

Análisis y Virtualización con Xen Hypervisor y QEMU en Distribuciones Linux

Vasquez Lopez, Aldo

Arequipa, Perú

avasquezl@ulasalle.edu.pe

Pinazo Vera, María Fernanda

Arequipa, Perú

mpinazov@ulasalle.edu.pe

Machaca Meléndez, Alvaro

Arequipa, Perú

amachacam@ulasalle.edu.pe

Chura Pacci, Jerson

Arequipa, Perú

jchurap@ulasalle.edu.pe

Resumen — En esta actividad se realizó la configuración y prueba de conectividad entre dos máquinas virtuales (VMs) conectadas en una red privada. Se tomaron en cuenta los datos sobre la CPU, memoria y tamaño en disco asignados a cada VM. Se utilizaron herramientas de virtualización como Qemu y VirtualBox.

Palabras clave — Virtualización, Red Privada, Conectividad, Máquinas Virtuales, Qemu, VirtualBox.

I. INTRODUCTION

“La virtualización ha revolucionado la gestión de redes y la computación en la nube, facilitando la transición hacia arquitecturas más flexibles y eficientes. Tecnologías como Software Defined Networking (SDN) y Network Function Virtualization (NFV) son esenciales para soportar redes de próxima generación, como las redes 5G [1].”

Este documento se centra en la implementación de estas tecnologías en sistemas operativos de distribución Linux, utilizando hipervisores como Xen y herramientas de emulación como QEMU.

II. INVESTIGACIÓN ESTABLECIDA HACIA XEN HYPERVISOR Y QEMU

A. Arquitecturas y Diseños de SDN y NFV

SDN y NFV representan una evolución significativa en la gestión y operación de redes. SDN separa el plano de control del plano de datos, permitiendo una gestión centralizada y dinámica de la red. NFV, por su parte, permite que las funciones de red que tradicionalmente se ejecutaban en hardware dedicado se ejecuten en máquinas virtuales sobre hardware genérico. Esto no solo reduce costos operativos, sino que también mejora la flexibilidad y escalabilidad de la red.

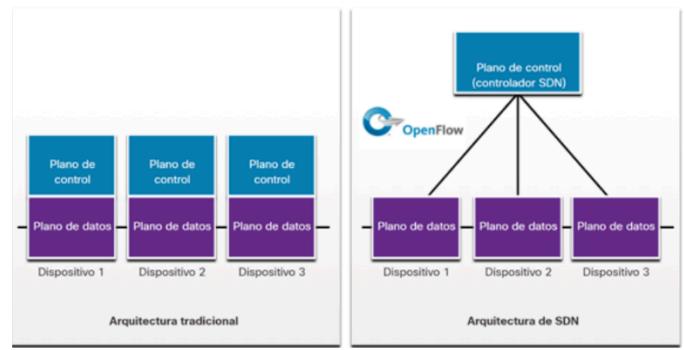


Fig. 1. Diagrama SDN

a.)

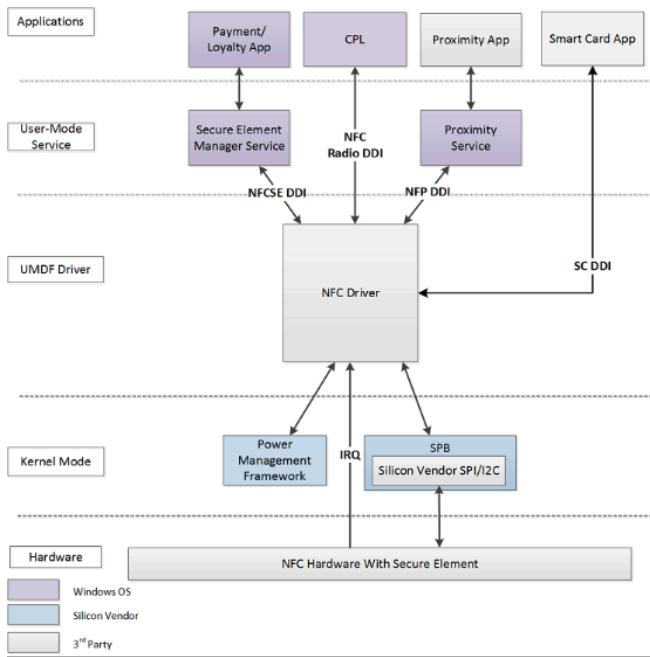


Fig. 2. Diagrama NFC

B. Integración y Orquestación SDN-NFV

La integración de SDN y NFV es crucial para optimizar el rendimiento de la red. La orquestación implica la coordinación de recursos y servicios en la red para asegurar que se cumplan las necesidades de los usuarios finales de manera eficiente. Esto incluye la gestión de la ubicación de las funciones de red, la asignación de recursos y la adaptación dinámica a las condiciones cambiantes de la red.

C. Optimización y Gestión de Redes

Un aspecto crítico de SDN y NFV es la capacidad de optimizar el uso de recursos de la red. Esto incluye la optimización del tráfico, la asignación dinámica de ancho de banda y la gestión eficiente de recursos computacionales. Además, la gestión de redes con SDN permite una respuesta rápida a problemas de red, mejorando la resiliencia y la calidad del servicio.

D. Seguridad en SDN y NFV

La seguridad es un tema crucial en la implementación de SDN y NFV. La centralización del control en SDN puede ser una vulnerabilidad, ya que un ataque exitoso podría comprometer toda la red. Por lo tanto, es vital implementar medidas de seguridad robustas, como la autenticación de dispositivos, el cifrado de datos y la detección de intrusiones.

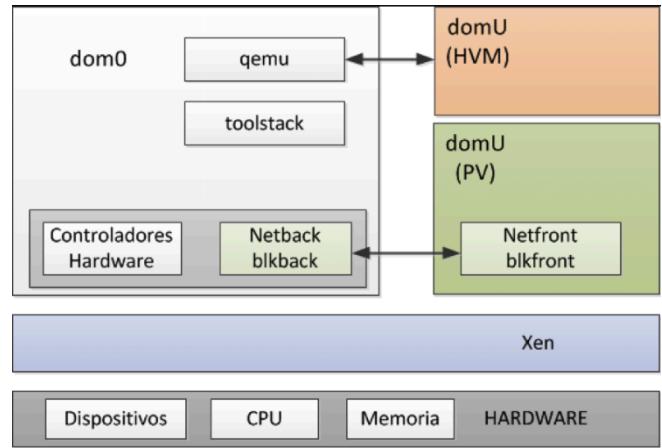


Fig. 3. Evaluación de la Capacidad de Particionamiento Adaptativo y Tiempo Real para Virtualización con Xen

"La arquitectura de Xen es modular y permite la separación estricta de dominios, lo que reduce significativamente el riesgo de compromisos de seguridad. Schulz y Annighöfer (2022) destacaron la capacidad de particionamiento adaptativo y la capacidad en tiempo real del hipervisor Xen, lo que es crucial para aplicaciones en tiempo real [4]."

E. Arquitectura de Xen

Xen es un hipervisor de código abierto que permite la separación estricta de dominios y la ejecución de componentes críticos en entornos aislados. Esto reduce significativamente el riesgo de compromisos de seguridad. La arquitectura modular de Xen permite que el hipervisor y el kernel de Linux operen de manera separada, lo que limita el impacto de las vulnerabilidades del sistema huésped sobre el hipervisor.

F. Introspección de Máquina Virtual (VMI)

VMI es una técnica avanzada de seguridad que permite la monitorización y protección contra intrusiones y ataques de malware utilizando APIs asistidas por hardware. Esta técnica añade una capa adicional de seguridad, permitiendo la detección y respuesta a amenazas en tiempo real sin afectar el rendimiento de las máquinas virtuales.

G. Desprivilegiación de QEMU

QEMU es una herramienta utilizada para la emulación de hardware en entornos virtuales. Ejecutar QEMU como un usuario no privilegiado, una técnica conocida como sandboxing, minimiza el impacto de posibles vulnerabilidades. Esta técnica ayuda a contener cualquier compromiso dentro de un entorno controlado, mejorando la seguridad general del sistema.

H. Parcheo en Vivo

El parcheo en vivo es una capacidad crítica para mantener la seguridad en entornos virtualizados. Permite aplicar parches de seguridad sin necesidad de reiniciar el sistema, minimizando el tiempo de inactividad y asegurando que el sistema permanezca protegido contra vulnerabilidades conocidas sin interrupciones significativas.



Fig. 4. Xen y el Arte de la Virtualización

I. Interfaz de Memoria de Máquina Virtual

Xen introduce el concepto de paravirtualización, que requiere modificaciones mínimas en los sistemas operativos invitados para mejorar significativamente el rendimiento. Esta técnica facilita la virtualización eficiente del x86, una arquitectura que no fue diseñada originalmente para soportar la virtualización.

J. Interfaz de CPU de Máquina Virtual

Xen modifica los sistemas operativos invitados para que operen en un nivel de privilegio más bajo que el hipervisor, utilizando los anillos de privilegio del x86 (anillos 0 a 3). Esta técnica mejora la seguridad y el rendimiento al gestionar las excepciones de llamadas al sistema y fallos de página directamente a través de manejadores rápidos registrados por los sistemas operativos invitados.

III. VIRTUALIZACIÓN DEL SISTEMA CON QEMU Y XEN HYPERVISOR Y PRUEBAS

"Xen introduce el concepto de paravirtualización, que requiere modificaciones mínimas en los sistemas operativos invitados para mejorar significativamente el rendimiento. Barham et al. (2003) explican cómo Xen mejora la seguridad y el rendimiento al gestionar las excepciones de llamadas al sistema y fallos de página directamente a través de manejadores rápidos registrados por los sistemas operativos invitados[5]."

Debido a errores de versionamiento, Xen Hypervisor no puede ser ejecutado para virtualizar un Sistema Operativo. Por ello, se realizó la virtualización desde QEMU y se compararon dichas características al final de los resultados.

A. Instalando Qemu

Para descargar Qemu, debemos ingresar a la página de descarga del archivo ejecutable. En nuestro caso desarrollaremos la virtualización en Windows, si la instalación se hiciera desde linux, entonces simplemente se utilizarían los comandos que se presentarán a continuación de frente en la consola.

<https://qemu.weilnetz.de/>

En la página oficial descargamos el archive .exe más reciente y la aplicación de escritorio que nos permitirá ejecutar virtualizar con QEMU, de una manera más rápida y sencilla

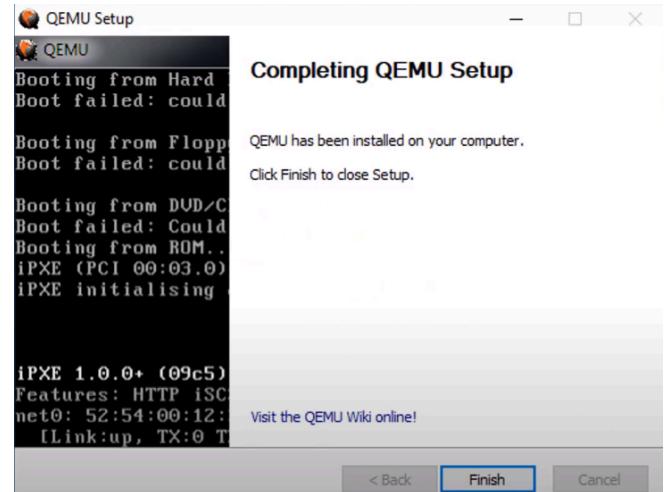


Fig. 5. Aceptamos todos los aspectos solicitados por qemu.exe y completamos la instalación.

Luego de que se haya completado la instalación de QEMU, procedemos a buscar la carpeta donde se procedió con la instalación y la añadimos al path.

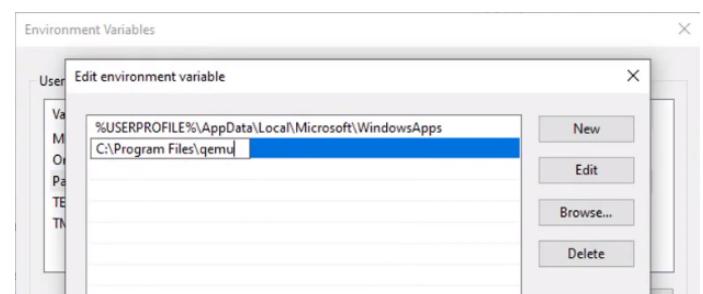


Fig. 6. Añadiendo la ruta de QEMU al path.

Una vez que ya se tenga configurado el path, nos dirigimos al cmd, y escribimos lo siguiente.

```
PS C:\Users\Aldo> qemu-system-x86_64.exe -boot d -cdrom debian.iso -m 2048
```

Fig. 7. Virtualizando el Sistema operativo Debian

Para ello, se debe tener instalado con anterioridad la imagen iso del Sistema Operativo que se desea. Si la instrucción realizada no da errores, entonces, se podrá observar la virtualización del S.O. deseado.

```

9: > #5 lógica 1.8 GB f ext4 /var
10: > #6 lógica 1.0 GB f intercambio intercambio
11: > #7 lógica 396.4 MB f ext4 /tmp
12: > #8 lógica 14.3 GB f ext4 /home
13:

14: Deshacer los cambios realizados a las particiones,
15: Finalizar el particionado y escribir los cambios en el disco [*],
Indicador: <?> para obtener ayuda, por omisión=15>
15

Se escribirán en los discos todos los cambios indicados a continuación si continúa. Si no lo hace
podrá hacer cambios manualmente.

Se han modificado las tablas de particiones de los siguientes dispositivos:
  SCSI3 (0,0,0) (sda)

Se formatearán las siguientes particiones:
  partición #1 de SCSI3 (0,0,0) (sda) como ext4
  partición #5 de SCSI3 (0,0,0) (sda) como ext4
  partición #6 de SCSI3 (0,0,0) (sda) como intercambio
  partición #7 de SCSI3 (0,0,0) (sda) como ext4
  partición #8 de SCSI3 (0,0,0) (sda) como ext4

Desea escribir los cambios en los discos?
 1: Sí 2: No [*]
Indicador: <?> para obtener ayuda, por omisión=2>
1

Formato de particiones ... 33z
Formato de particiones
Instalando el sistema base ... 0%... 17%... 20%

```

Fig. 8. Virtualizando Debian desde QEMU.

Una vez que hayan culminado las acciones de virtualización, podremos observar que nuestro Sistema Operativo Debian, ya está listo para ser usado.

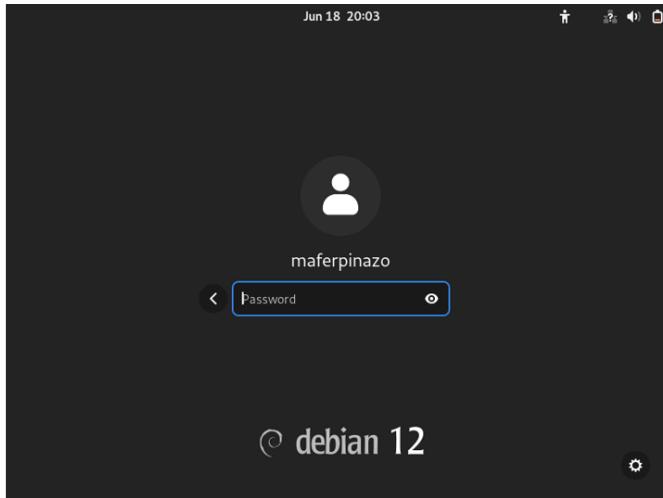


Fig. 9. Debian como Maquina Virtual desde QEMU.

En este momento, ya que se tiene la Máquina Virtual, debemos instalar la paquetería correspondiente. Primero, usamos la cuenta root, e instalamos los paquetes directamente con permisos root. Para aquello, utilizamos el comando `$ su -`, para instalar los paquetes `sudo apt update`, seguido de esto, `sudo apt install gcc g++ screenfetch vim`.

Fig. 10. Instalando la paquetería correspondiente.

Ahora, comprobamos la instalación correcta de los paquetes instalados, con los siguientes comandos: `gcc--version`, `g++--version`, `screenfetch`, `vim --version`.

```
root@Debian:~# gcc --version
gcc (Debian 12.2.0-14) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

root@Debian:~# g++ --version
g++ (Debian 12.2.0-14) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

root@Debian:~# screenfetch
      _.,met$$$$$gg.
     ,g$$$$$$$$$$$$$$$$$SP.
    ,gSSP""        """YSSS".
   ,SSP'          '$$.
  ',SSP'        .ggs.    '$$:
 `d$$'        ,$$"    .  $$$
  $$P d$`       ,  $$P
  $$: $$. -  ,d$$'
  $$\` YSB_. __d$P'
  YSS. > ``Y$$$$$SP"
  '$$b
  `YSS
  '$$b.
  `YSSb.
  ``Ysb_
  ``Ysb_``

root@Debian
OS: Debian 12 bookworm
Kernel: x86_64 Linux 6.1.0-21-amd64
Uptime: 1h 0m
Packages: 1630
Shell: bash 5.2.15
Disk: 5.3G / 21G (28%)
CPU: 11th Gen Intel Core i5-1135G7 @ 2.419GHz
GPU: VMware SVGA II Adapter
RAM: 1393MiB / 3823MiB

root@Debian:~# vim --version
```

Fig. 11. Comprobando la versión de la paquetería.

```
#include <pthread.h>
#include <stdio.h>
#include <unistd.h>

#define NumFilosofos 5

pthread_mutex_t tenedores[NumFilosofos];
pthread_t filosofos[NumFilosofos];

void* filosofo(void* numero){
    int id = *(int*)numero;
    while(1){
        printf("Filosofo %d esta pensando...\n", id);
        sleep(1);
        printf("Filosofo %d tiene hambre...\n", id);
        pthread_mutex_lock(&tenedores[id]);
        pthread_mutex_lock(&tenedores[(id + 1) % NumFilosofos]);
        printf("Filosofo %d esta comiendo...\n", id);
        sleep(2);
        pthread_mutex_unlock(&tenedores[(id + 1) % NumFilosofos]);
        pthread_mutex_unlock(&tenedores[id]);
    }
}

int main(){
    int ids[NumFilosofos];
    for (int i = 0; i < NumFilosofos; i++){
        pthread_mutex_init(&tenedores[i], NULL);
    }
    for (int i = 0; i < NumFilosofos; i++){
        ids[i] = i;
        pthread_create(&filosofos[i], NULL, filosofo, &ids[i]);
    }
    for (int i = 0; i < NumFilosofos; i++)
        pthread_join(filosofos[i], NULL);
    return 0;
}
```

Fig. 12. Creando un archive en el lenguaje de programación C.

Para comprobar nuestro Código, se debe implementar un algoritmo que nos permita verificar, en este caso, se trata de Filosofos. Una vez que se haya creado el archive utilizando vim, procedemos a ejecutarlo.

```
Filosofo 2 esta pensando...
Filosofo 3 esta pensando...
Filosofo 4 esta pensando...
Filosofo 1 esta pensando...
Filosofo 0 esta pensando...
Filosofo 2 tiene hambre...
Filosofo 2 esta comiendo...
Filosofo 4 tiene hambre...
Filosofo 4 esta comiendo...
Filosofo 3 tiene hambre...
Filosofo 1 tiene hambre...
Filosofo 0 tiene hambre...
Filosofo 2 esta pensando...
Filosofo 1 esta comiendo...
Filosofo 4 esta pensando...
Filosofo 3 esta comiendo...
Filosofo 2 tiene hambre...
```

Fig. 13. Ejecutando Código de los Filósofos en C.

```
root@Debian:~# sudo apt-get update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm-updates InRelease
Hit:3 http://security.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
root@Debian:~# sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Fig. 14. Utilizamos los siguientes comandos sudo apt-get update y sudo apt-get upgrade, para mantener todo actualizado.

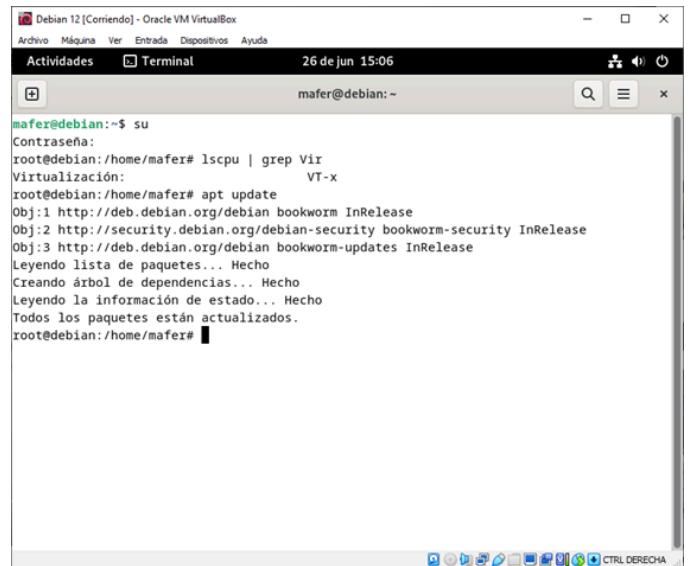


Fig. 15. Ingresamos al superusuario, para verificar que la virtualización en nuestro equipo esté habilitada, usamos el comando lscpu | grep Virt.

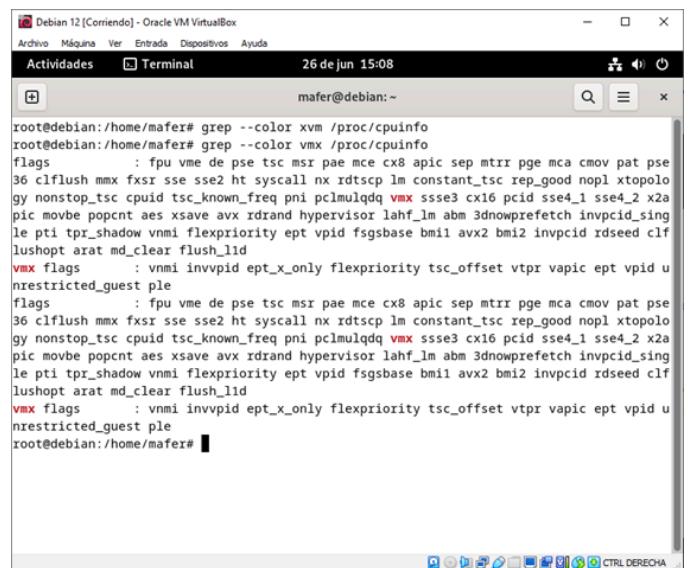


Fig. 16. Actualizamos y, nos vamos a asegurar que la verificación esté activa, para esto, utilizamos el comando grep -color vmx /proc/cpuinfo.

```

mafer@debian:~/home/mafer# apt-get -y install xen-hypervisor-4.8-amd64 xen-tools
...eyendo lista de paquetes... Hecho
...reando árbol de dependencias... Hecho
...eyendo la información de estado... Hecho
El paquete xen-hypervisor-4.8-amd64 no está disponible, pero algún otro paquete hace referencia a él. Esto puede significar que el paquete falta, está obsoleto o sólo se encuentra disponible desde alguna otra fuente
sin embargo, los siguientes paquetes lo reemplazan:
  xen-hypervisor-common

E: El paquete «xen-hypervisor-4.8-amd64» no tiene un candidato para la instalación
root@debian:~/home/mafer# apt search xen-hypervisor
...ndiendo... Hecho
3uscar en todo el texto... Hecho
xen-hypervisor-4.17-amd64/stable 4.17.3+10-g091466ba55-1-deb12u1 amd64
  Xen Hypervisor on AMD64

xen-hypervisor-4.17-amd64-dbg/stable 4.17.3+10-g091466ba55-1-deb12u1 amd64
  debug symbols for Xen Hypervisor on AMD64

xen-hypervisor-common/stable 4.17.3+10-g091466ba55-1-deb12u1 amd64
  Xen Hypervisor - common files

root@debian:~/home/mafer# apt-get -y install xen-hypervisor-4.17-amd64 xen-tools
...eyendo lista de paquetes... Hecho
...reando árbol de dependencias... Hecho
...ndiendo la información de estado... Hecho

```

Fig. 17. Instalamos Xen Hypervisor, con el siguiente comando: sudo apt-get -y install xen-hypervisor-4.17-amd64 xen-tools.

```

Debian 12 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 26 de jun 15:20
mafer@debian:~
```

```

root@debian:/home/mafer# sudo update-grub
Including Xen overrides from /etc/default/grub.d/xen.cfg
Warning: GRUB_DEFAULT changed to boot into Xen by default! Edit /etc/default/grub.d/xen.cfg to avoid this warning.
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-6.1.0-21-amd64
Found initrd image: /boot/initrd.img-6.1.0-21-amd64
Found linux image: /boot/vmlinuz-6.1.0-18-amd64
Found initrd image: /boot/initrd.img-6.1.0-18-amd64
Found linux image: /boot/vmlinuz-6.1.0-21-amd64
Found initrd image: /boot/initrd.img-6.1.0-21-amd64
Found linux image: /boot/vmlinuz-6.1.0-18-amd64
Found initrd image: /boot/initrd.img-6.1.0-18-amd64
Found linux image: /boot/vmlinuz-6.1.0-21-amd64
Found initrd image: /boot/initrd.img-6.1.0-21-amd64
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
root@debian:/home/mafer# grep -i "menuentry" "/boot/grub/grub.cfg|sed -r "s|--class .* $||g|nl -v 0
  0 menuentry 'Debian GNU/Linux'
```

Fig. 17. Instalamos Xen Hypervisor, con el siguiente comando: sudo apt-get -y install xen-hypervisor-4.17-amd64 xen-tools.

```

Debian 12 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 26 de jun 15:19
mafer@debian:~
```

```

root@debian:/home/mafer# dpkg-divert --divert /etc/grub.d/08_linux_xen --rename /etc/grub.d/20_linux_xen
Añadiendo 'desviación local de /etc/grub.d/20_linux_xen a /etc/grub.d/08_linux_xen'
root@debian:/home/mafer# grep -i "menuentry" "/boot/grub/grub.cfg|sed -r "s|--class .* $||g|nl -v 0
  0 menuentry 'Debian GNU/Linux'
    1     menuentry 'Debian GNU/Linux, with Linux 6.1.0-21-amd64'
    2     menuentry 'Debian GNU/Linux, with Linux 6.1.0-21-amd64 (recovery mode)'

    3     menuentry 'Debian GNU/Linux, with Linux 6.1.0-18-amd64'
    4     menuentry 'Debian GNU/Linux, with Linux 6.1.0-18-amd64 (recovery mode)'

    5 menuentry 'Debian GNU/Linux, with Xen hypervisor'
      6           menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-21-amd64'
      7           menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-21-amd64 (recovery mode)'
      8           menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-18-amd64'
      9           menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-18-amd64 (recovery mode)'
     10          menuentry 'Debian GNU/Linux, with Xen 4.17-amd64.efi and Linux 6.1.0-21-amd64'
     11          menuentry 'Debian GNU/Linux, with Xen 4.17-amd64.efi and Linux 6.1.0-21-amd64 (recovery mode)'
```

Fig. 18. Utilizamos el comando dpkg-divert --divert /etc/grub.d/08_linux_xen --rename /etc/grub.d/20_linux_xen, con la finalidad de cambiar el arranque del sistema.

Fig. 19. Actualizamos el grub, con sudo update-grub. Para que se guarden nuestros cambios de arranque del sistema.

```

Debian 12 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 26 de jun 15:20
mafer@debian:~
```

```

root@debian:/home/mafer# grep -i "menuentry" "/boot/grub/grub.cfg|sed -r "s|--class .* $||g|nl -v 0
  0 menuentry 'Debian GNU/Linux, with Xen hypervisor'
    1     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-21-amd64'
    2     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-21-amd64 (recovery mode)'
    3     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-18-amd64'
    4     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64 and Linux 6.1.0-18-amd64 (recovery mode)'
    5     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64.efi and Linux 6.1.0-21-amd64'
    6     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64.efi and Linux 6.1.0-21-amd64 (recovery mode)'
    7     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64.efi and Linux 6.1.0-18-amd64'
    8     menuentry 'Debian GNU/Linux, with Xen 4.17-amd64.efi and Linux 6.1.0-18-amd64 (recovery mode)'
    9     menuentry 'Debian GNU/Linux'
   10     menuentry 'Debian GNU/Linux, with Linux 6.1.0-21-amd64'
   11     menuentry 'Debian GNU/Linux, with Linux 6.1.0-21-amd64 (recovery mode)'

   12     menuentry 'Debian GNU/Linux, with Linux 6.1.0-18-amd64'
   13     menuentry 'Debian GNU/Linux, with Linux 6.1.0-18-amd64 (recovery mode)'
```

Fig. 20. El siguiente comando, nos ayuda a verificar que la primera línea está en 0, por lo tanto está configurado correctamente el arranque con Xen Hypervisor.

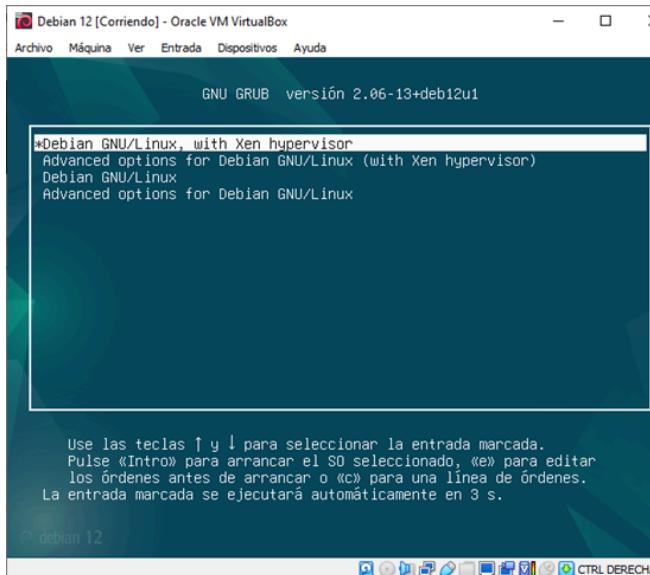


Fig. 21. Al reiniciar, comprobamos que la primera opción de arranque está con Debian GNU/Linux, con Xen Hypervisor.

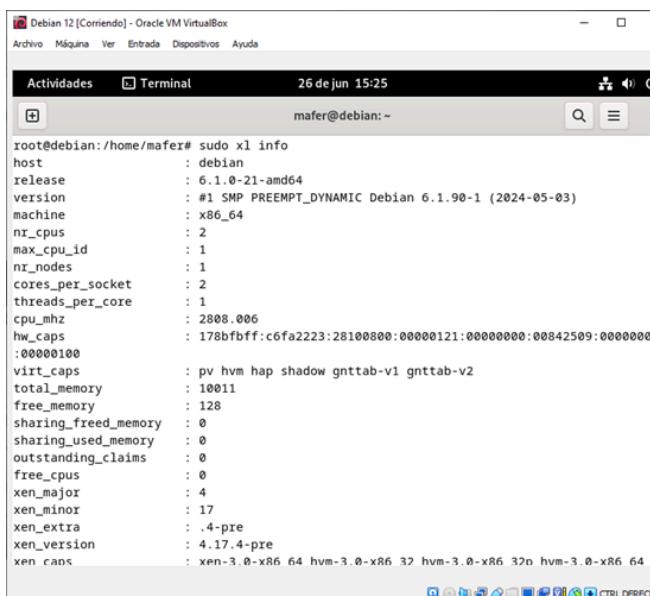


Fig. 22. Utilizamos el comando `xl info`, el cual debería mostrar la información sobre el Hypervisor Xen instalado, el host, etc.

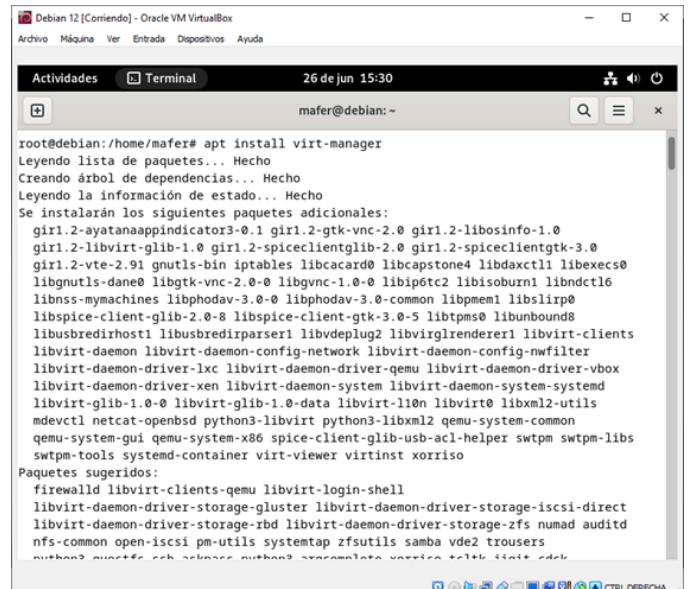


Fig. 23. Instalamos Virt Manager, que nos va a proporcionar su interfaz gráfica para trabajar con Xen Hypervisor.

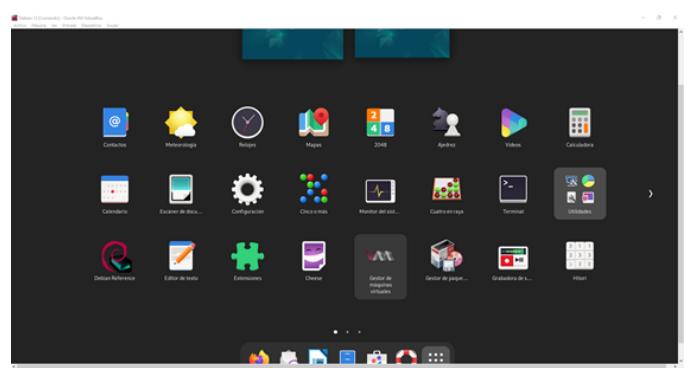


Fig. 24. Al reiniciar nuestra máquina virtual, podemos observar que tenemos nuestro gestor de máquinas virtuales Virt Manager.

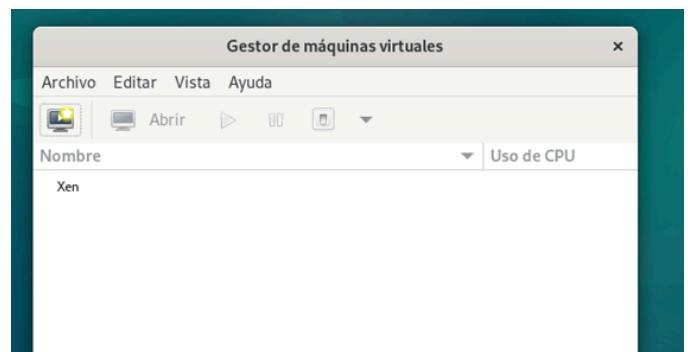


Fig. 25. Al estar activo, seguimos con el proceso de instalación.

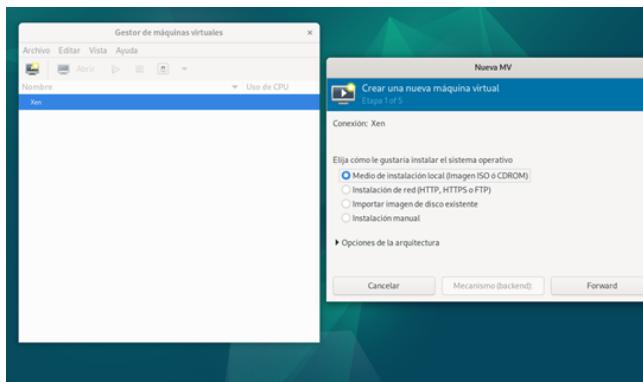


Fig. 26. Elegimos el método para instalar el sistema operativo, en este caso, Imagen ISO

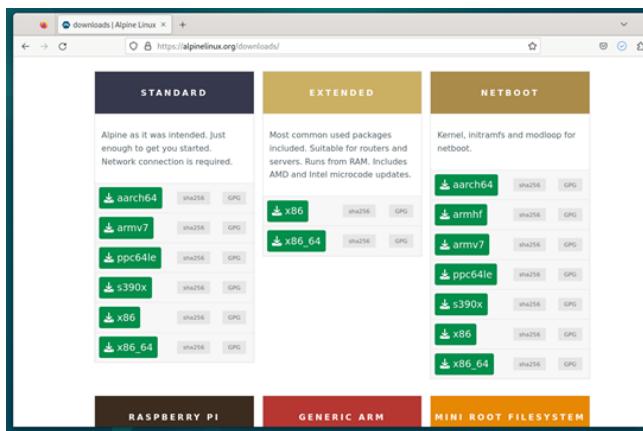


Fig. 27. Se ingresa al navegador predefinido, y se instala el sistema operativo de preferencia, en este caso, Alpine Linux, porque se requiere de un sistema operativo ligero.

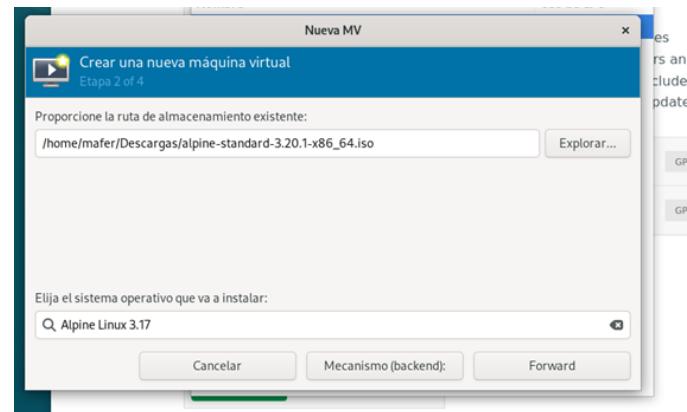


Fig. 28. Ya tenemos instalado nuestra imagen de Alpine Linux, vamos a añadirla como nuestra ruta de almacenamiento, definimos el sistema operativo, y damos a siguiente.

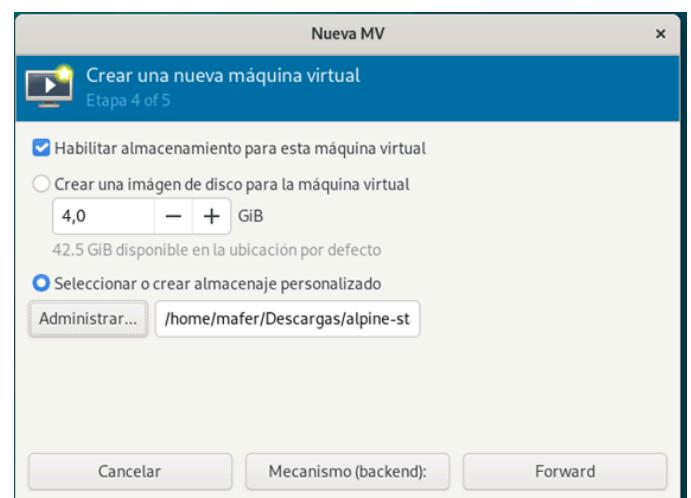
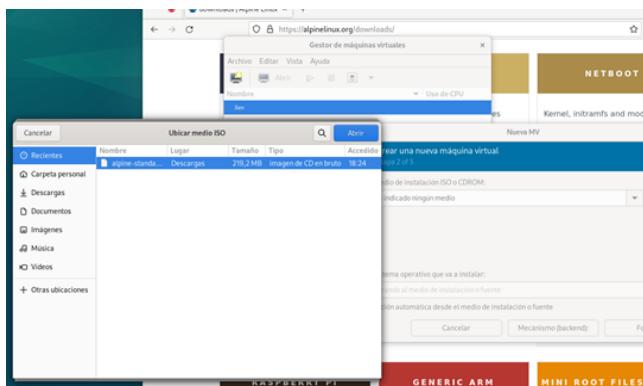
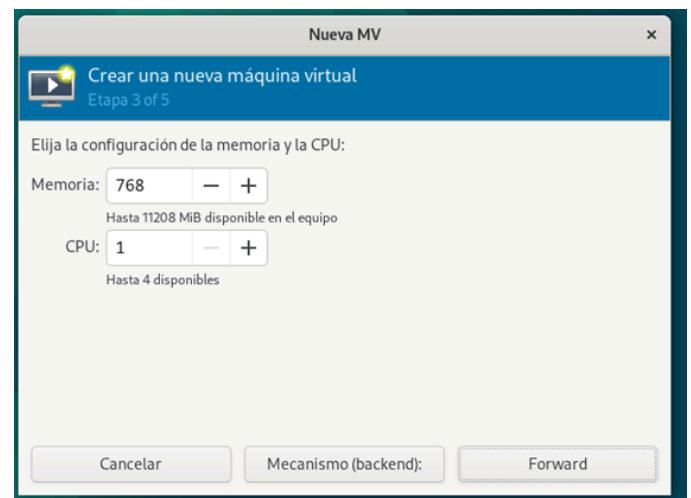


Fig. 29. Definimos los recursos necesarios para nuestra máquina virtual, y listo.

IV. EJECUCIÓN DE GUI.

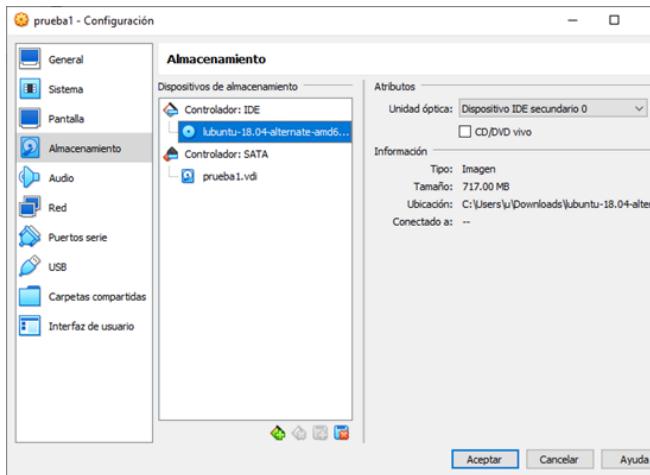


Fig. 30. Creamos nuestra máquina virtual, definimos los recursos, y para la instalación de la interfaz gráfica, definimos lubuntu, en este caso, como unidad de disco.

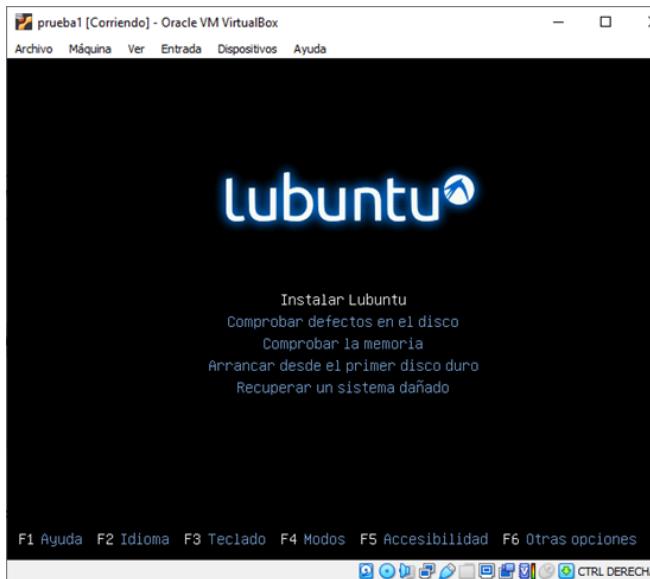


Fig. 31. Nuestra máquina virtual se está ejecutando con la interfaz gráfica de Lubuntu. Damos enter a Instalar Lubuntu.

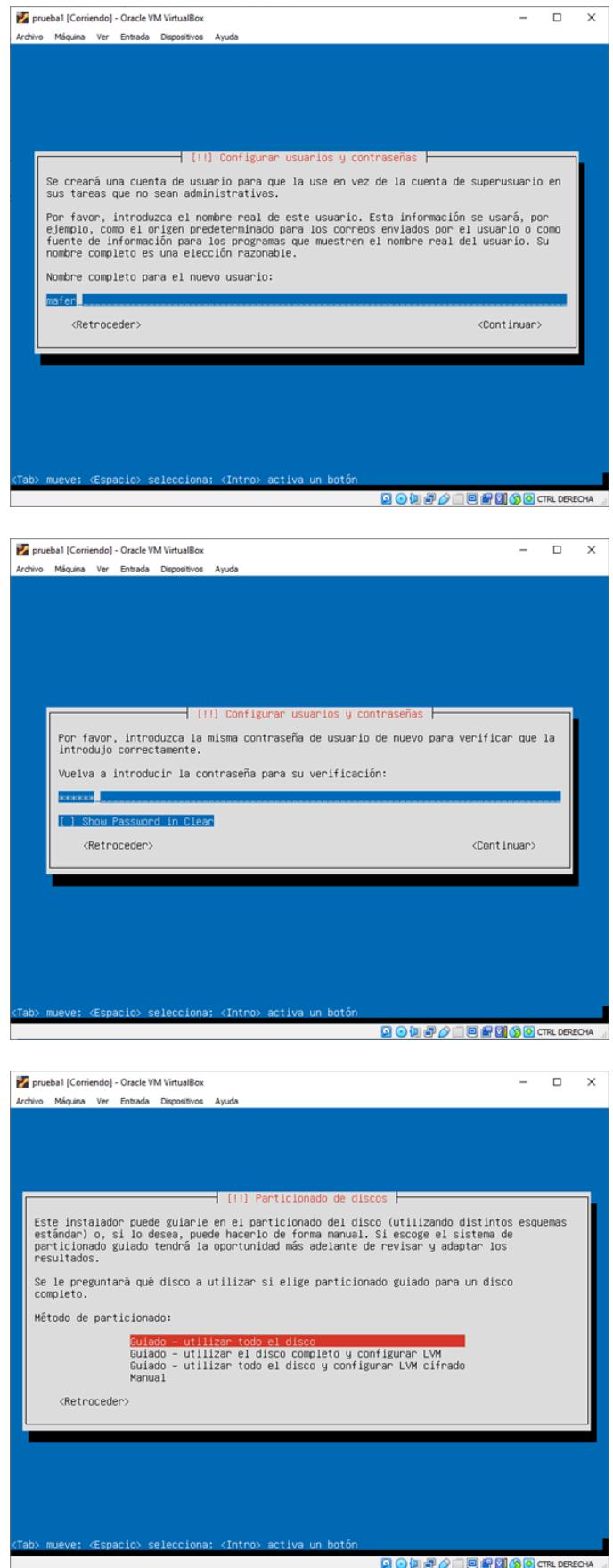


Fig. 32. Se hacen los pasos de configuración de nuestra máquina virtual, como contraseñas y usuarios, particionado de discos, instalación de paquetes y programas, etc.

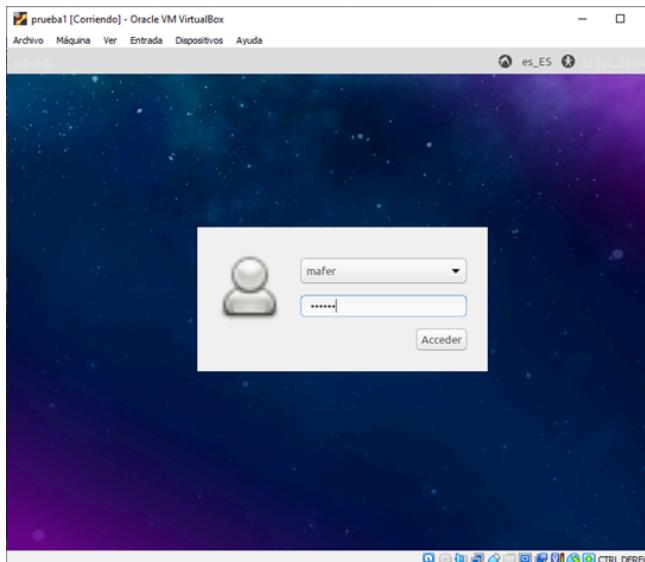


Fig. 33. Al finalizar podemos iniciar con nuestro usuario y contraseña creados.

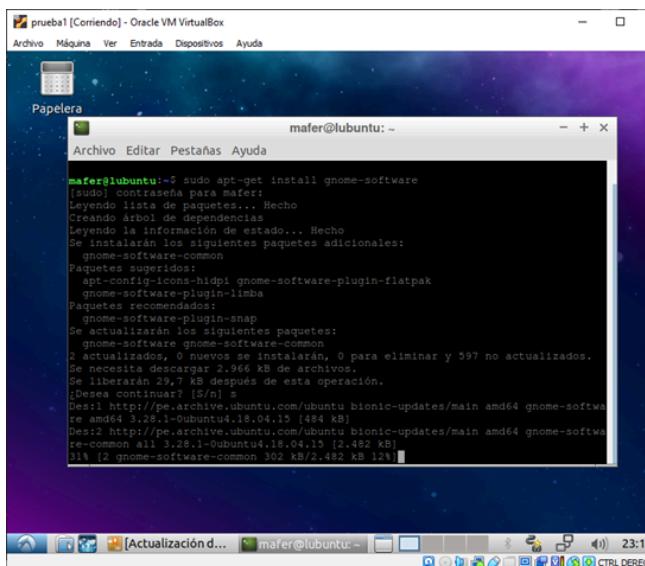


Fig. 34. Instalación de GNOME, para que funcione como gestor de escritorio.

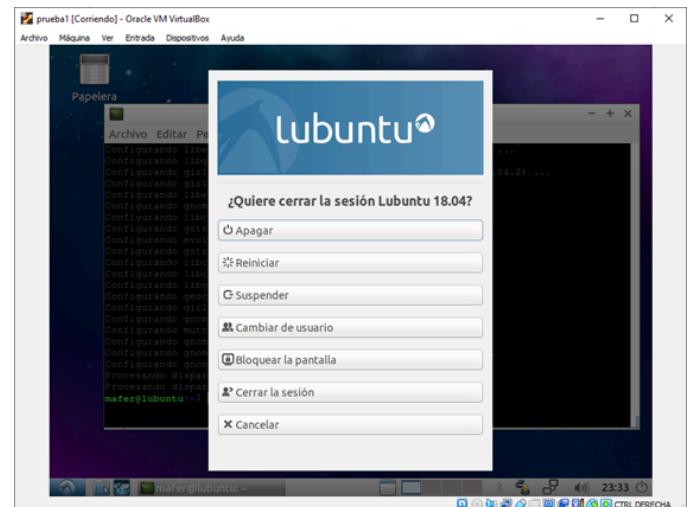


Fig. 35. Se cierra sesión, para ingresar, con el entorno de escritorio de gnome.

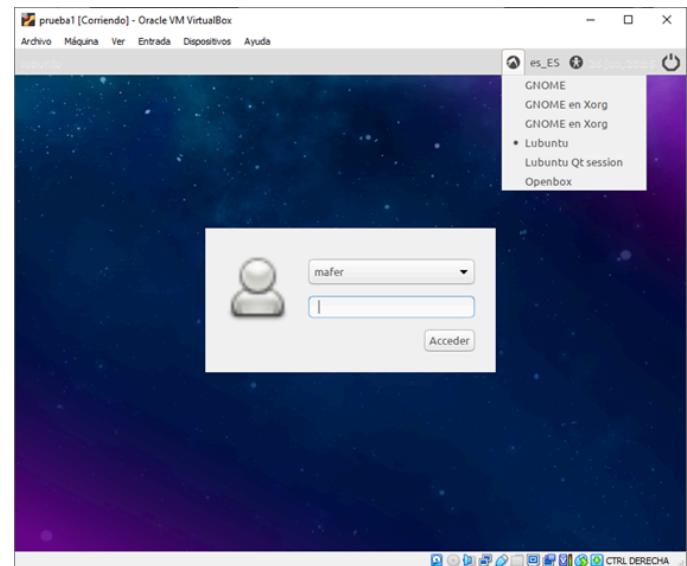


Fig. 36. Entre las opciones de inicio de sesión, podemos observar que tenemos las opciones GNOME, Lubuntu, Openbox. En este caso, utilizamos GNOME, anteriormente instalado.

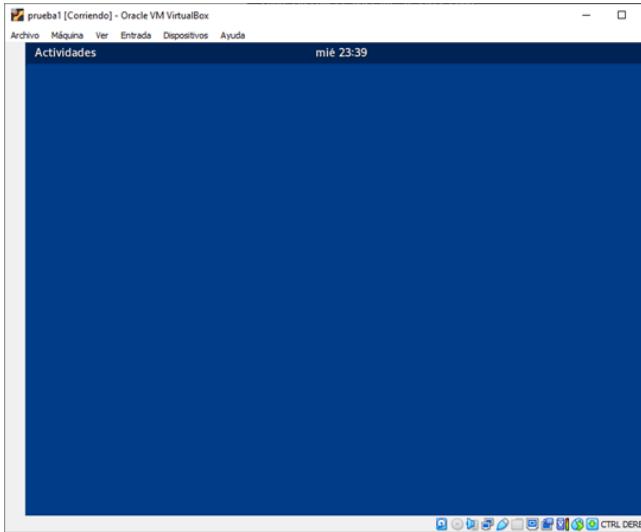


Fig. 37. Al momento de iniciar, podemos ver que tenemos el entorno de escritorio de GNOME. El cual, se puede mejorar con la instalación de un menú de gnome, etc. Para esto, ingresamos a nuestro navegador Mozilla Firefox.



Fig. 38. Ingresamos al Software de Gnome.

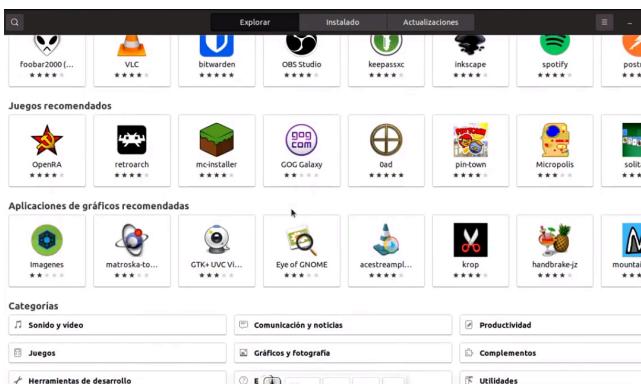


Fig. 39. Vemos las aplicaciones disponibles en el software de gnome, de la misma manera, las diversas opciones de instalación que existen.

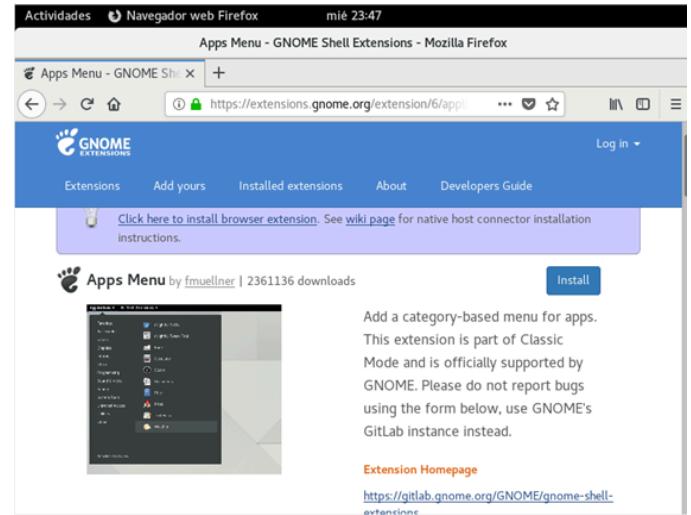


Fig. 40. De la misma manera, tenemos instalado el navegador web de forma gráfica Mozilla Firefox. E igualmente se pueden hacer diferentes instalaciones, por ejemplo, instalamos Apps Menu para adornar y darle vida a nuestro gestor de escritorio Gnome.

V. CONEXIÓN ENTRE DOS MÁQUINAS VIRTUALES

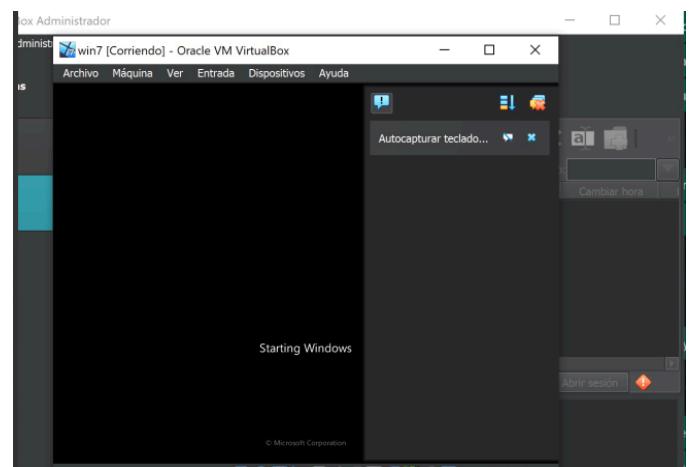


Fig. 41. Iniciando nuestra máquina virtual.

CARACTERÍSTICAS DE NUESTRA RED

- En la terminal de nuestra máquina virtual podemos apreciar todas las estadísticas de nuestra red, en especial nuestra IP, la cual nos servirá para realizar la conexión mediante la red.

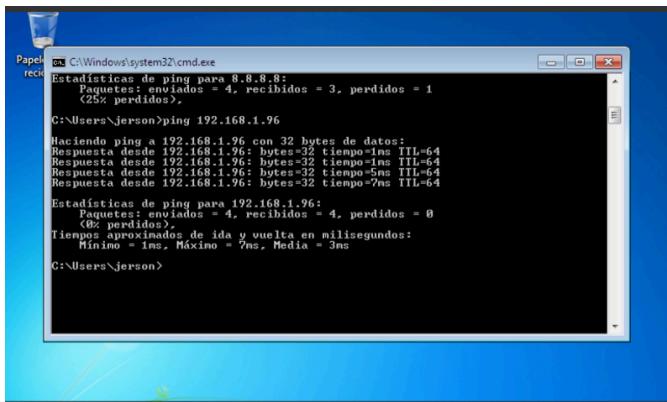


Fig. 42. Buscando el IP de la otra máquina virtual.

- Y en nuestra otra máquina virtual donde ejecutamos UBUNTU, realizamos la misma búsqueda, encontrando así la otra IP.

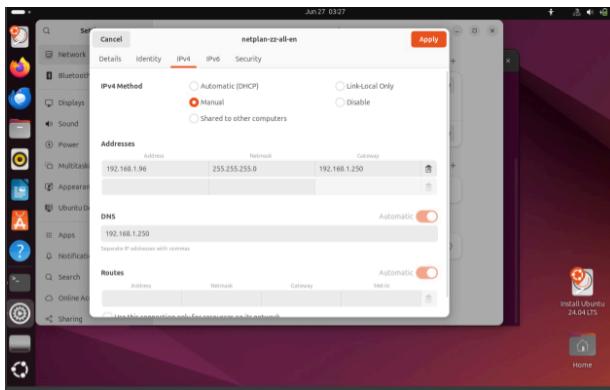


Fig. 43. Configurando las direcciones manualmente.

PROCEDIMIENTO PARA LA CONEXIÓN DE AMBAS MÁQUINAS VIRTUALES

- Para ello nos dirigimos a las propiedades de conexión de área local, y seleccionamos el Protocolo de Internet versión 4(TPC/IPv4)

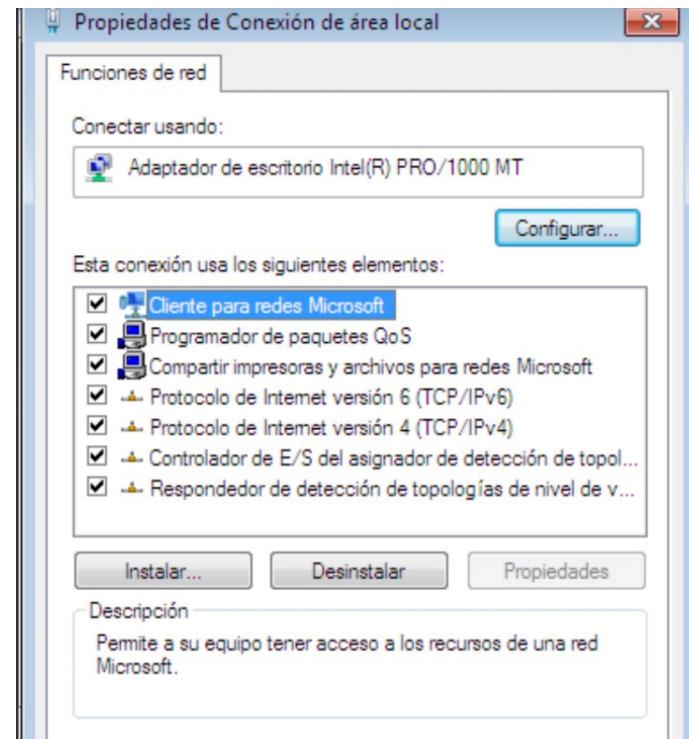


Fig. 44. Seleccionando el Protocolo de Internet versión 4(TPC/IPv4).

- Una vez allí, seleccionamos la opción de usar otras direcciones de servidor DNS, y colocamos la IP de nuestra máquina virtual donde se ejecutó UBUNTU. Para luego dar clic en “Aceptar”.

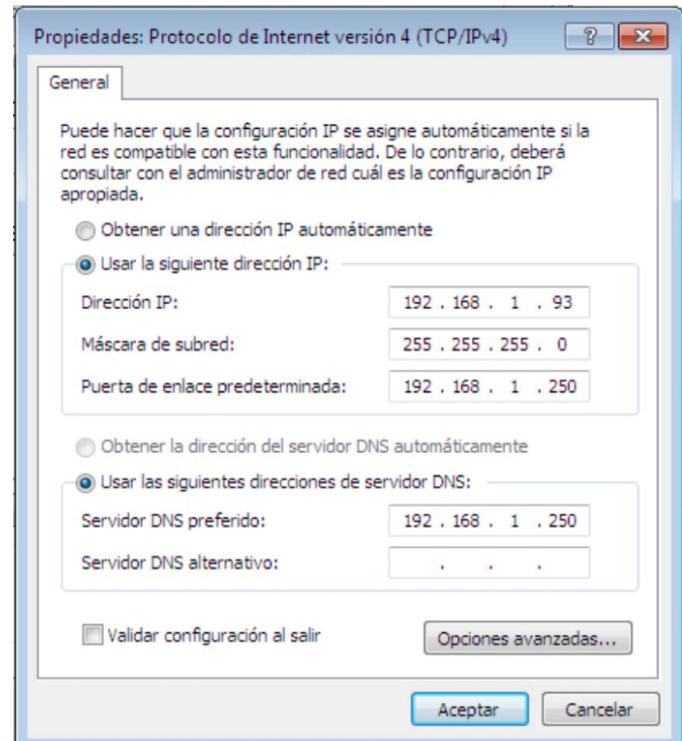


Fig. 45. Culminando configuración de conexión entre máquinas virtuales.

COMPROBAR DE LA CONEXIÓN

- Luego de realizar todos los pasos anteriores nos dirigimos a las configuraciones de nuestro sistema Windows 7, para comprobar que hemos realizado la conexión de red correctamente.

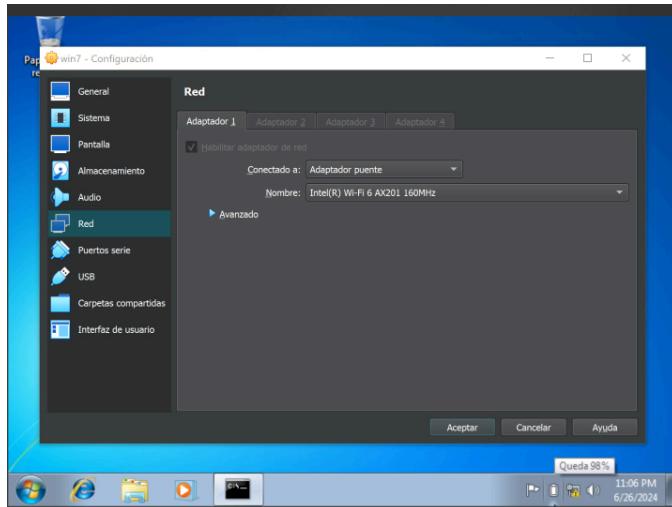


Fig. 46. Comprobando que la conexión sea exitosa.

- Y lo mismo hacemos con el sistema de UBUNTU

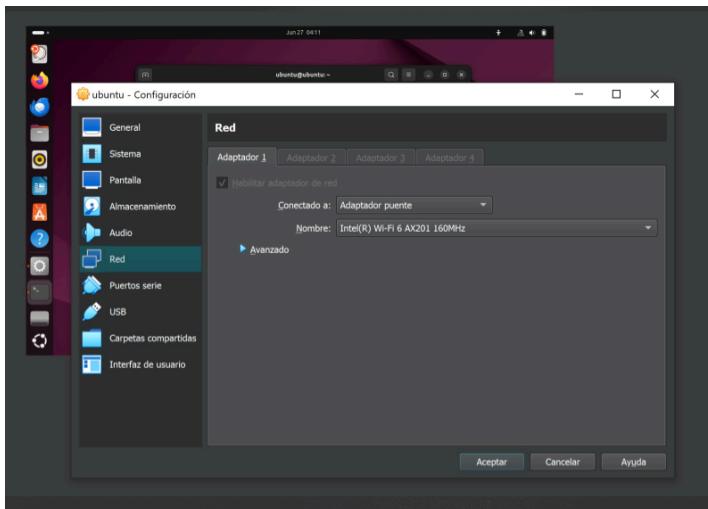


Fig. 47. Comprobando que la conexión sea exitosa.

CONCLUSIONES

EFICIENCIA Y DESEMPEÑO

Xen Hypervisor y QEMU, cuando se utilizan en conjunto, ofrecen una solución de virtualización eficiente en términos de desempeño. Xen, al ser un hypervisor de tipo 1 (bare-metal), proporciona una capa de virtualización que interactúa directamente con el hardware, lo que permite un uso más eficiente de los recursos. QEMU, como

emulador y hypervisor de tipo 2 (hosted), complementa esta capacidad al permitir la emulación de hardware, ampliando la compatibilidad y flexibilidad en entornos heterogéneos.

COMPATIBILIDAD Y FLEXIBILIDAD

La combinación de Xen Hypervisor y QEMU permite a los usuarios ejecutar múltiples sistemas operativos en una única máquina física, aprovechando la compatibilidad de QEMU con diversas arquitecturas de hardware y sistemas operativos. Esta flexibilidad es crucial en entornos de desarrollo y prueba, donde es necesario ejecutar diferentes configuraciones y versiones de software de manera simultánea y aislada.

SEGURIDAD Y AISLAMIENTO

Xen Hypervisor proporciona un alto grado de seguridad y aislamiento entre las máquinas virtuales, lo cual es esencial para entornos que requieren estrictas políticas de seguridad. La arquitectura de Xen permite la separación de dominios privilegiados y no privilegiados, minimizando el riesgo de ataques y vulnerabilidades. QEMU, al ejecutarse sobre Xen, se beneficia de esta arquitectura, proporcionando un entorno virtual seguro y confiable.

REFERENCES

- [1] D. Martin, "How to use QEMU," YouTube, 24-Nov-2021. [Online]. Available: https://www.youtube.com/watch?v=XX_F_pnT3eM. [Accessed: 27-Jun-2024].
- [2] TechExpert, "Network setup with QEMU," YouTube, 15-Jan-2023. [Online]. Available: <https://www.youtube.com/watch?v=jXGxhVbnJKk>. [Accessed: 27-Jun-2024].
- [1] Abbasi, A.A., Jin, H. (2018). v-Mapper: An Application-Aware Resource Consolidation Scheme for Cloud Data Centers. *Future Internet*, 10, 90.
- [2] Schulz y B. Annighöfer, "Evaluation of Adaptive Partitioning and Real-Time Capability for Virtualization With Xen Hypervisor," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 1, pp. 206-217, 2022.
- [3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, y A. Warfield, "Xen and the Art of Virtualization," en *Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03)*, 2003.