

Trabajo Final

Entorno de escritorio con gestor de ventanas XMonad en Haskell

Universidad: UNaM

Facultad: FCEQyN

Carrera: Lic. en Sist. de Información

Cátedra: Paradigmas y Leng. de Prog.

Alumno: Antoniak, Rodrigo Lionel

Profesores:

Pautsch, Germán

Martini, Esteban

Año: 2022

Índice general

I	Introducción	5
II	Desarrollo	9
1.	Justificación	11
2.	Objetivos	15
2.1.	General	16
2.2.	Específicos	16
3.	Alcance	17
4.	Recursos	19
5.	Instrucciones	37

6. Resultados	41
A. Glosario	49
Bibliografía	55

Índice de cuadros

6.1. Tabla de resultados . . .	42
--------------------------------	----

Índice de figuras

4.1. Computadora huésped .	20
4.2. Máquina virtual	20
4.3. Vistazo de MV	21
4.4. Información de SO de MV	21
4.5. CPUs de MV	22
4.6. Memoria de MV	22
4.7. Opciones de arranque de MV	23
4.8. Disco VirtIO de MV . .	23
4.9. Lector de discos en MV	24
4.10. Tableta de MV	24
4.11. Ratón de MV	25
4.12. Teclado de MV	25
4.13. Servidor de pantalla de MV	26
4.14. Sonido de MV	26

4.15. Dispositivo Serial 1 de MV	27
4.16. Canal qemu-ga de MV	27
4.17. Canal spice de MV	28
4.18. Controlador eSPI en MV	28
4.19. Controlador de gestión de energía en MV	29
4.20. Controlador de Audio HD en MV	29
4.21. Bus de gestión del siste- ma en MV	30
4.22. Controlador de Ethernet en MV	30
4.23. Controlador de WiFi en MV	31
4.24. Modelo de video de MV	31
4.25. Controlador USB de MV	32
4.26. Controlador PCIe de MV	32
4.27. Controlador PCI de MV	33
4.28. Controlador SATA de MV	33
4.29. Controlador Serial de MV	34

4.30. Controlador de lector de disco de MV	34
4.31. Redoreccionador USB 1 de MV	35
4.32. Redoreccionador USB 2 de MV	35
4.33. Generador de números aleatorios de MV	36
6.1. Primera captura de MV	43
6.2. Segunda captura de MV	43
6.3. Tercera captura de MV	44
6.4. Cuarta captura de MV .	44
6.5. Quinta captura de MV .	45
6.6. Sexta captura de MV .	45
6.7. Séptima captura de MV	46
6.8. Octava captura de MV .	46
6.9. Novena captura de MV .	47

Parte I

Introducción

Este trabajo práctico final se ha realizado con el objetivo de completar la cátedra de Paradigmas y Lenguajes de Programación (en el año 2022), perteneciendo a la carrera de Licenciatura en Sistemas de Información; correspondiente a la Facultad de Ciencias Exactas, Químicas y Naturales, que ofrece la Universidad Nacional de Misiones.

En él, se desarrolla una implementación de un Entorno de escritorio desde el gestor de ventanas XMonad [1]; donde cada ventana se encuentra en mosaicos y está escrito en el lenguaje Haskell. En conjunto, se utiliza la barra de estado XMobar [2], el emulador de terminal Termonad [3], el lanzador de aplicaciones Rofi [4] y una aplicación para gestionar una sesión activa (con acciones como apagar, reiniciar y cerrar sesión).

Parte II

Desarrollo

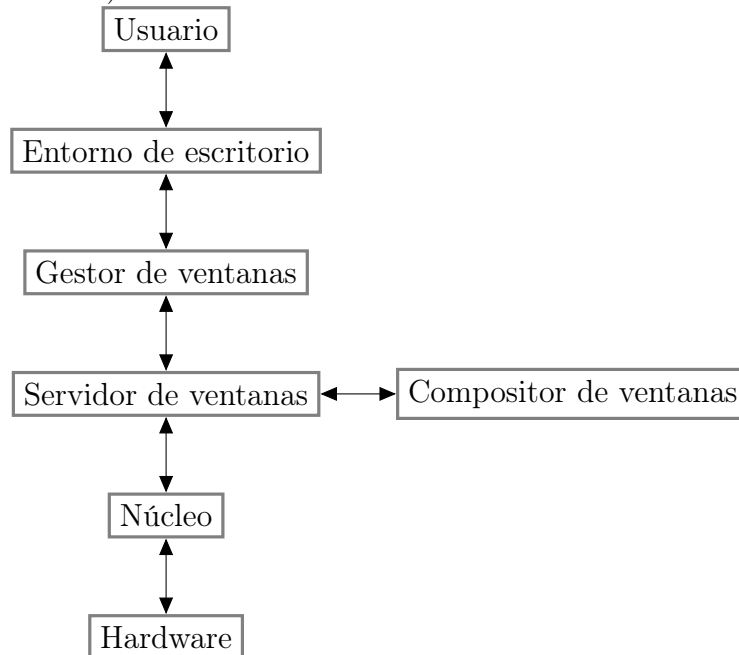
Capítulo 1

Justificación

En la actualidad, todas las computadoras personales utilizan una interfaz gráfica para interactuar con el usuario. Más allá del sistema operativo que se utilice, el entorno visual que se utilice en el espacio personal es esencial; tomando en cuenta la facilidad de uso, comodidad y personalización, entre otros aspectos.

Dependiendo de la arquitectura del sistema operativo, podrán existir distintas opciones para los entornos de escritorio de la computadora personal. En el caso de Windows [5], el motor gráfico está definido a nivel del kernel (como se muestra en <https://learn.microsoft.com/es-ar/windows-hardware/drivers/display/images/dpy1.png> y https://en.wikipedia.org/wiki/Architecture_of_Windows_NT#/media/File:Windows_2000_architecture.svg); esto limita a que deba utilizarse el entorno gráfico que define Microsoft para su sistema operativo. En cambio, los sistemas operativos basados en UNIX [6] pueden elegir su propio entorno de escritorio (por lo tanto, su gestor de ventanas) porque su arquitectura en capas lo permite.

A continuación, se mostrará cómo se gestiona el entorno gráfico en un sistema operativo Linux para computadoras personales (se aclara esto porque los servidores no cuentan normalmente con ninguna interfaz gráfica por defecto):



Un ejemplo de entorno gráfico completo para el esquema anteriormente mostrado es el siguiente:

- Servidor de ventanas: X.Org.
- Compositor de ventanas: picom.

- Gestor de ventanas: xfwm4.
- Entorno de escritorio: Xfce.

En la actualidad, existe una gran cantidad de entornos de escritorio para GNU/Linux; desde GNOME y KDE, hasta Xfce, LXQt y otros. Cada uno de ellos se caracteriza por su aspecto, conjunto de herramientas y gestor de ventanas; además de características comparables entre sí, como el consumo eléctrico y de memoria.

Más allá de los pros y contras que posee cada uno de los anteriormente citados, todos comparten la propiedad de ser muy conducidos al uso del ratón o pad táctil. Sin embargo, el usuario puede decidir por interactuar directamente con el gestor de ventanas; en consecuencia, la cantidad de capas que interviene en la interacción humano-computadora es menor y se favorece al menor consumo de recursos.

Además, debe considerarse la salud de las manos con el uso del ratón [7]; donde existen estudios que demuestran los efectos negativos del mouse promedio [8]. A pesar de la existencia de ratones ergonómicos [9], estos últimos son personales; cada uno se adapta a un tamaño particular de mano, incluyendo el hecho de haber varias formas ergonómicas y un costo económico elevado.

Por lo tanto, se puede concretar que la utilización de un gestor de ventanas en mosaico es factible por varias razones; algunas de las cuales son el consumo de recursos y la disminución del uso del mouse, entre otros (ciertas motivaciones son más enfocadas para desarrolladores, como la programación y la personalización; mientras que otras pueden observarse como subjetivas, evitándose incluirse para mantener la objetividad de la justificación).

Capítulo 2

Objetivos

2.1. Objetivo General

- Implementar un entorno de escritorio utilizable con XMonad [1] en Haskell.

2.2. Objetivos Específicos

- Ajustar los atajos de teclado a gusto personal.
- Agregar un menú que se asemeje a la función del menú de inicio de Windows.
- Cambiar la barra de estado por defecto en XMonad (Polybar) por XMonobar [2].
- Reemplazar el emulador de terminal por defecto por Termonad [3].
- Personalizar la aplicación para salir de la sesión actual con Haskell.

Capítulo 3

Alcance

En este trabajo, se abarcará la implantación del gestor de ventanas en mosaico XMonad; en conjunto con una configuración personalizada, la cual pueda satisfacer la mayoría de necesidades de los usuarios en promedio. Considerando que muchos usuarios utilizan entornos de escritorio completos, se contempla las utilidades que las personas generalmente necesitan a continuación:

- Menú que permita acceder a las aplicaciones deseadas de un menú de inicio.
- Barra de estado que indica los parámetros actuales del sistema (XMonad).
- Accesibilidad hacia el control de los periféricos de la computadora, como la salida de sonido, WiFi, Bluetooth, almacenamiento extraíble, entre otros.
- Menú que permita realizar distintas acciones con la sesión activa, tales como apagar, reiniciar, cerrar sesión suspender, hibernar y dormir.

No se pretende demostrar todas las capacidades que posee XMonad, las cuales son varias; sino que se alcanza hasta las funciones básicas requeridas para la adaptación de un usuario, el cual está acostumbrado a un entorno de escritorio completo.

Para el lector que desee explorar las configuraciones y personalizar su sistema al máximo, con objetivo de lograr el entorno más cómodo y productivo para sí mismo; se encomienda que busque obtener sus metas personales por su cuenta, ya que los gestores de ventanas en mosaico dan la posibilidad de ajustar el escritorio a gusto de cada usuario.

Adicionalmente, se utilizará el lanzador de aplicaciones Rofi; en consecuencia, se podrá ejecutar programas sin necesidad de personalizarlo en el menú de inicio. También, se cambiará la terminal por defecto de la distribución de Linux a seleccionar por Termonad; buscando mostrar las capacidades del lenguaje de programación Haskell.

Capítulo 4

Recursos

Las características de los sistemas operativos utilizados son:

```

      _.;=====;._
    _.=+++++++=+====;.
      -+++++=====+=====+_
    .-=: ``-----+=+====.
      _vi,                --+=+++++.
      .uvnvi.              _.-==+====+.
      .vnvnnI`             .;|=|==;.           :|=| |=|.
+QmQQmPpvvvv;_yYsyQQWUUQQQm #QmQ#::QQQUUV$QQQm.
-QQQWQvpvwowZ?.wQQQE==<QWWQ/QWQW.QQWW(: jQWQE
-$QQQQmmU'   jQQQ@+=<QWQQ)mQQQ.mQQQC+;jWQQ@'
-$WQ8YnI: QWQWgQQWV`mWQQ.jQWQQgyyWW@!
-lvnnvv.     `~+++`       ++|+++
+vnvnnv,     +-.|===
+vnnvvnns.         .:=--
-Invnnvnsi.._____.=sv=.
+Invnnvnnnnnnnnnnvnn;.
~|Invnnvnnvvvvnnv}+`
~-|{*l}*|~

voidrodrigo
-----
OS: Void Linux x86_64
Host: HP Laptop 15-bs1xx
Kernel: 6.1.12_1
Uptime: 9 hours, 51 mins
Packages: 815 (xbps-query)
Shell: bash 5.1.16
Resolution: 1366x768
WM: dwm
Theme: gnome [GTK2], Adwaita [GTK3]
Icons: Adwaita [GTK2/3]
Terminal: st
Terminal Font: Liberation Mono
CPU: Intel i5-8250U (8) @ 3.400GHz
GPU: Intel UHD Graphics 620
Memory: 8953MiB / 11890MiB

```


```

./+00SSSS00+/-.`
    `!+SSSSSSSSSSSSSSSSSSS+:`
++SSSSSSSSSSSSSSSSSSSVVSSS+-
.oSSSSSSSSSSSSSSSSSSdMMMNYSSSSO.
 /SSSSSSSSSSShdmNNmymMMMMHhSSSSS/
 +SSSSSSSSShnydMMMMMMNdddySSSSSSS+
 /SSSSSSShNMMMyhyyyymhNMMMNhSSSSSSS/
 .SSSSSSdMMMNHSSSSSSShNMMDSSSSSSS.
+SSShhyNMMNySSSSSSSSSYNMMMYSSSSSS+
 pSYNMMMNyMMhSSSSSSSSSSShmmhSSSSSO
 pSYNMMMNyMMhSSSSSSSSShmmhSSSSSO
 +SSShhyNMMNySSSSSSSSSYNMMMYSSSSSS+
 +SSShhyNMMNySSSSSSSSSYNMMMYSSSSSS+
 .SSSSSSdMMMNHSSSSSSShNMMDSSSSSS.
 /SSSSSSShNMMMyhyyyhdNMMMNhSSSSSSS/
 +SSSSSSSSdnydMMMMMMMdddySSSSSSS+
 /SSSSSSSSShdmNNNNmyNMMMHhSSSSS/.
.oSSSSSSSSSSSSSSSSSSdMMMNYSSSSO.
++SSSSSSSSSSSSSSSSVVSSS+-
    `!+SSSSSSSSSSSSSSSSS+:`
./+00SSSS00+/-.`

```

rdriago@rdriago-ubuntu

OS: Ubuntu 22.04.1 LTS x86_64
Host: KVM/QEMU (Standard PC (Q35 + ICH9, 2009) pc-q35-7.1)
Kernel: 5.15.0-58-generic
Uptime: 4 hours, 20 mins
Packages: 1907 (dpkg), 11 (snap)
Shell: bash 5.1.16
Resolution: 1366x768
WM: xmonad
Theme: Adwaita [GTK3]
Icons: Adwaita [GTK3]
Terminal: termonad-linux-
CPU: Intel i5-8250U (4) @ 1.799Ghz
GPU: 00:01.0 Red Hat, Inc. Virtio GPU
Memory: 1373MiB / 3923MiB



La Máquina Virtual se encuentra configurada de la siguiente forma:

Figura 4.3: Vistazo de MV

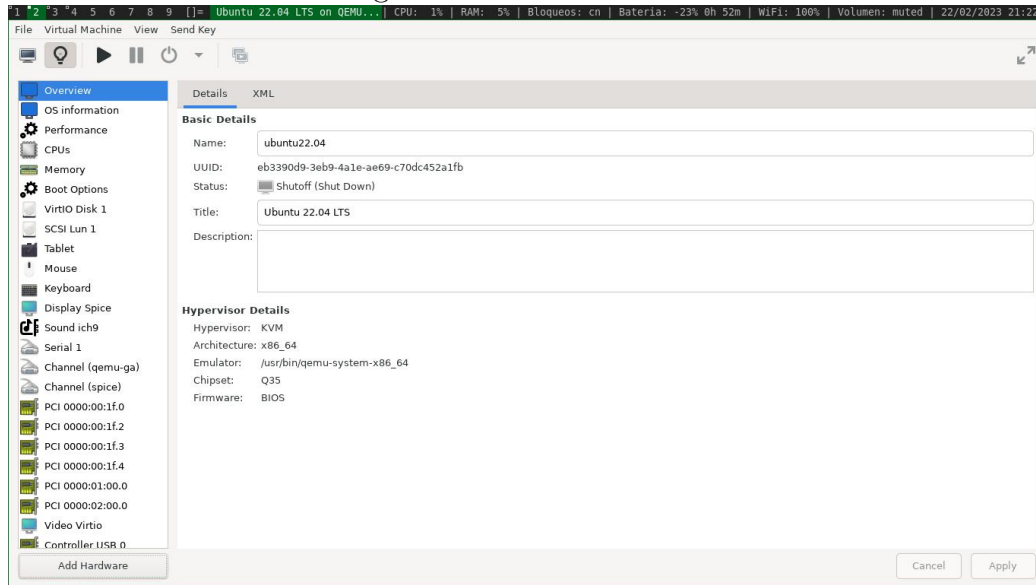


Figura 4.4: Información de SO de MV

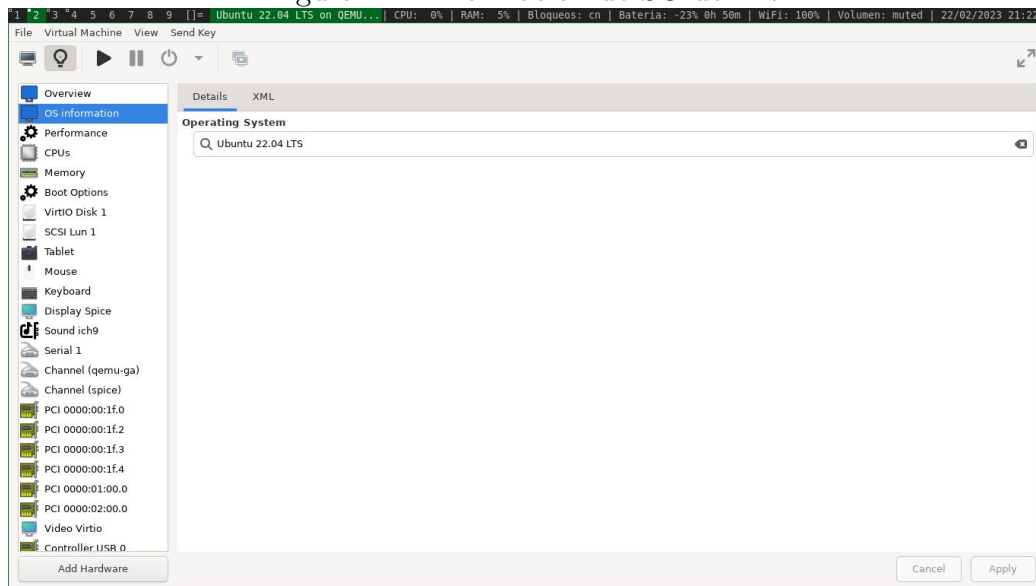


Figura 4.5: CPUs de MV

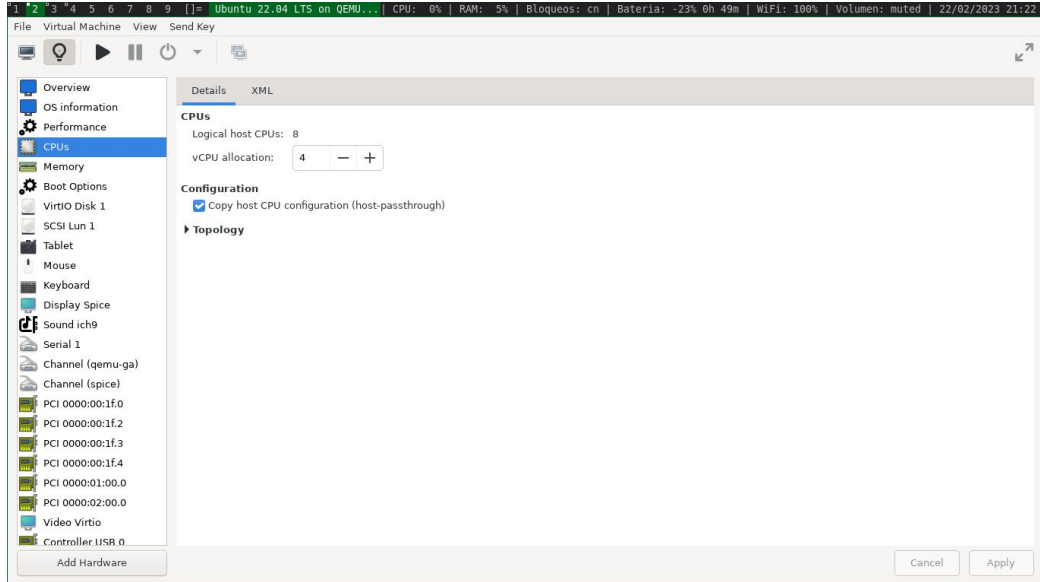


Figura 4.6: Memoria de MV

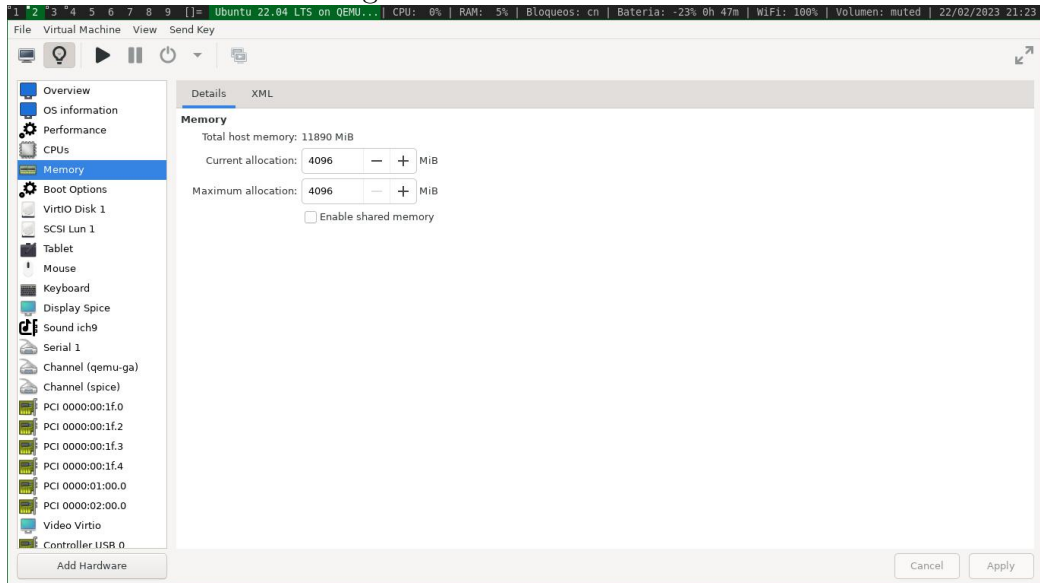


Figura 4.7: Opciones de arranque de MV

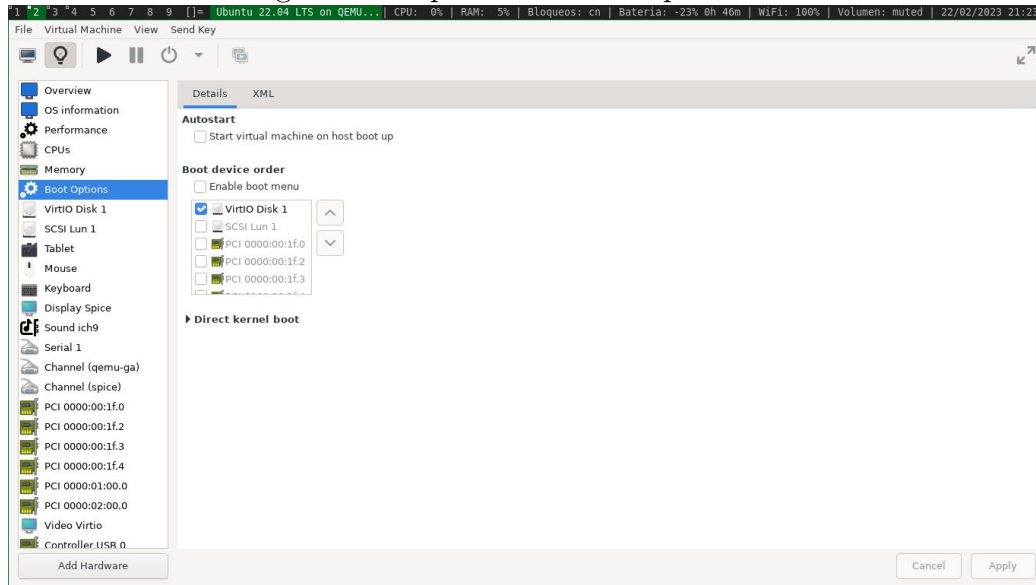


Figura 4.8: Disco VirtIO de MV

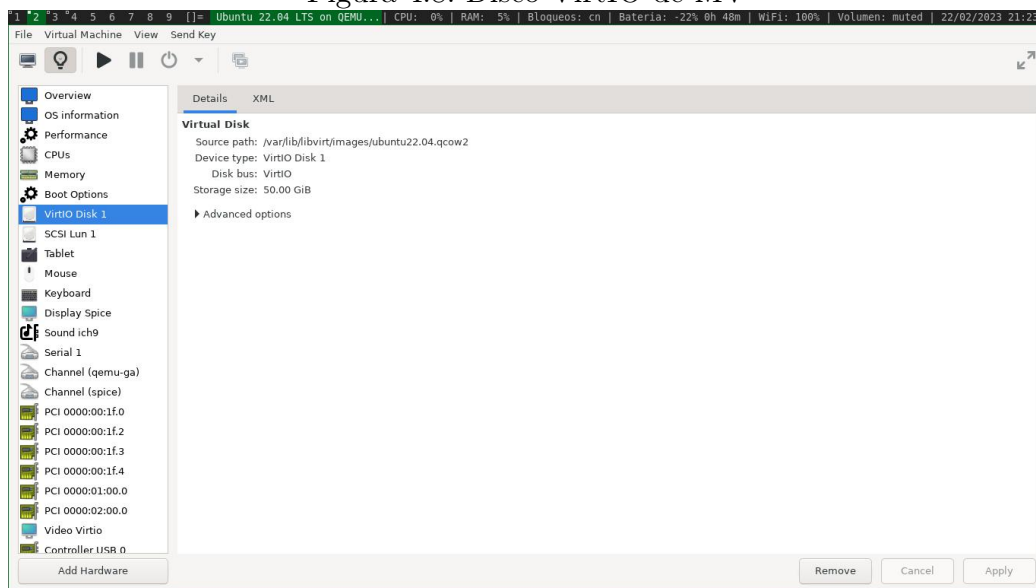


Figura 4.9: Lector de discos en MV

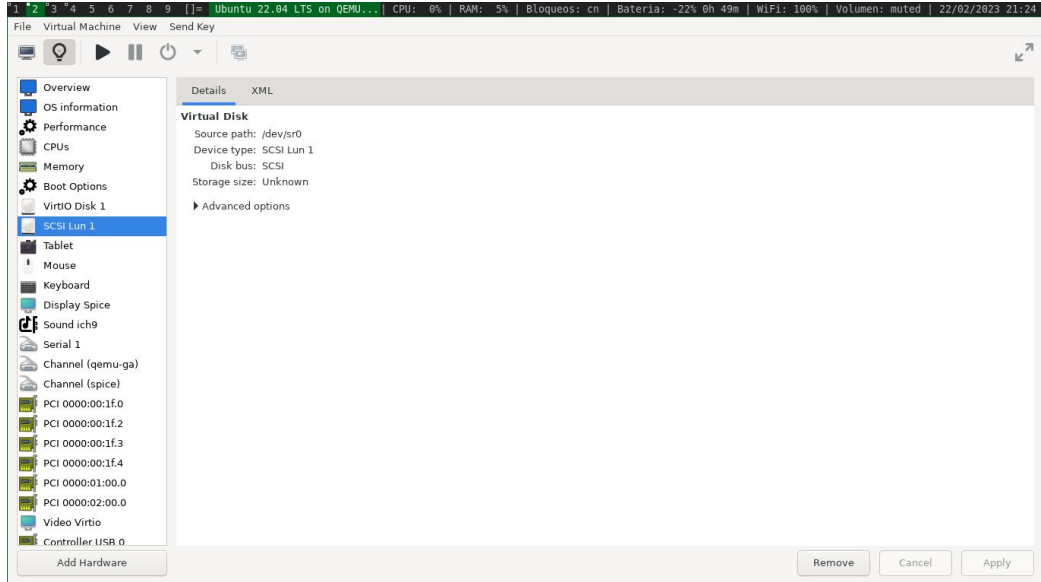


Figura 4.10: Tableta de MV

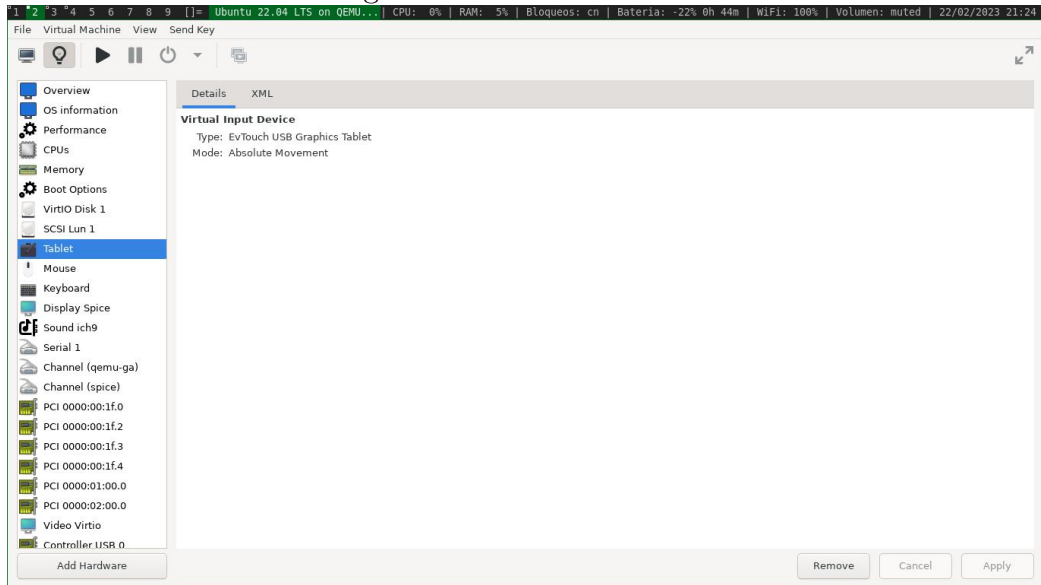


Figura 4.11: Ratón de MV

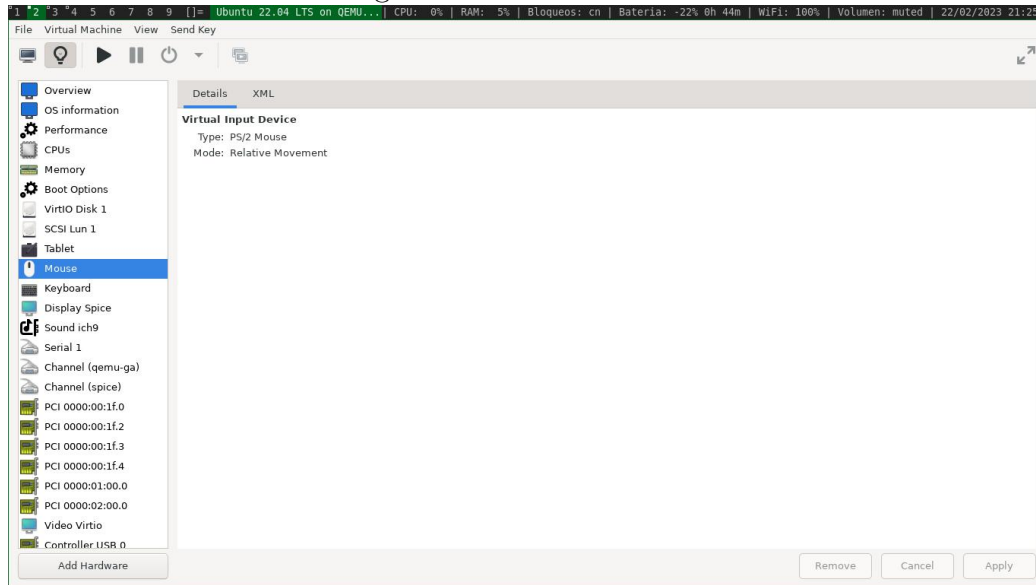


Figura 4.12: Teclado de MV

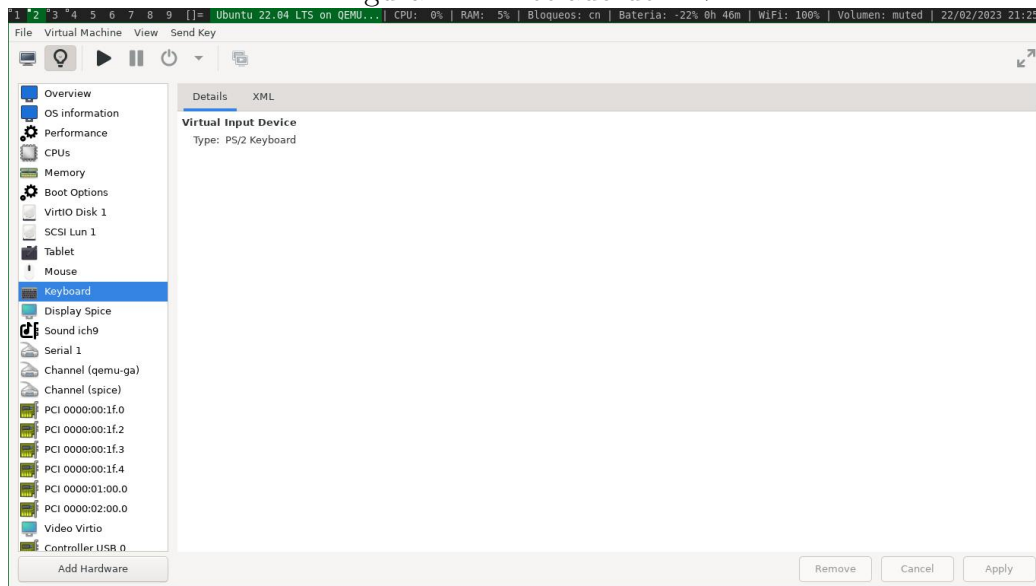


Figura 4.13: Servidor de pantalla de MV

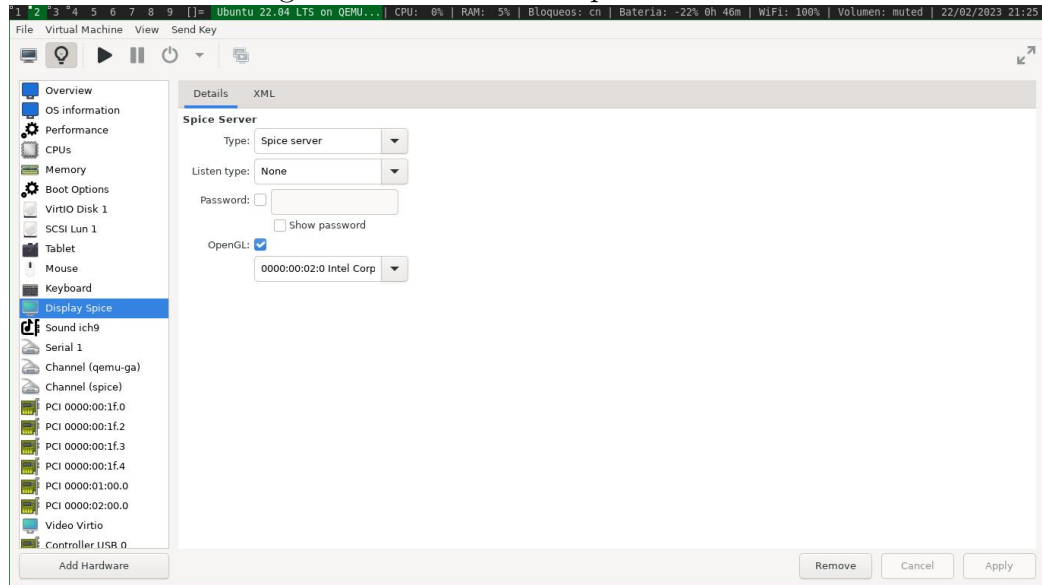


Figura 4.14: Sonido de MV

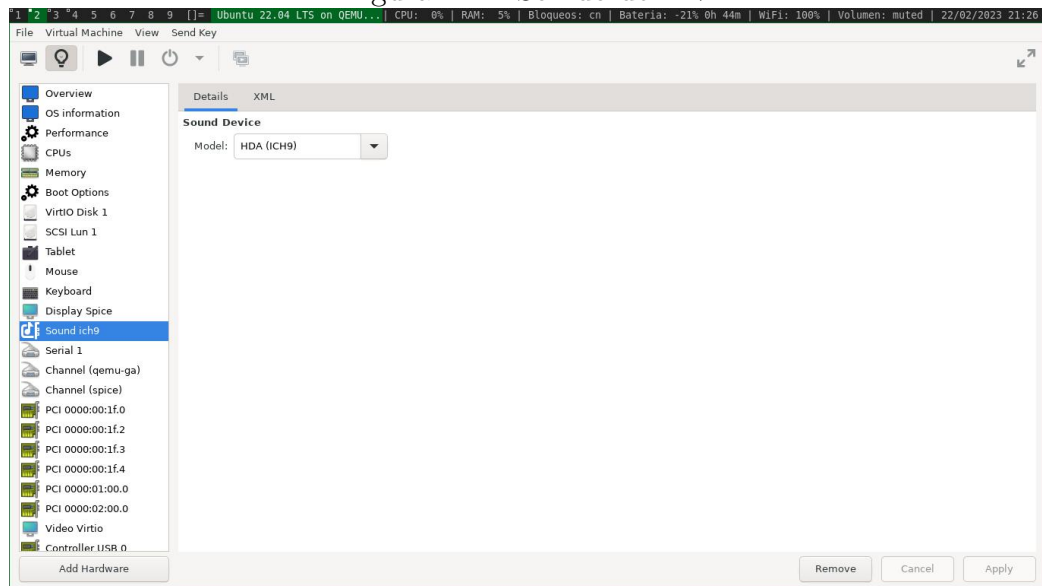


Figura 4.15: Dispositivo Serial 1 de MV

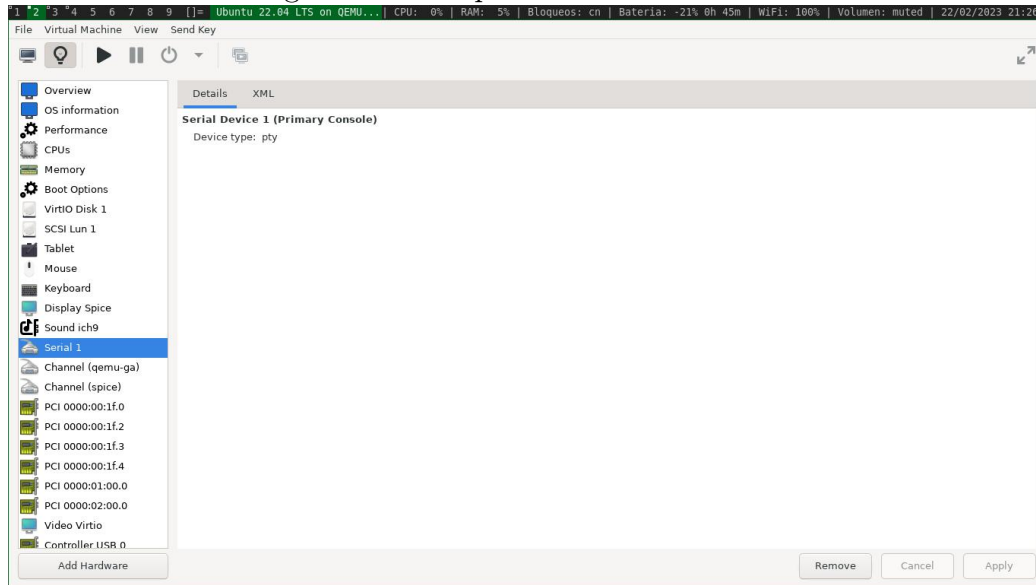


Figura 4.16: Canal qemu-ga de MV

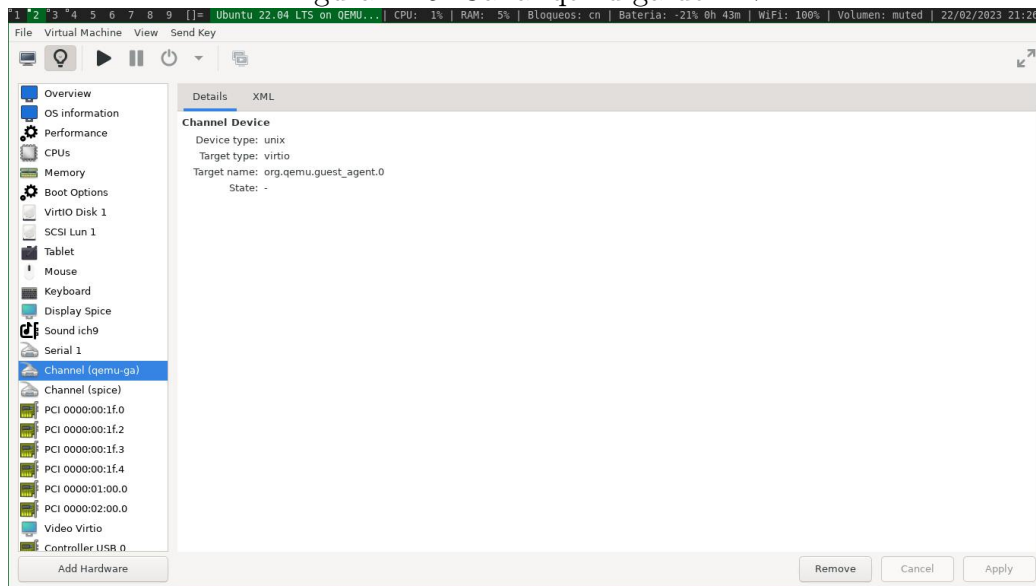


Figura 4.17: Canal spice de MV

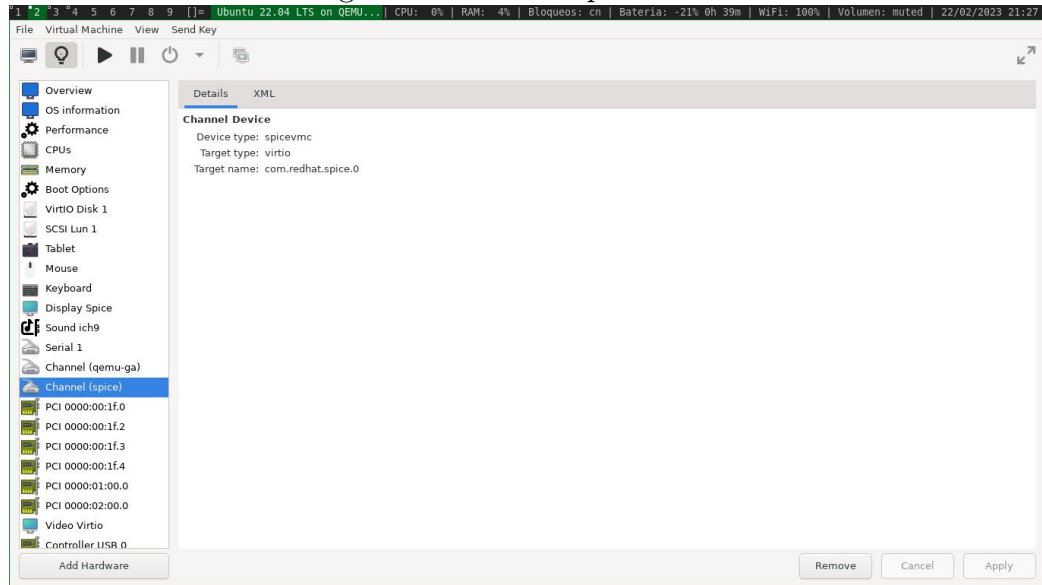


Figura 4.18: Controlador eSPI en MV

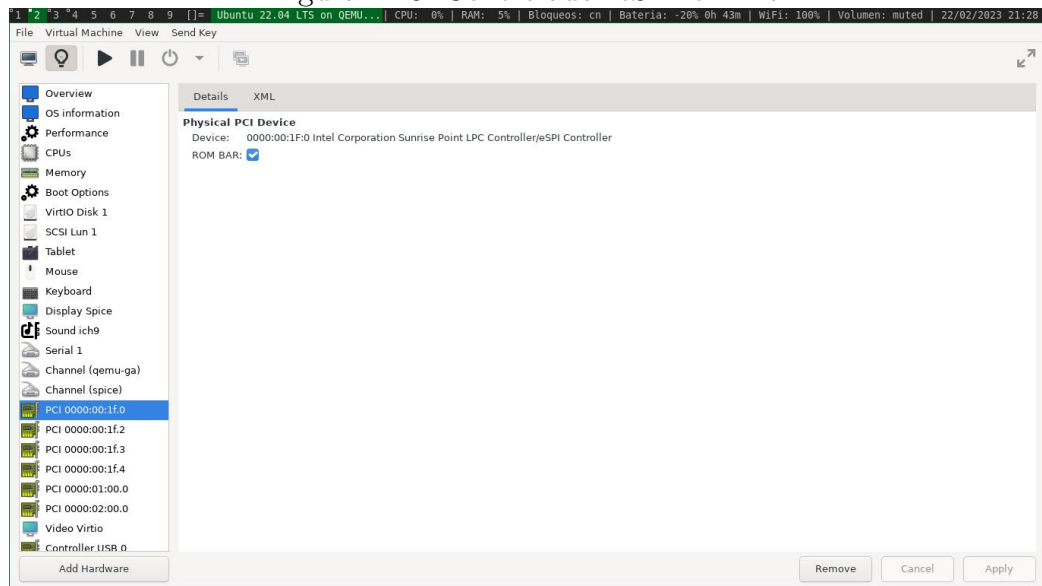


Figura 4.19: Controlador de gestión de energía en MV

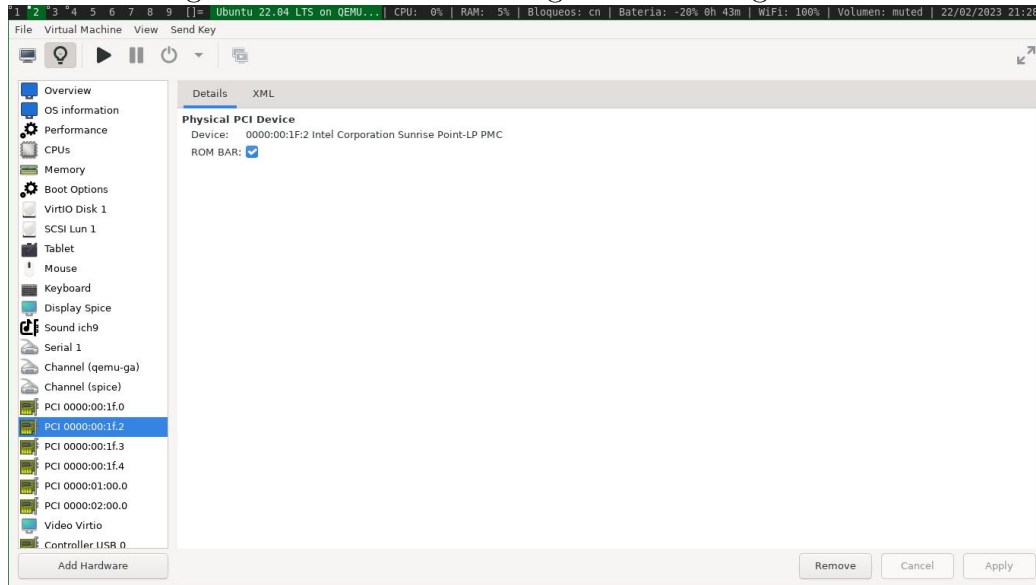


Figura 4.20: Controlador de Audio HD en MV

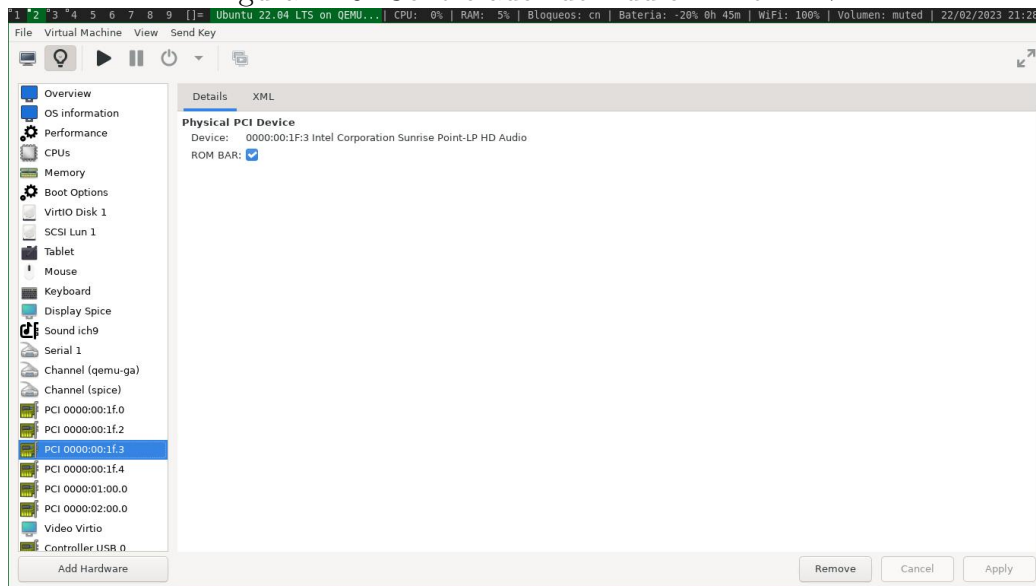


Figura 4.21: Bus de gestión del sistema en MV

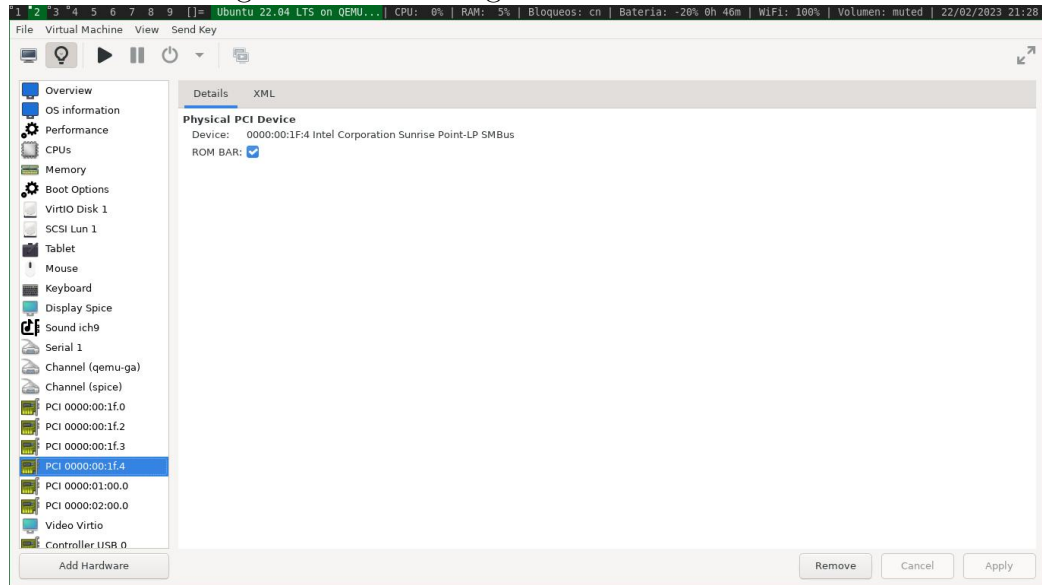


Figura 4.22: Controlador de Ethernet en MV

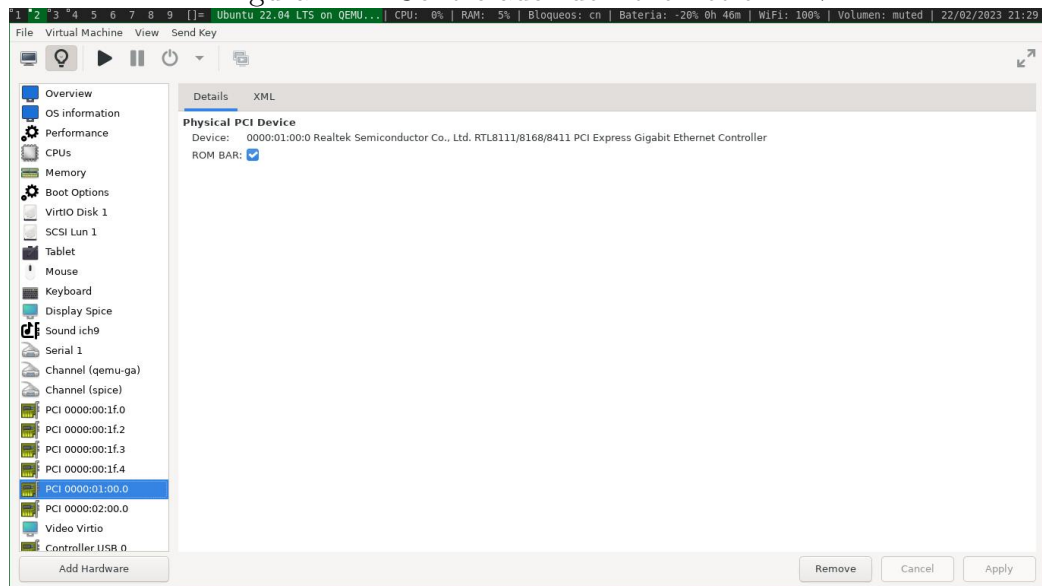


Figura 4.23: Controlador de WiFi en MV

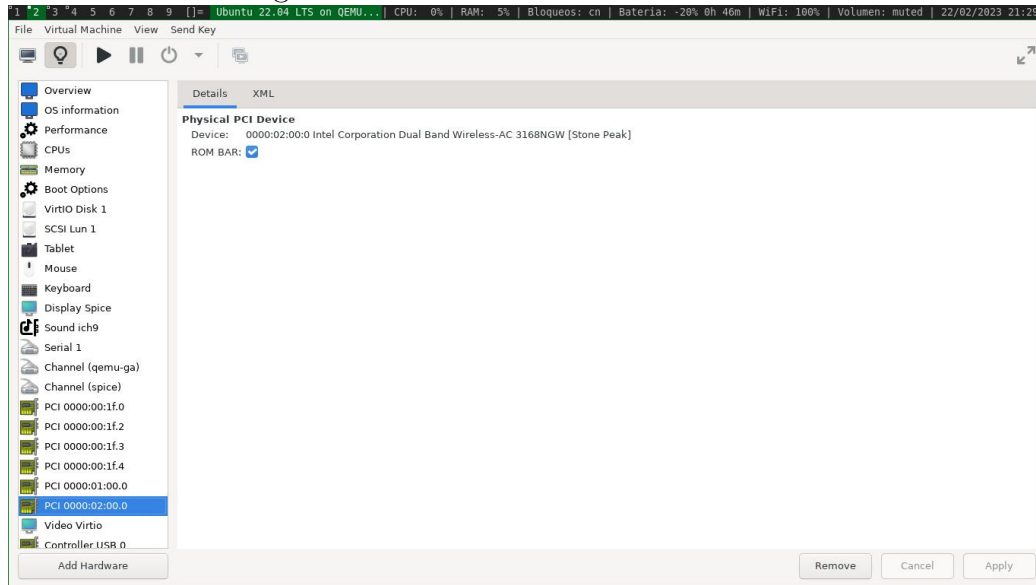


Figura 4.24: Modelo de video de MV

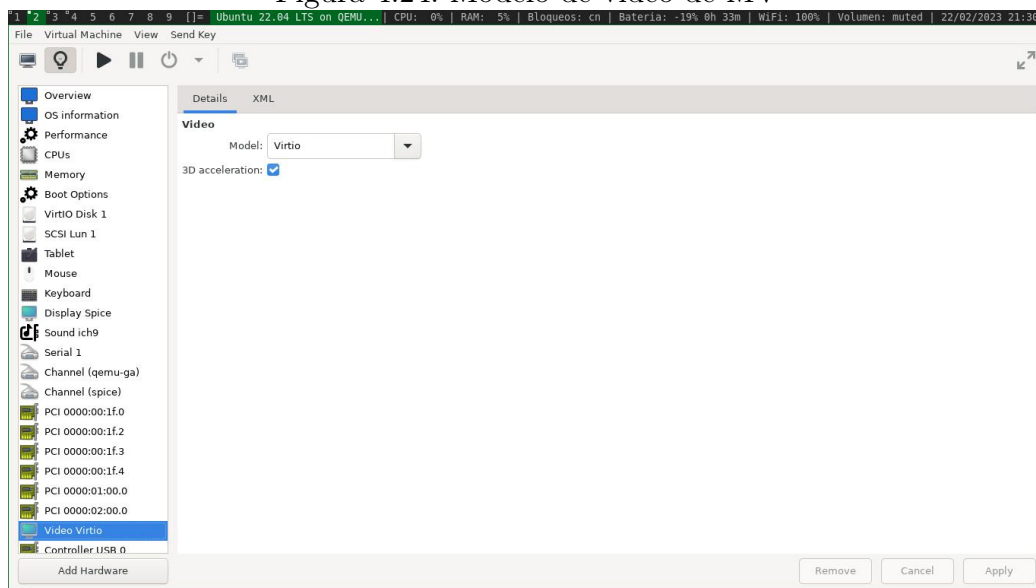


Figura 4.25: Controlador USB de MV

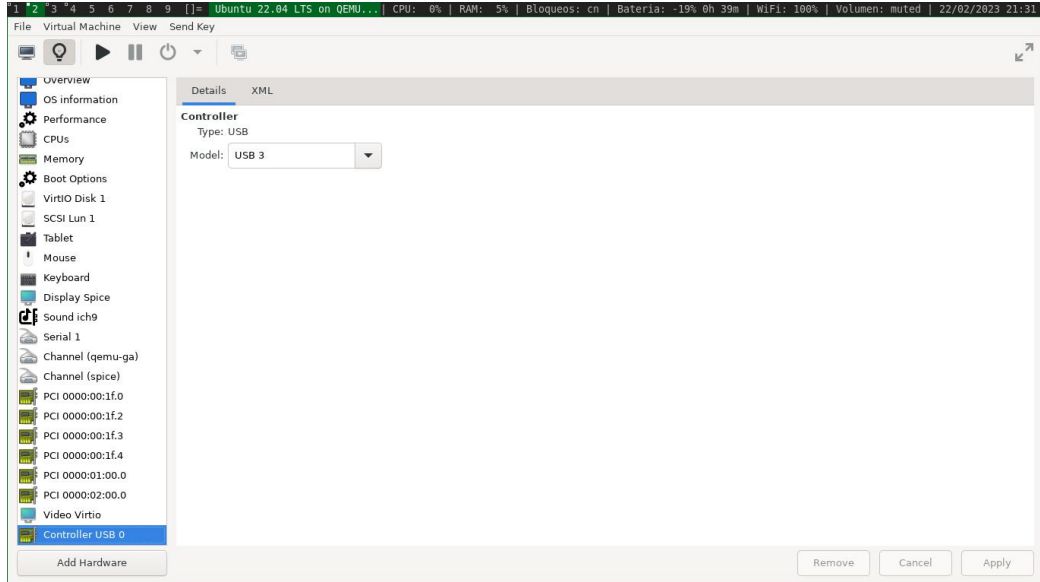


Figura 4.26: Controlador PCIe de MV

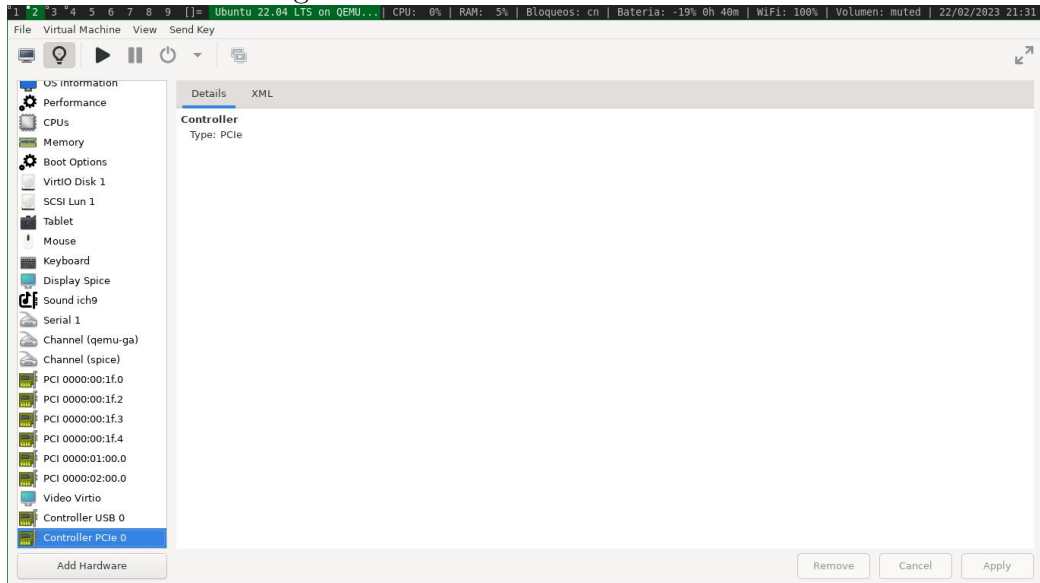


Figura 4.27: Controlador PCI de MV

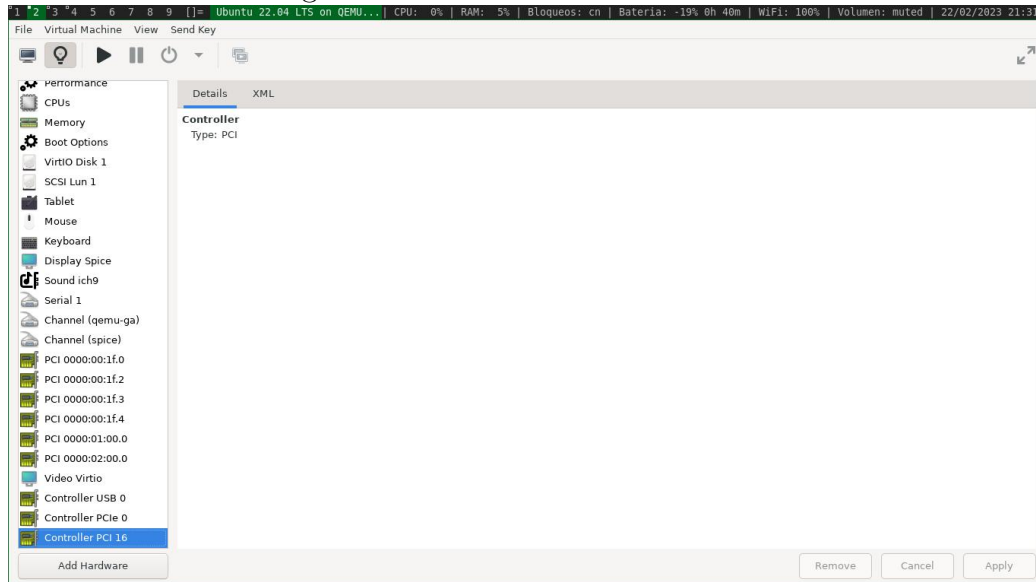


Figura 4.28: Controlador SATA de MV

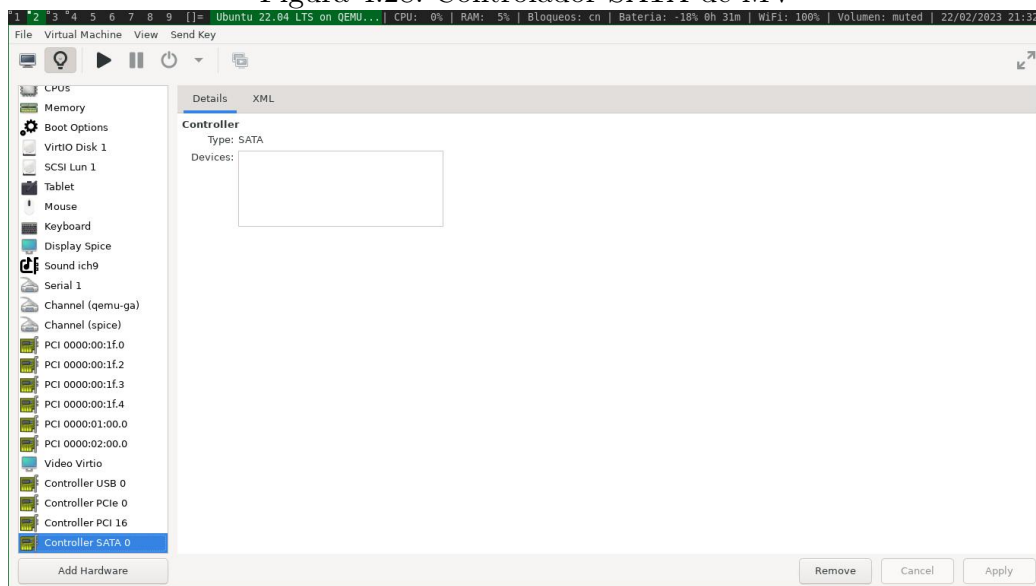


Figura 4.29: Controlador Serial de MV

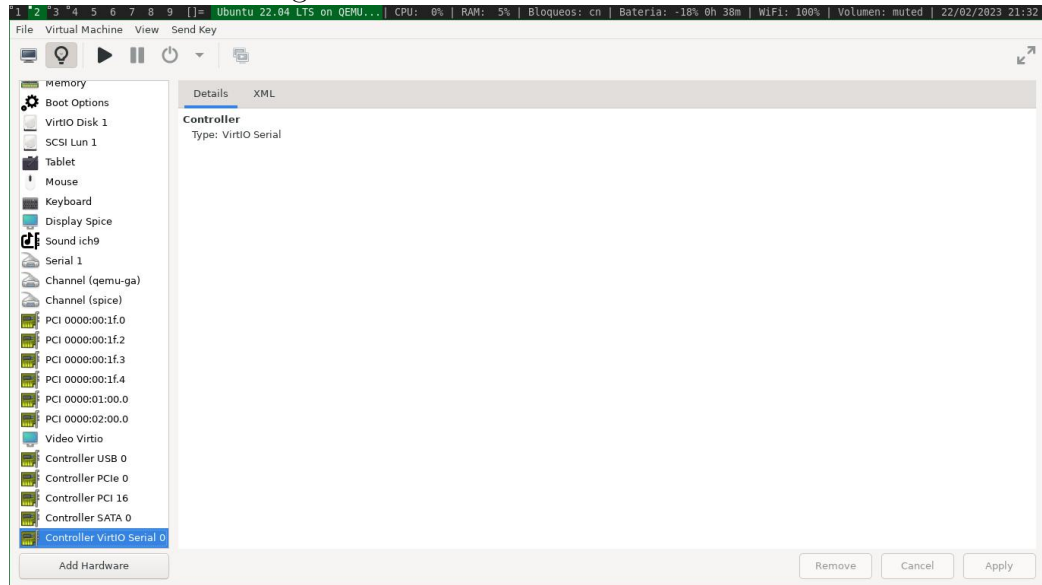


Figura 4.30: Controlador de lector de disco de MV

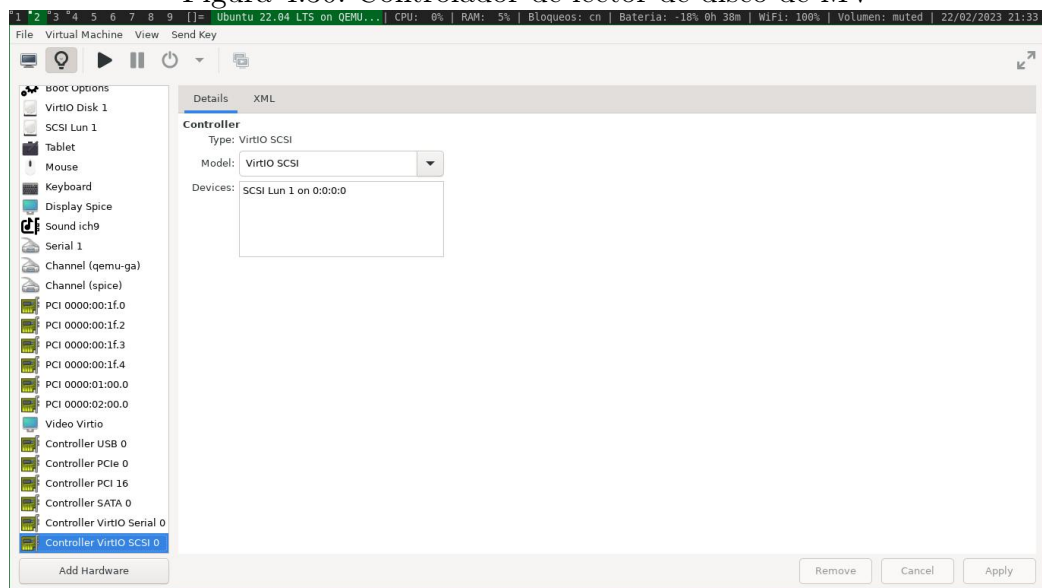


Figura 4.31: Redireccionador USB 1 de MV

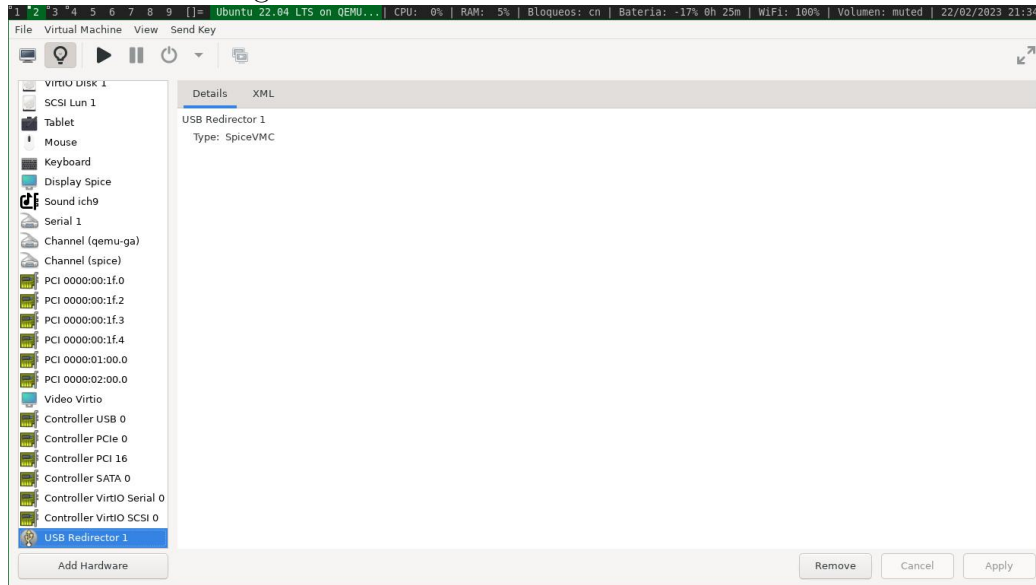


Figura 4.32: Redireccionador USB 2 de MV

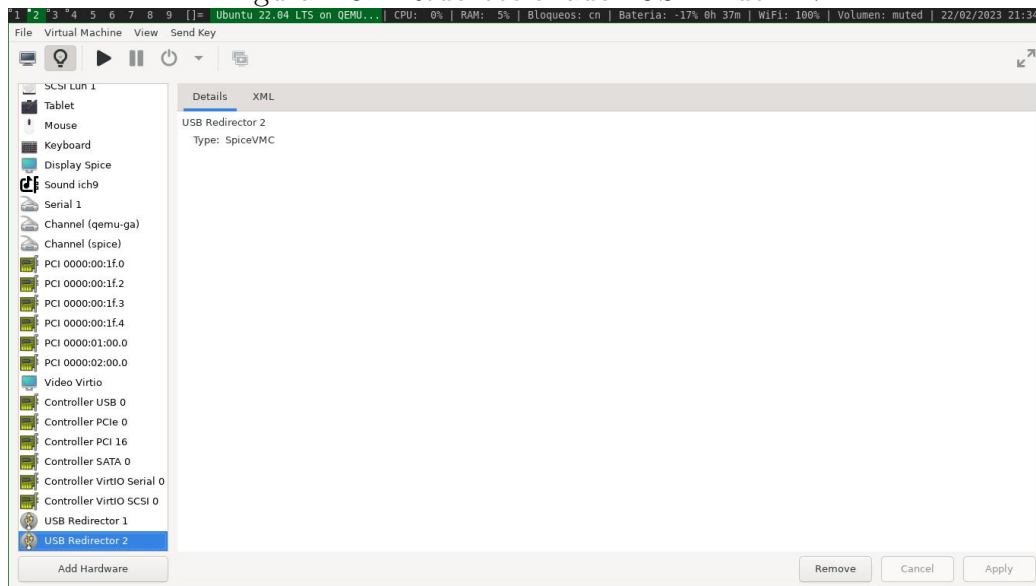
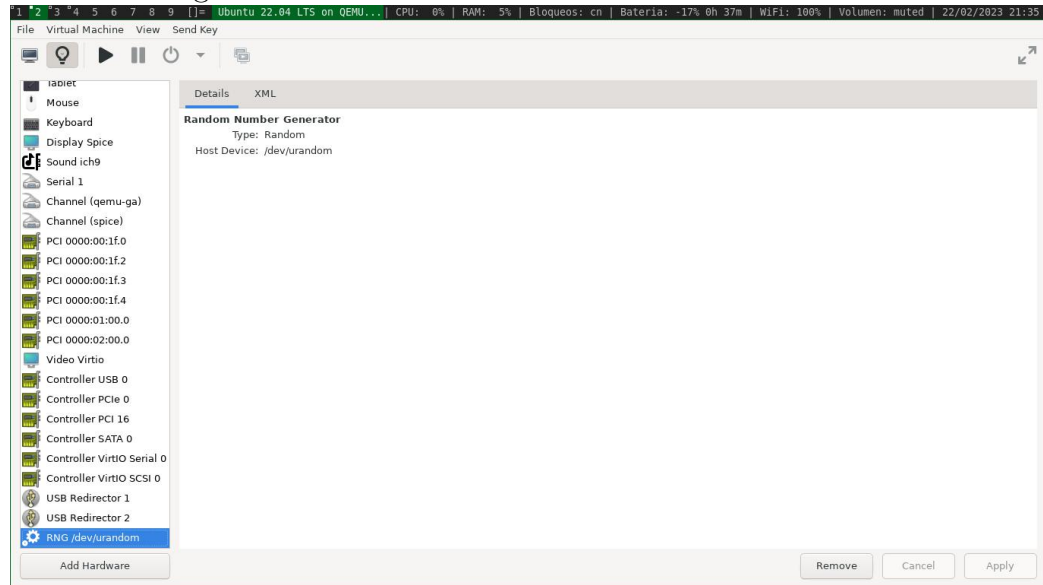


Figura 4.33: Generador de números aleatorios de MV



Para la instalación de los paquetes de Haskell, se utilizará el gestor de paquetes Apt; de todas formas, es posible mejorar la experiencia de ciertas características utilizando cabal (aunque para lo que se realizará, no hará falta).

Si bien, puede instalarse Termonad por el gestor de paquetes; la única forma de que el programa sea modificable con un archivo de Haskell es compilando el programa (revisar <https://github.com/cdepillabout/termonad#running-with-stack>), por lo que se necesitará adicionalmente Stack [10].

Capítulo 5

Instrucciones

Para instalar todos los componentes necesarios en Ubuntu 22.04 LTS, se siguieron los siguientes pasos:

- `sudo apt install xmonad libghc-extra-dev \`
`libghc-xmonad-contrib-dev libghc-xmonad-wallpaper-dev \`
`xmobar compton curl rofi scrot stterm git gnome-screensaver \`
`g++ gcc libc6-dev libffi-dev libgmp-dev make xz-utils \`
`zlib1g-dev gnupg netbase`
 Donde:
 - xmonad es el ejecutable del gestor de ventanas.
 - libghc-extra-dev provee librerías extras para GHC, necesarias en el archivo xmonad.hs modificado.
 - libghc-xmonad-contrib-dev es la librería de XMonad Contrib.
 - libghc-xmonad-wallpaper-dev provee la función de seleccionar aleatoriamente el fondo de pantalla desde xmonad.hs.
 - xmobar es el ejecutable de la barra de estado.
 - compton es el compositor de ventanas.
 - curl servirá para una instrucción posterior.
 - rofi es el lanzador de aplicaciones.
 - scrot es una herramienta para capturar pantallas.
 - stterm es una terminal minimalista que servirá para su ejecución en casos distintos a la principal (Termonad).
 - git servirá para clonar el repositorio de una aplicación
 - gnome-screensaver sirve para bloquear la pantalla en un entorno con GDM.
 - El resto de los paquetes corresponde a las dependencias dentro de un comando a realizar posteriormente.
- `curl -sSL https://get.haskellstack.org/ | sh`
- `sudo apt install gobject-introspection libgirepository1.0-dev \`
`libgtk-3-dev libvte-2.91-dev libpcre2-dev`
 Donde todos los paquetes son dependencias para poder construir Termonad.
- `cd directorio/a/gusto`
- `git clone https://github.com/cdepillabout/termonad`

- `cd termonad/`
- `stack build`
- `stack run`

Posteriormente, para poder configurar de forma correcta; es necesario recompilar todos los archivos de Haskell, donde:

- XMonad se compila con el comando “`xmonad --recompile`” y se reinicia con “`xmonad --restart`”. Esta última acción afecta a XMobar, por lo que no se necesita más acciones para la barra de estado.
- Termonad se configura realizando “`stack exec --package termonad --package colour -- termonad`” desde `~/local/bin`. Adicionalmente, se copia el archivo `~/cache/termonad/termonad-linux-x86_64` dentro del directorio `/usr/local/bin` (puede ser cualquier otro directorio dentro del PATH, lo único que se busca es evitar añadir el directorio origen del archivo)

Para el caso de Rofi, no habrá necesidad de ninguna compilación; con actualizar el archivo `~/config/rofi/config.rasi` es suficiente, ya que el programa lee esta configuración cada vez que se ejecuta.

Por siguiente, para la aplicación que permite hacer acciones sobre la sesión actual; se clona desde el repositorio de Distrotube <https://gitlab.com/dwt1/byebye>, así se cambia lo necesario y se obtiene el ejecutable. Dentro de lo modificado, se encuentra la traducción de las opciones en español y la adaptación de algunos comandos para que funcionen en la máquina virtual.

Para construir la aplicación, se utiliza “`stack build`” desde el directorio en donde se clonó el repositorio; para obtener el ejecutable en `.stack-work/install/unicaCarpetaAqui/hash/version/bin/byebye-exe`, que se copia dentro de `/usr/local/bin` por las mismas razones que Termonad.

Adicionalmente, se ha ajustado la salida de audio controlada por defecto desde alsamixer; para ello, se ha seguido los siguientes pasos:

- Listar las fuentes de sonido con “`pactl list short sources`”.
- Seleccionar la fuente de sonido correcta con “`pactl set-default-source nombreFuenteSonido`”.
- Listar los disipadores de sonido con “`pactl list short sinks`”.

- Seleccionar el disipador de sonido correcta con “`pactl set-default-sink nombreDisipadorSonido`”.

Con todos estos pasos realizados, se consigue el entorno de escritorio deseado en Ubuntu con XMonad. Para obtener las configuraciones de cada programa y la aplicación personalizada, se puede acceder al siguiente repositorio: <https://github.com/rodrigoantoniak/pylp>

Capítulo 6

Resultados

Por cada área de los objetivos a abarcar, se ha obtenido los siguientes resultados:

Menú de Inicio	Usando TreeSelect en XMonad, se obtiene un menú personalizado que otorga las opciones que se desea; incluyendo las esperadas en un menú de inicio.
Barra de Estado	XMobar es más que satisfactoria en la transmisión de información sobre el sistema.
Accesibilidad a Periféricos	Hay distintas formas de cubrir estas necesidades. Para el área de almacenamiento extraíble, todas las operaciones de un usuario promedio se cubren en el gestor de archivos Nautilus (propio de GNOME); si se ocupara otro que fuera gráfico, también cubriría esta necesidad. Para WiFi y salida de sonido, se utiliza TUIs para poder cubrir la necesidad; ejecutándose dentro de una terminal que se activa con ScratchPads. Para Bluetooth, es imposible inspeccionar la funcionalidad; considerando que no se puede trasladar esta capacidad a la máquina virtual, aunque haya formas de trabajarlo
Menú de acciones para la Sesión	Con la aplicación <i>byebye</i> de Derek Taylor (conocido en YouTube como DistroTube, o DT abreviadamente) y pocas modificaciones; se logra el objetivo, incluso teniendo otra forma de acceso a estas acciones desde el menú que se asemeja al menú de Inicio.

Cuadro 6.1: Tabla de resultados

El único problema que se ha encontrado es que Firefox se fuerza a ir para el tercer espacio de trabajo, lo cual puede ser un error propio de XMonad o porque el archivo binario utilizado conserva la memoria de lo usado por defecto; más allá de eso, no se ha encontrado fallas.

A continuación, se mostrará algunas capturas del entorno de escritorio resultante:

Figura 6.1: Primera captura de MV

The screenshot shows a terminal window with the following content:

```

[1] 2 3 4 5 6 7 8 9 | Cpu: 13% | Memoria: 22% | Swap: 1% | Volumen: activo ##### | Wlft: 10KB 10KB | 22/02/2023 20:46:04

1. rodrigo@rodrigo-ubuntu: ~
n b ] < # * z ; a j F z F +
v 1 5 ) ^ G T 2 ' 9 ' i v ] Q
- J b \ ^ G T 2 ' 9 ' i v ] Q
* ( w x q ) y s ; k N d 3 '
A % ? H t Y T L I 7 _ % Y : #
ld 6 9 . c f [ _ % 2 * B w
w ; F v V w - c e > 3 Y B
Z B J N t A q u > 3 Y B
R V u ) r g y ^ 0 N / \ a
hd 0 5 G @ G u l J J
E ; 1 D @ G u l J J
ad M c X % i l l k ; J a
V G 1 c X % i l l k ; J a
z > ? Q 9 / 7 p a
E w < v T e K 9 . s M
U / ^ Y Y G G o . ^ h r B
# d a x N C a . 4 h r B
& > ( d : P < A K K
s t t i A A K K
v X
o X
X s R

```

System statistics:

```

0[|||||] 15.6% Tasks: 96, 106 thr, 1 running
1[|||||] 20.5% Load average: 0.68 0.25 0.14
2[|||||] 9.3% Uptime: 13:05:14
3[|||||] 13.4%
Mem[|||||] [597M/3.83G]
Swp[|||||] [23.7M/3.81G]

```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
75732	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:02.53	termonad-linux
75733	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:00.00	termonad-linux
75734	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:00.00	termonad-linux
75735	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:00.00	termonad-linux
75736	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:00.01	termonad-linux
75737	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:00.00	termonad-linux
75738	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:00.03	termonad-linux
75770	rodrigo	20	0	1024G	100M	85732	S	0.0	2.6	0:00.00	termonad-linux

System information:

```

rodrigo@rodrigo-ubuntu
-----
OS: Ubuntu 22.04.1 LTS x86_64
Host: KVM/QEMU (Standard PC (Q35 + ICH9, 2009) pc-q35-7.1)
Kernel: 5.19.0-32-generic
Uptime: 13 hours, 4 mins
Packages: 1917 (dpkg), 13 (snap)
Shell: bash 5.1.16
Resolution: 1366x768
WM: xmonad
Theme: Adwaita [GTK3]
Icons: Adwaita [GTK3]
Terminal: termonad-linux
CPU: Intel i5-8250U (4) @ 1.799GHz
GPU: 00:01.0 Red Hat, Inc. Virtio GPU
Memory: 592MiB / 3922MiB

```

Figura 6.2: Segunda captura de MV

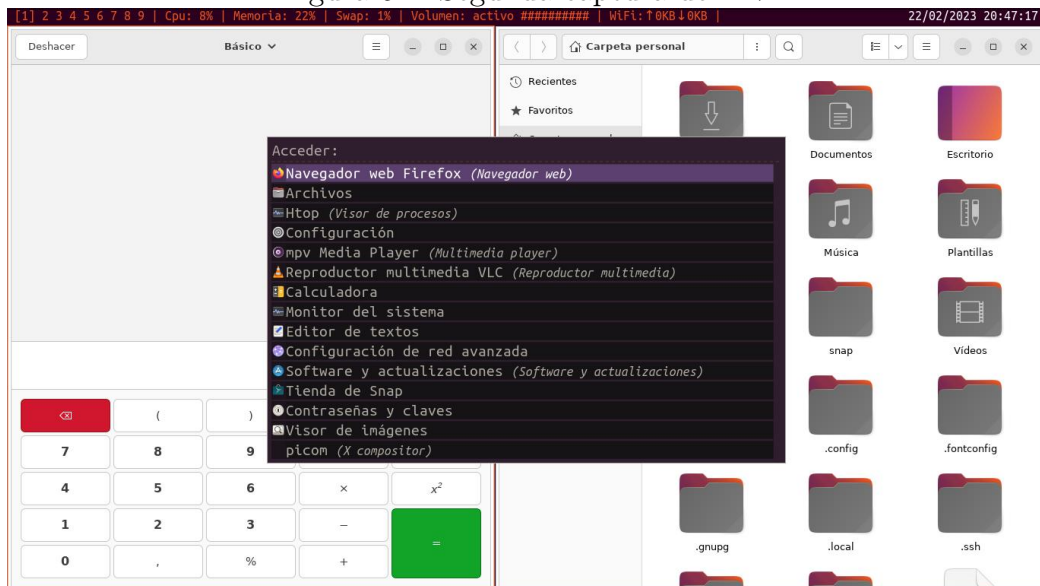


Figura 6.3: Tercera captura de MV

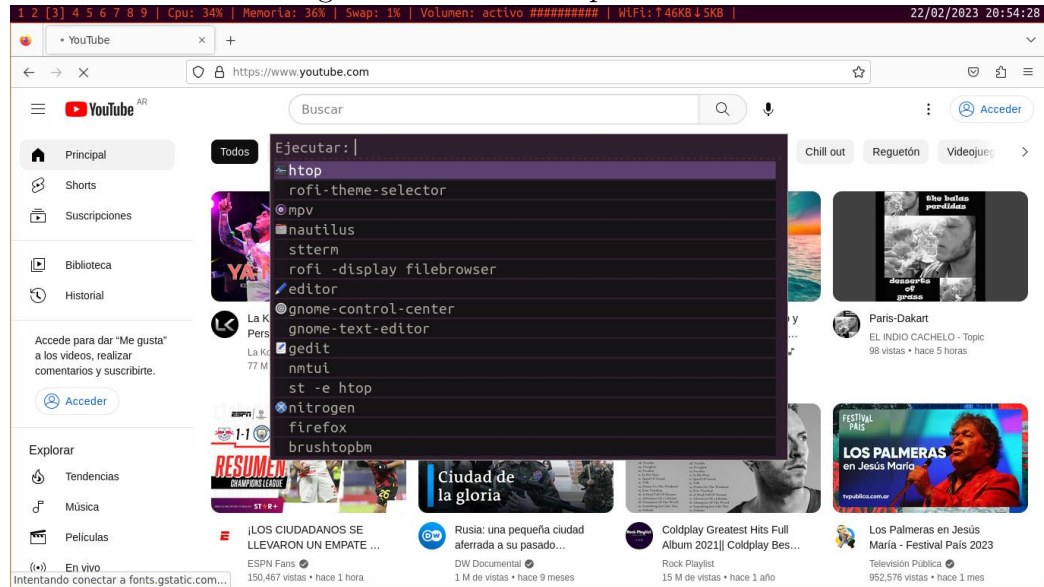


Figura 6.4: Cuarta captura de MV

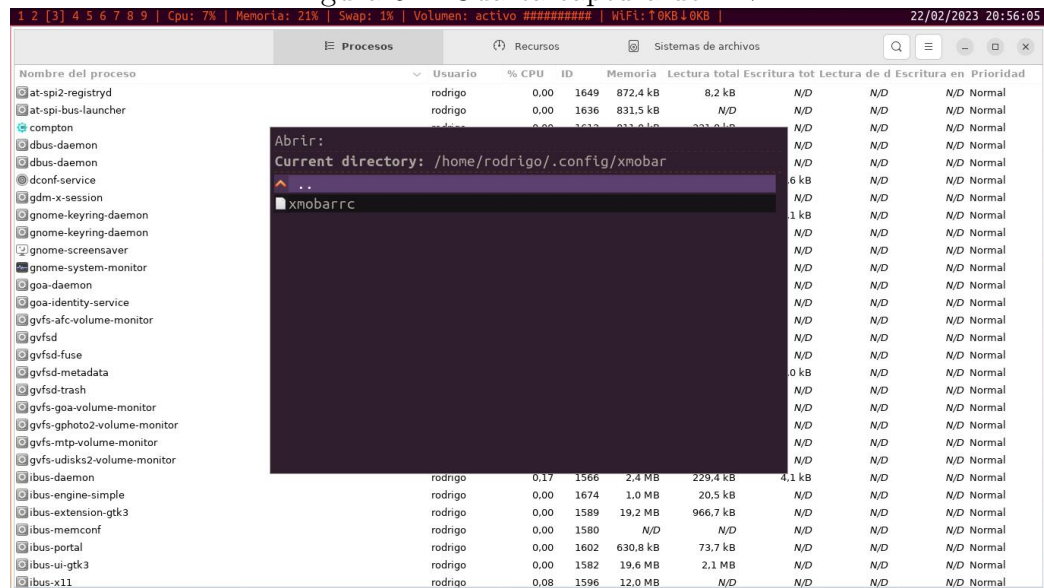


Figura 6.5: Quinta captura de MV

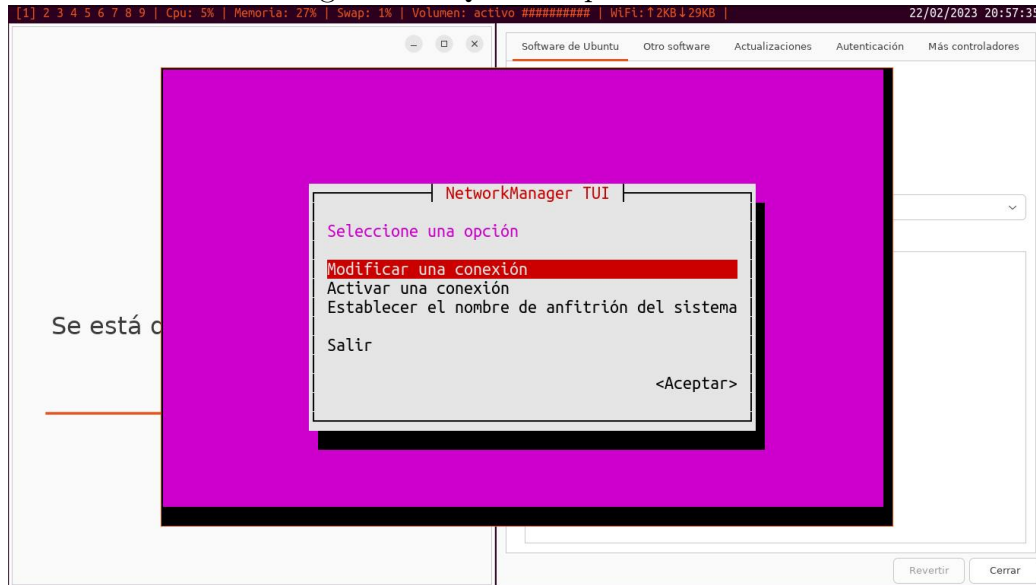


Figura 6.6: Sexta captura de MV

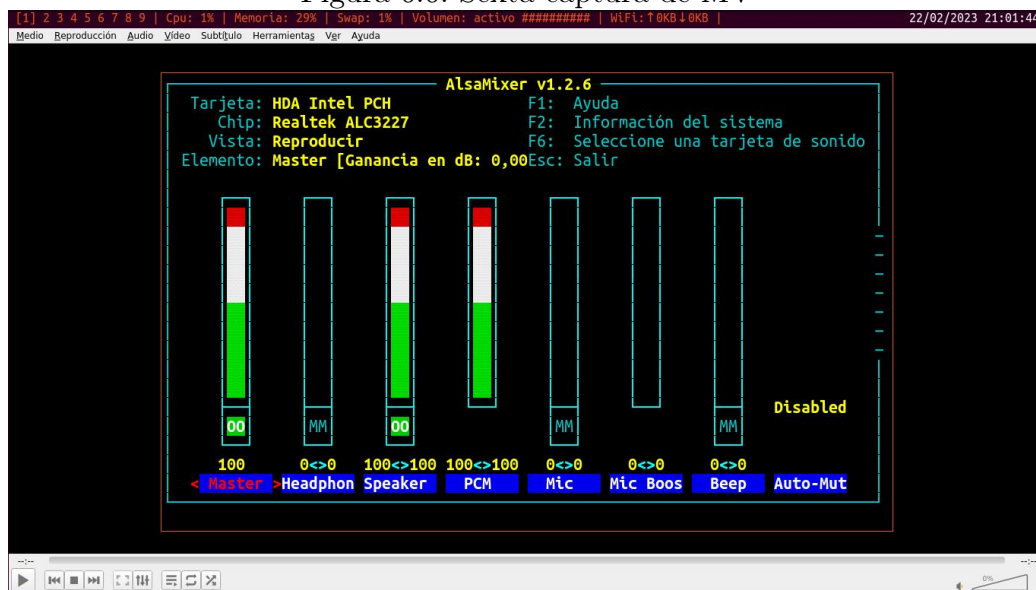


Figura 6.7: Séptima captura de MV

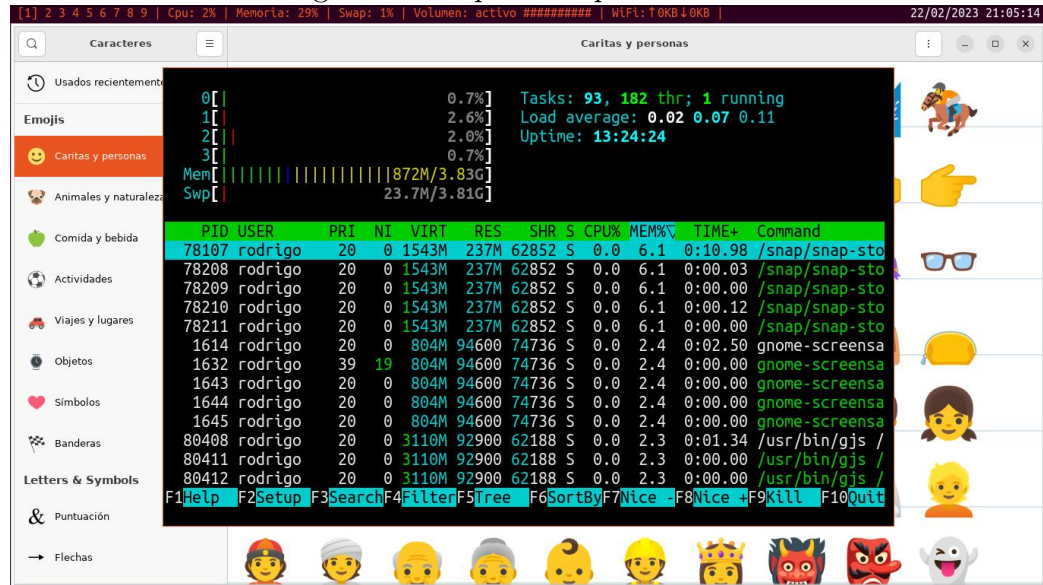
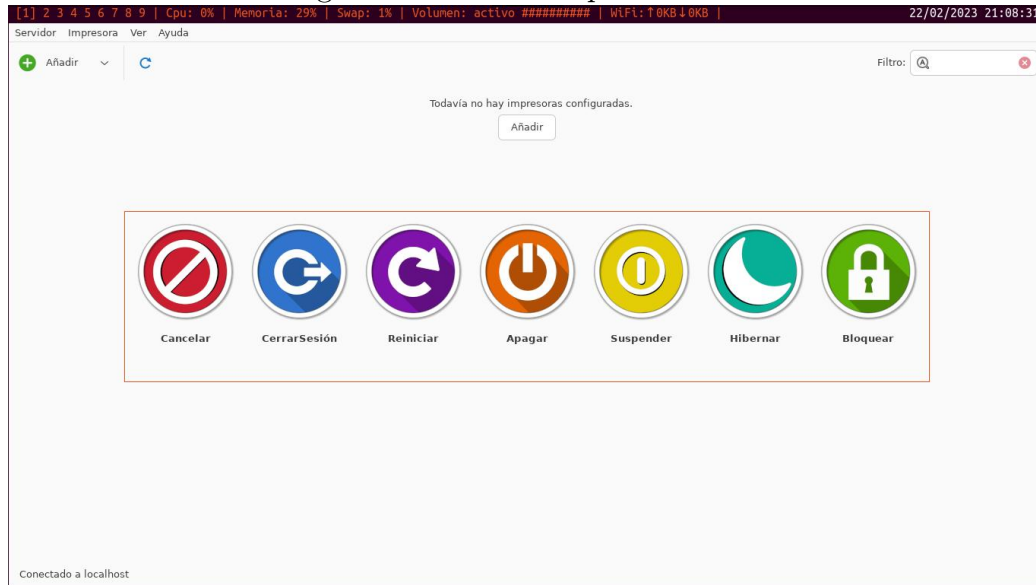


Figura 6.8: Octava captura de MV



Figura 6.9: Novena captura de MV



Apéndice A

Glosario de Términos

Abreviaturas y acrónimos

CPU Central Processing Unit.

FCEQyN Facultad de Ciencias Exactas, Químicas y Naturales.

GDM GNOME Display Manager.

GHC GNU Haskell Compiler.

GNOME GNU Network Object Model Environment.

GNU GNU's Not Unix.

KDE K Desktop Environment.

Lic. en Sist. de Información Licenciatura en Sistemas de Información.

LXQt Lightweight X11 Qt.

MV Máquina Virtual.

Paradigmas y Leng. de Prog. Paradigmas y Lenguajes de Programación.

SO Sistema Operativo.

TUI Terminal User Interface.

UNaM Universidad Nacional de Misiones.

Virt-manager Virtual Machine Manager.

Xfce X Free Cholesterol Environment.

Definiciones

Compositor de ventanas Componente de la interfaz gráfica de una computadora que dibuja las ventanas o sus bordes. Éste también controla cómo éstas son mostradas y cómo interactúan con otras ventanas y el resto del entorno del escritorio.

Entorno de escritorio Conjunto de software para ofrecer al usuario de una computadora una interacción amigable y cómoda. Es una implementación de interfaz gráfica de usuario que ofrece facilidades de acceso y configuración, como barras de herramientas e integración entre aplicaciones con habilidades como arrastrar y soltar.

Gestor de ventanas Programa informático que controla la ubicación y apariencia de las ventanas bajo un sistema de ventanas en una interfaz gráfica de usuario.

Servidor de ventanas Programa cuya tarea principal es coordinar la entrada y la salida de sus clientes hacia y desde el resto del sistema operativo, el hardware, y otros. El servidor gráfico se comunica con sus clientes con el protocolo de servidor gráfico.

Referencias bibliográficas

Bibliografía

- [1] "Documentación de XMonad". Dirección: <https://xmonad.org/documentation.html>.
- [2] "Paquete XMobar". Dirección: <https://hackage.haskell.org/package/xmobar>.
- [3] "Paquete Termonad". Dirección: <https://hackage.haskell.org/package/termonad>.
- [4] "Documentación de Rofi". Dirección: <https://davatorium.github.io/rofi/>.
- [5] "Arquitectura y componentes de Windows". Dirección: <https://learn.microsoft.com/es-mx/windows/win32/directcomp/architecture-and-components>.
- [6] "Arquitectura X vs. Wayland". Dirección: https://wayland.freedesktop.org/docs/html/ch03.html#sect-Wayland-Architecture-wayland_architecture.
- [7] M. Fogleman y G. Brogmus, "Computer mouse use and cumulative trauma disorders of the upper extremities", vol. 38, n.º 12, 24 de oct. de 2007. DOI: 10.1080/00140139508925280. Dirección: <https://www.tandfonline.com/doi/abs/10.1080/00140139508925280>.
- [8] P. Ghasri y S. R. Feldman, "Frictional lichenified dermatosis from prolonged use of a computer mouse: Case report and review of the literature of computer-related dermatoses", vol. 16, n.º 12, 1 de dic. de 2010. DOI: 10.5070/D39bs5w7c3. Dirección: <https://escholarship.org/uc/item/9bs5w7c3>.
- [9] J. Reste, T. Zvagule, N. Kurjane y col., "Wrist Hypothermia Related to Continuous Work with a Computer Mouse: A Digital Infrared Imaging Pilot Study", vol. 12, n.º 9265, 7 de ago. de 2015. DOI: 10.3390/ijerph120809265. Dirección: <https://www.mdpi.com/1660-4601/12/8/9265>.

- [10] "Documentación de Stack (Haskell)". Dirección: <https://docs.haskellstack.org/en/stable/>.