

# Tarea 1

Jaime Mendez, Carolay Rodriguez,Laura Baez

Marzo 2025

## 1 Introducción

En este informe se describe el proceso que se realizó paso a paso para la instalación y configuración de distintas herramientas en las máquinas virtuales. Se abordan tres tareas principales: la instalación de QEMU para la virtualización de sistemas operativos, el uso de Nmap para el escaneo de redes y la automatización de tareas con Cron y Crontab. Lo primero que realizamos, se hace la explicación de cómo instalar QEMU en diferentes sistemas operativos y crear máquinas virtuales con Windows, CentOS y Linux. Luego, se desarrolló el uso de Nmap para identificar dispositivos en la red y analizar puertos abiertos. Por último, se configuran tareas automáticas con Crontab, como la ejecución de scripts y la realización de copias de seguridad.

## 2 Desarrollo

Instalación de QEMU: Para poder descargar y instalar QEMU en Ubuntu, primero se debe abrir una terminal y se ejecuta en sudo apt update para poder realizar la actualización de los repositorios. Despues, se instala QEMU y las herramientas con el comando sudo apt install qemu-kvm qemu-system qemu-utils libvirt-daemon-system libvirt-clients virt-manager. Esto hace que se descargue y se instale los paquetes necesarios para poder ejecutar y administrar las máquinas virtuales.

1. Instalación de Windows en QEMU:

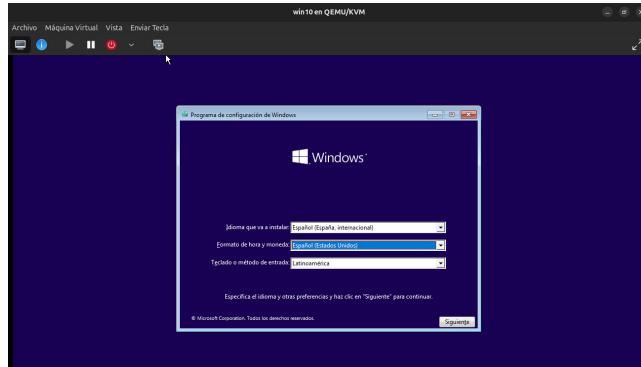


Figure 1: Proceso de instalación de Windows en QEMU

En la imagen que se proporcionó se realiza la instalación de Windows en una máquina virtual creada con QEMU. La imagen muestra la pantalla de configuración inicial del instalador de Windows, donde se seleccionan las opciones de idioma, formato de hora y moneda, así como el método de entrada del teclado. Estas configuraciones son primordiales para garantizar la correcta localización del sistema operativo antes de proceder con la instalación.

### 2.Gestor de maquinas virtuales:

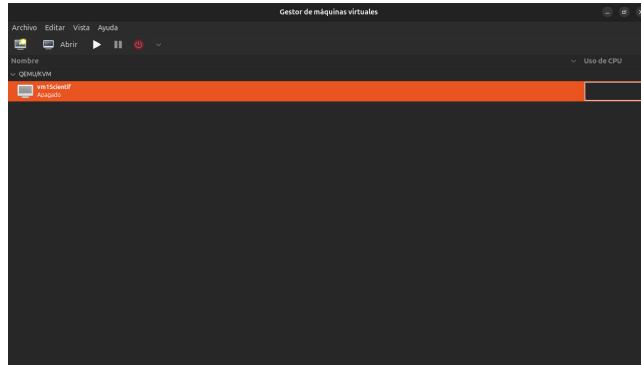


Figure 2: Gestor de maquinas virtuales

La imagen muestra el Gestor de máquinas virtuales en QEMU, donde se administra el estado de las máquinas virtuales disponibles. En este caso, la máquina denominada 'vm1Scientif' se encuentra apagada. Desde esta interfaz, es posible iniciar, pausar, detener o modificar las configuraciones de las máquinas virtuales, permitiendo gestionar los entornos virtualizados.

### 3.Gestor de máquinas virtuales en QEMU:

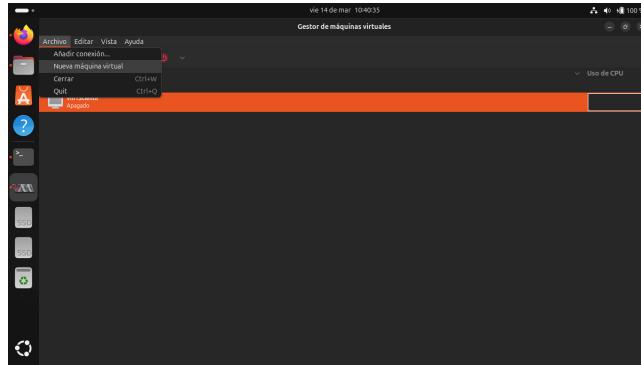


Figure 3: Gestor de máquinas virtuales en QEMU

La imagen muestra la interfaz del Gestor de máquinas virtuales en QEMU, en un sistema operativo basado en Linux. En la captura, se ha desplegado el menú 'Archivo', donde se observan opciones clave para la administración de máquinas virtuales, como la creación de nuevas instancias y la conexión con hipervisores remotos. Esta herramienta permite gestionar entornos virtualizados de manera eficiente, facilitando la ejecución y control de sistemas operativos dentro de un host principal.

#### 4.Nueva maquina virtual (Windows) en QEMU:

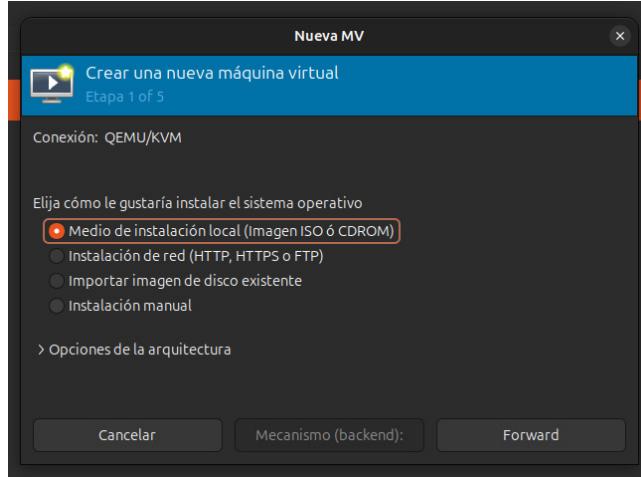


Figure 4: Primera parte de la nueva maquina virtual

La imagen muestra la ventana de creación de una nueva máquina virtual en QEMU, está en la etapa 1 de 5 del proceso de configuración. En esta fase, el usuario debe seleccionar el método de instalación del sistema operativo. La opción marcada indica que se usará un medio de instalación local, como una imagen ISO o un CD-ROM. También se presentan otras opciones, como

instalación desde red (HTTP, HTTPS o FTP), importación de una imagen de disco existente o una instalación manual. Este paso es fundamental para definir la fuente del sistema operativo que se ejecutará en la máquina virtual.

5.Nueva maquina virtual (Windows) en QEMU:

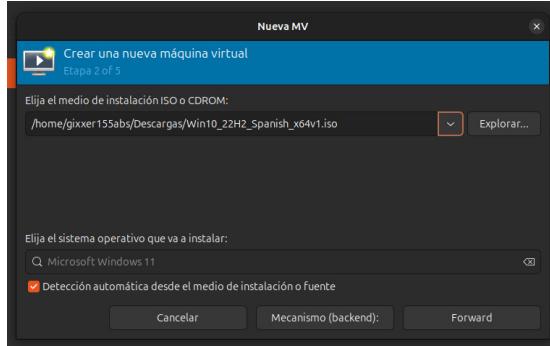


Figure 5: Segunda parte

La imagen muestra la segunda etapa del asistente para la creación de una nueva máquina virtual en QEMU. En esta fase, el usuario selecciona el medio de instalación ISO o CD-ROM para el sistema operativo Windows. Se ha elegido un archivo ISO localizado en el directorio de descargas, específicamente 'Win10-22H2-Spanish-x64v1.iso', lo que indica que se instalará Windows 10. El asistente ha detectado incorrectamente el sistema operativo como 'Microsoft Windows 11'. La opción de detección automática está activada, lo que eso puede decir que en la configuración recomendada de hardware. El usuario puede proceder con la instalación haciendo clic en 'Forward'

6.Nueva maquina virtual (Windows) en QEMU:

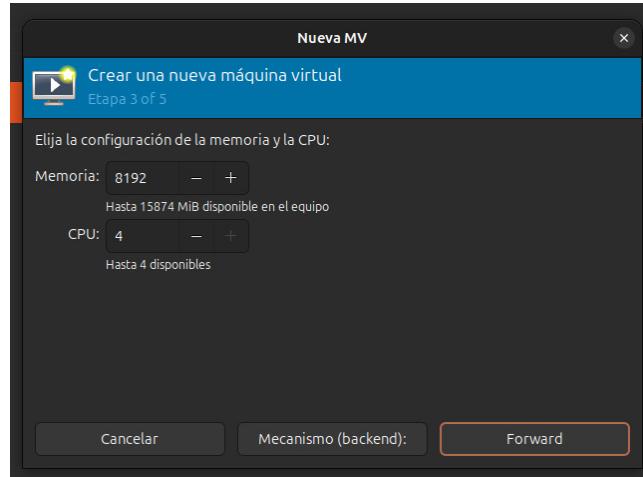


Figure 6: Tercera parte

La imagen muestra la tercera etapa del asistente de creación de una máquina virtual en QEMU. En esta fase, se configuran los recursos de memoria RAM y CPU asignados a la máquina virtual. Memoria: se configuran los recursos de memoria RAM y CPU asignados a la máquina virtual. Se ha seleccionado 8192 MB (8 GB) de memoria RAM, de un total disponible de 15874 MB en el sistema anfitrión. Cpu: Se han seleccionado 4 núcleos de CPU, con la opción de ajustar este número. Se indica que hay hasta 4 núcleos disponibles. La configuración asignada es adecuada para la instalación y ejecución de Windows 10, asegurando un rendimiento óptimo dentro de los límites del hardware del anfitrión. El usuario puede continuar con la configuración presionando 'Forward'.”

7.Nueva maquina virtual (Windows) en QEMU:

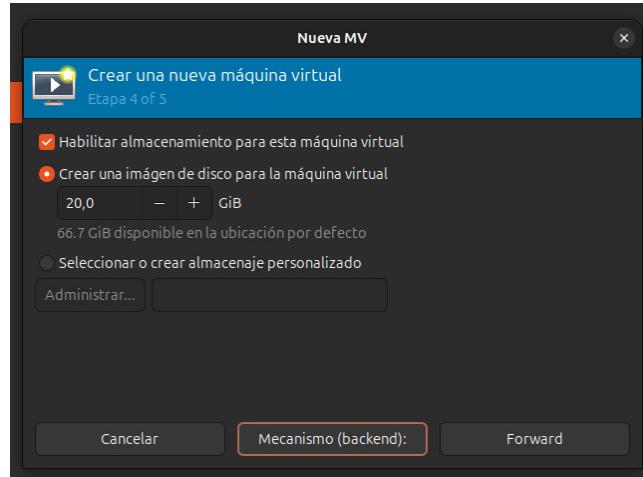


Figure 7: Cuarta parte

La imagen muestra la cuarta etapa del asistente de creación de una máquina virtual en QEMU. En esta fase, se configura el almacenamiento de la máquina virtual. Se ha habilitado el almacenamiento y seleccionado la opción de crear una imagen de disco virtual con una capacidad de 20 GiB, de un total disponible de 66.7 GiB en la ubicación. También se da la opción de seleccionar o crear un almacén de almacenamiento personalizado. El usuario puede proceder con la configuración presionando 'Forward'.”

8.Nueva maquina virtual (Windows) en QEMU:

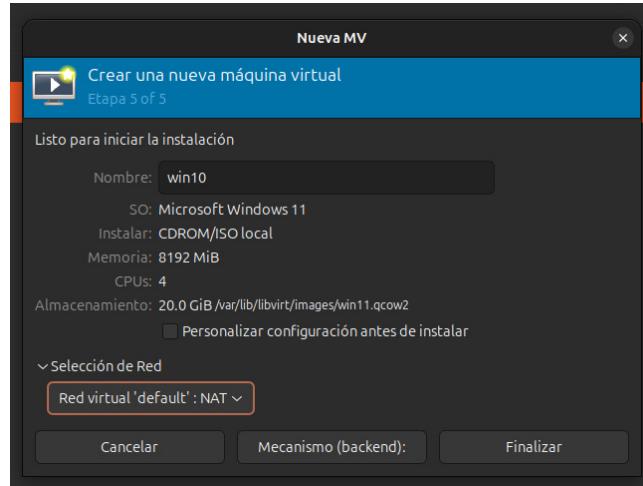


Figure 8: Quinta parte

La imagen muestra la quinta etapa del asistente de creación de una máquina virtual en QEMU. En esta ultima etapa, se presenta un resumen de la configuración antes de hacer la instalación. La máquina virtual ha sido nombrada 'centos-stream9' y utilizará CentOS Stream 9 como sistema operativo, instalado desde un archivo ISO local. Se ha asignado 1536 MiB de memoria RAM, 2 CPUs y un almacenamiento de 20 GiB en el formato qcow2. La red está configurada con NAT bajo la red virtual 'default'. El usuario puede utilizar la opción de finalizar la configuración o personalizarla antes de instalar.

9.Inicio del firmware:

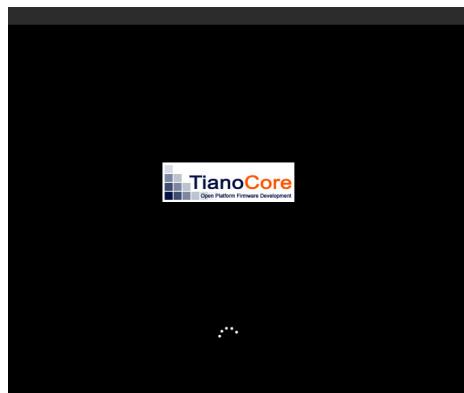


Figure 9: Inicio del firmware

En la imagen se muestra la pantalla de inicio del firmware TianoCore durante el inicio de la máquina virtual en QEMU. TianoCore es una implementación de código abierto de UEFI (Unified Extensible Firmware Interface), se utiliza

como firmware en entornos de virtualización. En la pantalla se observa el logo de TianoCore en el centro y esta cargando, señalando que el proceso de inicio está en curso.

#### 10. Selección del sistema operativo:

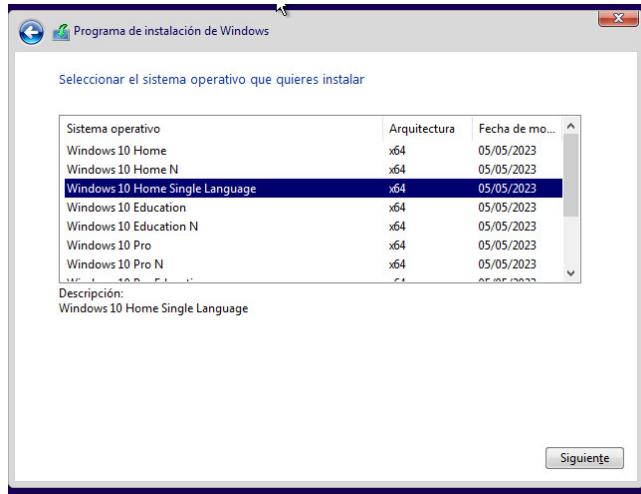


Figure 10: Selección del sistema operativo

En la imagen se muestra la pantalla de selección del sistema operativo en el programa de instalación de Windows. En este proceso, se puede elegir entre diferentes ediciones de Windows 10, todas con arquitectura de 64 bits (x64) y fecha de modificación del 05/05/2023. La opción resaltada es 'Windows 10 Home Single Language'. En la parte inferior derecha, se encuentra el botón 'Siguiente' para continuar con la instalación después de realizar la selección.

#### 11. Proceso de instalación de Windows:

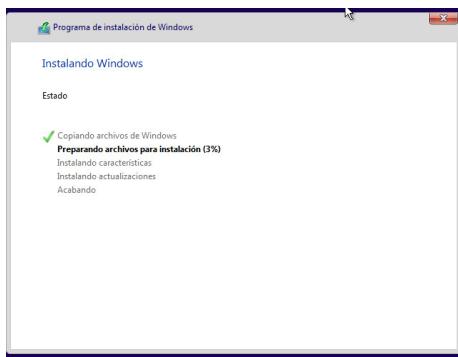


Figure 11: Proceso de instalación de Windows

En la imagen se muestra el proceso de instalación de Windows en curso.

En este proceso, el programa de instalación está copiando y preparando los archivos necesarios para la instalación del sistema operativo. Se observa que la copia de archivos ha finalizado y que la fase actual es 'Preparando archivos para instalación', con un progreso del 3 porciento. Despues, el instalador procederá con la instalación de características, actualizaciones y la finalización del proceso.

12. Proceso de instalación de Windows:

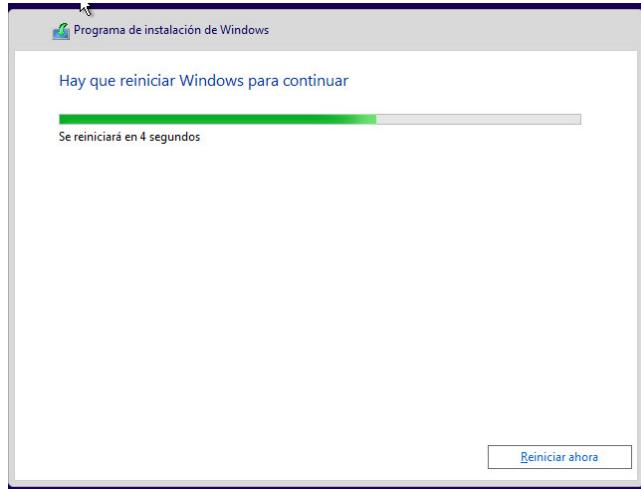


Figure 12: Proceso de instalación de Windows

En la imagen se muestra una etapa avanzada del proceso de instalación de Windows. En este proceso el sistema indica que es necesario reiniciar para continuar con la instalación. Se encuentra una barra de progreso en color verde y un contador regresivo que indica que el reinicio ocurrirá automáticamente en unos segundos. También se ofrece la opción de reiniciar manualmente de inmediato mediante el botón 'Reiniciar ahora', se puede utilizar cualquiera de las dos.

13. Inicio de TianoCore:

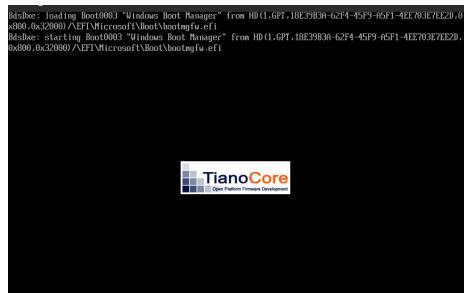


Figure 13: Arranque de TianoCore

En la imagen se muestra la pantalla de inicio de TianoCore, un firmware de código abierto basado en UEFI, que se utiliza en entornos de virtualización. En este proceso, se está cargando y ejecutando el 'Windows Boot Manager' desde una parte del disco identificada por su estructura GPT. Se observa la ruta de inicio dentro del directorio EFI de Microsoft, lo que se indica que el sistema operativo Windows está siendo iniciado en un entorno virtualizado.

14. Configuración de Window:

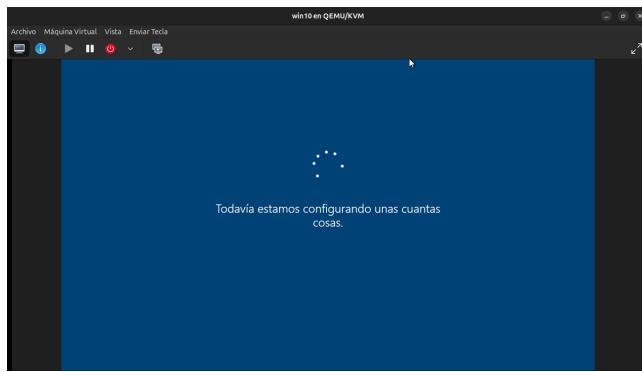


Figure 14: Configuración de Windows

"En la imagen se muestra una pantalla de configuración de Windows en una máquina virtual ejecutándose en QEMU. El mensaje que aparece indica que el sistema sigue configurando algunos parámetros antes de completar la instalación. Lo cual indica de que el proceso de instalación de Windows está en su etapa final y pronto estará listo para ser utilizado."

15. Configuración final:

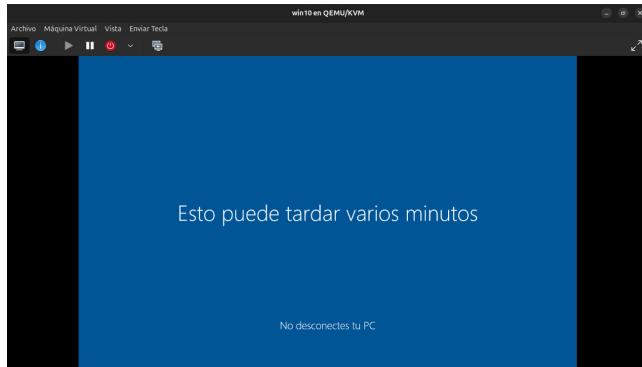


Figure 15: Configuración final

En la imagen se muestra una pantalla de configuración final de Windows en una máquina virtual ejecutándose en QEMU. El mensaje indica que el proceso

aún está en curso y puede tardar varios minutos en completarse.

16.Escritorio de Windows 10:

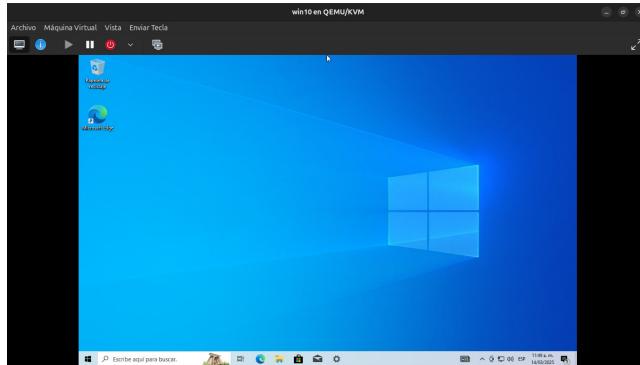


Figure 16: Escritorio de Windows 10

En la imagen se muestra el escritorio de Windows 10 ejecutándose dentro de una máquina virtual en QEMU. La instalación que se ha realizado finalizo con éxito y el sistema está listo para su uso.

17.Nueva maquina virtual (Linux) en QEMU: La imagen muestra la ventana

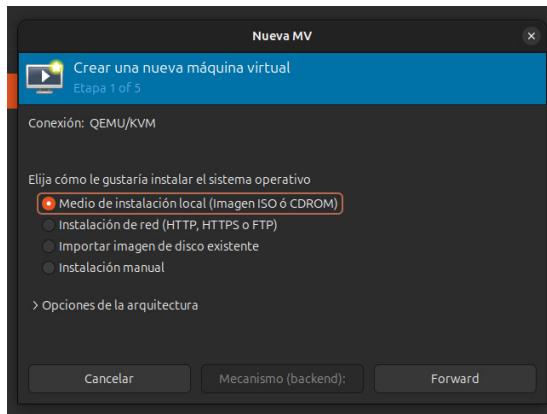


Figure 17: Primera parte

de creación de una nueva máquina virtual en QEMU, está en la etapa 1 de 5 del proceso de configuración.Se realiza el mismo proceso que se realizó con Windows.

18.Nueva maquina virtual (Linux) en QEMU:

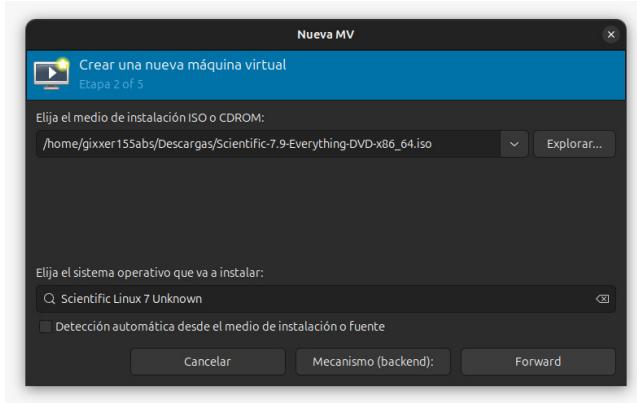


Figure 18: Segunda parte

La imagen muestra la segunda parte del asistente de creación de una máquina virtual en QEMU. Se selecciona el medio de instalación, el cual corresponde a un archivo ISO localizado en la ruta /home/gixxer155abs/Descargas/Scientific-7.9-Everything-DVD-x86-64.iso. El asistente detecta el sistema operativo como 'Scientific Linux 7 Unknown', permitiendo la detección automática de la configuración. Se puede seleccionar manualmente el sistema operativo o también la configuración se puede hacer automática.

19.Nueva maquina virtual (Linux) en QEMU: La imagen muestra la tercera

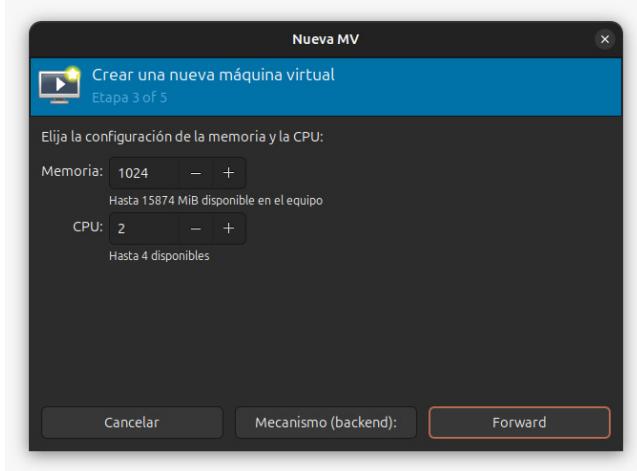


Figure 19: Tercera parte

parte del asistente de creación de una máquina virtual en QEMU. Se asignaron los siguientes datos: Memoria RAM: 1024 MiB (1 GB), con un máximo disponible de 15874 MiB en el equipo. CPU: 2 núcleos, con un máximo de 4

disponibles. Se puede proceder con la configuración haciendo clic en "Forward" o ajustar los valores antes de continuar con el proceso.

20. Nueva maquina virtual (Linux) en QEMU:

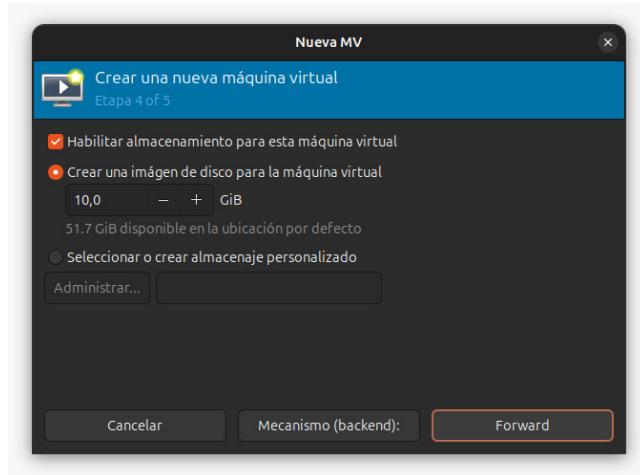


Figure 20: Cuarta parte

La imagen muestra la cuarta parte del asistente de creación de una máquina virtual en QEMU. Se habilita el almacenamiento para la máquina virtual, el cual está activado. Se puede crear una imagen de disco: Seleccionado, con un tamaño de 10 GiB El espacio disponible en la ubicación por defecto: 51.7 GiB La opción alternativa: Seleccionar o crear un almacenamiento personalizado (no seleccionado) Se puede proceder haciendo clic en "Forward" o ajustar el tamaño del disco antes de continuar.

21. Nueva maquina virtual (Linux) en QEMU:

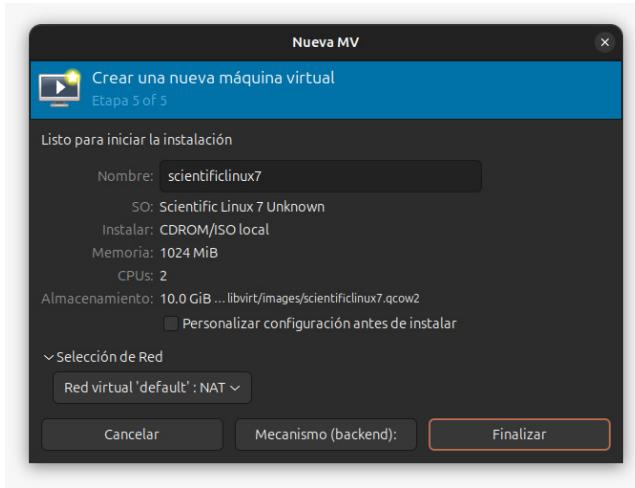


Figure 21: Quinta parte

La imagen muestra la quinta parte del asistente de creación de una máquina virtual en QEMU. Se definen los parámetros finales antes de iniciar la instalación del sistema operativo. Nombre: scientificlinux7 Sistema Operativo: Scientific Linux 7 Unknown Fuente de instalación: CDROM/ISO local Memoria RAM: 1024 MiB CPUs: 2 Almacenamiento: 10.0 GiB (qcow2 en libvirt) Red: Red virtual 'default': NAT Opción para personalizar la configuración antes de instalar: No seleccionada Para iniciar la instalación, el usuario debe hacer clic en "Finalizar".

## 22.Inicio del instalador:

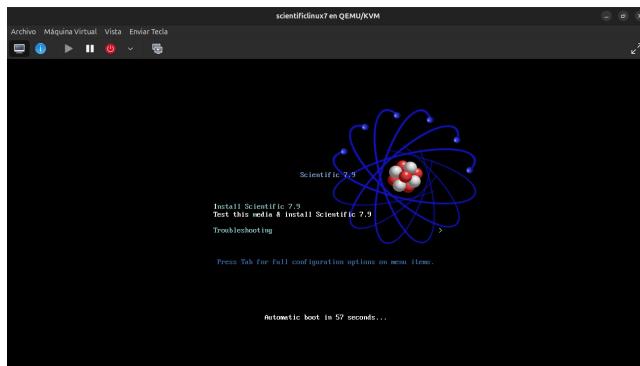


Figure 22: Inicio del instalador

En la imagen se muestra la pantalla de inicio del instalador de Scientific Linux 7.9 en una máquina virtual de QEMU, donde se puede elegir instalar el sistema, verificar la integridad del medio antes de la instalación o acceder a

opciones avanzadas de solución de problemas, el arranque automático ocurrirá en 57 segundos.

### 23. selección de idioma:

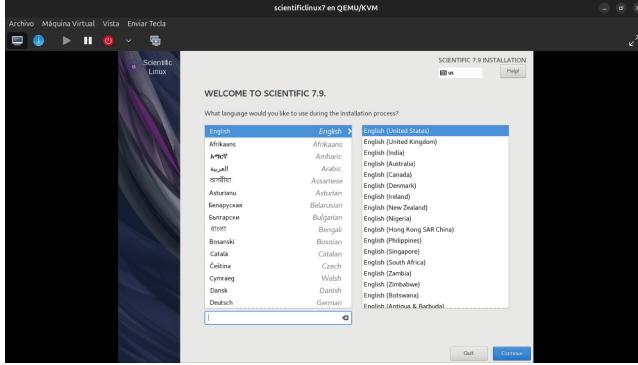


Figure 23: selección de idioma

La imagen muestra la pantalla de selección de idioma durante la instalación de Scientific Linux 7.9, un sistema operativo basado en Linux. En esta etapa, el usuario debe elegir el idioma que se usará durante el proceso de instalación. La interfaz presenta una lista de idiomas en la parte izquierda y, al seleccionar "English", se despliega una lista adicional a la derecha con variantes de inglés según la región.

### 24. Resumen de instalación:



Figure 24: Resumen de instalación

Se muestra en la pantalla el resumen de la instalación de Scientific Linux 7.6 en la máquina virtual con QEMU, donde se ha configurado la regionalización, selección de software y opciones del sistema. La instalación aún no ha comenzado, y el botón "Empezar instalación" está disponible.

### 25. Configuración de usuario:

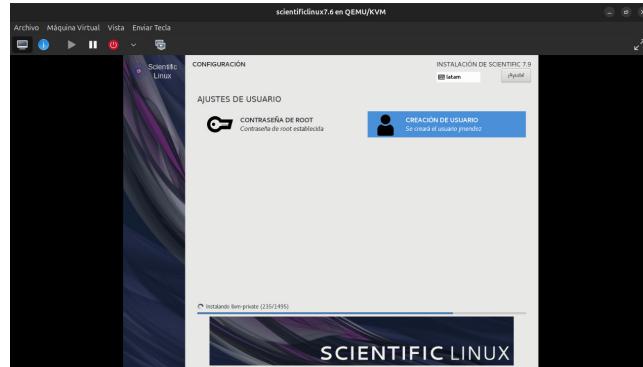


Figure 25: Configuración de usuario

La instalación está en proceso, y se están configurando los ajustes de usuario, se esta incluyendo la creación del usuario jmendez y la contraseña de root. En la parte inferior, se muestra la barra de progreso, indicando la instalación del paquete llvm-private (235/1495).

26.Pantalla de inicio de sesión:

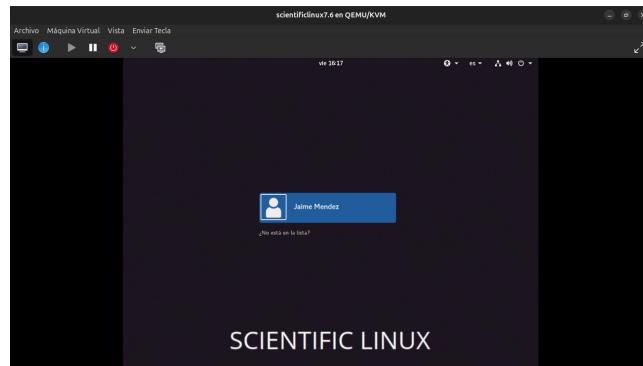


Figure 26: Pantalla de inicio

La instalación ha finalizado y el sistema muestra en la pantalla de inicio de sesión al usuario Jaime Mendez listo para acceder.

27.Pantalla de inicio:

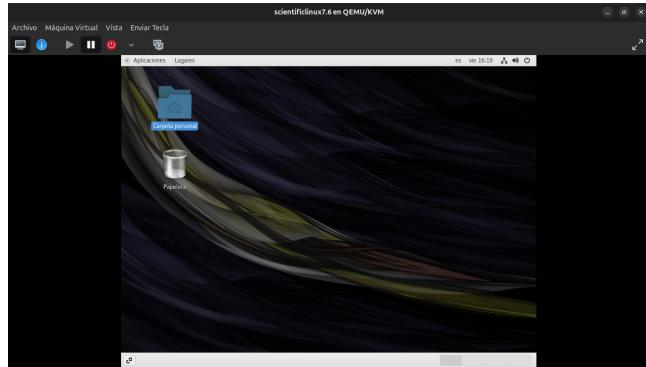


Figure 27: Pantalla de inicio

El sistema muestra la pantalla de inicio de sesión.  
28.Nueva máquina virtual (CentOS):

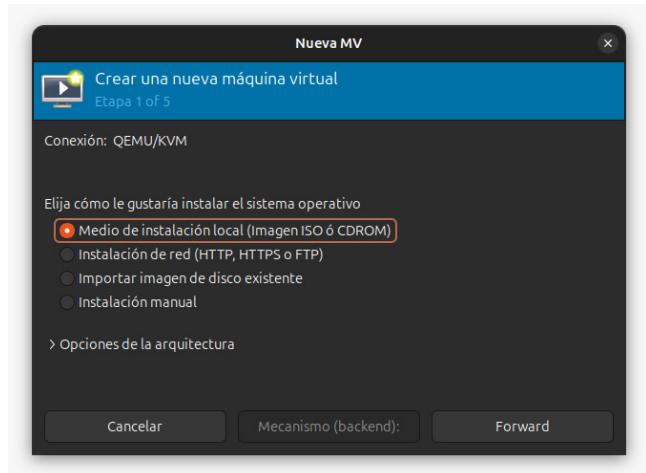


Figure 28: Primera parte

La imagen muestra la ventana de creación de una nueva máquina virtual en QEMU, está en la etapa 1 de 5 del proceso de configuración. Se realiza el mismo proceso que se realizó con Windows y en scientificlinux.

29.Nueva máquina virtual (CentOS):

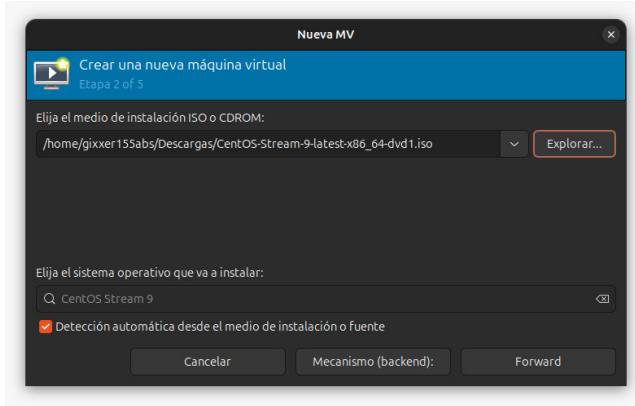


Figure 29: Segunda parte

En la imagen está en la segunda parte de la nueva máquina virtual. Se selecciona un medio de instalación (ISO o CDROM) y se elegió el sistema operativo a instalar, en este caso, CentOS Stream 9. También se da la opción de detección automática del sistema operativo desde el medio de instalación. Hay botones para cancelar, configurar el backend y avanzar.

### 30.Nueva máquina virtual (CentOS):

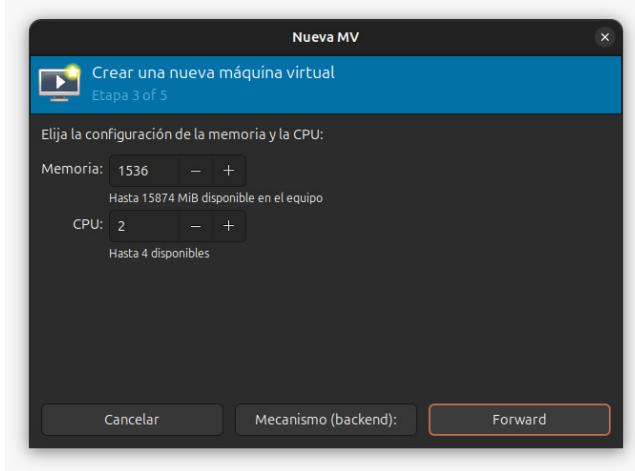


Figure 30: Tercera parte

En la imagen está en la tercera parte de la nueva máquina virtual. Se selecciona configurar la memoria y la CPU, con opciones para asignar hasta 1536 MiB de memoria (de un máximo de 15874 MiB disponible) y 2 CPUs (de un máximo de 4 disponibles).

### 31.Nueva máquina virtual (CentOS):

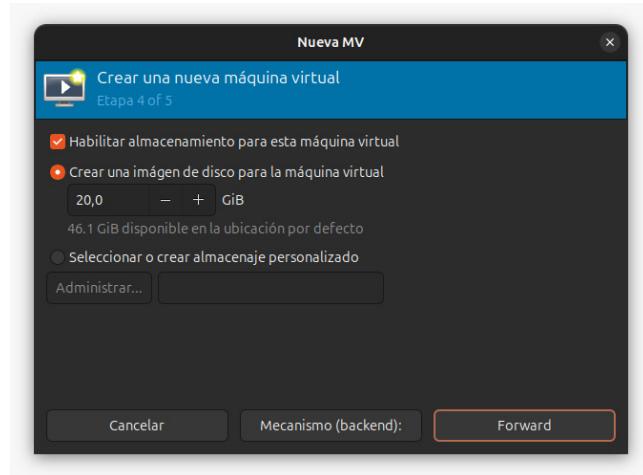


Figure 31: Cuarta parte

En la imagen se encuentra la cuarta parte de la máquina virtual. Se da la opción de habilitar almacenamiento, con la opción de crear una imagen de disco de 20,0 GiB (de 46,1 GiB disponibles). También se ofrece la opción de seleccionar o crear almacenamiento personalizado.

32.Nueva máquina virtual (CentOS):

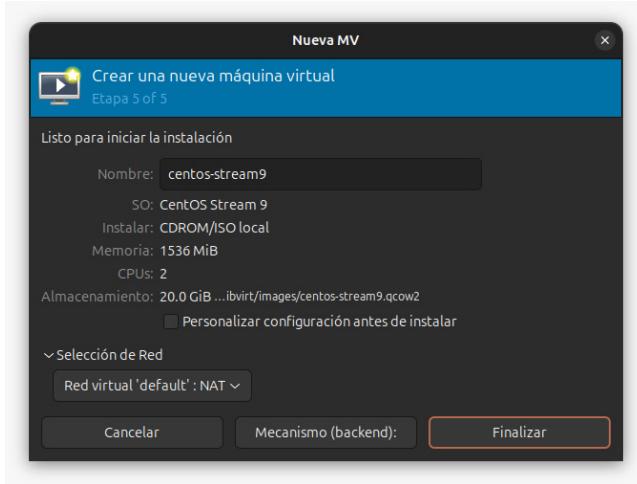


Figure 32: Quinta parte

En la imagen se encuentra la ultima parte de la máquina virtual. Se indica que la red virtual 'default' está configurada en modo NAT. Hay opciones para personalizar la configuración antes de instalar y con eso finalizara la creación de la máquina virtual.

### 33.Creación de maquina virtual (CentOS):

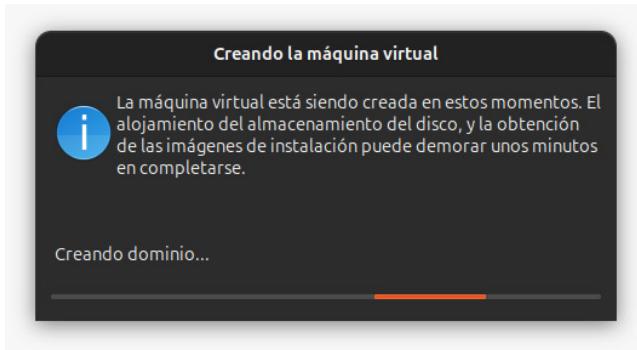


Figure 33: Creando maquina virtual

En la imagen se muestra el proceso de creación de la nueva máquina virtual que se esta creando. Se indica que la máquina virtual está siendo creada, lo que incluye la configuración del almacenamiento en disco y la obtención de las imágenes de instalación necesarias. Este proceso puede tardar unos minutos. Además, se menciona que se está creando el dominio, lo que implica la configuración y preparación del entorno virtual para la máquina.

### 34.Inicio de instalación (CentOS):

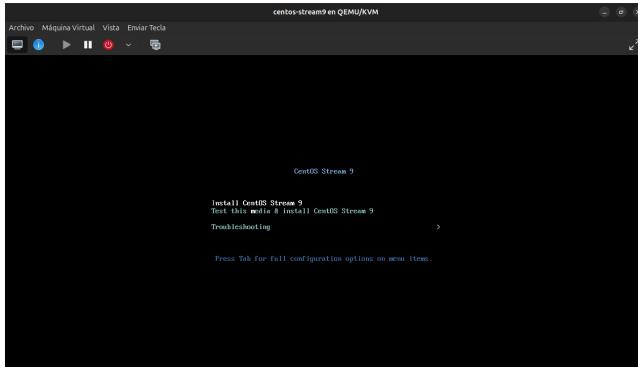


Figure 34: Inicio de instalación

En la imagen se muestra el menú de inicio de la instalación de CentOS Stream 9, además, se indica que al presionar la tecla "Tab" se pueden ver todas las opciones de configuración disponibles en los elementos del menú.

### 35.Registro:

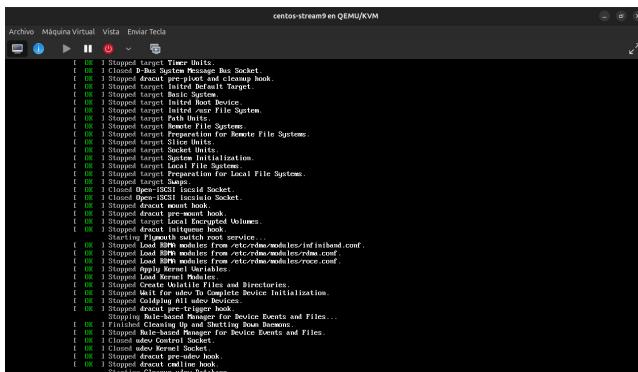


Figure 35: Registro

En la imagen se muestra un registro detallado de las operaciones realizadas durante el proceso de apagado o reinicio de un sistema, dentro de una máquina virtual. Se enumeran una serie de acciones que indican la detención de varios servicios, objetivos (targets) y sockets del sistema. Estas acciones incluyen:

- Detener servicios como Timer Units, System Message Bus Socket, y Rule-based Manager for Device Events and Files.
- Cerrar sockets relacionados con udev y Open-ISCSI.
- Detener hooks de dracut relacionados con diferentes fases del proceso de arranque y montaje.
- Finalizar la limpieza y el apagado de daemons.

El proceso se está finalizando la operación de apagado o reinicio, cerrando y limpiando todos los componentes del sistema de manera ordenada.

36.Selección de idioma: En la imagen se muestra el primer paso interactivo

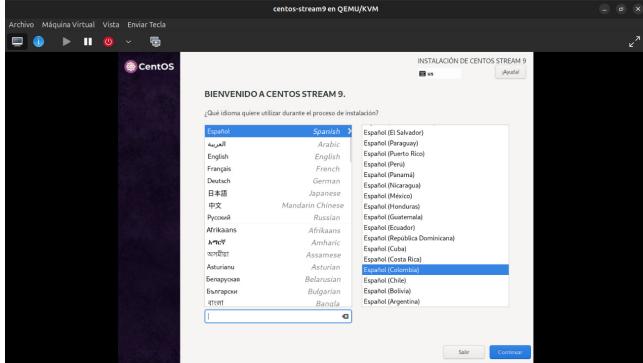


Figure 36: Selección de idioma

en el proceso de instalación de CentOS Stream 9, donde se debe seleccionar su idioma preferido. Se realiza el mismo proceso que se realizó con Windows y en scientificlinux.

37.Configuración final:

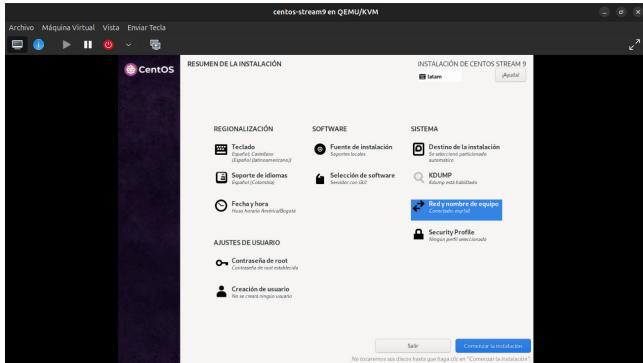


Figure 37: Configuración final

En la imagen se muestra la pantalla de configuración final antes de comenzar la instalación de CentOS Stream 9.

38.Proceso de instalación:

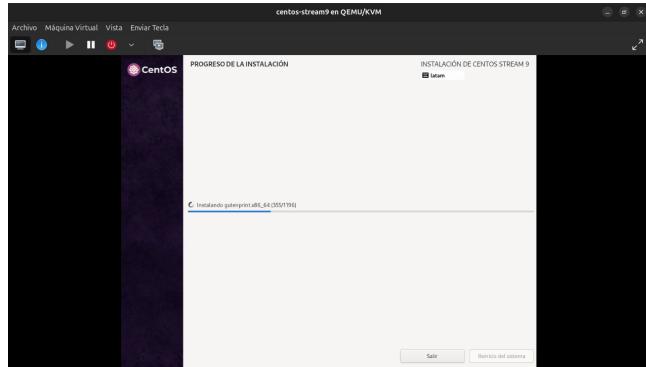


Figure 38: Proceso de instalación

En la imagen se muestra el proceso de la instalación de CentOS Stream 9 en la máquina virtual utilizando QEMU. Se indica que se está instalando el paquete "gueraprínt.365-64" y se muestra el progreso actual (355 de 1196 paquetes instalados). Esto es la fase de instalación de software del sistema operativo.

#### 39.Finalización de instalación:

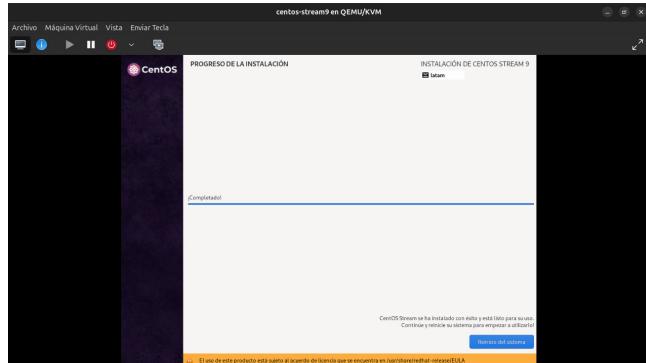


Figure 39: Finalización de instalación

En la imagen se muestra la finalización de la instalación.

#### 40.Pantalla de bienvenida:

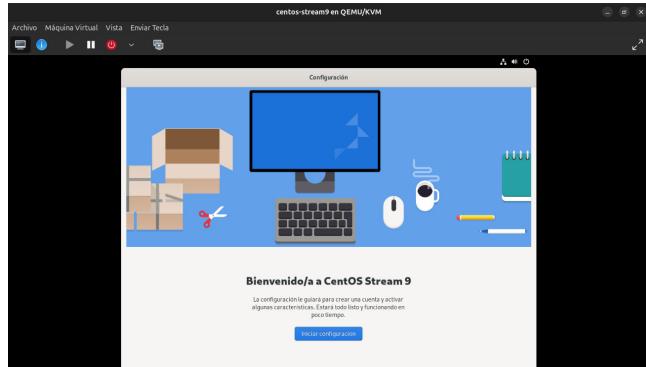


Figure 40: Pantalla de bienvenida

En la imagen se muestra la máquina virtual ejecutando CentOS Stream 9 dentro de QEMU. En la pantalla de la máquina virtual se observa la pantalla de bienvenida del asistente de configuración del sistema operativo, indicando que se guiará al usuario para crear una cuenta y activar algunas características. En la parte inferior, hay un botón azul que dice "Iniciar configuración", listo para que el usuario comience el proceso de configuración inicial del sistema.

#### 41.Escritorio:

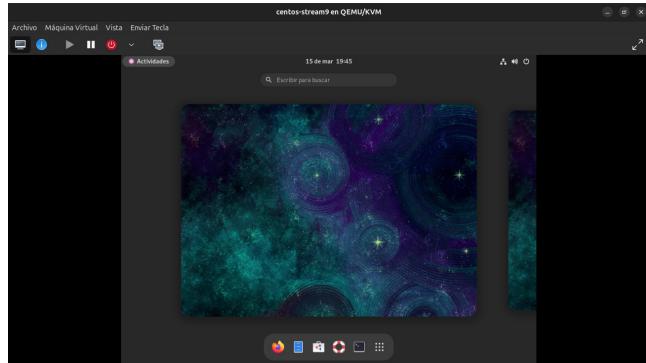


Figure 41: Escritorio

La imagen muestra la interfaz gráfica de CentOS Stream 9 ejecutándose en una máquina virtual a través de QEMU. Se observa el escritorio del sistema. Esto indica que la instalación y configuración inicial de CentOS Stream 9 ya se han completado con éxito.

#### 42.IP en terminal LINUX:

Figure 42: IP en terminal LINUX

La imagen muestra la salida del comando ip a ejecutado en una terminal de Linux. Este comando se usa para listar las interfaces de red y sus configuraciones. Se puede observar varias interfaces de red, incluyendo:

- lo (Loopback): Tiene la dirección 127.0.0.1, usada para comunicación interna dentro del sistema.
  - en0 (Ethernet cableada): Está activa (UP) con una dirección IPv4 192.168.1.9/24 y una dirección MAC 24:b6:fd:c9:cd.
  - wlp2s0 (Wi-Fi): También está activa (UP) con la dirección IPv4 192.168.1.10/24 y una dirección MAC 8c:70:5a:0d:53:4b.
  - wlp9s0b1 (Otra interfaz Wi-Fi o virtual): Está en estado DOWN, por lo que no está activa.
  - virbr0 (Interfaz de red virtual): Utilizada para máquinas virtuales, con la dirección 192.168.122.1/24.
  - vnet interfaces (vnet8, vnet3, vnet22, etc.): Asociadas a máquinas virtuales gestionadas por herramientas como libvirt o QEMU.

La salida indica que el equipo está conectado a una red tanto por Ethernet como por Wi-Fi, y también tiene configuraciones para redes virtuales.

43.Ifconfig -a en terminal LINUX:

```

gixxer15Sabs@gixxer15Sabs:~$ ifconfig -a
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.9 netmask 255.255.255.0 broadcast 192.168.1.255
                inet6 fe80::4671:f655:fcbe:63b prefixlen 64 scopeid 0x20<link>
                    ether 24:b6:fd:fb:c9:cd txqueuelen 1000 (Ethernet)
                    RX packets 25302281 bytes 36530831199 (36.5 GB)
                    RX errors 0 dropped 5566 overruns 0 frame 0
                    TX packets 2504478 bytes 227153091 (227.1 MB)
                    TX errors 0 dropped 12 overruns 0 carrier 0 collisions 0
                    device interrupt 20 memory 0xe1d00000-e1d20000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                    loop txqueuelen 1000 (Bucle local)
                    RX packets 87020 bytes 7012275 (7.0 MB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 87020 bytes 7012275 (7.0 MB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
                ether 52:54:00:fd:c4:34 txqueuelen 1000 (Ethernet)
                RX packets 137549 bytes 42257917 (42.2 MB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 427124 bytes 1809776108 (1.8 GB)
                TX errors 0 dropped 41 overruns 0 carrier 0 collisions 0

vnet5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet6 fe80::fc54:ff:fea:47aa prefixlen 64 scopeid 0x20<link>
            ether fe:54:00:a4:47:aa txqueuelen 1000 (Ethernet)
            RX packets 115288 bytes 40909422 (40.9 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1216701 bytes 1740852609 (1.7 GB)
            TX errors 0 dropped 40420 overruns 0 carrier 0 collisions 0

vnet16: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet6 fe80::fc54:ffffe95:9d9b prefixlen 64 scopeid 0x20<link>
            ether fe:54:00:95:9d:9b txqueuelen 1000 (Ethernet)
            RX packets 4564 bytes 447804 (447.8 KB)

```

Figure 43: ifconfig -a en terminal LINUX

En la imagen se muestra la salida del comando ifconfig -a que se ha ejecutado en una terminal de Linux. Este comando se utilizo para mostrar la configuración de todas las interfaces de red, incluyendo aquellas que están inactivas. Se van a observar varias interfaces:

1. eno1 (Interfaz Ethernet cableada)
  - Activa (UP), con dirección IPv4 192.168.1.9 y máscara de subred 255.255.255.0.
  - Dirección MAC: 24:b6:fd:c9:cd.
  - Tiene tráfico de red significativo en RX y TX.
1. lo (Interfaz Loopback)
  - Utilizada para comunicación interna del sistema (127.0.0.1).
  - No tiene errores ni paquetes descartados.
1. virbr0 (Interfaz de puente virtual)
  - Se usa para redes de máquinas virtuales, con dirección IPv4 192.168.122.1.

- Tiene actividad en RX y TX.
1. vnet5 y vnet16 (Interfaces virtuales para máquinas virtuales)
    - Usadas para conexiones de red en entornos de virtualización (libvirt/QEMU/KVM).
    - Ambas tienen direcciones MAC y actividad en la red.
    - vnet5 ha transferido más datos ( 1.7 GB TX).

La salida indica que el sistema está conectado a la red mediante Ethernet y tiene configuraciones de red virtual para entornos de virtualización.

44.Ifconfig -a en terminal LINUX:

```
vnet5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::fc54:ff:fea:47aa prefixlen 64 scopeid 0x20<link>
        ether fe:54:00:aa:47:aa txqueuelen 1000 (Ethernet)
          RX packets 115288 bytes 40909422 (40.9 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 1216701 bytes 1740852609 (1.7 GB)
          TX errors 0 dropped 40420 overruns 0 carrier 0 collisions 0

vnet16: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::fc54:ff:fe95:9d9b prefixlen 64 scopeid 0x20<link>
        ether fe:54:00:95:9d:9b txqueuelen 1000 (Ethernet)
          RX packets 4564 bytes 447804 (447.8 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 23669 bytes 15159542 (15.1 MB)
          TX errors 0 dropped 31326 overruns 0 carrier 0 collisions 0

vnet28: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::fc54:ff:fe7d:27af prefixlen 64 scopeid 0x20<link>
        ether fe:54:00:7d:27:af txqueuelen 1000 (Ethernet)
          RX packets 17081 bytes 2765139 (2.7 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 44782 bytes 96075963 (96.0 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::e397:b009:7f92:6239 prefixlen 64 scopeid 0x20<link>
        ether 8c:70:5a:0d:53:48 txqueuelen 1000 (Ethernet)
          RX packets 3172328 bytes 4516185109 (4.5 GB)
          RX errors 0 dropped 1984 overruns 0 frame 0
          TX packets 1008386 bytes 102332313 (102.3 MB)
          TX errors 0 dropped 16 overruns 0 carrier 0 collisions 0

wlp9s0b1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether e0:ca:94:5e:85:d0 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

gixxer155abs@gixxer155abs-Latitude-E6220:~
```

Figure 44: ifconfig -a en terminal LINUX

En esta imagen se muestra la salida del comando ifconfig -a, coloca en una lista todas las interfaces de red, activas e inactivas, en un sistema Linux. Análisis de las interfaces listadas:

1. vnet5, vnet16, vnet28 (Interfaces virtuales)

- Estas interfaces se relacionan con redes virtuales, probablemente usadas por libvirt/QEMU/KVM.
  - Tienen direcciones MAC únicas y direcciones IPv6 asignadas (fe80::...).
  - vnet5 se ha transmitido y recibido un gran volumen de datos ( 1.7 GB TX).
  - vnet16 tiene menos tráfico ( 447 KB RX, 15.1 MB TX).
  - vnet28 también tiene actividad ( 2.7 MB RX, 96 MB TX).
1. wlp2s0 (Interfaz Wi-Fi)
    - Es una interfaz de red inalámbrica (wlp2s0).
    - Tiene una dirección IPv4 (192.168.1.25).
    - Está activa (UP).
    - Ha recibido una gran cantidad de datos ( 4.5 GB RX) y ha transmitido ( 102 MB TX).
  1. wlp9s0b1 (Otra interfaz inalámbrica)
    - Aparece como BROADCAST,MULTICAST, pero no RUNNING, lo que indica que no está activa.
    - No tiene dirección IPv4 asignada.
    - No muestra el tráfico de red.

Este sistema usa una conexión Wi-Fi (wlp2s0) y tiene varias interfaces virtuales para máquinas virtuales. También hay una interfaz Wi-Fi inactiva (wlp9s0b1).

#### 45.Ip rout:

```
gixxer155abs@gixxer155abs-Latitude-E6220:~$ ip route
default via 192.168.1.1 dev eno1 proto dhcp src 192.168.1.9 metric 100
default via 192.168.1.1 dev wlp2s0 proto dhcp src 192.168.1.10 metric 600
192.168.1.0/24 dev eno1 proto kernel scope link src 192.168.1.9 metric 100
192.168.1.0/24 dev wlp2s0 proto kernel scope link src 192.168.1.10 metric 600
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
gixxer155abs@gixxer155abs-Latitude-E6220:~$
```

Figure 45: Ip rout

En la imagen se muestra la salida del comando ip route, que alista las rutas de red configuradas en el sistema. Se mostrara el análisis:

1. Rutas por defecto (default via 192.168.1.1)

- Hay dos rutas por defecto a través del mismo gateway (192.168.1.1):
    - Por eno1 (Ethernet) con métrica 100.
    - Por wlp2s0 (Wi-Fi) con métrica 600.
  - La métrica indica prioridad: eno1 tiene prioridad más alta (menor métrica), por lo que el tráfico saldrá por Ethernet cuando esté disponible.
  1. Rutas de red local (192.168.1.0/24)
    - eno1 tiene una ruta para 192.168.1.0/24, con dirección IP 192.168.1.9.
    - wlp2s0 también tiene una ruta para 192.168.1.0/24, con IP 192.168.1.10 y métrica 600.
  1. Red de virbr0 (192.168.122.0/24)
    - virbr0 es una interfaz virtual, generalmente usada por libvirt para máquinas virtuales.
    - Su dirección IP es 192.168.122.1 y permite la comunicación entre VMs y el host.

El sistema tiene conexión por Ethernet (eno1) y Wi-Fi (wlp2s0) con rutas por defecto al mismo gateway, priorizando Ethernet. También se tiene una red virtual para las máquinas virtuales (virbr0).

## 46.Nmap:

Figure 46: Nmap

En la iamgen el escaneo de red con Nmap en el rango 192.168.122.0/24 dio tres hosts activos, todos con interfaces de red virtualizadas mediante QEMU. Dos se ejecutan en Linux, con uno identificado como un NAS (Synology o Netgear RAIDiator) en 192.168.122.183, mientras que el segundo, en 192.168.122.152, tiene un servidor OpenSSH 8.7 en el puerto 22. El tercer host, 192.168.122.221, es un sistema Windows XP SP3 con el servicio Microsoft HTTPAPI 2.0 en el puerto 5357. Las máquinas virtuales tienen un entorno de pruebas.

#### 47.Nmap:

Figure 47: Nmap

En la imagen se muestra el análisis de la red realizada con Nmap muestra varios dispositivos activos en el rango de direcciones 192.168.122.0/24. Se identificarán ciertos tipos de sistemas operativos y servicios, como OpenSSH en equipos Linux y HTTPAPI en un dispositivo con Windows. También se pudo detectar dispositivos con Synology DiskStation Manager y Netgear RAIDiator. Hubieron detecciones de SO que no son precisas.

48.curl ifconfig.me:

```
Mostrar aplicaciones s-Latitude-E6220:~$ curl ifconfig.me  
abs@gixxer155abs-Latitude-E6220:~$
```

Figure 48: curl ifconfig.me

En la imagen se puede ver que se ejecuto el comando curl ifconfig.me, el cual se utiliza para obtener la dirección IP pública del dispositivo.

#### 49. Escaneo de puertos:

```
152.204.177.221gixxer155abs@giixer155abs-Latitude-E6220: $ sudo nmap -p- 152.204.177.221
Starting Nmap 7.94VN ( https://nmap.org ) at 2025-03-15 21:34 -05
Nmap scan report for 152.204.177.221
Host is up (0.022s latency).

Not shown: 65523 closed tcp ports (reset)
PORT      STATE    SERVICE
21/tcp    filtered  ftp
22/tcp    filtered  ssh
23/tcp    filtered  telnet
53/tcp    filtered  domain
80/tcp    filtered  http
161/tcp   filtered  snmp
1024/tcp  open      kdm
1990/tcp  filtered  stun-p1
7547/tcp  filtered  ccmp
8000/tcp  filtered  http-alt
16667/tcp open      unknown
44401/tcp filtered  unknown

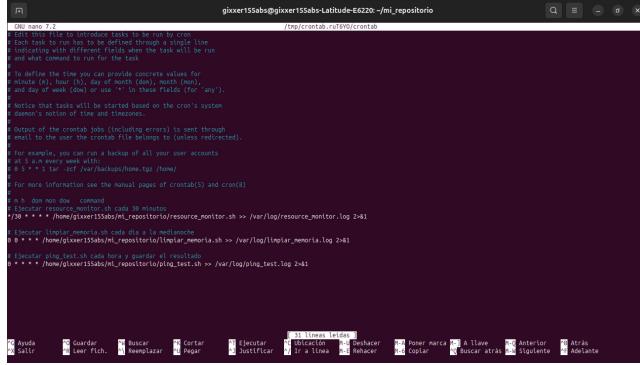
Nmap done: 1 IP address (1 host up) scanned in 23.24 seconds
gixxer155abs@giixer155abs-Latitude-E6220: $
```

Figure 49: Escaneo de puertos

En la imagen se muestra un escaneo de puertos realizado con nmap sobre la dirección IP 152.204.177.221. El escaneo muestra que la mayoría de los puertos

están filtrados, lo hace que estén bloqueados por un firewall o dispositivo de seguridad. Sin embargo, se identificaron algunos puertos abiertos

#### 50.crontab:



```

gioxer155abs@giixer155abs-Latitude-E6220:~/mi_repositorio
crontab -e

# Only run /2
# This file tells the scheduler tasks to be run by cron.
# Each task to run has to be defined through a single line
# consisting of the time to run, the command to run, where the task will be run
# and what command to run for the task

# To define the time you can provide concrete values
# for the hour (0-23), minute (0-59), month (0-11),
# and day of the week (0-6) or use wildcards (* for "any").

# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.

# Output of the cronjob (including errors) is sent through
# email to the user the cronjob file belongs to (unless redirected).

# For example, you can run a backup of all your user accounts
# every 3 hours (you may need to change 2:30 to 23:30)
# 0 3 * * * tar -zcf /var/backups/home.tgz /home

# For more information see the manual pages of crontab(5) and cron(8)

# h m dia mes año comand
# 0 0 * * * comando que se ejecuta cada hora
*/30 * * * * /home/gioxer155abs/mi_repositorio/resource_monitor.sh >> /var/log/resource_monitor.log 2>1
0 2 * * * /home/gioxer155abs/mi_repositorio/limpiar_memoria.sh >> /var/log/limpiar_memoria.log 2>1
0 * * * * /home/gioxer155abs/mi_repositorio/ping_test.sh >> /var/log/ping_test.log 2>1

```

Figure 50: crontab

En la imagen se ve el archivo crontab, el cual contiene tres tareas programadas en un sistema Linux mediante cron. La primera tarea se ejecuta resource-monitor.sh cada 30 minutos, la segunda tarea se ejecuta limpiar-memoria.sh diariamente a la medianoche, y la tercera y ultima tarea se ejecuta ping-test.sh cada hora. Cada tarea se redirige su salida y errores a archivos de log específicos en /var/log/, asegurando un registro de su ejecución.

#### 51.Limpiar-memoria.sh:



```

gioxer155abs@giixer155abs-Latitude-E6220:~/mi_repositorio
/home/gioxer155abs/mi_repositorio/limpiar_memoria.sh

.: "Liberando memoria caché..."
root sync
root sysctl -w vm.drop_caches=3
Memoria liberada.

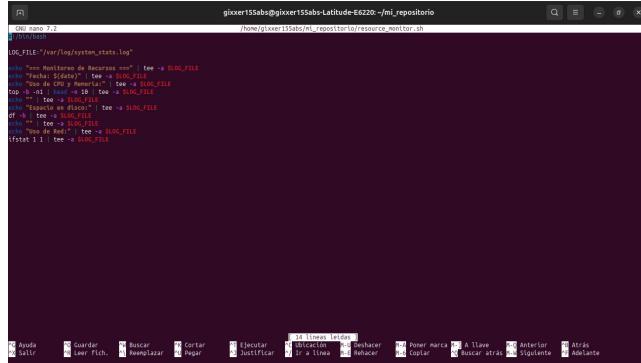
```

Figure 51: Limpiar-memoria.sh

En la imagen se muestra el script limpiar-memoria.sh está diseñado para liberar la memoria caché en un sistema Linux. Primero, se imprime un mensaje indicando que se iniciará la limpieza, luego ejecuta sync para escribir los datos pendientes en el disco y poder utilizar sysctl -w vm.drop-caches=3 con privilegios de superusuario para liberar el caché del sistema. Finalmente, imprime un

mensaje confirmando que la memoria ha sido liberada.

#### 52.Resource-monitor.sh:



```
gixxer155bs@gixxer155bs-Latitude-E6220:~/mi_repositorio
[1] 1004/gixxer155bs/mi_repositorio/resource-monitor.sh
LOG_FILE="/var/log/system-stats.log"
date >> $LOG_FILE
echo "Fechas: $(date)" | tee -a $LOG_FILE
free -m | tee -a $LOG_FILE
top -b -n 1 | head -n 10 | tee -a $LOG_FILE
df -h | tee -a $LOG_FILE
df -h --total | tee -a $LOG_FILE
ifstat -t 1 | tee -a $LOG_FILE
```

Figure 52: Resource-monitor.sh

En la imagen se muestra el script resource-monitor.sh el cual captura y registra información del sistema en el archivo /var/log/system-stats.log. Se registra la fecha, el uso de CPU y memoria con top, el espacio en disco con df -h y el uso de red con ifstat. Utiliza tee -a para que se pueda añadir la salida de los comandos al archivo de log, permitiendo la revisión posterior.

#### 53.Ping-test.sh:



```
gixxer155bs@gixxer155bs-Latitude-E6220:~/mi_repositorio
[1] 1014/gixxer155bs/mi_repositorio/ping-test.sh
read -p "Introduce la dirección a hacer ping:" url
ping -c 4 $url
```

Figure 53: Ping-test.sh

En la imagen se muestra el script ping-test.sh que solicita una dirección para hacer ping, la almacena en la variable url y luego se ejecuta ping -c 4 \$url para enviar cuatro paquetes ICMP a la dirección ingresada, verificando la conectividad correspondiente.

### **3 Conclusiones**

1. Instalación de QEMU y Máquinas Virtuales: Se logró el propósito de instalar QEMU y configurar máquinas virtuales con Windows, CentOS y Scientific Linux, asegurando el correcto funcionamiento y documentación de cada paso del proceso.
1. Escaneo de Red con Nmap: Se realizó el análisis correspondiente de la red tanto a nivel local como global mientras que las máquinas virtuales estaban encendidas, identificando dispositivos que estuvieran conectados y se hubiera realizado la verificación de la información que este expuesta a la red.
1. Automatización con Cron: Se realizó la implementación de tres tareas automáticas con Cron, programando acciones como la actualización del sistema, el monitoreo de red y la ejecución de scripts personalizados, lo que optimiza la administración del entorno virtual, teniendo en cuenta todo lo aprendido durante las clases.