V, siendo estos algunos de los factores que contribuyeron para elegirlo sobre ESXi:

+ Cuenta con consolas de administración y herramientas para personal de TI familiarizado con interfaces basadas en Windows.

+ Debido al convenio de la universidad con Microsoft, se pudieron disponer de las mismas licencias para la activación de Windows Server

2016.

+ La virtualización anidada no está soportada oficialmente con VMware. ESXi tiene restricciones tiempo real muy estrictas que no siempre se pueden cumplir en un entorno virtualizado, en donde el hipervisor tiene que compartir el control del hardware físico con el sistema operativo del host (VMware, Support for running ESXi/ESX as a nested virtualization solution, 2017).

La virtualización anidada en Hyper-V es una funcionalidad que permite correr Hyper-V o algún otro hipervisor dentro de una máquina virtual creada con Hyper-V. En otras palabras, es posible que un hipervisor host pueda ser virtualizado y a su vez poder crear otras máquinas virtuales en este. Correr un contenedor en un host virtualizado, probar escenarios con múltiples máquinas sin contar con hardware dedicado o hasta desplegar un laboratorio de cómputo completo en un ambiente virtualizado son algunos casos de uso para virtualización anidada (Thompson y Cooley, 2016), siendo este último la finalidad de este trabajo.

### • Configuración del servidor físico

Para poder implementar virtualización anidada sobre Hyper-V, primero se tuvieron que tener en cuenta los siguientes prerequisites en el servidor físico donde se harían las pruebas:

- + Procesador(es) que soportara(n) virtualización asistida por hardware.
- + Sistema operativo Windows Server 2012 R2 o posterior.
- + Tener habilitado el rol de Hyper-V.
- + Hyper-V debía estar actualizado a la versión 8.0 o posterior.

Posteriormente se crearon las máquinas virtuales donde se crearon los ambientes de prueba (véase siguiente sección), para luego hacer una configuración adicional a cada una de manera que soportaran virtualización anidada. Hecho lo anterior, se deshabilitó la memoria dinámica para cada máquina virtual debido a que no tendría efecto al momento que los hipervisores virtualizados estuvieran corriendo (es necesario apagar la máquina virtual para hacer el ajuste de memoria manual). Luego se habilitó el MAC spoofing en el adaptador virtual de red de cada máquina, de manera que permitiera que los paquetes de red pudieran transferirse entre switches virtuales; esto para que las máquinas virtuales de cada hipervisor pudieran comunicarse con el resto de la red. Por último, se le asignó a cada máquina virtual un mínimo de dos procesadores virtuales por máquina (Thompson y Cooley, 2016).

## Implementación de máquinas virtuales estándar: nivel de virtualización 1

Después de haber configurado el servidor físico y creado las dos máquinas virtuales, se instalaron Hyper-V y ESXi en cada máquina respectivamente. Ambas máquinas se conectaron a un mismo switch virtual de tipo externo, de manera que se les pudiera proveer de direccionamiento estático y conectividad a internet. Adicional, se creó una máquina virtual que fungiría como cliente de VMware (Windows 10), para así gestionar

las máquinas virtuales contenidas en VMware Server. El rango de direcciones IP estáticas utilizado para todo el ambiente fue del 148.210.68.231 (host físico) al .239. En la Figura 1 se muestra el nivel de virtualización 1 corriendo por encima del nivel físico.

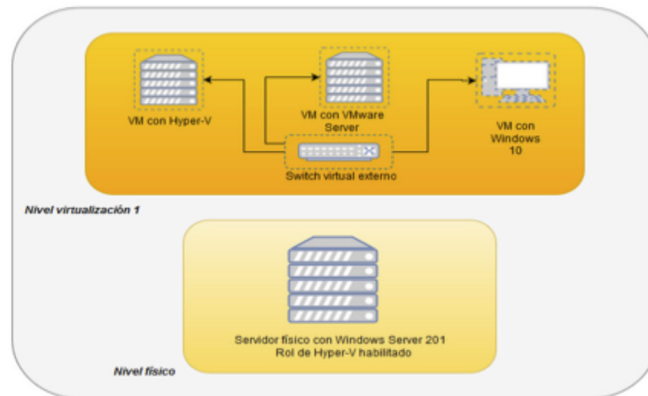


Figura 1. Esquema de virtualización nivel 1 (virtualización sobre host físico).

### • Máquina virtual con Hyper-V (LC-Hyper-V)

Se creó una máquina virtual con las siguientes características:

- + 10240 MB (10 GB) de memoria RAM.
- + 80 GB de disco duro.
- + Imagen (.iso) montada de Windows Server 2016 (x64).
- + Conectado al switch virtual externo.

En esta máquina virtual se instaló el sistema operativo de Windows Server 2016, de manera que se pudiera habilitar el rol de Hyper-V. Para que la administración del sistema operativo pudiera ser más sencilla, este se instaló en modo estándar con experiencia de escritorio. La única configuración adicional fue asignarle la dirección IP estática 148.210.68.232, de manera que se pudiera comunicar con el exterior. Una vez habilitado Hyper-V, se creó una máquina virtual con Windows 10 que serviría como template para el resto de las máquinas del laboratorio. Una vez creada la máquina virtual, se instaló y actualizó el sistema operativo para posteriormente preparar la máquina para hacerla template mediante la herramienta sysprep. La función de esta herramienta es crear una imagen personalizada del Windows, que limpia la información de seguridad del equipo (SIDs) donde está alojado y la prepara para un arranque desde cero, tomando en cuenta las configuraciones y aplicaciones instaladas previamente en este. Luego que se replicara la máquina virtual original, se creó un switch privado para interconectar las máquinas en una red aislada, de manera que se le pudieran asignar direcciones IP de un rango privado.

### • Máquina virtual con VMware (LC-ESXi)

Se creó una segunda máquina en el hipervisor principal para alojar ESXi, con las siguientes características:

+ 10240 MB (10 GB) de memoria RAM.

+ 80 GB de disco duro.

Debido a que por default el kernel de ESXi no cuenta con los controladores necesarios para poder correr sobre una NIC virtual (en el caso de Hyper-V los Microsoft Networking Adapters), fue necesario “inyectar” los controladores necesarios en la copia del .iso de ESXi a utilizar en esta máquina virtual. Para ello fue necesario contar con lo siguiente:

+ VMware PowerShell CLI, la cual es una línea de comandos basada en PowerShell, la cual se utiliza para gestionar y automatizar ambientes de VSphere.

+ ESXi-Customizer-PS, el cual es un script de PowerShell para inyectar los controladores requeridos en un .iso de ESXi 5.x/6.x. Teniendo en cuenta lo anterior, se corrió el siguiente comando en Windows PowerShell sobre la ruta del directorio donde se encontraba almacenado el script: `ESXi-Customizer-PS-v2.4.ps1 -v60 -vft -load net-tulip`. Esto hizo que se descargara una copia de ESXi v6.0 de un repositorio de imágenes a la cual se le inyectaron los controladores necesarios (Scott-Raynsford, 2016).

Al haber descargado la imagen, se montó en la máquina virtual para su instalación. Adicional a esta imagen, se le agregó un adaptador de red virtual Legacy en vez de uno convencional, ya que VMware no soporta los adaptadores de red virtuales convencionales. Los adaptadores Legacy emulan un adaptador de red físico (Microsoft, Configure Networking, 2017).

### • Máquina virtual con Windows 10 (LC-ESXi Cliente)

Por último, se creó una tercera máquina virtual en el hipervisor principal para fungir como el cliente de ESXi, desde el que se haría la administración de las máquinas virtuales, con las siguientes características:

+ 4096 MB (4 GB) de memoria RAM.

+ 50 GB de disco duro.

+ Imagen (.iso) montada de Windows 10 Education (x64).

+ Adaptador de red Legacy (para establecer comunicación con el servidor ESXi).

Una vez instalado y configurado el sistema operativo, se accedió al servidor ESXi por medio de la aplicación cliente de vSphere, mediante el cual se crearon las máquinas virtuales contenidas en el servidor ESXi. De igual manera que en la máquina con Hyper-V, primero se creó una máquina virtual con Windows 10 que serviría como template para el resto de las máquinas del laboratorio, con las siguientes características base:

+ 2 GB de memoria RAM.

+ 20 GB de disco duro.

+ Imagen (.iso) montada de Windows 10 Education (x86).

Sin embargo, debido a que ESXi no soporta la virtualización anidada de manera nativa, se tuvo que agregar el parámetro `vmx.allowNested = TRUE` a la máquina virtual, la cual permite la instalación de un sistema operativo (VMware, Installing a nested virtual machine fails with the error, 2015). El sistema operativo se tuvo que instalar en su versión de 32 bits, debido a que Hyper-V no virtualiza las características de virtualización asistida por hardware con las que cuenta el CPU físico (Mattson, 2015).

Al igual que en la máquina virtual con Hyper-V, se creó un switch privado para interconectar las máquinas virtuales que se crearan en ESXi en una red aislada.

## ▶ Resultado de pruebas de máquinas virtuales anidadas: nivel de virtualización 2

Luego de realizar las configuraciones correspondientes en cada hipervisor virtualizado, se procedió a replicar el template en tres máquinas virtuales por hipervisor. En la Figura 2, se muestra la arquitectura completa, con el nivel de virtualización 2. Inicialmente se hicieron pruebas para ver cuanta memoria había que asignarle a cada máquina, de manera que pudieran correr todas a la vez sin causar conflicto con la memoria necesaria para el funcionamiento de la máquina virtual/ hipervisor que las alojaba.

Una vez que se aseguró que todas las máquinas pudieran correr simultáneamente sin problemas, se les asignaron direcciones IP privadas, de manera que se creara una red aislada para cada grupo de máquinas por hipervisor.

### Direcciones red Hyper-V:

Segmento: 10.0.1.0/24

Gateway: 10.0.1.1

### Direcciones red ESXi:

Segmento: 10.0.2.0/24

Gateway: 10.0.2.

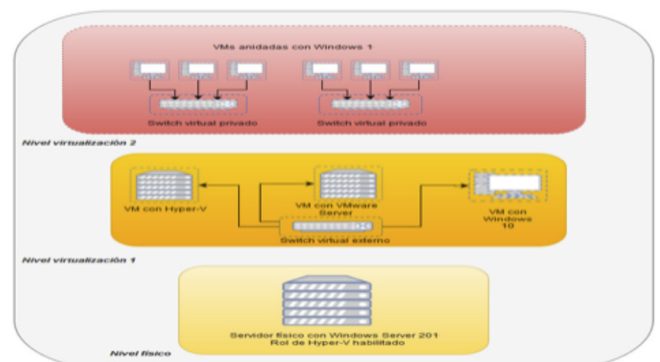


Figura 2. Esquema de virtualización nivel 2 (virtualización sobre hosts virtuales).

La primera máquina de cada grupo (HV-VM1, ES\_VM1) recibió la dirección .1 de su segmento, de manera que fungieran como gateway para interconectar el resto de las máquinas de su red. Posteriormente se hizo ping entre las máquinas para confirmar que hubiera comunicación entre ellas. Finalmente se probó que la conexión de red entre máquinas era correcta (Figura 3 y Figura 4), lo cual pudo proveer un ambiente virtualizado aislado el cual no tuviera salida hacia las otras máquinas hipervisores o hacia el host físico. Cabe resaltar que el desempeño de las máquinas virtuales de segundo nivel no fue óptimo, pero esto se debe a los recursos limitados del servidor físico. En el caso de las máquinas de ESXi el desempeño fue aún más pobre que en Hyper-V; esto debido a la naturaleza del hipervisor, el cual no está diseñado para funcionar correctamente en modo anidado.

```

C:\Users\Lab1_U2>ping 10.0.1.2
Pinging 10.0.1.2 with 32 bytes of data:
Reply from 10.0.1.2: bytes=32 time=1ms TTL=128
Reply from 10.0.1.2: bytes=32 time=1ms TTL=128
Reply from 10.0.1.2: bytes=32 time=1ms TTL=128
Reply from 10.0.1.2: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 3ms, Average = 0ms

C:\Users\Lab1_U3>ping 10.0.1.3
Pinging 10.0.1.3 with 32 bytes of data:
Reply from 10.0.1.3: bytes=32 time=0ms TTL=128
Reply from 10.0.1.3: bytes=32 time=0ms TTL=128
Reply from 10.0.1.3: bytes=32 time=2ms TTL=128
Reply from 10.0.1.3: bytes=32 time=2ms TTL=128
Ping statistics for 10.0.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 8ms, Average = 3ms

C:\Users\Lab1_U2>ping 10.0.1.1
Pinging 10.0.1.1 with 32 bytes of data:
Reply from 10.0.1.1: bytes=32 time=1ms TTL=128
Reply from 10.0.1.1: bytes=32 time=1ms TTL=128
Reply from 10.0.1.1: bytes=32 time=2ms TTL=128
Reply from 10.0.1.1: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Users\Lab1_U3>ping 10.0.1.1
Pinging 10.0.1.1 with 32 bytes of data:
Reply from 10.0.1.1: bytes=32 time=1ms TTL=128
Reply from 10.0.1.1: bytes=32 time=1ms TTL=128
Reply from 10.0.1.1: bytes=32 time=1ms TTL=128
Reply from 10.0.1.1: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
    
```

Figura 3. Confirmando comunicación mediante pings en Hyper-V.

```

C:\Users\Lab2_U3>ping 10.0.2.2
Pinging 10.0.2.2 with 32 bytes of data:
Reply from 10.0.2.2: bytes=32 time=1ms TTL=128
Reply from 10.0.2.2: bytes=32 time=2ms TTL=128
Reply from 10.0.2.2: bytes=32 time=8ms TTL=128
Reply from 10.0.2.2: bytes=32 time=8ms TTL=128
Ping statistics for 10.0.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 7ms, Average = 25ms

C:\Users\Lab2_U3>ping 10.0.2.3
Pinging 10.0.2.3 with 32 bytes of data:
Reply from 10.0.2.3: bytes=32 time=1ms TTL=128
Reply from 10.0.2.3: bytes=32 time=1ms TTL=128
Reply from 10.0.2.3: bytes=32 time=1ms TTL=128
Reply from 10.0.2.3: bytes=32 time=2ms TTL=128
Ping statistics for 10.0.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 3ms

C:\Users\Lab2_U2>ping 10.0.2.1
Pinging 10.0.2.1 with 32 bytes of data:
Reply from 10.0.2.1: bytes=32 time=4ms TTL=128
Reply from 10.0.2.1: bytes=32 time=30ms TTL=128
Reply from 10.0.2.1: bytes=32 time=2ms TTL=128
Reply from 10.0.2.1: bytes=32 time=3ms TTL=128
Ping statistics for 10.0.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 45ms, Average = 20ms

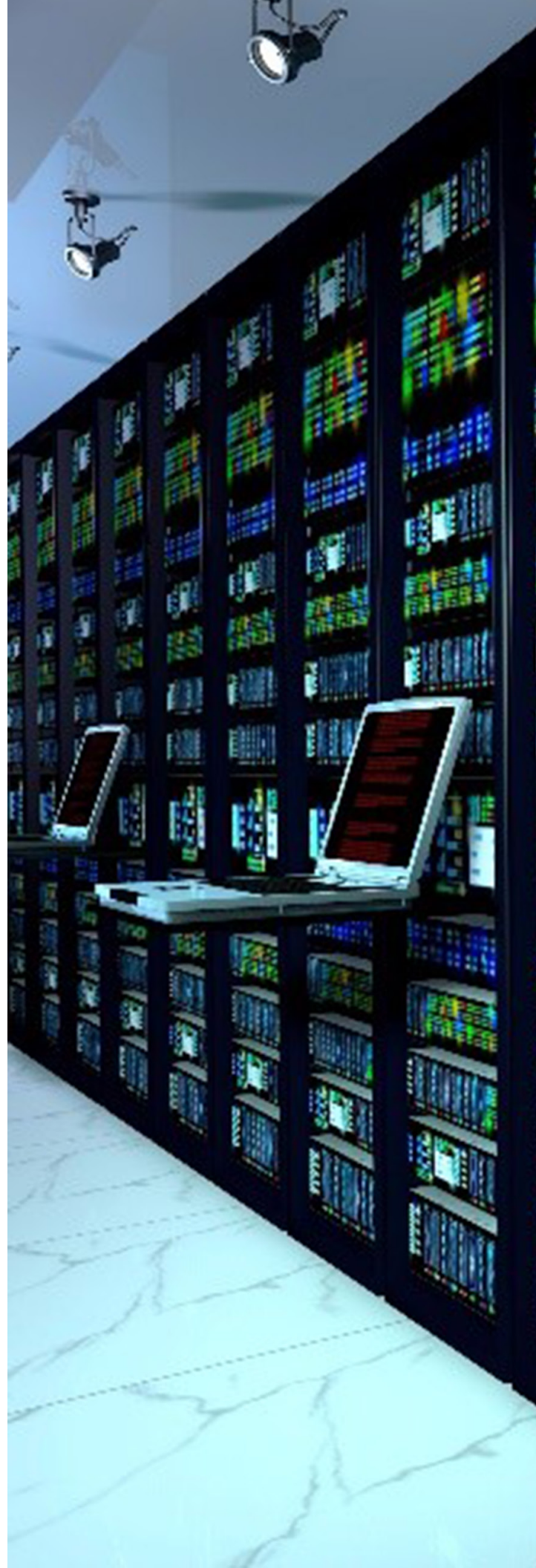
C:\Users\Lab2_U2>ping 10.0.2.1
Pinging 10.0.2.1 with 32 bytes of data:
Reply from 10.0.2.1: bytes=32 time=47ms TTL=128
Reply from 10.0.2.1: bytes=32 time=10ms TTL=128
Reply from 10.0.2.1: bytes=32 time=1ms TTL=128
Reply from 10.0.2.1: bytes=32 time=3ms TTL=128
Ping statistics for 10.0.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 47ms, Average = 15ms
    
```

Figura 4. Confirmando comunicación mediante pings en ESXi.

## Conclusiones

Con este trabajo se probó que la virtualización anidada funciona de manera efectiva como un medio viable para crear ambientes de prueba. En dichos ambientes, los alumnos pueden disponer de su propio hipervisor de manera que puedan experimentar y comprender los conceptos de virtualización. Cabe resaltar que la robustez del equipo físico determinará cuántos ambientes de prueba se pueden crear a la vez, así como el desempeño que tendrán estos.

Es importante destacar que la implementación de la virtualización anidada representa un importante ahorro de recursos y por lo tanto de dinero, lo cual convierte a este tipo de tecnología en la solución ideal no solo para laboratorios de cómputo, sino para todo tipo de organizaciones que requieren dar servicio a una cantidad grande de usuarios.





## Bibliografía

- + Al Kaab, S.; Al Kindi, N.; Al Fazarri, S. y Trabelsi, Z. (2016). *Virtualization based Ethical Educational Platform for Hands-on Lab Activities on DoS Attacks*. IEEE Global Engineering Education Conference.
- + ESXi 6 install stuck on “Relocating modules and starting up the kernel...” (2016). Obtenido de *Tales from the Datacenter*: <http://talesfromthedatacenter.com/archives/582>
- + IBM (2010). *IBM Developer Works*. (Biblioteca Tecnica) Obtenido de <https://www.ibm.com/developerworks/ssa/websphere/library/ws-vcl/>
- + Mattson, J. (2015). *Failed to start the virtual machine*. Obtenido de *VMware Communities*: <https://communities.vmware.com/thread/511199>
- + Microsoft (2017). *¿Qué es Sysprep?* Obtenido de *Technet*: [https://technet.microsoft.com/es-es/library/cc721940\(-v=ws.10\).aspx](https://technet.microsoft.com/es-es/library/cc721940(-v=ws.10).aspx)
- + Microsoft. (2017). *Configure Networking*. Obtenido de *Technet*.
- + Misevicien, R.; Ambrazien, D.; Tuminauskas, R. y Pazereckas, N. (2012). *Educational Infrastructure Using Virtualization Technologies: Experience at Kaunas University of Technology*. *Informatics in Education*, 11(2), 227–240.
- + Murphy, M. C. y McClelland, M. K. (2009). *My Personal Computer Lab: Operating in the “Cloud”*. *Information Systems Education Journal*, 7(93), 1-11.
- + Oracle® (2014). *What are Hypervisors?* Obtenido de *Concepts Guide for Release 3.3*: [https://docs.oracle.com/cd/E50245\\_01/E50249/html/vmcon-hypervisor.html](https://docs.oracle.com/cd/E50245_01/E50249/html/vmcon-hypervisor.html)
- + Scott-Raynsford, D. (2016). *Install a VMWare ESXi 6.0 Hypervisor in a Hyper-V VM*. Obtenido de *PowerShell, Programming and DevOps*: <https://dsco-trraynsford.wordpress.com/2016/04/22/install-a-vmware-esxi-6-0-hypervisor-in-a-hyper-v-vm>
- + Stuart (2012). *Second Edge*. Obtenido de *The Impact of Nested Virtualization*: <https://www.second-edge.com/?p=101>
- + Syrewicze, A. (2016). *How to enable Nested Virtualization on Hyper-V y Windows Server 2016*. Obtenido de *Hyper-V Hub by Altaro*: <http://www.altaro.com/hyper-v/nested-virtualization-hyper-v-windows-server-2016/>
- + Thompson, T. y Cooley, S. (2016). *Run Hyper-V in a Virtual Machine with Nested Virtualization*. Obtenido de *Microsoft Docs*: <https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/nested-virtualization>
- + VMware. (2015). *Installing a nested virtual machine fails with the error*. Obtenido de *VMware Knowledge Base*: [https://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=2108724](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2108724)
- + VMware (2017). *Support for running ESXi/ESX as a nested virtualization solution*. Obtenido de *VMware Knowledge Base*: [https://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=k-c&docTypeID=DT\\_KB\\_1\\_1&externalId=2009916](https://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=k-c&docTypeID=DT_KB_1_1&externalId=2009916)