

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Grado

**DESARROLLO DE UNA INTERFAZ
GRÁFICA PARA GESTIÓN DE MÁQUINAS
VIRTUALES**

(Development of a graphical interface for
management of virtual machines)

Para acceder al Título de

Graduado en

Ingeniería de Tecnologías de Telecomunicación

Autor: María Isabel Sciddurlo Juaristi

Septiembre -
2024

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: MARÍA ISABEL SCIDDURLO JUARISTI

Director del TFG: ALBERTO ELOY GARCIA GUTIERREZ

Título: “DESARROLLO DE UNA INTERFAZ GRÁFICA PARA GESTIÓN DE MÁQUINAS VIRTUALES”

**Title: “DEVELOPMENT OF A GRAPHICAL INTERFACE FOR VIRTUAL MACHINES
MANAGEMENT”**

Presentado a examen el día: 21-10-2024

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): M. MERCEDES GRANDA MIGUEL

Secretario (Apellidos, Nombre): FELIX FANJUL VELEZ

Vocal (Apellidos, Nombre): JOSE ANGEL IRASTORZA TEJA

Este Tribunal ha resuelto otorgar la calificación de:

Fdo: El Presidente

Fdo: El Secretario

Fdo: El Vocal

Fdo: El Director del TFG

(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº

(a asignar por Secretaría)

Resumen

Esta interfaz gráfica ha sido diseñada para interactuar con máquinas virtuales gestionadas por VirtualBox, cubriendo las deficiencias que tiene la aplicación nativa a la hora de representar, de forma gráfica, el estado y conectividad de las máquinas virtuales gestionadas. Haciendo uso de Python como lenguaje de programación y aprovechando las capacidades nativas del hipervisor para ser controlado mediante comandos de línea a través de su Software Development Kit. Mediante el uso de bibliotecas como Tkinter y Matplotlib, se ha desarrollado una aplicación que incluye gráficos de red y menús de funciones para la gestión de las máquinas virtuales instaladas. Así, la interfaz tiene como objetivo principal plasmar el mapa de redes y facilitar la gestión y manipulación de las máquinas virtuales (VMs) dentro de VirtualBox. Se ha incluido una librería de funciones entre las que caben destacar: encender y apagar, cambiar el nombre, importar archivos OVA, eliminar VMs, añadir y eliminar adaptadores de red, cambiar VMs de red o abrir una ventana de terminal.

Abstract

This graphical interface has been designed to interact with virtual machines managed by VirtualBox, covering the deficiencies that the native application has when it comes to graphically representing the status and connectivity of the managed virtual machines. Using Python as a programming language and taking advantage of the native capabilities of the hypervisor to be controlled by line commands through its Software Development Kit. Using libraries such as Tkinter and Matplotlib, an application has been developed that includes network graphics and function menus for the management of installed virtual machines. Thus, the interface's main objective is to map the networks and facilitate the management and manipulation of virtual machines (VMs) within VirtualBox. A library of functions has been included, including: turning on and off, changing the name, importing OVA files, deleting VMs, adding and removing network adapters, changing network VMs or opening a terminal window.

Agradecimientos

En primer lugar, deseo agradecer a mi familia, en especial a mi madre, a mi tío y a mi hermano, que hicieron lo imposible para que yo esté donde estoy, por el apoyo que me han dado durante todo el grado y sobre todo para este último proyecto. Por alentarme a seguir adelante frente a los diversos problemas que se me presentaron durante el desarrollo de este trabajo y del grado.

También quiero dar las gracias a todos mis amigos de la carrera, en especial a Raquel Martín y a Andrés Montero, quienes procuraron estar siempre ahí desde el primer momento para brindarme su ayuda en cuanto la necesitaba.

Por último, pero no menos importante, agradecer a mi tutor del TFG Alberto Eloy García por haberme propuesto este tema tan interesante y haberme ayudado a resolver toda duda que me surgía durante el desarrollo del proyecto.

Tabla de contenido

Índice de figuras.....	12
Acrónimos.....	14
1 Introducción	16
1.1 Motivación y objetivos.....	17
1.2 Organización del documento	18
2 Marco teórico	19
2.1 Virtualización	19
2.2 Máquina virtual	20
2.3 Hipervisor	21
2.4 Red virtual	22
2.4.1 NAT.....	23
2.4.2 Internal Network	24
2.4.3 Host-Only.....	25
2.4.4 Bridged	26
2.5 VirtualBox	27
2.5.1 Deficiencias de la interfaz gráfica nativa	30
2.5.2 VboxManage.....	31
2.5.3 Python SDK para VirtualBox	32
2.6 Python	33
2.6.1 Tkinter	34
2.6.2 Matplotlib.....	35
2.6.3 NetworkX.....	35
3 Marco práctico.....	37
3.1 una interfaz gráfica para la gestión de máquinas virtuales	37
3.2 Elección de VirtualBox	38
3.3 Elección de Python.....	39
3.3.1 Librerías y módulos de código.....	39
3.4 Preparación del entorno de desarrollo.....	40
3.4.1 VirtualBox.....	40
3.4.2 Python	40
3.4.3 Pywin32.....	41
3.4.4 Pyvbox.....	41

3.4.5	SDK de VirtualBox	42
3.4.6	PyPI	42
3.4.7	Otras librerías	43
3.5	Funciones de la interfaz gráfica	43
3.5.1	Funciones para pintar el grafo	43
3.5.1.1	Funciones para obtener información	43
3.5.1.2	Funciones para representar el mapa de redes y máquinas virtuales	44
3.5.2	Funciones para el menú	44
3.5.3	Funciones para la gestión de VMs	45
3.5.3.1	Obtener información de la VM	45
3.5.3.2	Operaciones sobre las VMs	45
3.5.3.3	Funciones para la conexión SSH	47
3.5.4	Funciones para el estado de VMs	48
3.6	Flujo de la interfaz	48
3.6.1	Obtención de la información de VirtualBox	48
3.6.2	Modificación de la configuración o estado de una VM	50
3.6.3	Establecer conexión SSH	58
3.6.4	Importar o eliminar una VM	62
3.6.5	Mostrar información de la VM	64
3.6.6	Refrescar	64
4	Conclusiones, Aplicaciones y Líneas futuras	65
4.1	Conclusiones	65
4.2	Aplicaciones	66
4.3	Líneas futuras	66
5	Bibliografía y referencias	67

ÍNDICE DE FIGURAS

Figura 1: Virtualización.....	19
Figura 2: Sistemas Operativos	20
Figura 3: Modelo hipervisor tipo 1	21
Figura 4: Modelo hipervisor tipo 2	22
Figura 5: NAT	23
Figura 6: NAT Network	24
Figura 7: Internal Network	25
Figura 8: Host-Only	26
Figura 9: Bridged	27
Figura 10: Interfaz de VirtualBox	28
Figura 11: Opciones del menú de la máquina virtual de VirtualBox	29
Figura 12: Menú avanzado de VirtualBox	29
Figura 13: Opción “Configuración de VirtualBox”	30
Figura 14: Resultado del comando VboxManage list vms	32
Figura 15: Comando para configurar en el adaptador una red Host-Only	32
Figura 16: Resultado de aplicar el comando de configurar la red.....	32
Figura 17: Ventana del menú hecha con Tkinter.....	34
Figura 18: Cuadro de diálogo hecho con Tkinter.....	34
Figura 19: Combobox hecho con Tkinter	35
Figura 20: Mapa de redes hecho con Matplotlib y NetworkX	36
Figura 21: Interfaz gráfica.....	38
Figura 22: Obtención de información de VirtualBox	49
Figura 23: Menú de la interfaz	50
Figura 24: Flujo básico para la modificación de una VM	51
Figura 25: Panel de adaptadores habilitados	51
Figura 26: Panel de redes	52
Figura 27: Advertencia de apagado para cambiar de red.....	52
Figura 28: Advertencia de misma red seleccionada	52
Figura 29: Diagrama de cambio de red del adaptador	53
Figura 30: Panel para añadir un nuevo adaptador	53
Figura 31: Adaptadores habilitados.....	54
Figura 32: Adaptadores deshabilitados	54
Figura 33: Tipos de red.....	54
Figura 34: Selección de red Host-Only	55
Figura 35: Advertencia de apagado para habilitar un adaptador	55
Figura 36: Advertencia de apagado para deshabilitar un adaptador	55
Figura 37: Diagrama para añadir o eliminar un adaptador.....	56
Figura 38: Panel para escribir el nuevo nombre.....	56
Figura 39: Advertencia de apagado para cambiar el nombre de una VM	56
Figura 40: Diagrama para cambiar el nombre de una VM.....	57
Figura 41: Advertencia de VM encendida	57
Figura 42: Advertencia de VM apagada	57

Figura 43: Diagrama de cambio de estado encendido o apagado	58
Figura 44: Advertencia de apagado para establecer una sesión SSH.....	58
Figura 45: Selección de IP para establecer la sesión SSH.....	58
Figura 46: Panel para el usuario	59
Figura 47: Panel para la contraseña.....	59
Figura 48: Repetición de la contraseña en el terminal	59
Figura 49: Sesión establecida correctamente	60
Figura 50: Fallo de conexión por permiso denegado	60
Figura 51: Diagrama para establecer una sesión SSH.....	61
Figura 52: Ventana para la búsqueda del archivo OVA	62
Figura 53: Panel para el nombre de la VM importada.....	62
Figura 54: Grafo de red con la nueva VM	63
Figura 55: Diagrama para importar una VM	63
Figura 56: Advertencia de apagado para eliminar una VM	63
Figura 57: Diagrama para eliminar una VM	64
Figura 58: Panel de información	64

ACRÓNIMOS

API: Application Programming Interface

VM: Virtual Machine

CLI: Command Line Interface

SDK: Software Development Kit

OVA: Open Virtual Appliance

SSH: Secure Shell

NIC: Network Interface Card

NAT: Network Address Translation

SCP: Secure Copy

SFTP: SSH File Transfer Protocol

1 INTRODUCCIÓN

La virtualización es una tecnología que permite la creación de múltiples máquinas virtuales (VMs) que funcionan sobre un hardware físico, lo que facilita y aprovecha el uso de los recursos, la gestión de los entornos de desarrollo y pruebas y la división del de trabajo. VirtualBox, un software de virtualización, es la aplicación que he elegido para este proyecto. Sin embargo, gestionar máquinas virtuales a través de la interfaz de VirtualBox o utilizando comandos de línea puede resultar algo complicado y hasta poco intuitivo como, por ejemplo, cuando se maneja un gran número de VMs y redes virtuales.

Para abordar esta necesidad que llega a convertirse en ocasiones en problemas, se ha desarrollado una interfaz gráfica utilizando el lenguaje de programación Python y la biblioteca Tkinter y Matplotlib, entre otras. Esta interfaz tiene el propósito de facilitar y automatizar la gestión de máquinas virtuales en VirtualBox, proporcionando una herramienta sencilla para los usuarios con diferentes niveles de experiencia en este tipo de entornos.

La interfaz gráfica, programada en Python, permite a los usuarios realizar una variedad de tareas relacionadas con la gestión de VMs de manera más fácil e intuitiva, como por ejemplo la creación, modificación, eliminación de máquinas virtuales, así como la administración de adaptadores de red de redes Host-Only y NAT y el establecimiento de conexiones SSH.

Una de las características más importantes de la interfaz es su capacidad para interactuar con VirtualBox mediante el uso de Python SDK y los comandos de VBoxManage, lo que permite ejecutar comandos para tratar a las VMs.

A través de menús intuitivos y ventanas emergentes, los usuarios pueden realizar acciones como cambiar el nombre de una VM, añadir o eliminar adaptadores de red o incluso importar nuevas máquinas virtuales. La interfaz también incluye funcionalidades un poco más avanzadas, como la posibilidad de iniciar sesiones SSH para poder conectarse a la VM de manera remota y poder trabajar en su consola de comandos. También se implementan advertencias y mensajes de error.

1.1 MOTIVACIÓN Y OBJETIVOS

La motivación detrás de este proyecto es poder facilitar el aprendizaje de conceptos de máquina y red virtual, y simplificar la gestión de estos, como principal referencia en el apoyo de asignaturas que hacen uso intensivo de este tipo de sistemas, sin ninguna introducción previa. La mayor parte de los programas de virtualización no incluyen ningún apartado en su interfaz gráfica que permita mostrar cómo se relacionan las máquinas virtuales creadas entre sí, a través de las correspondientes redes, físicas o virtuales. Así, los usuarios echan en falta esta interfaz, en la que quede plasmado un mapa de red, en el cual se vea claramente las interconexiones entre las redes y las máquinas virtuales.

Los objetivos del proyecto son los siguientes:

1. **Desarrollo de una interfaz gráfica intuitiva:** Las herramientas de virtualización presentan interfaces básicas mediante las cuales crear y acceder a las máquinas virtuales. Las operaciones relacionadas con la gestión de la red a la que se conectan las máquinas es mínima, salvo que se realice a través de comandos y un terminal. Una interfaz gráfica que acceda a esos comandos, facilitando su uso, y que permita a los usuarios realizar tareas complejas sin que ellos las perciban como tal, solo puede realizarse de forma gráfica, mediante un mapa de red en el que se observen todas las interconexiones y estados de las diferentes máquinas virtuales.
2. **Integración con VirtualBox a través de Python:** En el caso concreto de VirtualBox, se incluyen dos utilidades: VBoxManage y Python SDK. La primera es una API de comandos, que permite acceder a todas las funcionalidades del programa desde un terminal, mientras que la segunda actúa como API para el acceso a las utilidades de VirtualBox directamente desde aplicaciones desarrolladas en lenguaje Python. Ambos mecanismos permiten interactuar con VirtualBox con sus diversas funciones y crear una herramienta accesible. Sin embargo, en ambos casos la comunidad de desarrolladores bien no las conoce, o no las han considerado aún en sus aplicaciones.
3. **Facilitar la gestión de máquinas virtuales:** Realizar operaciones sobre una máquina virtual, más allá del uso en sí del ordenador como tal, puede llegar a ser farragoso, especialmente si se tiene en cuenta que dichas operaciones se deben realizar a través de interfaces de comandos más o menos complejas. Es fundamental reducir esta complejidad, normalmente mediante el uso de los atajos que suelen ofrecer las aplicaciones visuales. Por lo tanto, como objetivo general se presenta la necesidad de facilitar la gestión de máquinas virtuales, haciendo que las tareas sean más fáciles para el usuario.

A título personal, desde que empecé la carrera de telecomunicaciones sentí gran interés por la virtualización, utilizada ampliamente en las primeras asignaturas. En este proyecto he podido combinar mi habilidad de programación con mi interés por la

virtualización, llevándome a desarrollar una interfaz gráfica que facilita la gestión de máquinas virtuales de una manera bastante intuitiva.

1.2 ORGANIZACIÓN DEL DOCUMENTO

Este documento está estructurado de la siguiente forma:

En el capítulo 2 se desarrolla el marco teórico donde se explican los conceptos básicos que nos ayudan a entender qué es la virtualización, las VMs, las redes virtuales, VirtualBox... y así poder comprender el uso y complejidad de esta interfaz gráfica.

En el capítulo 3 se desarrolla la explicación de por qué se elige VirtualBox y el lenguaje de programación, así como por qué surge la necesidad de crear una herramienta para gestionar VMs virtuales. También abordará la explicación de las funciones de la interfaz, las librerías que se han usado en el código y qué es lo necesario para poder ejecutarla.

Finalmente, en el capítulo 4 se exponen las conclusiones y líneas futuras que puede ofrecer la interfaz gráfica desarrollada en este trabajo.

El presente capítulo tiene como objetivo explicar los conceptos básicos de la virtualización de sistemas, punto clave de este proyecto.

2.1 VIRTUALIZACIÓN

La virtualización es una tecnología que permite la creación de versiones virtuales de recursos tecnológicos, como redes y sistemas operativos. A través de la virtualización, es posible ejecutar múltiples sistemas operativos y aplicaciones en un solo servidor físico, lo que optimiza el uso de los recursos. En la Figura 1 se muestra la composición de la virtualización de un sistema. [1]

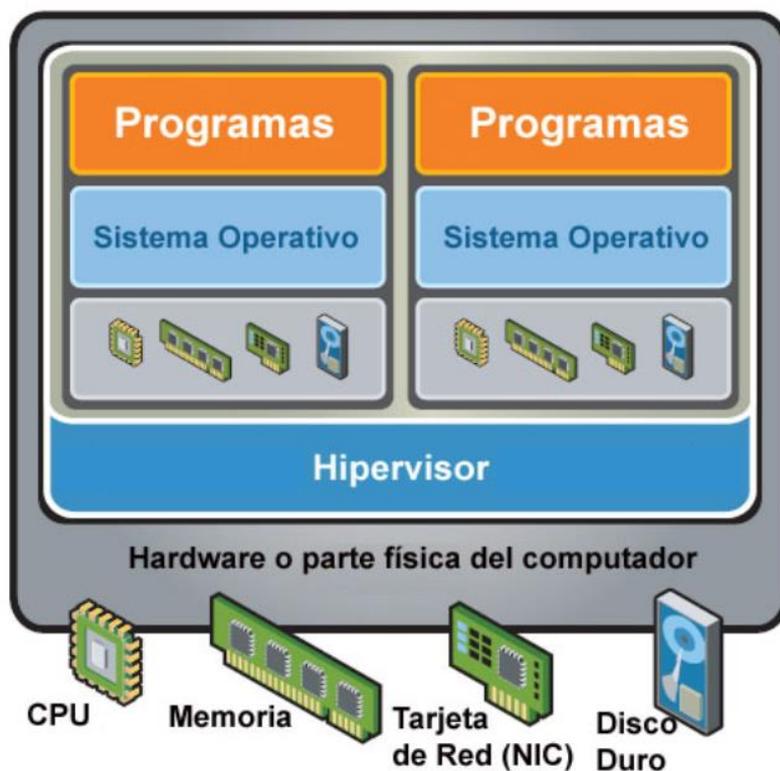


Figura 1: Virtualización [1]

La gestión de recursos en un entorno virtualizado implica la asignación y optimización de recursos físicos como CPU, memoria, almacenamiento...

- **Virtualización de datos:** facilita a las empresas la integración de múltiples fuentes de datos en un suministro único y dinámico.
- **Virtualización de servidores:** permite a los administradores dividir los servidores en particiones que desempeñan funciones específicas.
- **Virtualización de sistemas operativos:** permite la ejecución de varios sistemas operativos en una misma máquina.

- **Virtualización de funciones de red:** separa las funciones de la red, como los servicios de directorio, el uso compartido de archivos y la configuración de IP, para distribuir las entre distintos entornos. [2]

2.2 MÁQUINA VIRTUAL

Una máquina virtual (VM) es una emulación de un sistema de computación físico que se ejecuta en un entorno virtualizado que funciona como un sistema aislado. Las VMs permiten ejecutar múltiples sistemas operativos en un solo hardware físico, cada uno con su propio conjunto de aplicaciones y recursos asignados [2].

El sistema asociado con la máquina virtual se denomina "Sistema Operativo Invitado" (Guest OS). Este es el sistema operativo que se ejecuta dentro de una máquina virtual. Actualmente es posible virtualizar casi cualquier sistema operativo, aunque a nivel de aprendizaje los usuarios nos limitamos a los más conocidos o comerciales, como los indicados en la Figura 2.

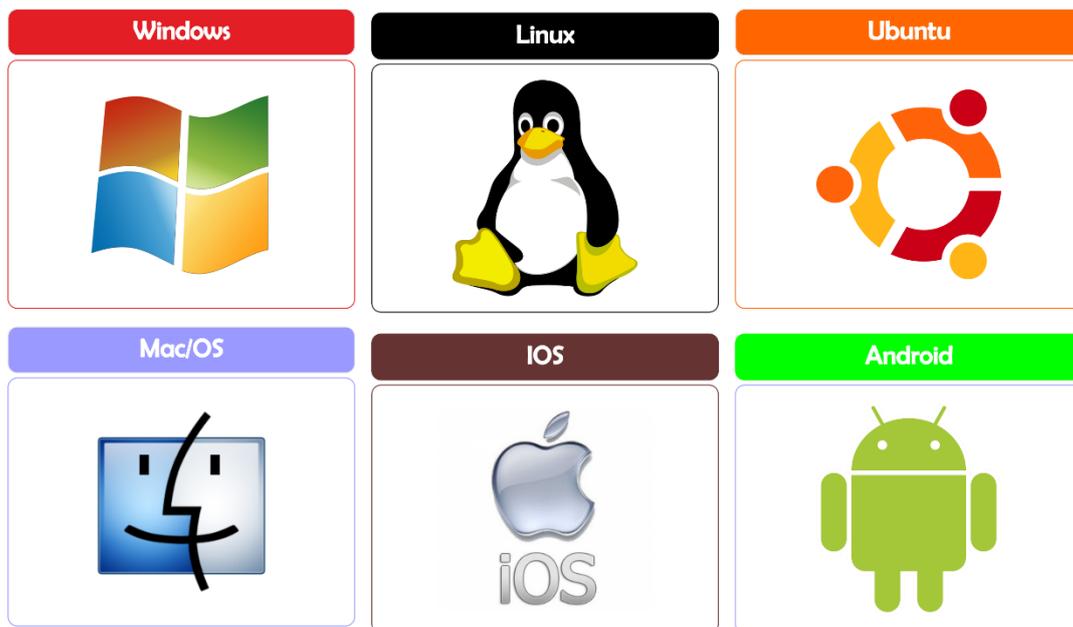


Figura 2: Sistemas Operativos [3]

El sistema operativo anfitrión, por otro lado, es el que se ejecuta directamente en el hardware del host, el equipo físico. En general, se denomina "host" o "anfitrión", y se refiere a la máquina física que alberga los sistemas virtuales. Es el equipo físico que tiene los recursos necesarios (CPU, memoria, interfaces de red...) para ejecutar sistemas operativos o aplicaciones virtuales.

El host ejecuta un software llamado hipervisor que permite la creación y gestión de las máquinas virtuales. Cada máquina virtual actúa como un sistema independiente, pero comparte los recursos físicos proporcionados por el host.

2.3 HIPERVISOR

Un hipervisor es un software que crea y ejecuta máquinas virtuales. El hipervisor gestiona los recursos del hardware, como la CPU, memoria, y almacenamiento, y los distribuye entre las máquinas virtuales. Existen hipervisores de tipo 1 y 2:

- **Hipervisor de tipo 1:** Esta modalidad se distingue porque el software se instala directamente en el hardware, asumiendo tanto las funciones de sistema operativo (SO) como de plataforma de virtualización. La Figura 3 muestra el modelo de este tipo de hipervisor.

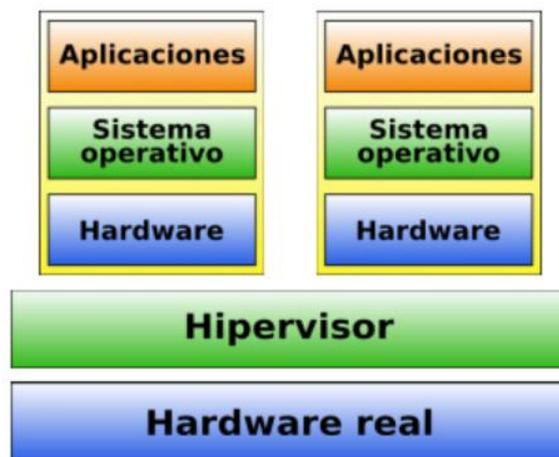


Figura 3: Modelo hipervisor tipo 1 [1]

Entre los hipervisores de tipo 1 más populares se encuentran:

- VMware ESXi (gratis)
 - VMware ESX Server (de pago)
 - Xen (de código abierto)
 - Citrix XenServer (gratis)
 - Microsoft Hyper-V Server (gratis) [1]
- **Hipervisor de tipo 2:** Esta modalidad se distingue porque debe instalarse en un equipo que ya tenga un sistema operativo (SO) previamente instalado, como Linux, Windows, IOS, entre otros. La Figura 4 muestra el modelo de este tipo de hipervisor.



Figura 4: Modelo hipervisor tipo 2 [1]

Aunque este método de virtualización es menos eficiente que la virtualización del hipervisor de tipo 1, permite que el equipo físico continúe utilizando otras aplicaciones y servicios mientras se ejecutan las máquinas virtuales.

Entre los hipervisores de tipo 2 más comunes se encuentran:

- Oracle VM VirtualBox (gratuito con extensiones gratuitas bajo licencia PUEL)
- VMware Workstation (de pago)
- VMware Server (gratuito)
- VMware Player (gratuito)
- QEMU (de código abierto)
- Microsoft Virtual PC (gratuito)
- Microsoft Virtual Server (gratuito) [1]

En este trabajo se toma como referencia VirtualBox, es decir, un hipervisor de tipo 2, tanto por su sencillez, como por su condición de “Software Open Source”. No es de extrañar que sea uno de los más utilizados en los entornos académicos.

2.4 RED VIRTUAL

Las redes virtuales permiten la interconexión de máquinas virtuales y otros recursos dentro de un entorno virtualizado. Estas redes simulan el comportamiento de una red física.

La NIC, o tarjeta de red, es el componente físico que permite a la máquina virtual conectarse a la red externa y comunicarse con otros dispositivos. Sin embargo, el hipervisor puede configurar varias NICs virtuales (VNIC) que son asignadas a una máquina virtual, permitiendo no solo conectar hacia el exterior, sino hacia otras VNIC,

formando redes virtuales. Existen diferentes tipos de redes virtuales, aunque VirtualBox solo incluye las redes NAT, interna, Host-Only y Bridge, cada una con sus características.

2.4.1 NAT

Este tipo de red permite que las VNIC de las máquinas virtuales se conecten a la red externa utilizando la dirección IP del host. Es útil cuando se necesita conectividad externa sin exponer las VMs directamente a la red. En la Figura 5 se muestra el esquema de red NAT.

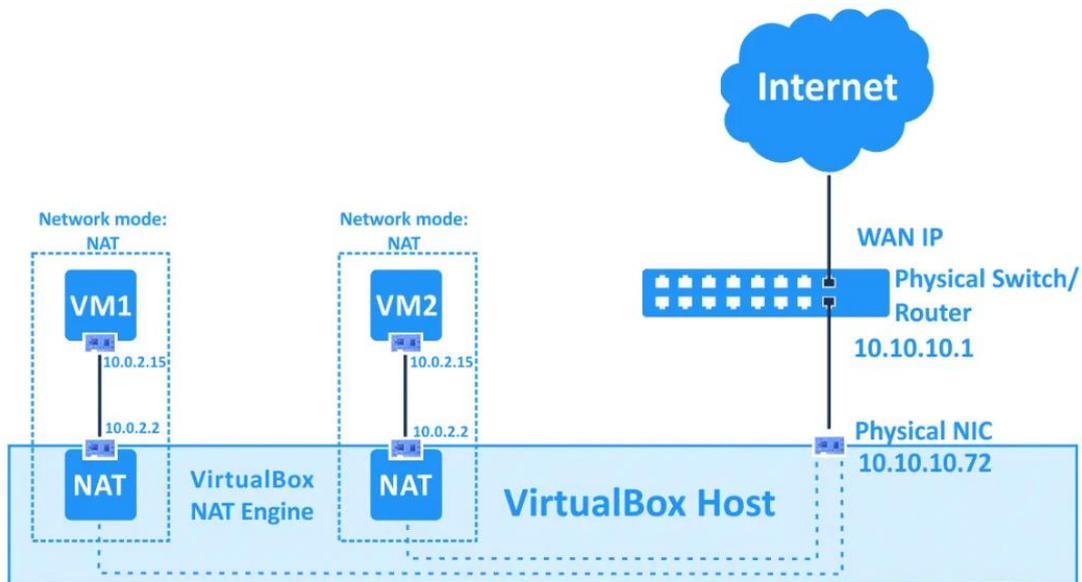


Figura 5: NAT[4]

Utilidades:

- NAT es útil cuando se quiere permitir a las VMs acceder a Internet, pero no se requiere que otros dispositivos en la red local puedan acceder a las VMs.
- Las máquinas virtuales están aisladas entre sí, pero pueden acceder a Internet a través del host.

Usos comunes:

- Navegación segura en Internet desde una VM, sin exposición directa a la red pública.
- Descarga de actualizaciones o software desde Internet en una VM que no necesita estar accesible desde otras redes.
- Despliegue de aplicaciones cliente que necesitan acceso a Internet, pero no requieren comunicación directa con otras VMs. [5]

VirtualBox incluye una variante de la red NAT, denominada NAT Network. La red NAT permite que los dispositivos dentro de la misma red NAT puedan comunicarse entre sí. En la Figura 6 se muestra el esquema de la red NAT.

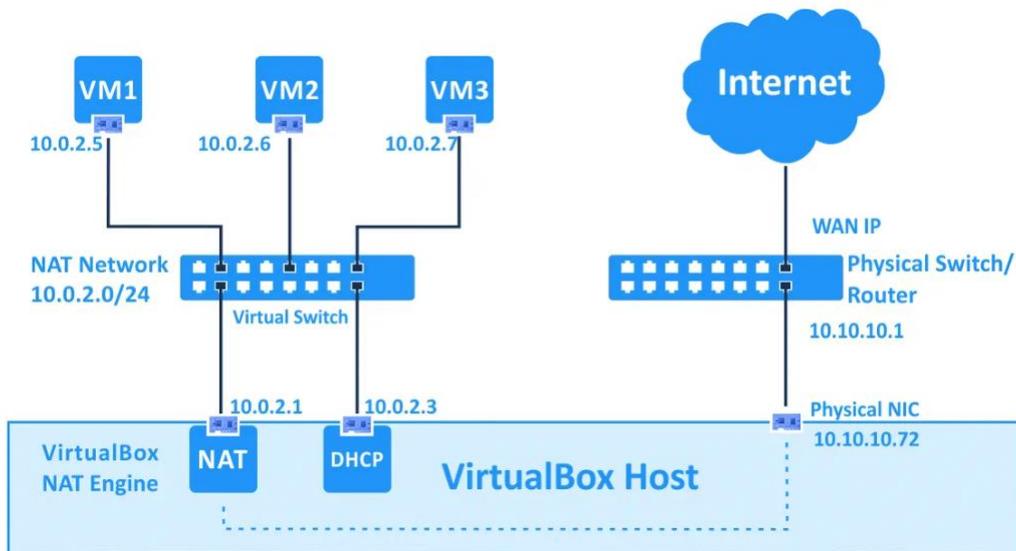


Figura 6: NAT Network [4]

Utilidades:

- Similar al modo NAT básico, pero permite la comunicación entre VMs dentro de la misma red NAT.
- Ideal para configuraciones donde múltiples VMs necesitan comunicarse entre sí y también con Internet.

Usos comunes:

- Configuración de servicios en una VM (como servidores web o bases de datos) que deben ser accesibles desde otras VMs en la misma red NAT.

Uno u otro modo puede ser útil en función del aislamiento que queramos dar a cada máquina o conjunto de máquinas.

2.4.2 Internal Network

Es una configuración de red que permite crear redes completamente aisladas, donde solo las máquinas virtuales (VMs) que forman parte de la misma red interna pueden comunicarse entre sí. En la Figura 7 se ve el esquema de red de Internal Network.

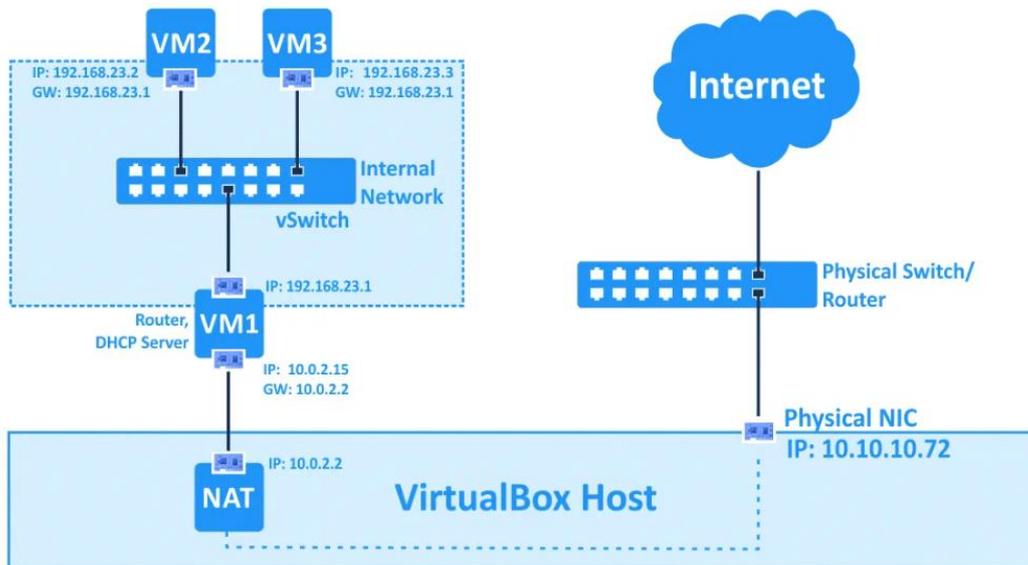


Figura 7: Internal Network [4]

Utilidades:

- Aislamiento Completo
- Seguridad: Es útil para simular un entorno de red seguro donde las VMs deben interactuar entre sí sin riesgo de intrusión externa.

Usos comunes:

- Laboratorios Virtuales: En entornos educativos se pueden crear laboratorios virtuales donde se pueden experimentar con configuraciones de red sin afectar el entorno real.
- Entornos de Pruebas: Configuración de entornos de prueba donde los desarrolladores o equipos de TI pueden realizar experimentos sin riesgos de seguridad.
- Esta configuración es recomendada para establecer, por ejemplo, laboratorios de redes aislados, evitando la interacción ya sea del exterior, o directamente desde las máquinas virtuales.

2.4.3 Host-Only

Este tipo de red aísla las máquinas virtuales del host al igual que la red interna, pero en este caso se permite la comunicación entre la VM y el host. En la Figura 8 se ve el esquema de este tipo de red.

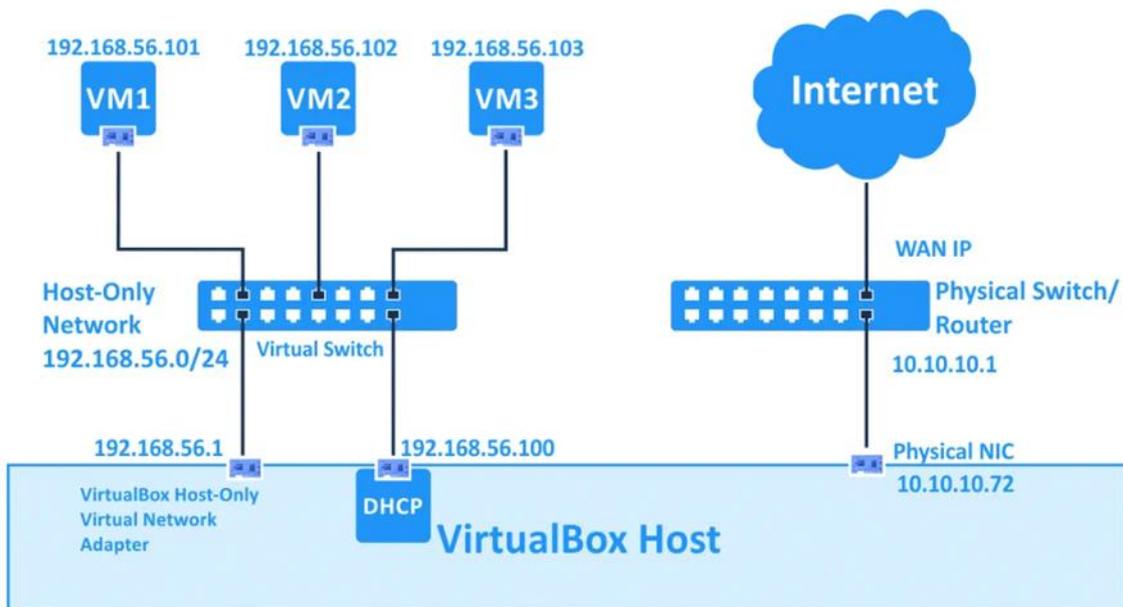


Figura 8: Host-Only [4]

Utilidades:

- Comunicación entre una VM y el host pero no con Internet.
- Entorno de desarrollo o pruebas donde se necesita aislamiento por seguridad.

Usos comunes:

- Entornos de desarrollo o pruebas donde varias VMs necesitan comunicarse entre sí y compartir recursos, pero sin riesgo de acceso externo.
- Simulación de redes privadas. [5]
- Esta configuración es parecida a la NAT network, salvo que cada VNIC tiene sus propias características. En este caso, un laboratorio de red así creado, mantiene su conectividad hacia Internet, permitiendo, por ejemplo, realizar instalaciones y actualizaciones de los programas.

2.4.4 Bridged

Este modo conecta las máquinas virtuales directamente a la red física del host, permitiéndoles obtener direcciones IP directamente del router o servidor DHCP. Para ello, el hipervisor coloca las VNIC al mismo nivel de la NIC, permitiendo que otros equipos puedan “ver” cada una de ellas de forma totalmente independiente. En la Figura 9 se ve un esquema de este tipo de red.

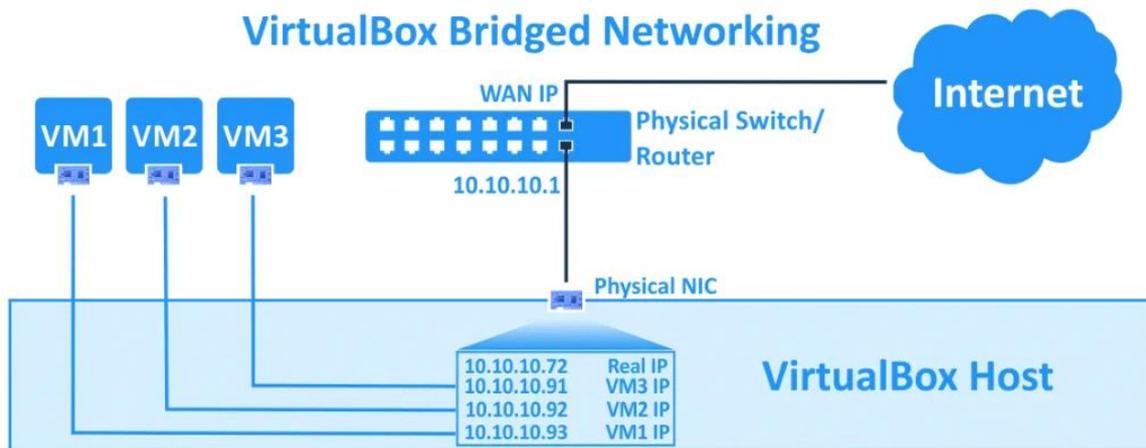


Figura 9: Bridged [4]

Utilidades:

- La red en modo puente conecta directamente las VMs a la red física a la que está conectado el host, como si fueran dispositivos físicos adicionales en esa red.
- Las VMs obtienen sus propias direcciones IP de la red física, lo que les permite interactuar con otros dispositivos en la misma red, como cualquier otro equipo físico.

Usos comunes:

- Configuración de un entorno de desarrollo o pruebas que requiere acceso completo a otros dispositivos en la misma red física. [5]

Esta configuración es adecuada, por ejemplo, para el despliegue de servicios, asociados cada uno con una determinada máquina virtual.

2.5 VIRTUALBOX

Tal como se ha indicado anteriormente, VirtualBox es un software de virtualización que permite la creación y gestión de máquinas virtuales. VirtualBox es un hipervisor de tipo 2, lo que significa que se ejecuta como una aplicación en un sistema operativo anfitrión y permite la creación de VMs que pueden ejecutar otros sistemas operativos.

Algunas características destacadas de VirtualBox son:

- Multiplataforma: Se puede instalar en diferentes sistemas operativos
- Multihuéspedes: Es capaz de virtualizar varios sistemas operativos
- Software libre
- Portabilidad: VirtualBox mantiene una funcionalidad similar en todas las plataformas soportadas. Además, las máquinas virtuales se pueden importar y exportar fácilmente utilizando el formato OVA.
- No siempre requiere virtualización asistida por hardware

- Guest Additions: Son paquetes de software que se instalan en los sistemas operativos invitados para mejorar el rendimiento y la comunicación con el anfitrión.
- Snapshots: Permiten guardar el estado actual de la máquina virtual y restaurarla a ese estado en cualquier momento.
- Crear grupos de máquinas virtuales
- Montaje de imágenes ISO: Se pueden montar imágenes ISO en las unidades de CD o DVD virtuales, eliminando la necesidad de usar medios físicos. [1]

VirtualBox dispone de una interfaz gráfica donde se muestran las máquinas virtuales instaladas a modo de lista y un panel de información donde se muestra aspectos de la configuración de la máquina virtual seleccionada, tal como se ve en la Figura 10.

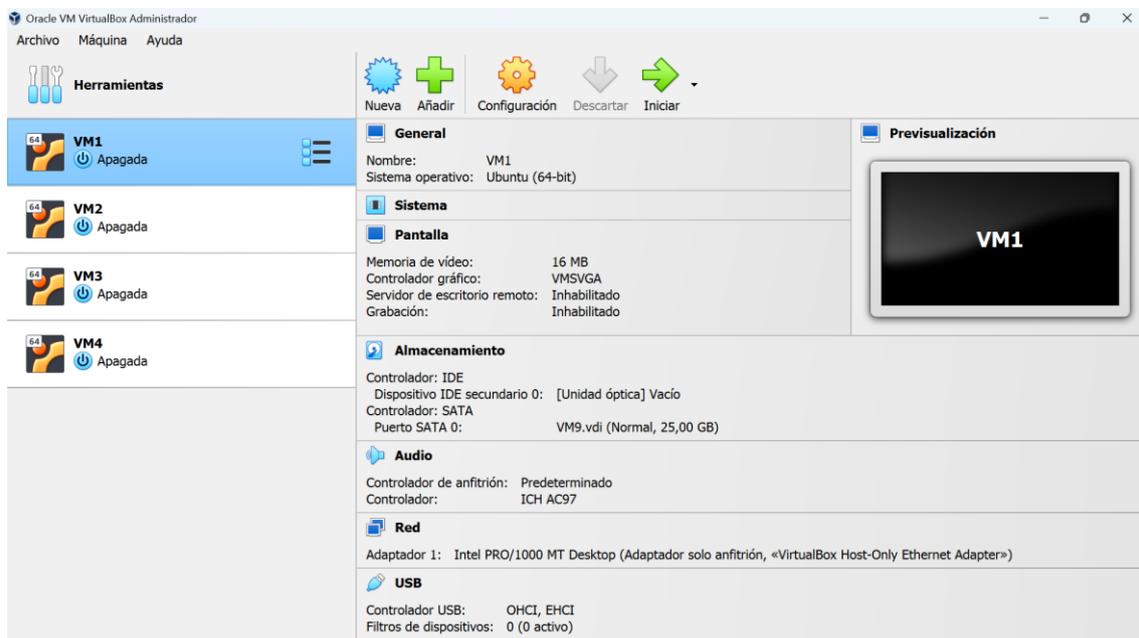


Figura 10: Interfaz de VirtualBox

En el panel de información se muestran datos generales de la máquina virtual como el nombre, el sistema operativo, configuración de la pantalla, del almacenamiento, los tipos de adaptadores de red, entre otros datos. La interfaz ofrece más información aparte de los detalles de las máquinas virtuales, como por ejemplo la capacidad de ver las trazas, tomar snapshots, ver la actividad, etc como se ve en la Figura 11:

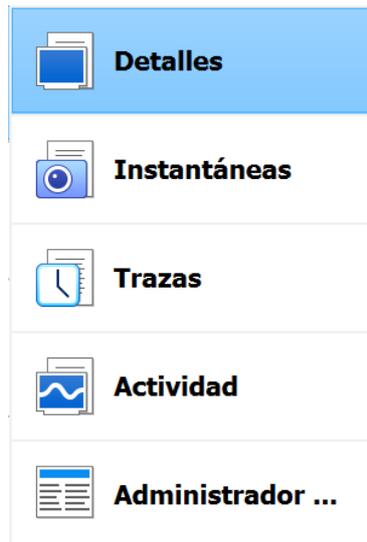


Figura 11: Opciones del menú de la máquina virtual de VirtualBox

También se muestra otra ventana con opciones más completas haciendo click derecho en la máquina virtual, tal como se ve en la Figura 12:



Figura 12: Menú avanzado de VirtualBox

Este menú muestra opciones para poder operar en la máquina virtual, como por ejemplo poder clonarla, eliminarla, iniciarla, apagarla, etc.

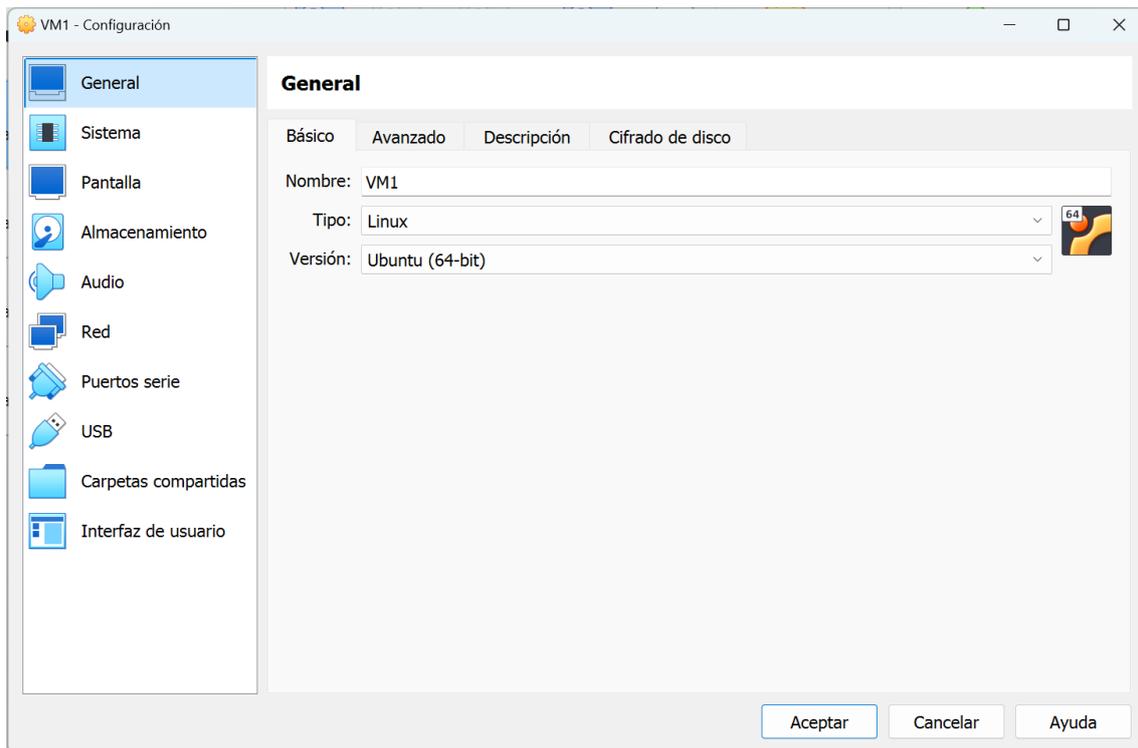


Figura 13: Opción “Configuración de VirtualBox”

En concreto, el menú proporciona una opción muy interesante, la opción de “Configuración” (Figura 13). Con esta opción se puede modificar la configuración de la máquina virtual como, por ejemplo, configurar los puertos series, los adaptadores de red, modificar la memoria, etc.

2.5.1 Deficiencias de la interfaz gráfica nativa

Aunque la interfaz que proporciona VirtualBox es bastante completa y no resulta excesivamente compleja a la hora de configurar máquinas virtuales, no deja de tener posibilidades de mejora, que se evidencian conforme el usuario adquiere experiencia en el manejo del entorno. Así por ejemplo, la configuración de las máquinas virtuales solamente puede realizarse de, una en una, a través del menú mostrado anteriormente en la Figura 13.

El menú de configuración implica la necesidad de navegar por diferentes submenús, a través de los cuales acceder tanto a los parámetros básicos, como a los parámetros avanzados de la máquina. En la figura anterior se observan hasta diez categorías de submenús, comenzando por “General”. A su vez, como es éste el caso, los parámetros se encuentran repartidos en hasta cuatro tarjetas, es decir, cuatro nuevos submenús. con algunos de los cuales, como condición fundamental antes de ser modificados, requieren la parada previa de la máquina en cuestión, lo que obliga a salir de todos los submenús, volver al menú principal, parar la máquina, y repetir el acceso al parámetro a modificar.

De hecho, la condición de realizar cambios de configuración “en caliente”, esto es, mientras la máquina está arrancada, puede resultar realmente frustrante, como por ejemplo cuando necesitamos modificar la configuración de los adaptadores de red para poder “mover” una máquina a otra red diferente de la original.

Por otro lado, hay otro conjunto de parámetros que no pueden ser consultados, como por ejemplo las direcciones de red (las IPs) de las máquinas activas, lo cual obliga a acceder directamente a la consola de dicha máquina para obtener dicho valor y poder acceder así desde cualquier otro cliente de terminales.

Por último, como el acceso a los parámetros de cada máquina se realiza de forma individual, con su correspondiente cascada de submenús, hace realmente difícil poder tener una visión general del escenario de máquinas activas y disponibles, más allá de conocer si están activas o no. Así, por ejemplo, para poder conocer qué máquinas están conectadas a la misma red, necesitamos acceder al menú de configuración de cada una de ellas para, al menos, conocer el adaptador de red, el cual nos indica solamente la red a la que se conectará, o a la que ya está conectada, según el caso.

Uno de los objetivos de este trabajo es precisamente crear una interfaz gráfica que se complementa con la interfaz nativa de VirtualBox, que incluya aquella información que la interfaz de VirtualBox no proporciona, como por ejemplo mostrar las direcciones IPs de las máquinas virtuales. Pero, además, poder visualizar la interconexión de las diferentes redes y máquinas virtuales, o dar la posibilidad de configurar la máquina virtual de manera más intuitiva que la que se muestra en la interfaz original.

2.5.2 VboxManage

Aunque la interfaz de VirtualBox no de acceso a todos los comandos básicos y avanzados del hipervisor, la aplicación sí que incluye una herramienta que da acceso a los mismos, denominada VBoxManage.

VBoxManage es la interfaz de línea de comandos (CLI) para VirtualBox, que permite a los usuarios gestionar máquinas virtuales y otros aspectos del entorno virtual directamente desde el terminal.

Una interfaz de línea de comandos (CLI) es una herramienta que permite a los usuarios interactuar con el software introduciendo comandos en un terminal o consola. En este proyecto, VBoxManage es la CLI utilizada para gestionar las máquinas virtuales de VirtualBox.

Los comandos que VboxManage maneja son variados y van de lo básico a una estructura más compleja, todos empiezan por “VboxManage”, luego se escribe el comando y luego el subcomando, se pueden añadir más atributos y combinaciones dependiendo de la acción que se quiera realizar.

Vamos a poner ejemplo este comando básico:

VBoxManage list vms

Muestra las VM y su UUID, tal como se ve en la Figura 14:

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\titis>VBoxManage list vms
"VM4" {ae929d6a-a594-4b5f-8807-d49acdf4e325}
"VM1" {0bf4a38d-6264-42cf-9e7e-ea65d5244179}
"VM2" {e894cc11-9b58-4b93-b60f-4561dd3b46d8}
"VM3" {7165148d-5a2b-40bf-a6b3-050186fe72d2}

C:\Users\titis>
```

Figura 14: Resultado del comando `VboxManage list vms`

Existen comandos con una sintaxis un poco más compleja, como por ejemplo el siguiente comando que configura el adaptador de red para que tenga una red Host-Only (Figura 15):

VBoxManage modifyvm "VM5" --nic1 hostonly --hostonlyadapter1 "VirtualBox Host-Only Ethernet Adapter"

```
C:\Users\titis>
C:\Users\titis>VBoxManage modifyvm "VM5" --nic1 hostonly --hostonlyadapter1 "VirtualBox Host-Only Ethernet Adapter"
```

Figura 15: Comando para configurar en el adaptador una red Host-Only

Con el comando **VBoxManage showvminfo "VM5" --details**, vemos que efectivamente la NIC1 tiene como red VirtualBox Host-Only Ethernet Adapter (Figura 16):

```
Default Frontend:
VM process priority: default
Storage Controllers:
#0: 'IDE', Type: PIIX4, Instance: 0, Ports: 2 (max 2), Bootable
  Port 1, Unit 0: Empty
#1: 'SATA', Type: IntelAhci, Instance: 0, Ports: 1 (max 30), Bootable
  Port 0, Unit 0: UUID: c68e2c6a-73a5-4ede-acdf-a2e18202d476
  Location: "C:\Users\titis\VirtualBox VMs\VM5\VM_New-disk001.vmdk"
NIC 1: MAC: 080027E03552, Attachment: Host-only Interface 'VirtualBox Host-Only Ethernet Adapter', Cable connected: on, Trace: off (file: none), Type: 82540EM, Reported speed: 0 Mbps, Boot priority: 0, Promisc Policy: deny, Bandwidth group: none
NIC 2: disabled
NIC 3: disabled
NIC 4: disabled
NIC 5: disabled
NIC 6: disabled
NIC 7: disabled
NIC 8:
```

Figura 16: Resultado de aplicar el comando de configurar la red

2.5.3 Python SDK para VirtualBox

Además de una aplicación funcional, y de una herramienta en modo comando, los desarrolladores de VirtualBox también han incluido herramientas de programación mediante las cuales los usuarios, y otros desarrolladores, puedan hacer uso de todo el potencial del programa. Comúnmente a estas herramientas se las denomina SDK (Software Development Kit), y suelen ser librerías de código que permiten acceder a otros programas, a todas o parte de las funcionalidades de la aplicación en cuestión.

En concreto, el Python SDK para VirtualBox permite escribir código para interactuar con VirtualBox desde scripts de Python. Con este SDK, los desarrolladores pueden crear, configurar, y gestionar máquinas virtuales y otros recursos de VirtualBox directamente desde el código Python. [8]

A diferencia del SDK, una API (Application Program Interface) es un conjunto de definiciones y protocolos que permiten la comunicación entre diferentes sistemas de software. Las APIs permiten que las aplicaciones interactúen entre sí, compartan datos y utilicen servicios sin necesidad de conocer los detalles internos de cómo están implementados esos servicios. [6]

Según lo anterior, VirtualBox ofrece una API nativa a través de VBoxManage, que es la herramienta de línea de comandos proporcionada por VirtualBox. Esta API nos permite controlar y gestionar las máquinas virtuales directamente a través de un terminal. Sin embargo, no resulta tan sencillo acceder a sus funcionalidades desde otra aplicación.

Para poder acceder a dichas funcionalidades es necesario utilizar un lenguaje de programación como intermediario, que virtualice a su vez la operativa del terminal para acceder a la API de VBoxManage y obtener los resultados correspondientes. Aquí es donde entra en juego Python SDK para VirtualBox, logrando hacer las siguientes tareas directamente desde una aplicación:

- **Controlar VM:** Se puede iniciar, apagar, cambiar el nombre y eliminar máquinas virtuales
- **Gestionar adaptadores de red:** permite habilitar o deshabilitar adaptadores, y cambiar entre diferentes tipos de redes (NAT o Host-Only).
- **Automatizar tareas:** Por ejemplo, la importación de máquinas virtuales desde archivos OVA o establecer una sesión SSH.

Aunque existen multitud de funciones incluidas en VBoxManage, éstas son las más utilizadas, por lo que han sido consideradas fundamentales en el desarrollo de este trabajo.

2.6 PYTHON

Python, creado por Guido van Rossum en 1991, es un lenguaje de programación de código abierto orientado a objetos conocido por su simplicidad y eficiencia, permitiendo a los programadores desarrollar conceptos en menos líneas de código en comparación con otros lenguajes. Python es ampliamente utilizado en desarrollo web y software, análisis de datos, inteligencia artificial, automatización y muchas otras áreas. Python sirve como lenguaje de punto de partida ya que enseña conceptos fundamentales de programación que pueden aplicarse a otros lenguajes de programación distintos.

Una de las características de Python es su modularidad. Los desarrolladores exponen constantemente nuevas librerías que pueden incluirse en desarrollos propios, y que

permiten realizar los trabajos más diversos y facilitando la programación del código haciéndolo menos voluminoso y más sencillo.

En este trabajo caben destacar los siguientes paquetes, que resultan fundamentales para la elaboración de una interfaz gráfica.

2.6.1 Tkinter

Tkinter es la biblioteca Python para crear interfaces gráficas de usuario (GUI). En este proyecto, Tkinter se utiliza para construir la interfaz gráfica que permite a los usuarios interactuar con las máquinas virtuales gestionadas por VirtualBox.

En nuestro código se ha usado para:

- **Ventanas y botones:** Creación de las ventanas principales, menús y botones para realizar acciones como encender, apagar, cambiar el nombre de la máquina virtual, habilitar o deshabilitar adaptadores de red, etc. Tal como se ve en la Figura 17:

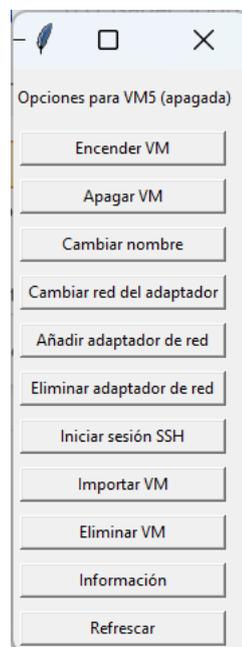


Figura 17: Ventana del menú hecha con Tkinter

- **Cuadros de diálogo:** Mostrar mensajes de advertencia, éxito (Figura 18) o error, además de permitir la entrada de texto (por ejemplo, pedir al usuario el nombre de una VM o credenciales SSH).

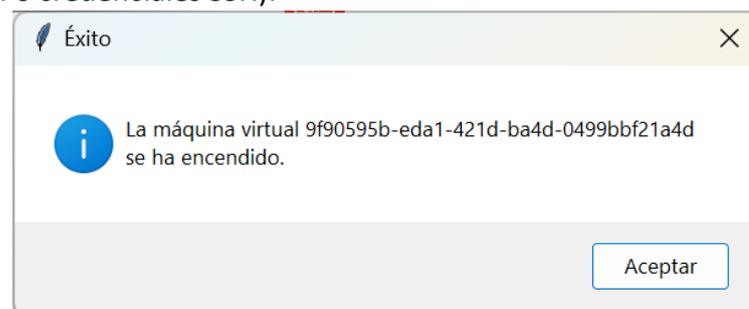


Figura 18: Cuadro de diálogo hecho con Tkinter

- **Selección de opciones:** Usar comboboxes para seleccionar redes (Figura 19), adaptadores de red, o IPs para conexiones SSH.

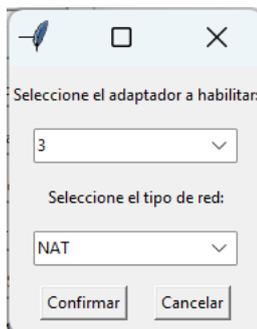


Figura 19: Combobox hecho con Tkinter

- **Disposición de elementos:** Organizar la interfaz de manera que los elementos (botones, campos de texto, etc.) estén correctamente colocados.

2.6.2 Matplotlib

Matplotlib es una librería de Python especializada en la creación de gráficos en dos dimensiones, permite crear y personalizar todo tipos de gráficos. En el desarrollo de este proyecto lo usé junto con otras librerías como NetworkX y para dibujar el mapa de red de las máquinas virtuales, dibujando las redes Host-Only como routers, la interfaz NAT como un switch y las máquinas virtuales siendo iconos de computadoras rojas o verdes según estén apagadas o encendidas enlazadas con las respectivas redes que tenga cada uno de sus adaptadores, creando así un mapa de las redes de VirtualBox.

En resumen, esta librería se ha usado para las siguientes funciones:

- **Visualización gráfica de la red:** Crear un gráfico que muestre las conexiones entre las máquinas virtuales y los adaptadores de red (Host-Only o NAT) con iconos representando routers, switches, máquinas encendidas y apagadas.
- **Renderizado del gráfico en la interfaz:** La función `FigureCanvasTkAgg` de Matplotlib permite incrustar gráficos generados en Matplotlib directamente en una ventana de Tkinter para integrarlo dentro de la interfaz gráfica.

2.6.3 NetworkX

NetworkX es una librería de Python para la creación, manipulación y análisis de la estructura y funciones de redes complejas. En este proyecto se ha usado para:

- **Creación de un grafo de la red:** Representa las redes y las máquinas virtuales como nodos en un grafo, con los adaptadores de red como enlaces que conectan los nodos.
- **Interacción:** Permite al usuario hacer clic en los nodos de la red visualizada para interactuar con las máquinas virtuales.

En este código, Matplotlib se ha empleado junto con NetworkX para visualizar la topología de la red de máquinas virtuales, trabajando en conjunto y complementándose

entre ellas para poder conseguir el mapa de redes de las máquinas virtuales que vemos en la Figura 20:

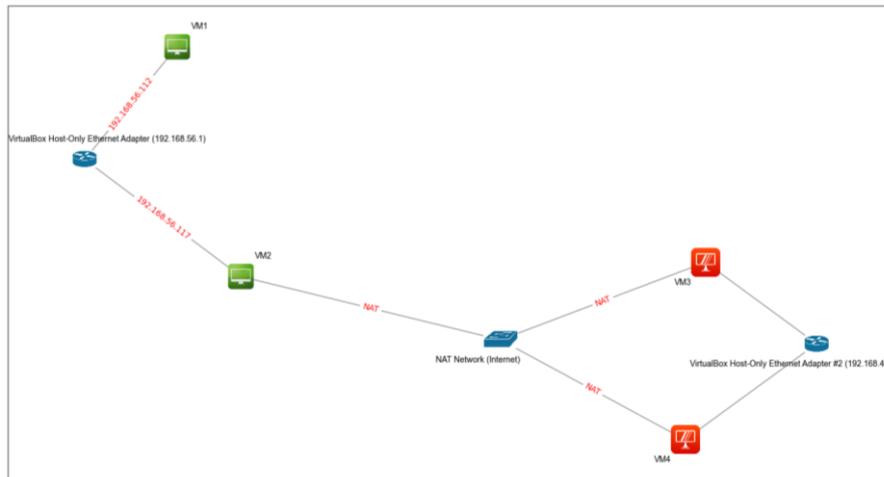


Figura 20: Mapa de redes hecho con Matplotlib y NetworkX

3 MARCO PRÁCTICO

A continuación, en este capítulo se expone la necesidad de crear una interfaz gráfica para gestionar VMs y redes virtuales, las razones del porqué he elegido VirtualBox para el desarrollo de este proyecto de fin de grado, o porqué utilizar Python como lenguaje de programación para este proyecto.

También se tratará las librerías y módulos necesarios para que funcione el código, las herramientas que se necesitan instalar y se explicarán las funciones de la interfaz gráfica.

3.1 UNA INTERFAZ GRÁFICA PARA LA GESTIÓN DE MÁQUINAS VIRTUALES

Cuando mi tutor del trabajo de fin de grado me propuso esta idea me fascinó desde el primer momento: nadie había creado una interfaz gráfica que mostrara un mapa de red y además que se pudiese configurar y gestionar máquinas virtuales.

Una interfaz de este tipo resulta ideal para una persona que no esté involucrada en el mundo de la virtualización y no sepa cómo configurar aspectos básicos de máquinas virtuales. Debe proporcionar un menú intuitivo con pasos fáciles de seguir a la hora de modificar una máquina virtual y permitir la interacción con la interfaz simplemente seleccionando las opciones que se van mostrando, sin necesidad de que el usuario utilice el teclado.

Por ejemplo, a la hora de añadir un nuevo adaptador de red a una máquina virtual, se necesita presentar una lista de los adaptadores de red deshabilitados. El usuario solo debe de escoger uno de los que se muestran y a continuación se le indica con un nuevo panel qué tipo de red desea en el adaptador de entre las opciones Host-Only o NAT. Si elige la red Host-Only, se abre otro panel con un listado de las redes disponibles para que se seleccione una de ellas. De esta forma, el usuario en ningún momento ha tenido que usar el teclado. Además de esta manera no hay posibilidad de error humano. Toda la lógica de esta interfaz sigue ese camino.

Como opción gráfica, resulta muy útil para entender los conceptos de virtualización, ya que esta interfaz puede plasmar en un mapa de red todos los elementos de VirtualBox, representando las redes Host-Only como routers, la interfaz NAT como un switch, las máquinas virtuales como computadores y las interconexiones de las máquinas virtuales con las distintas redes e IPs asignadas, tal como se ve en la Figura 14. Además, el mapa tiene que irse actualizando con todas las modificaciones que se van haciendo a tiempo real.

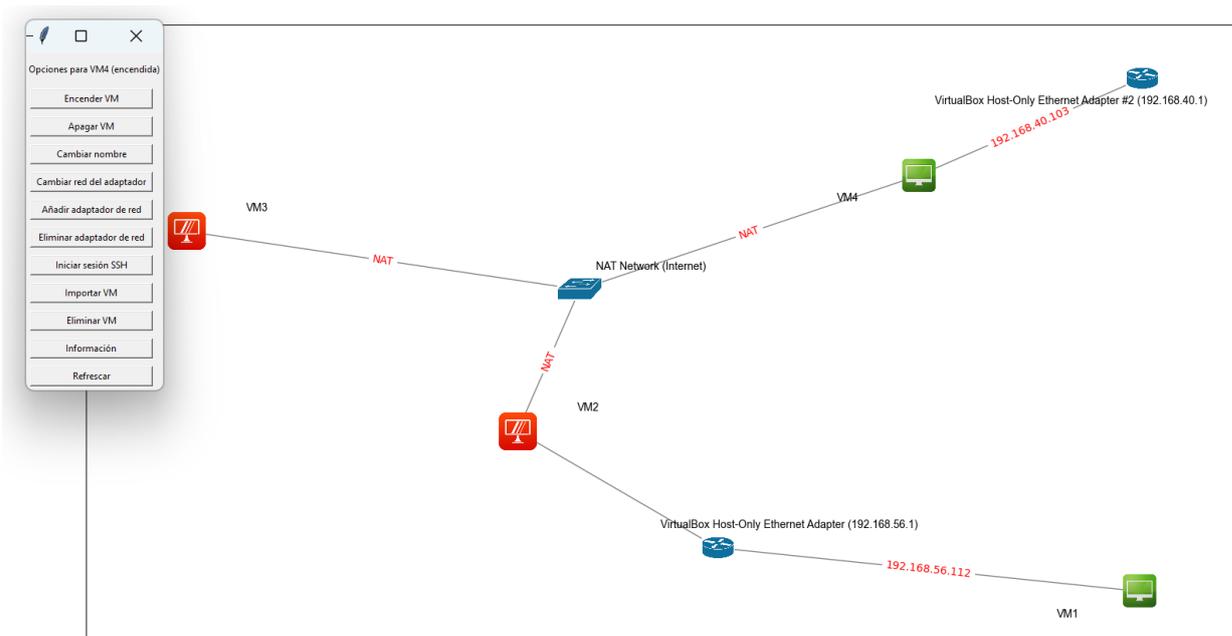


Figura 21: Interfaz gráfica

3.2 ELECCIÓN DE VIRTUALBOX

Para el desarrollo de este trabajo de fin de grado se ha elegido VirtualBox. VirtualBox es la aplicación más usada durante el transcurso de la carrera de Telecomunicaciones e incluso hasta en el instituto, por lo cual los alumnos suelen estar ya familiarizada con ella, es decir, que es una preferencia lógica ante otras aplicaciones que existen pero que no son tan usadas.

El uso de esta aplicación es a través de una interfaz gráfica básica, que incluye la mayor parte de operaciones de creación, configuración y manejo de máquinas virtuales. Además, VirtualBox tiene una interfaz de línea de comandos (CLI), VboxManage, necesaria para realizar operaciones más especializadas y, sobre todo, poder interactuar con las máquinas virtuales sin necesidad de activar la interfaz gráfica nativa. En pocas palabras, el VboxManage permite acceder a todas las funciones disponibles en la GUI de VirtualBox y además, a muchas más funciones que no están en la GUI. En el caso particular de este trabajo, se requiere acceder a las siguientes funcionalidades:

- Obtener las IPs de las redes y máquinas virtuales
- Obtener los adaptadores de red de cada máquina virtual
- Poder hacer configuraciones sobre la máquina virtual
 - Apagarla o encenderla
 - Cambiar su nombre
 - Añadir o eliminar adaptadores
 - Cambiar la red de cada adaptador
 - Mostrar información

- Establecer una sesión SSH para poder ejecutar comandos en la consola de la máquina virtual de manera remota
- Importar y eliminar máquinas virtuales

Si bien algunas de las acciones anteriores pueden estar incluidas en el menú de la GUI, una vez la máquina virtual ha sido creada, es prácticamente imposible acceder a dichas funcionalidades, obligando, en el mejor de los casos, apagar la máquina para poder hacer los cambios, o incluso obligando a volver a crear dicha máquina.

3.3 ELECCIÓN DE PYTHON

La interfaz desarrollada utiliza el lenguaje de programación Python. Es un lenguaje relativamente moderno, que no se incluye en los itinerarios formativos, por lo que, al inicio, es un verdadero reto. Al ser un lenguaje de programación orientado a objetos, se necesita tener claro todos los conceptos antes de lanzarse a la programación.

Pese a existir otros lenguajes más potentes, como C++ y Java, que también disponen de un SDK para VirtualBox, Python resulta la opción más simple, tanto por su sintaxis, como por el amplio repertorio de librerías y módulos, en continuo desarrollo y de acceso gratuito. Así, las tareas de creación de la interfaz gráfica, de los del menú de operaciones y del dibujo del grafo de las redes resulta hasta cierto punto, sencilla.

3.3.1 Librerías y módulos de código

Caben destacar las librerías Tkinter, Matplotlib y NetworkX, que son las librerías principales elegidas. Sin embargo, para el desarrollo de este código se han usado muchas más librerías y módulos.

- **Tkinter** es la biblioteca Python para crear interfaces gráficas de usuario (GUI). Se ha explicado anteriormente con más detalle en el punto 2.8.2.
- **Matplotlib** es una librería de Python especializada en la creación de gráficos en dos dimensiones, permite crear y personalizar todo tipos de gráficos. Se ha explicado anteriormente con más detalle en el punto 2.8.3.
- **NetworkX** es una librería de Python para la creación y manipulación de redes.
- **Os** es un módulo que proporciona las funciones para interactuar con el sistema operativo. [9]
- **Subprocess**: El módulo de Subprocess permite generar nuevos procesos, conectarse a sus canales de entrada/salida/error y obtener sus códigos de retorno. [10]
- **Paramiko** es una librería para establecer sesiones SSH que proporciona la funcionalidad tanto de cliente como de servidor. [11]
- **Pandas** es una librería de Python especializada en el manejo y análisis de estructuras de datos. Define nuevas estructuras de datos basadas en los arrays de la librería NumPy pero con nuevas funcionalidades, permite leer y escribir ficheros en formato CSV, Excel y bases de datos SQL y permite acceder a los datos

mediante índices o nombres para filas y columnas, entre otras características. [12]

- **NumPy** es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos. Incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación. [13]
- **Re** es módulo que proporciona operaciones de coincidencia de expresiones regulares. [14]
- **PIL** es una librería que permite la edición de imágenes directamente desde Python. Soporta una variedad de formatos, incluidos los más utilizados como GIF, JPEG y PNG. [15]
- **Time** Es un módulo que proporciona varias funciones relacionadas con el tiempo. [16]

Todos los módulos y librerías están en continuo desarrollo, por lo que las funcionalidades van aumentando y mejorando con cada actualización, lo cual hace de Python una opción muy potente para el desarrollo rápido de aplicaciones para múltiples usos.

3.4 PREPARACIÓN DEL ENTORNO DE DESARROLLO

Una vez seleccionado el lenguaje de programación, es necesario montar un entorno de desarrollo adecuado para la implementación de la interfaz gráfica deseada. Para ello es necesario contar con un ordenador que cuente, al menos, con una distribución operativa de VirtualBox y del entorno de desarrollo de Python.

3.4.1 VirtualBox

Lo primero que se debe instalar es VirtualBox, se puede descargar desde la página oficial <https://www.virtualbox.org/wiki/Downloads>. Para el desarrollo de este proyecto se ha usado la versión 7.0.14 de VirtualBox y como sistema operativo Windows 11, aunque se puede usar en cualquier sistema operativo ya que es multiplataforma, es decir, se puede usar también en IOS, Linux...

Una vez instalado VirtualBox, para poder realizar pruebas y ejemplos de aplicación de las utilidades desarrolladas, se deben instalar algunas máquinas virtuales por ejemplo, desde <https://ubuntu.com/download/desktop> se pueden instalar con SO de Linux, aunque cualquier otro tipo de sistema sería igualmente válido.

3.4.2 Python

Existen múltiples versiones de Python adaptadas a casi todos los sistemas operativos del mercado. Es tal su popularidad que incluso las distribuciones de Linux, como Ubuntu, lo incluyen ya en la instalación propia del sistema. En el caso de que el host sea Windows, es necesario acudir a la distribución de “Python en Windows”, que puede descargarse directamente de la página oficial de Python <https://www.python.org/downloads/> (en

este caso se recomienda utilizar el fichero EXE autoinstalable). Durante la instalación hay que asegurarse de seleccionar la opción “añadir Python al PATH”, para que pueda utilizarse desde cualquier ruta de llamada.

La gran ventaja del sistema autoinstalable es que durante la instalación no solo se incluye Python, sino que también todas las utilidades de configuración y gestión de los entornos de programación. Una de estas herramientas es precisamente PIP, la cual permite incluir las diferentes librerías que puedan ser necesarias para el desarrollo del proyecto. Aun así, se recomienda comprobar que ambas utilidades están ya disponibles ejecutando los siguientes comandos en la CMD:

python –version

pip –version

Tenga en cuenta que es posible utilizar versiones diferentes tanto de Python como de PIP incluso en el mismo proyecto. Esto puede llegar a ser un problema, ya que las diferencias entre las diferentes versiones pueden hacer que un mismo script pueda comportarse de formas muy diferentes. Es por ello que es importante hacer un seguimiento de los requerimientos de la aplicación tanto en el proceso de desarrollo, como para el momento de su despliegue, y así evitar comportamientos no deseados o incluso el bloqueo de la aplicación final.

3.4.3 Pywin32

Precisamente una de las primeras librerías que no vienen incluidas en la distribución original es la denominada pywin32. Esta librería es una extensión para Python que permite acceder a muchas de las APIs de Windows desde Python. Esto significa que puedes usar funciones y características de Windows directamente en tus programas de Python, como manipular archivos, interactuar con el registro de Windows, automatizar tareas de Windows, y más. Es el elemento fundamental para poder acceder a la API de Virtualbox sin necesidad de abrir una ventana de terminal de comandos.

Para realizar la instalación de la librería se debe ejecutar el siguiente comando en la CMD:

python -m pip install pywin32

Un problema que puede surgir es el siguiente: Si una vez instalado PIP, éste no es reconocido dentro de Python, es necesario hacer un upgrade, ejecutando el siguiente comando en la CMD:

Python -m ensurepip –upgrade

Si los problemas persisten, es recomendable consultar las correspondientes ayudas que aparecen en las páginas oficiales de cada distribución.

3.4.4 Pyvbox

Hasta aquí ya es posible lanzar comandos de VBoxManage tanto desde una ventana de comandos, como de un script de Python. Aunque para determinadas operaciones,

especialmente las de consulta de información, éste podría ser un método más que suficiente, en otros casos, especialmente aquellos comandos que interactúan con las máquinas o redes virtuales, es necesario especificar parámetros y recoger las correspondientes salidas del comando, lo cual puede resultar realmente complejo. Es por ello que existe una librería, denominada pyvbox, que actúa como interfaz de Python para VirtualBox, y que permite automatizar tareas como la creación, configuración, inicio y parada de máquinas virtuales, así como la gestión de sus recursos.

Los detalles de su instalación aparecen directamente en la referencia de su página oficial (<https://pypi.org/project/pyvbox/>), sin más que ejecutar:

pip install pyvbox

La librería Pyvbox es en realidad lo que se denomina un envoltorio (o wrapper) de Python para la SDK de VirtualBox, que simplifica enormemente su uso facilitando tareas como la automatización y la gestión de máquinas virtuales sin necesidad de profundizar en la complejidad de dicha SDK, pero sin evitar que sea necesaria su instalación.

3.4.5 SDK de VirtualBox

La SDK de VirtualBox es una colección de herramientas y APIs que permite a los desarrolladores interactuar con VirtualBox a nivel de sistema, proporcionando un control detallado sobre las máquinas virtuales y sus configuraciones. Requiere conocimientos de programación y configuración más avanzados.

El SDK incluye documentación y ejemplos que facilitan su uso con varios lenguajes de programación, como Python, Java, y C++. Esto permite integraciones más complejas y eficientes.

Para su instalación, primero es preciso descargar de la página de VirtualBox el SDK (<https://www.virtualbox.org/wiki/Downloads>), y descomprimirlo en un directorio fácilmente localizable. Una vez hecho, de nuevo en una ventana de comandos, tras situar el prompt en el directorio anterior, y dentro de él, en el directorio “installer”, se ejecuta el siguiente comando:

python vboxapisetup.py install

Como se observa, es una llamada a Python para ejecutar el script de instalación del SDK.

Aunque las funciones más importantes aparecen incluidas en Pyvbox, la SDK es precisamente la opción necesaria para acceder a comandos y funciones de más bajo nivel, no contempladas, como por ejemplo la configuración de todos los dispositivos virtuales, es decir, discos, red, USB, etc.

3.4.6 PyPI

La Python Package Index (PyPI) es una librería que permite a los desarrolladores encontrar, instalar y distribuir paquetes de Python creados por la comunidad. Es una herramienta esencial para gestionar dependencias y compartir código, facilitando la colaboración y el desarrollo de proyectos en Python.

Para su instalación basta con ejecutar el siguiente comando:

```
python -m pip install virtualbox
```

Aunque no es estrictamente necesaria, gracias a esta librería es posible distribuir la aplicación Python finalmente desarrollada sin necesidad de que el usuario necesite establecer a priori todo el entorno de Python, es decir, instalar todas las dependencias asociadas a las librerías utilizadas, como es el caso de tkinter, pandas, etc.

3.4.7 Otras librerías

Aunque para el uso de la aplicación final no resulte necesario, durante el proceso de desarrollo es necesario incluir dentro del entorno todas las librerías a utilizar. Tal como se indicó anteriormente, la interfaz gráfica desarrollada hace uso de Tkinter, Matplotlib, Nertworkx, Subprocess, Paramiko, Pandas y Numpy, entre otras.

Para llevar a cabo su instalación basta con utilizar la misma sintaxis que en los casos anteriores:

```
python -m pip install <xxxx>
```

Sin más que sustituir <xxxx> por el nombre de la librería en cuestión.

3.5 FUNCIONES DE LA INTERFAZ GRÁFICA

El código cuenta con 43 funciones, en este punto vamos a explicar para qué sirve cada una de ellas y sus interacciones con el resto de las funciones.

3.5.1 Funciones para pintar el grafo

Existen dos tipos de funciones, las de obtención de datos y las de representación:

3.5.1.1 *Funciones para obtener información*

El programa ejecuta una serie de funciones para extraer la información de VirtualBox como por ejemplo cuántas VMs hay, si están encendidas o apagadas, los adaptadores que tiene cada una, si tienen dirección IP, etc.

- `guardar_VBoxManage_list_hostonlyifs`: Obtiene y guarda la lista de interfaces de red "Host-Only" configuradas en VirtualBox. Es usada en `actualizar_interfaz` para obtener la lista de redes "Host-Only".
- `guardar_VMs`: Obtiene y guarda la lista de máquinas virtuales registradas en VirtualBox. Es usada en `actualizar_interfaz` para listar todas las VMs.
- `guardar_VM`: Obtiene información detallada sobre una VM específica, como el sistema operativo, estado, y adaptadores de red. Es usada en `actualizar_interfaz` para obtener detalles de VMs específicas.

- `guardar_RunningVMs`: Obtiene y guarda la lista de VMs que están encendidas. Es usada en `actualizar_interfaz` para listar las VMs que están en ejecución.
- `guardar_MACs`: Obtiene las direcciones MAC y los tipos de red de los adaptadores de una VM específica. Es usada en `iniciar_sesion_ssh`, `actualizar_interfaz`, `Dibuja_redes` para obtener las MACs asociadas con una VM.
- `obtener_IP`: Obtiene la dirección IP asignada a cada adaptador de red de una VM. Es usada en `iniciar_sesion_ssh`, `actualizar_interfaz` para obtener las IPs de las VMs.
- `obtener_vm_id`: Obtiene el ID de una VM a partir de su nombre. Es usada en varias funciones como `open_menu`, `cambiar_nombre`, `iniciar_sesion_ssh`, etc.

3.5.1.2 *Funciones para representar el mapa de redes y máquinas virtuales*

El programa tiene 2 funciones que se encargan de representar el mapa de redes y de VMs y además de tenerlo siempre actualizado por si ha habido alguna modificación sobre las VMs, como por ejemplo cambiarlas de red, cambiarles el número de adaptadores, el nombre de la VM, etc. Las funciones que lo componen son:

- `Dibuja_redes`: Genera un gráfico de la topología de red de las VMs y redes "Host-Only" y "NAT" en la interfaz gráfica. Es usada en `actualizar_interfaz` para dibujar la topología de red en la interfaz gráfica.
- `actualizar_interfaz`: Actualiza la interfaz gráfica, obteniendo la última información sobre las redes y las VMs, y redibuja la topología de red. Es usada en varias funciones, como `encender_vm`, `apagar_vm`, `cambiar_nombre_vm`, `habilitar_adaptador`, `deshabilitar_adaptador`, etc., para refrescar la interfaz después de realizar cambios.

3.5.2 Funciones para el menú

La interfaz gráfica dispone de un menú para poder interactuar con las VMs, además este se debe de actualizar por si la máquina sufre algún cambio en su estado (apagada o encendida) y por si se le cambia el nombre. Las funciones que abordan estas tareas son:

- `open_menu`: Abre una ventana con opciones de administración para una VM específica. Es llamada desde la interfaz gráfica para abrir el menú de opciones de una VM.

- `cerrar_ventana_menu`: Cierra una ventana de menú y elimina su referencia del diccionario de ventanas abiertas. Es usada en `open_menu` y en el protocolo de cierre de la ventana del menú.
- `obtener_nombre_y_estado_vm`: Obtiene el nombre y el estado actual de una VM. Es usada en `open_menu`, `cerrar_y_reabrir_menu`.

3.5.3 Funciones para la gestión de VMs

En este apartado se explicarán las funciones que se emplean para la gestión de las VMs. Se subdividen en tres grandes bloques:

3.5.3.1 *Obtener información de la VM*

Para poder ejercer alguna modificación sobre una VM, primero se deben obtener los datos de la VM como por ejemplo los adaptadores de red que tiene y sus redes. Además, se debe de obtener la información de otros elementos disponibles como por ejemplo los tipos de redes que existen. Las funciones que se encargan de extraer la información para este cometido son:

- `obtener_info_vm`: Obtiene información detallada de una VM, incluyendo sus adaptadores de red. Es usada en `mostrar_informacion` para mostrar la información de una VM.
- `mostrar_informacion`: Muestra información detallada de una VM. Es usada en `open_menu` para mostrar información desde el menú.
- `obtener_adaptadores_vm`: Obtiene la lista de adaptadores de red habilitados en una VM. Es usada en `seleccionar_adaptador`, `deshabilitar_adaptador`, `habilitar_adaptador`.
- `obtener_red_actual`: Obtiene la red actual asociada a un adaptador de red en una VM. Es usada en `cambiar_red` para determinar la red actual del adaptador.

3.5.3.2 *Operaciones sobre las VMs*

En este apartado se explican las funciones que se encargan de modificar algún elemento de la VM. Las funciones son:

- `cambiar_nombre_vm`: Cambia el nombre de una VM si está apagada. Es usada en `cambiar_nombre` para cambiar el nombre de la VM seleccionada.

- `cambiar_red_interfaz`: Cambia la red asociada a un adaptador de red específico de una VM. Es usada en `cambiar_red` para modificar la red de un adaptador seleccionado.
- `encender_vm`: Enciende una VM. Es usada en `open_menu` para encender la VM desde el menú.
- `apagar_vm`: Apaga una VM. Es usada en `open_menu` para apagar la VM desde el menú.
- `deshabilitar_adaptador`: Deshabilita un adaptador de red específico de una VM. Es usada en `open_menu` para deshabilitar adaptadores desde el menú.
- `confirmar_deshabilitar_adaptador`: Es una función interna dentro de `deshabilitar_adaptador` que realiza la acción de deshabilitar el adaptador seleccionado y actualizar la interfaz.
- `habilitar_adaptador`: Habilita un adaptador de red específico en una VM y lo asocia a una red. Es usada en `open_menu` para habilitar adaptadores desde el menú.
- `actualizar_opciones_red`: Es una función interna dentro de `habilitar_adaptador` que actualiza las opciones de red disponibles en función del tipo de red seleccionado (NAT o Host-Only).
- `confirmar_habilitar_adaptador`: Es una función interna dentro de `habilitar_adaptador` que realiza la acción de habilitar el adaptador de red con el tipo de red seleccionado y actualizar la interfaz.
- `habilitar_adaptador_vm`: Habilita un adaptador de red y lo asocia con un tipo de red. Es usada en `habilitar_adaptador` para realizar la acción de habilitación.
- `seleccionar_adaptador`: Muestra un cuadro de diálogo para seleccionar un adaptador de red. Es usada en `cambiar_red`, `habilitar_adaptador` para seleccionar adaptadores en las acciones correspondientes.
- `cambiar_nombre`: Cambia el nombre de una VM y actualiza la interfaz. Es usada en `open_menu` para cambiar el nombre desde el menú.
- `solicitar_nombre_vm`: Solicita un nuevo nombre para una VM mediante un cuadro de diálogo. Es usada en `cambiar_nombre`, `display_import_options`.

- `cambiar_red`: Cambia la red de un adaptador de red específico en una VM. Es usada en `open_menu` para cambiar la red desde el menú.
- `confirmar_adaptador`: Tiene como objetivo confirmar la selección de un adaptador de red. Es usada en `seleccionar_adaptador`.
- `importar_vm_con_nombre_personalizado`: Importa una VM desde un archivo OVA, asignándole un nombre personalizado. Es usada en `display_import_options` para ejecutar la importación con el nombre especificado.
- `display_import_options`: Muestra un cuadro de diálogo para seleccionar un archivo OVA y asignar un nuevo nombre a la VM importada. Es usada en `open_menu` para importar una VM desde el menú.
- `confirmar_import`: Es una función interna dentro de `display_import_options` que maneja la confirmación de la importación de la VM con el nombre personalizado.
- `borrar_vm`: Borra una VM, desasociando primero sus discos virtuales. Es usada en `confirmar_borrado_vm` para ejecutar la eliminación de la VM.
- `confirmar_borrado_vm`: Solicita confirmación para borrar una VM y ejecuta la eliminación si el usuario confirma. Es usada en `open_menu` para borrar una VM desde el menú.

3.5.3.3 *Funciones para la conexión SSH*

En este apartado se explican las funciones que se usan para establecer la sesión SSH de una VM. Las funciones son las siguientes:

- `ssh_interactive_session`: Establece una sesión SSH interactiva con una VM. Es usada en `iniciar_sesion_ssh` para iniciar la sesión SSH.
- `iniciar_sesion_ssh`: Inicia una sesión SSH con una VM seleccionada, permitiendo al usuario elegir la IP para la conexión. Es usada en `open_menu` para iniciar una sesión SSH desde el menú.
- `conectar_ssh`: Es una función interna dentro de `iniciar_sesion_ssh` que establece la conexión SSH con la IP seleccionada, solicitando las credenciales al usuario y luego llamando a `abrir_terminal_ssh`.
- `abrir_terminal_ssh`: Abre un terminal y establece una conexión SSH con la VM seleccionada. Es usada en `iniciar_sesion_ssh` para abrir el terminal SSH.

3.5.4 Funciones para el estado de VMs

En este apartado se exponen las funciones para conocer el estado de una VM, ya que es importante verificar si está apagada para poder realizar alguna modificación sobre la VM o si está encendida para establecer una sesión SSH. También se maneja el estado de aborto. Las funciones son las siguientes:

- `obtener_estado_vm`: Determina el estado actual de una VM (apagada, encendida, abortada, etc.). Es usada en `cambiar_nombre_vm`, `cambiar_red_interfaz`, `encender_vm`, `apagar_vm`, `confirmar_borrado_vm`, `cerrar_y_reabrir_menu`, `actualizar_interfaz`.
- `manejar_estado_abortado`: Muestra una advertencia si la VM está en estado abortado y obtiene la razón del aborto. Es usada en `apagar_vm` cuando la VM se encuentra en estado abortado.
- `obtener_razon_aborto`: Obtiene información del log de la VM para determinar por qué está en estado abortado. Es usada en `manejar_estado_abortado` para proporcionar detalles sobre el estado abortado.

3.6 FLUJO DE LA INTERFAZ

En este apartado vamos a abordar el flujo del código de la interfaz, explicando con diagramas de flujo el funcionamiento.

3.6.1 Obtención de la información de VirtualBox

En este punto vamos a tratar las funciones que se ejecutan en la inicialización del programa para extraer los datos de VirtualBox. Con estas funciones la interfaz gráfica muestra siempre la información actualizada sobre las máquinas virtuales y sus redes. El flujo se representa en el diagrama de la Figura 22.

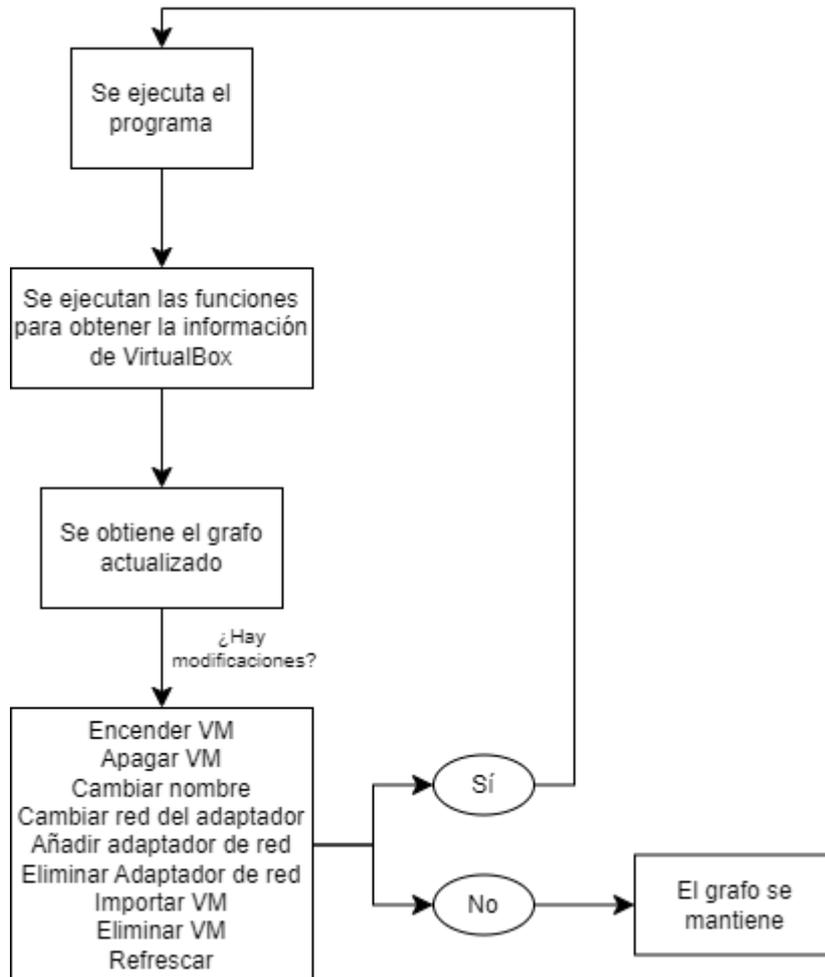


Figura 22: Obtención de información de VirtualBox

Las funciones que se usan para obtener la información de VirtualBox son:

1. **guardar_VBoxManage_list_hostonlyifs**: Esta función extrae información sobre las interfaces de red Host-Only configuradas en VirtualBox y la guarda en un archivo CSV. Devuelve un DataFrame con la información obtenida.
2. **guardar_VMs**: Esta función extrae una lista de todas las máquinas virtuales (VMs) registradas en VirtualBox, incluyendo sus nombres y UUIDs. La información se devuelve en un DataFrame.
3. **guardar_RunningVMs**: Esta función obtiene una lista de todas las máquinas virtuales que están actualmente en ejecución. Devuelve un DataFrame con los nombres y UUIDs de estas VMs.
4. **guardar_VM**: Esta función extrae información detallada de una máquina virtual específica, incluyendo su sistema operativo, estado, adaptadores de red, y otros detalles. Devuelve un DataFrame con la información de la VM.
5. **guardar_MACs**: Esta función obtiene las direcciones MAC de los adaptadores de red de una máquina virtual específica y devuelve un DataFrame con esta información.

6. **obtener_IP**: Esta función obtiene las direcciones IP asignadas a las MACs de las máquinas virtuales, basándose en la información obtenida de guardar_MACs. Devuelve un DataFrame con las IPs asociadas a las VMs.
7. **actualizar_interfaz**: Esta es la función central que llama a las funciones anteriores para extraer y organizar toda la información relevante sobre las VMs y las interfaces de red en VirtualBox. Esta función actualiza los datos y dibuja la red utilizando la información obtenida.

3.6.2 Modificación de la configuración o estado de una VM

La interfaz presenta varias opciones en el menú para poder interactuar con la VM bien sea modificando su estado, algún aspecto en la configuración o hasta establecer una conexión SSH. El menú se muestra en la Figura 23:

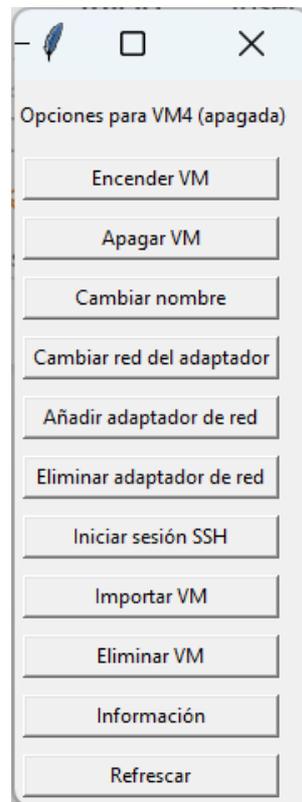


Figura 23: Menú de la interfaz

La interfaz presenta las siguientes opciones a la hora de interactuar con la configuración y el estado de una VM:

- Encender la VM
- Apagar la VM
- Cambiar el nombre
- Cambiar la red de un adaptador de red
- Añadir un nuevo adaptador de red
- Eliminar un adaptador de red

A la hora de realizar alguna modificación, el código por regla general antes de cualquier modificación lo primero que comprueba es si la máquina virtual está encendida o no, si es un requisito fundamental y, a partir de ahí, decide si proceder o no. Se basa en el flujo mostrado en la Figura 24.

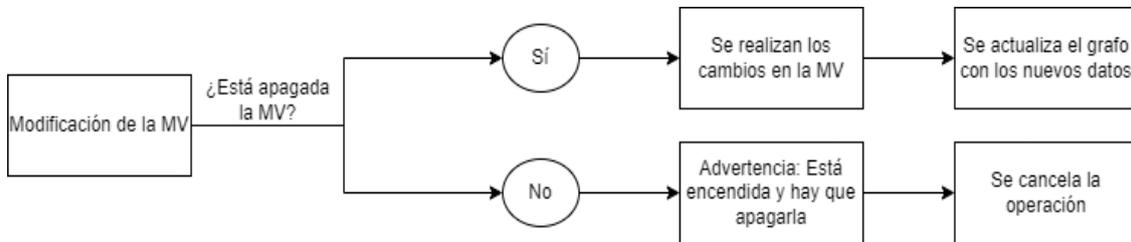


Figura 24: Flujo básico para la modificación de una VM

Sin embargo, éste es un requisito básico, y no para todos los casos es suficiente. En la mayoría de los casos se hacen comprobaciones específicas. Según la opción del menú que se seleccione se necesitaran más o menos comprobaciones, según la cantidad de parámetros que el usuario tenga que seleccionar. A continuación, se explican las diferentes opciones disponibles:

Cambiar red del adaptador de red

Si la máquina virtual tiene 2 adaptadores habilitados y se requiere cambiar la red de uno de ellos, al seleccionar la opción “Cambiar red del adaptador” se muestra un panel con los adaptadores de red habilitados, como se ve en la Figura 25. Una vez seleccionado el adaptador de red, se muestra otro panel para seleccionar la nueva red. La red que viene por defecto es la red configurada que tiene el adaptador actualmente, tal como se ve en la Figura 26.

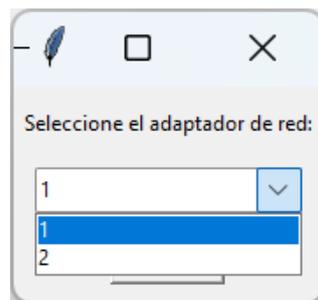


Figura 25: Panel de adaptadores habilitados

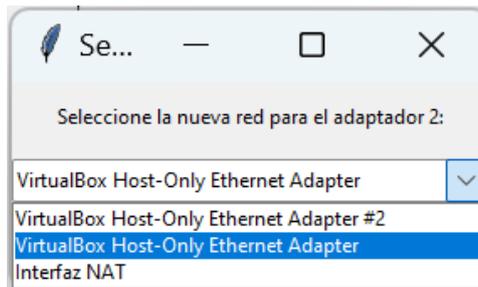


Figura 26: Panel de redes

Para realizar este cambio se requiere de una serie de comprobaciones, como por ejemplo, verificar si está apagada o no la VM: Se mostraría una advertencia, como se ve en la Figura 27, o que se seleccione una red distinta, como se ve en la Figura 28. Toda esta serie de comprobaciones se muestra en el diagrama de la Figura 29.

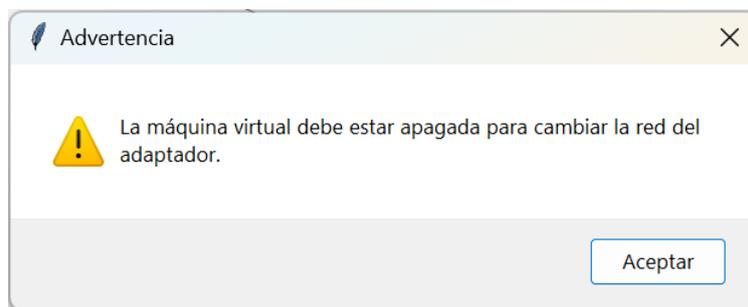


Figura 27: Advertencia de apagado para cambiar de red

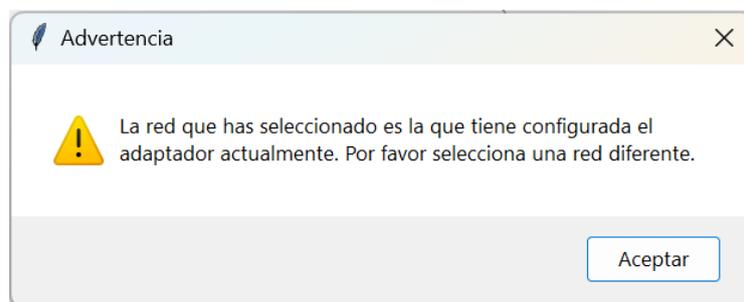


Figura 28: Advertencia de misma red seleccionada

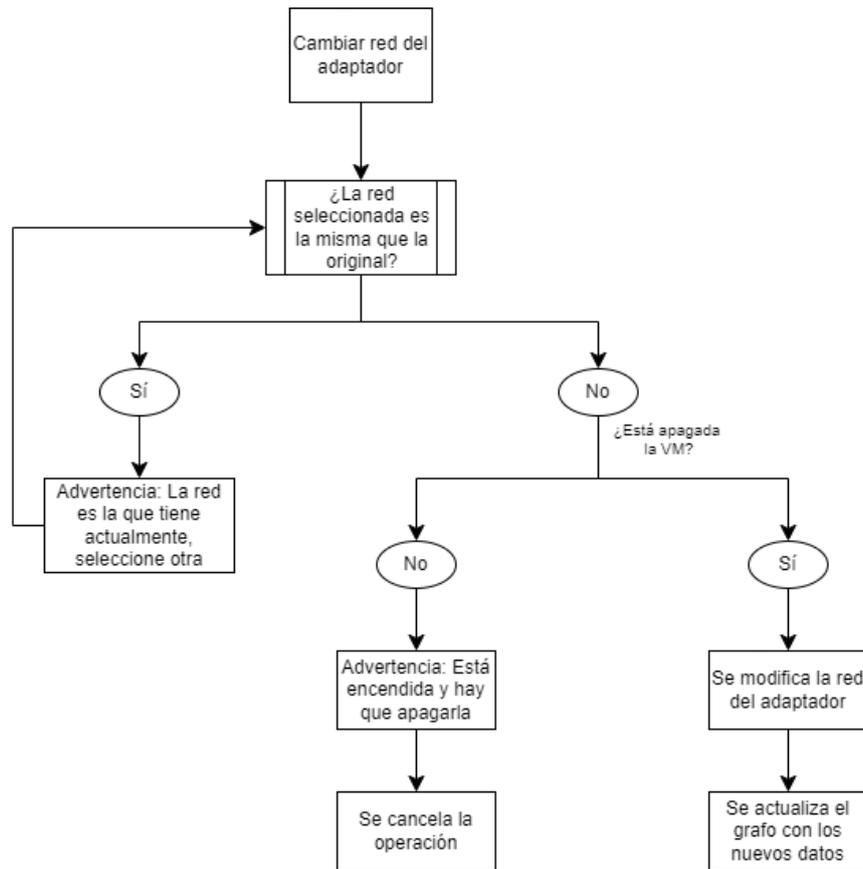


Figura 29: Diagrama de cambio de red del adaptador

Añadir o eliminar un adaptador de red

Para realizar estos cambios sobre la VM, el programa solo comprueba si la VM está encendida o no. Cuando se desea añadir un adaptador de red, los parámetros del panel, que se muestra en la Figura 30, son los que están disponibles. En el caso de querer eliminarlo, solo hay que seleccionar el adaptador que se muestra en la lista, como se ve en la Figura 31.

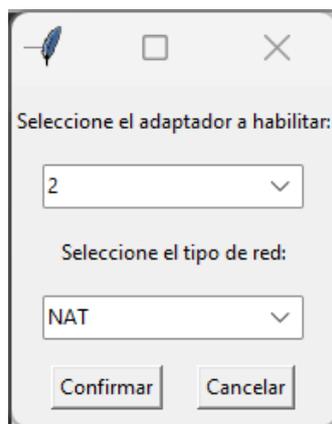


Figura 30: Panel para añadir un nuevo adaptador

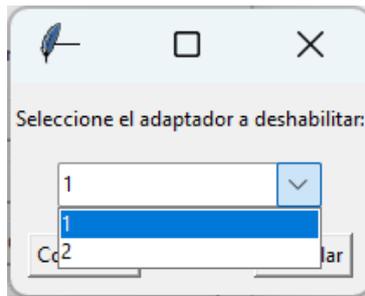


Figura 31: Adaptadores habilitados

En el caso de que se quiera añadir un nuevo adaptador, se muestran los adaptadores de red deshabilitados (Figura 32). Primero se selecciona uno de ellos, y luego se selecciona el tipo de red (Figura 33). Dependiendo si es NAT o Host-Only se ampliará el panel para seleccionar la red deseada (Figura 34).

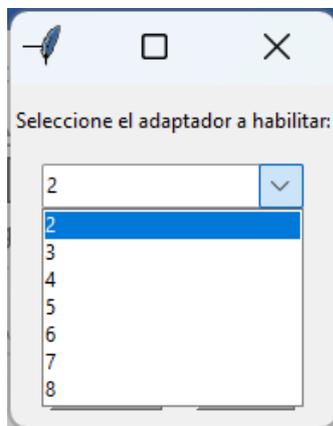


Figura 32: Adaptadores deshabilitados

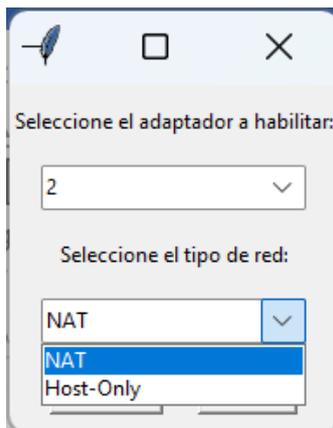


Figura 33: Tipos de red

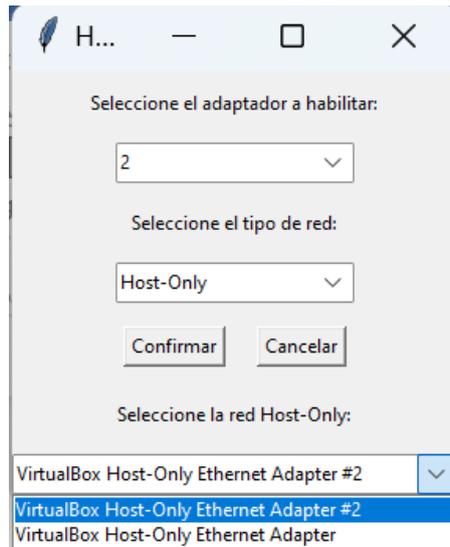


Figura 34: Selección de red Host-Only

Si la VM está encendida, se muestra una advertencia, tal como se ve en la Figura 35, en el caso de si se quiere añadir un adaptador, o bien la Figura 36 si, por el contrario, se quiere eliminar.

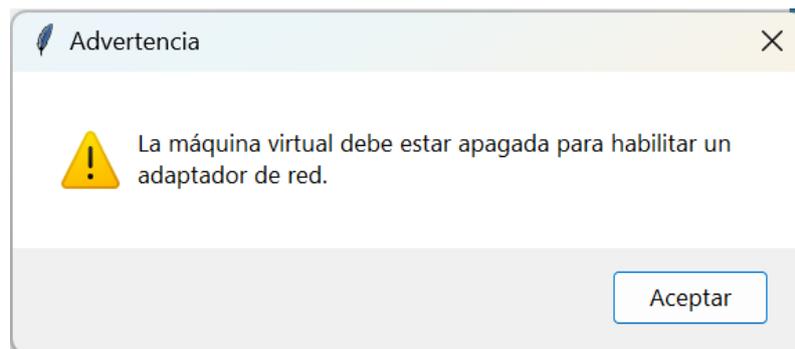


Figura 35: Advertencia de apagado para habilitar un adaptador

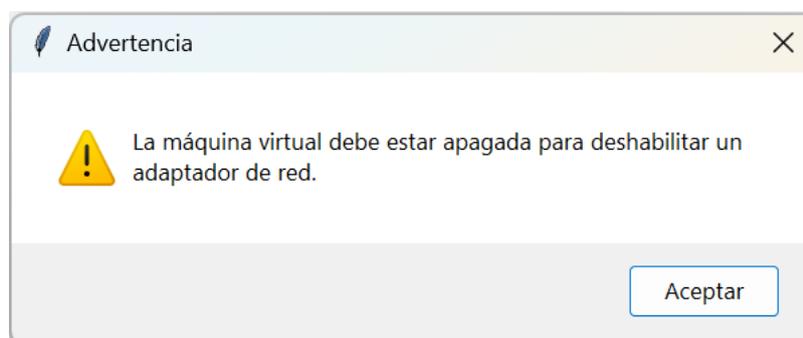


Figura 36: Advertencia de apagado para deshabilitar un adaptador

Las funciones de añadir o eliminar un adaptador de red siguen el flujo básico que se muestra en la Figura 37.

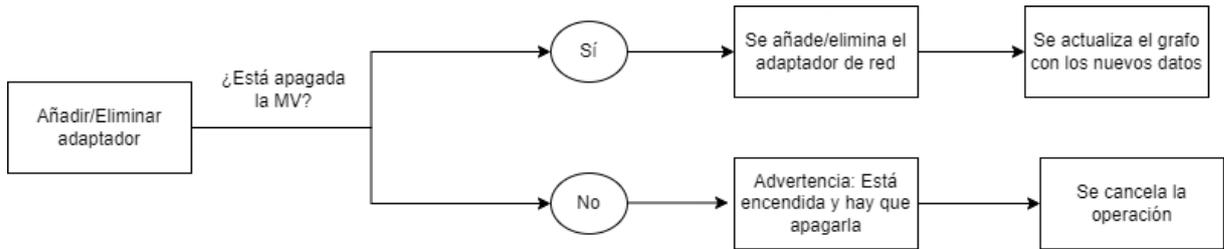


Figura 37: Diagrama para añadir o eliminar un adaptador

Cambiar nombre de la VM

Para esta función se muestra un panel para que el usuario escriba el nuevo nombre de la VM, como se ve en la Figura 38.

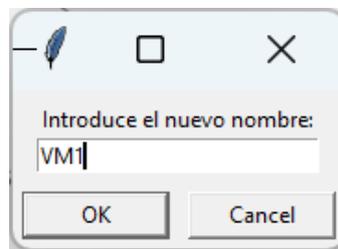


Figura 38: Panel para escribir el nuevo nombre

En este caso, también se verifica únicamente si la máquina virtual está encendida o apagada, ya que el único parámetro que necesita es el nuevo nombre. Si está apagada se cancelará la operación con un mensaje de advertencia, como se ve en la Figura 39.

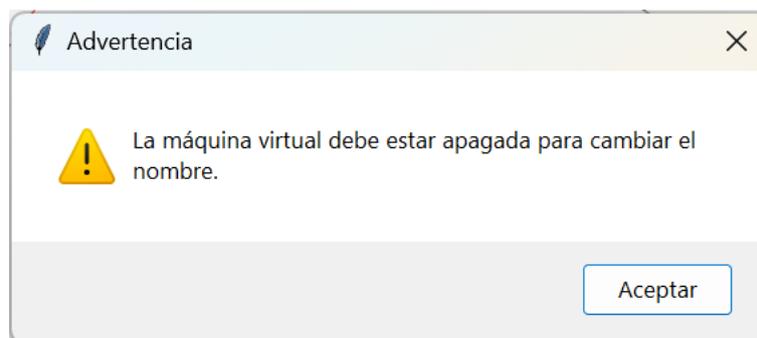


Figura 39: Advertencia de apagado para cambiar el nombre de una VM

El flujo para cambiar el nombre de la VM se muestra en la Figura 40.

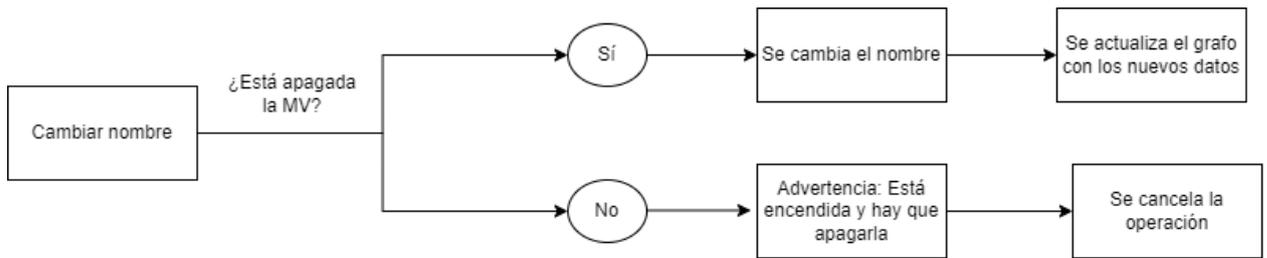


Figura 40: Diagrama para cambiar el nombre de una VM

Encender o apagar una VM

En ambas funciones simplemente se verifica si una VM está encendida o apagada, mostrando un mensaje de advertencia. Si se intenta encender una VM que ya está encendida es el caso de la Figura 41, mientras que si lo que se intenta es apagar una VM que ya está apagada sería el caso del Figura 42. El flujo de ambos procesos se muestra en la Figura 43.

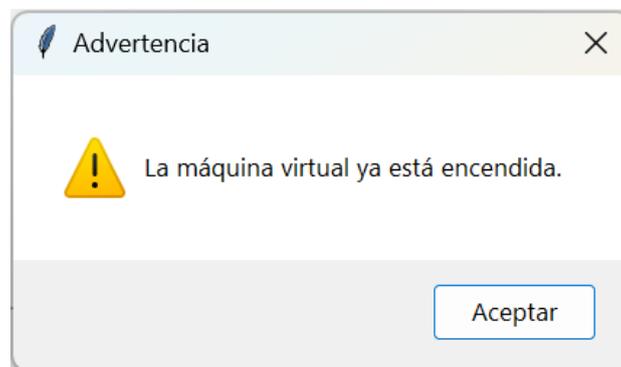


Figura 41: Advertencia de VM encendida

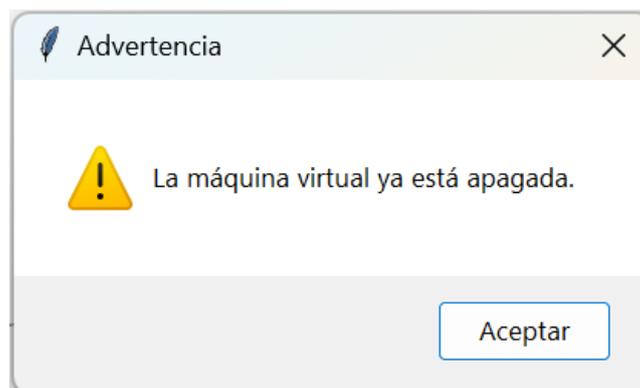


Figura 42: Advertencia de VM apagada

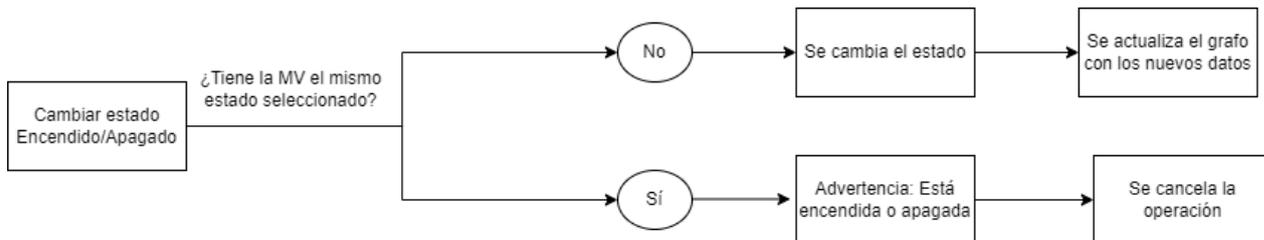


Figura 43: Diagrama de cambio de estado encendido o apagado

3.6.3 Establecer conexión SSH

En esta función, que también se proporciona en el menú gráfico, se establece una sesión SSH con la VM. Para ello, primero comprueba si está encendida o no, en cuyo caso se muestra una advertencia como se ve en la Figura 44.

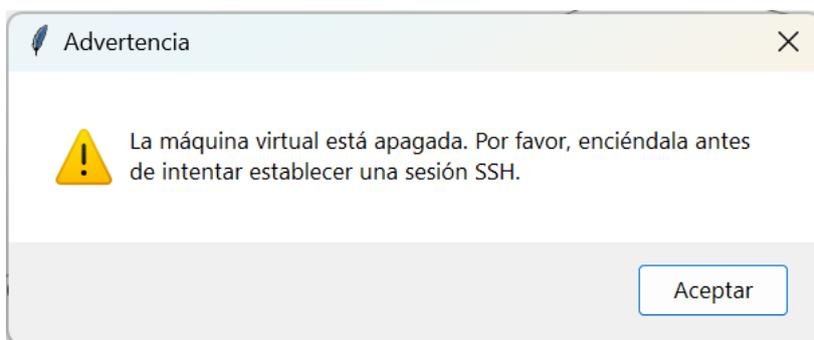


Figura 44: Advertencia de apagado para establecer una sesión SSH

Si la VM está encendida, se muestra un panel donde se puede seleccionar la IP de la dirección de red con la que se quiere establecer la conexión (Figura 45). En principio, cualquiera de las direcciones listadas es válida para establecer dicha conexión, aunque dependerá de la configuración final de cada red en cuestión.

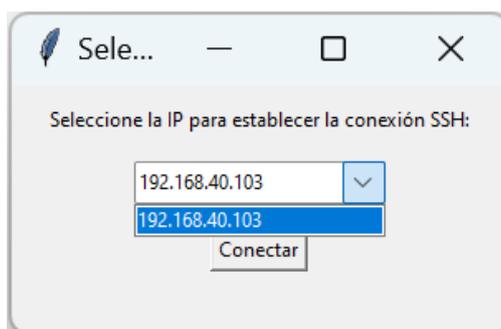


Figura 45: Selección de IP para establecer la sesión SSH

A continuación, se pide el usuario y contraseña de la VM, como se ve en la Figura 46 y la Figura 47, tras lo cual se abre una nueva terminal, en la cual se vuelve a pedir la contraseña (Figura 48).

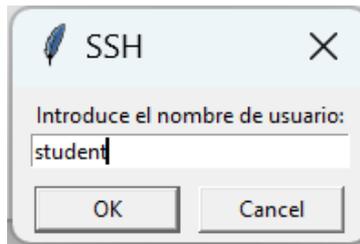


Figura 46: Panel para el usuario

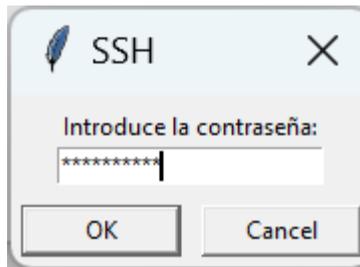


Figura 47: Panel para la contraseña

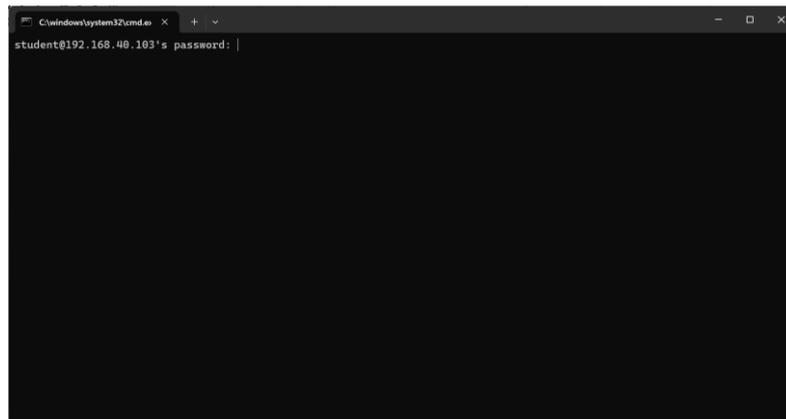


Figura 48: Repetición de la contraseña en el terminal

Si se han introducido bien las credenciales, estará listo para poder manejar la VM de manera remota (Figura 49). Si por el contrario no se han introducido bien las credenciales, fallará la conexión (Figura 50). El flujo se muestra en la Figura 51.

```
student@student: ~  
student@192.168.40.103's password:  
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.15.0-78-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
319 updates can be applied immediately.  
214 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Your Hardware Enablement Stack (HWE) is supported until April 2025.  
Last login: Thu Aug 15 23:28:07 2024 from 192.168.40.1  
student@student:~$
```

Figura 49: Sesión establecida correctamente

```
C:\windows\system32\cmd.exe  
UsuarioIncorrecto@192.168.40.103's password:  
Permission denied, please try again.  
UsuarioIncorrecto@192.168.40.103's password:  
Permission denied, please try again.  
UsuarioIncorrecto@192.168.40.103's password:  
UsuarioIncorrecto@192.168.40.103: Permission denied (publickey,password).  
C:\Users\titis\OneDrive\Documentos\Universidad\TFG>
```

Figura 50: Fallo de conexión por permiso denegado

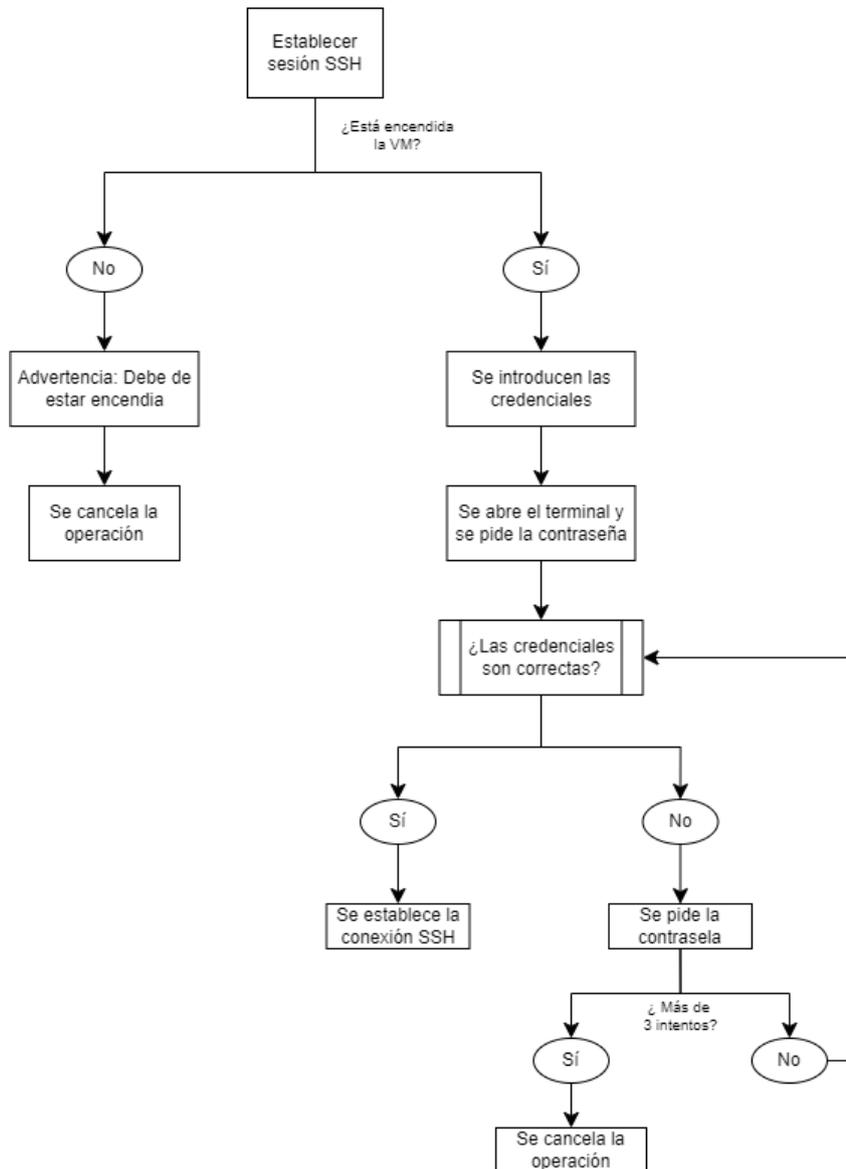


Figura 51: Diagrama para establecer una sesión SSH

3.6.4 Importar o eliminar una VM

La función para importar una VM permite crear una máquina virtual nueva de una manera muy sencilla: Para ello se abre una ventana para poder buscar el archivo OVA en nuestro equipo, y así poderlo instalarlo, tal como se ve en la Figura 52.

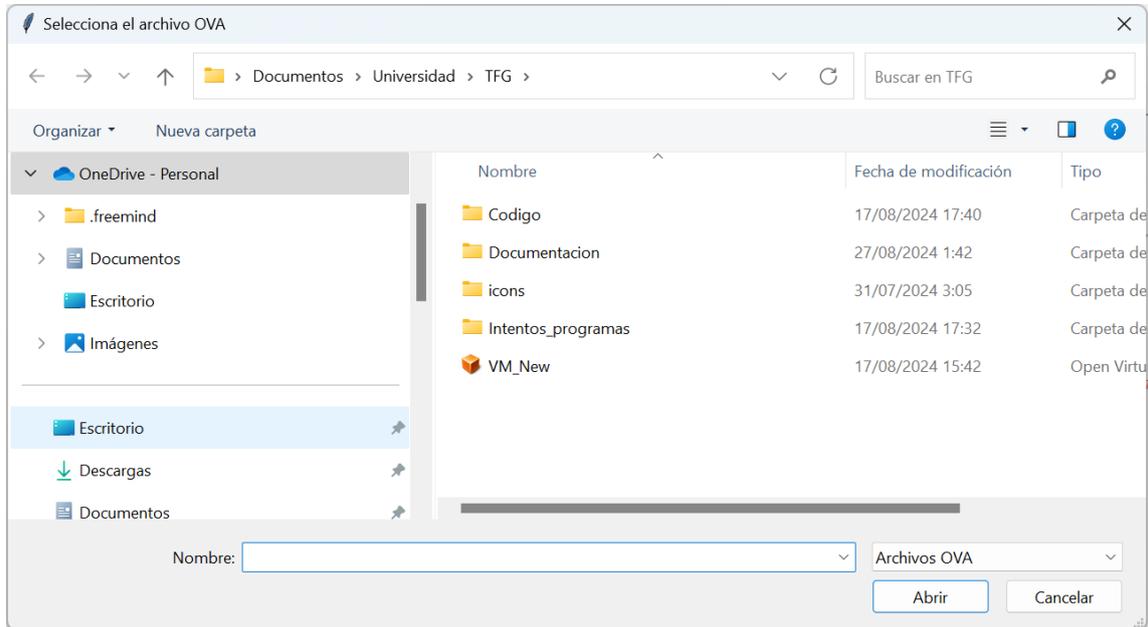


Figura 52: Ventana para la búsqueda del archivo OVA

Además, la interfaz sólo pide como parámetro el nombre de la nueva VM, en otro diálogo como el de la Figura 53. Por defecto, la nueva VM tiene un adaptador de red habilitado de tipo NAT, y aparecerá el grafo de redes sin más, tal como se muestra en la Figura 54. En el caso de que se quiera cambiar la red o añadir más adaptadores, se puede hacer seleccionando directamente las opciones del menú de la nueva VM.

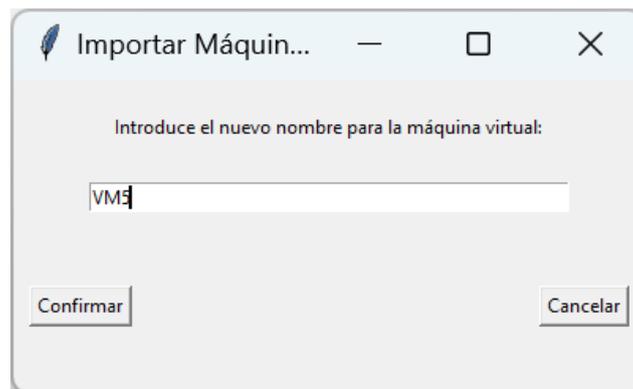


Figura 53: Panel para el nombre de la VM importada

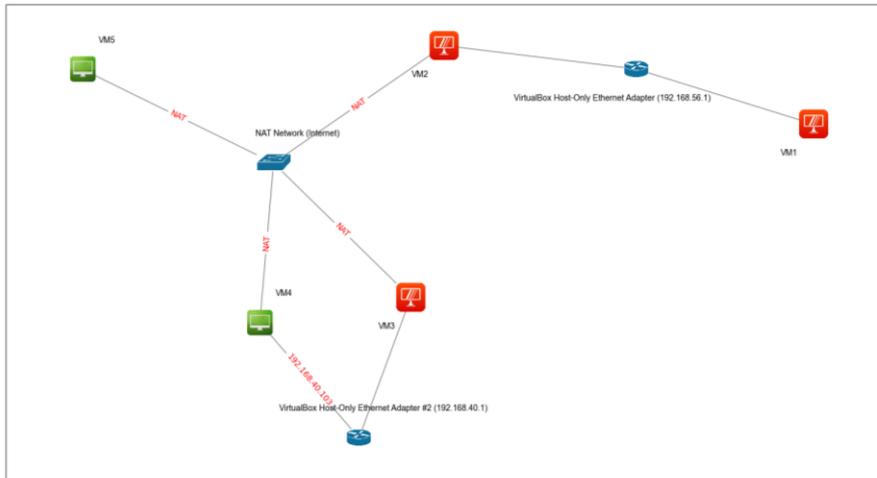


Figura 54: Grafo de red con la nueva VM

El flujo para importar una VM se muestra a continuación, en la Figura 55.



Figura 55: Diagrama para importar una VM

En el caso contrario, es decir, para eliminar una VM, sólo se debe seleccionar la opción del menú para eliminarla, aunque ésta deberá de estar apagada primero. De lo contrario, sale un mensaje de advertencia como se ve en la Figura 56. En la Figura 57 se ve el flujo para eliminar una VM.

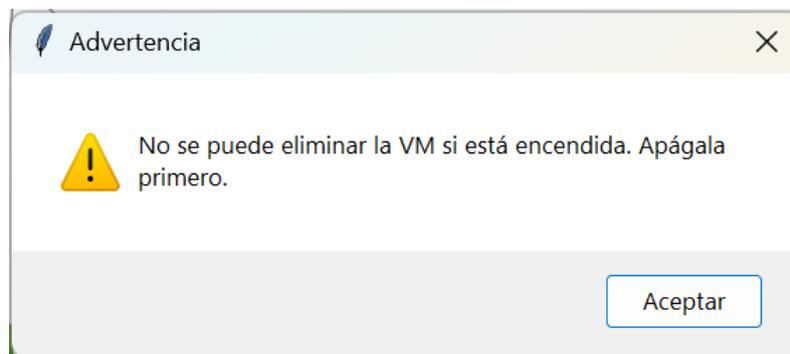


Figura 56: Advertencia de apagado para eliminar una VM

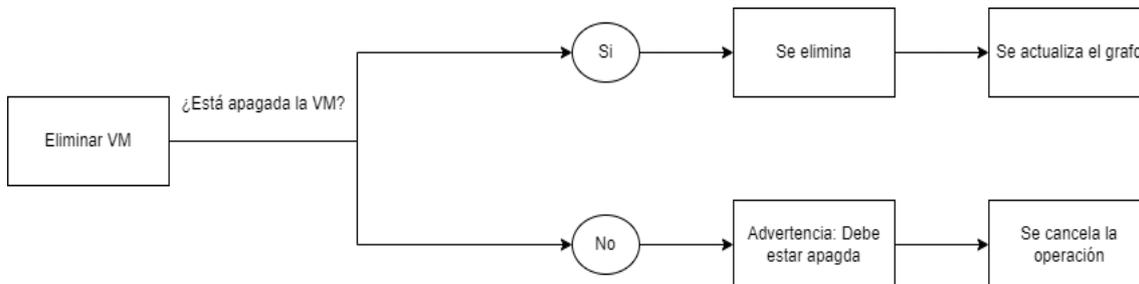


Figura 57: Diagrama para eliminar una VM

3.6.5 Mostrar información de la VM

Esta opción del menú muestra la información de los adaptadores de red habilitados tales como la dirección MAC, el tipo de red que tiene el adaptador, el tipo de adaptador... En la Figura 58 se muestra el panel de información.

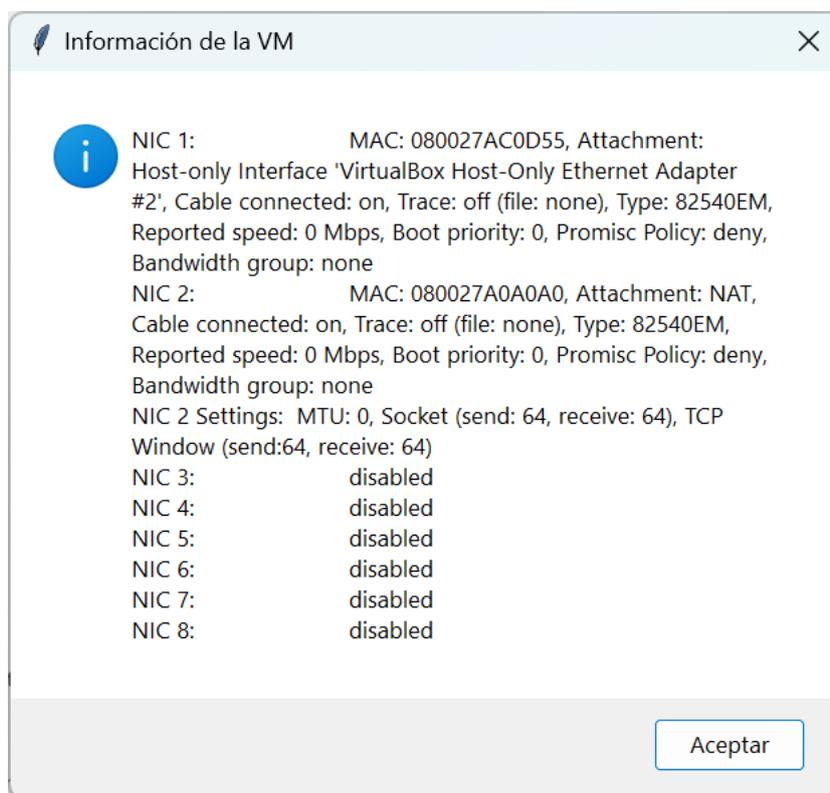


Figura 58: Panel de información

3.6.6 Refrescar

Esta última opción del menú permite actualizar manualmente el grafo de red, sin necesidad de que se haga ninguna modificación. En ciertos momentos el grafo puede aparecer de una manera poco clara o liosa. Mediante esta opción se fuerza a que se vuelva a generar una nueva representación.

4 CONCLUSIONES, APLICACIONES Y LÍNEAS FUTURAS

En esta parte del trabajo se exponen las principales conclusiones obtenidas tras el desarrollo completo de la interfaz gráfica, así como las posibles aplicaciones que se le puede dar a esta herramienta, así como las mejoras que se podrían hacer en futuros estudios.

4.1 CONCLUSIONES

Actualmente no existe ninguna interfaz gráfica para VirtualBox que muestre el mapa de red de sus máquinas virtuales y que, además, estas puedan ser gestionadas por el usuario.

La interfaz aquí propuesta, además de novedosa, cubre una necesidad de los usuarios de máquinas virtuales que, conforme aumentan en número, convierten la gestión de las mismas en un verdadero suplicio, lleno de continuos cambios entre ventanas o, en el peor de los casos, secuencias interminables de comandos con infinidad de parámetros.

Pese a que VirtualBox incluye en sus distribuciones paquetes de herramientas necesarias, como es la SDK (Software DevelopmentKit) y su correspondiente API (Application Program Interface), su uso se ha limitado tradicionalmente al acceso en modo terminal o de ventana de comandos, es decir, evitando cualquier representación gráfica. En este trabajo, aunque se basa en el lenguaje de programación Python y sus amplias librerías, se hace uso intensivo de la galería de comandos de VboxManage, la API nativa de VirtualBox, y las funciones integradas en la SDK de VirtualBox.

La interfaz resultante es capaz de plasmar el estado de las máquinas virtuales instaladas por el usuario, en un mapa de red detallado y, además, aportando un menú sencillo e intuitivo, que permite realizar las principales operaciones de gestión de cada una de las máquinas virtuales. Encender o apagar máquinas, configurar sus adaptadores de red o cambiar el nombre, crear nuevas máquinas, así como eliminarlas, son algunos de los ejemplos de funciones que ahora están integradas en una única ventana, con un grafo descriptivo tanto del estado de las mismas, como de la interconexión existente entre cada una de ellas a través de las redes virtuales configuradas en VirtualBox.

Por último, esta propuesta ha demostrado también la posibilidad de incluir funciones adicionales avanzadas que no son nativas de VirtualBox, como es el caso de la apertura de ventanas de terminal de las máquinas activas. Hasta ahora, a través de la interfaz nativa de VirtualBox solo es posible acceder a una ventana de terminal, bastante limitada según qué caso de sistema operativo se trate. Para abrir ventanas de terminal adicionales era necesario abrir aplicaciones cliente independientes, para lo cual antes era necesario conocer, al menos, la dirección IP de la máquina. Esta información que solo es posible obtener desde el terminal nativo, ya que VirtualBox no devuelve dicho valor en ninguno de sus menús por defecto, se recoge automáticamente del grafo de red, simplificando al extremo la operación completa.

4.2 APLICACIONES

La herramienta desarrollada en este proyecto simplifica enormemente la realización de operaciones rutinarias, normalmente de gestión, sobre las máquinas virtuales instaladas. Gracias a su representación gráfica de las VM y sus conectividades, así como de los menús y paneles, cualquier persona, incluso sin conocimientos básicos de VirtualBox, puede empezar a trabajar con redes de máquinas virtuales.

Desde el punto de vista del entorno educativo, esta herramienta puede ser un método de aprendizaje básico, para usuarios que quieren conocer de primera mano los conceptos de virtualización, tanto de máquinas como de redes. Pero, por otro lado, también puede convertirse en una herramienta fundamental para el despliegue y gestión de laboratorios virtuales, y las correspondientes prácticas de laboratorio, fundamentales en las titulaciones de ingenierías.

4.3 LÍNEAS FUTURAS

Aunque se hayan cumplido todos los objetivos planteados en el trabajo, una vez se prueba la herramienta a conciencia, siempre se acaba encontrando funcionalidades extra que podrían ser incorporadas en futuras versiones.

Una posible mejora es la incorporación de más tipos de redes virtuales, no solo los adaptadores de red Host-Only y NAT. VirtualBox incluye las redes de tipo Bridged y las NAT Network. Además de no ser muy utilizadas, plantean un pequeño reto en cuanto a la forma de representarlas en el entorno gráfico existente, por lo que, incluirlas, puede hacer revisar todo el sistema de representación gráfica.

Por otro lado, los menús nativos de VirtualBox incluyen muchas más funciones que no han sido implementadas en esta herramienta, simplemente por una cuestión de tiempo y prioridades. Las nuevas versiones de la aplicación pueden ir incluyendo dichas funcionalidades, poder gestionar las redes, es decir, poder crearlas, eliminarlas, cambiarles el nombre, asignarles el direccionamiento y configurar su servidor DHCP.

La complejidad de algunas funciones, por la cantidad de parámetros que deben incluirse, también ha sido un factor importante a la hora de elegir aquellas que se ha decidido implementar. Es el caso de la importación de una máquina a través del fichero OVA, que actualmente en este proyecto se dejan por defecto. De la misma forma, la creación de máquinas virtuales desde cero es otra funcionalidad que podría incorporarse en futuras versiones.

5 BIBLIOGRAFÍA Y REFERENCIAS

- [1] Periañez, Fco.: "Tutorial de VirtualBox". Recuperado en Sept. 2024, de <https://www.fpgenred.es/VirtualBox/index.html>
- [2] "El concepto de la virtualización". Recuperado en Agost. 2024, de <https://www.redhat.com/es/topics/virtualization/>
- [3] Romero, Andrés.: "¿Qué es sistema operativo?". Recuperado en 2022, de <https://www.goconqr.com/slide/38409872/tipos-de-sistemas-operativos>
- [4] Equipo NAKIVO.: "Ajustes de red VirtualBox: Guía completa". Recuperado en Mayo 2023, de <https://www.nakivo.com/es/blog/virtualbox-network-setting-guide/>
- [5] "Diferencias entre el modo Bridged, Host Only y NAT". Recuperado en Oct. 2014, de <https://brahimklamchiouat.blogspot.com/2014/10/diferencias-entre-el-modo-bridged-host.html>
- [6] Fernández, Yúbal.: "API: qué es y para qué sirve". Recuperado en Agost. 2019, de <https://www.xataka.com/basics/api-que-sirve>
- [7] "VirtualBox". Actualizado en Sept. 2024, de <https://es.wikipedia.org/wiki/VirtualBox>
- [8] "SDK para Python". Actualizado en Oct. 2024, de <https://docs.oracle.com/es-ww/iaas/Content/API/SDKDocs/pythonsdk.htm>
- [9] "OS Module in Python". Actualizado en Agost. 2024, de <https://www.geeksforgeeks.org/os-module-python-examples/>
- [10] Python Software Foundation.: "Subprocess management". Actualizado en Oct. 2024, de <https://docs.python.org/3/library/subprocess.html#module-subprocess>
- [11] Forcier, Jeff.: "Paramiko". Actualizado en 2024, de <https://www.paramiko.org/>
- [12] Sánchez, Alfredo.: "La librería Pandas". Actualizado en Jun. 2022, de <https://aprendeconalf.es/docencia/python/manual/pandas/>
- [13] Sánchez, Alfredo.: "La librería Numpy". Actualizado en Mayo 2022, de <https://aprendeconalf.es/docencia/python/manual/numpy/>
- [14] Python Software Foundation.: "Regular expression operations". Actualizado en Oct. 2024, de <https://docs.python.org/3/library/re.html>
- [15] "Instalar PILL". Recuperado en Jul. 2014, de <https://recursospython.com/guias-y-manuales/instalar-pil-pillow-efectos/>
- [16] Python Software Foundation.: "Time access and conversions". Actualizado en Oct. 2024, de <https://docs.python.org/3/library/time.html>

