

# **UT1-T2. Systemd.**

**16/10/2024**

**Creado por: Pedro José Riquelme Guerrero**



---

# INDICE

1. Arranque sin el servidor gráfico (sin GUI) .....	3
2. Diagnóstico del servicio SSH no arrancado .....	4
a) El servicio intentó arrancar, pero falló:.....	4
b) El servicio no intentó arrancar: .....	5
3. Aplicar cambios en la configuración del servicio `rsyslog` .....	5
4. Por qué SSL intenta arrancar automáticamente .....	6
5. Servicio en `multi-user.target` sin arrancar en `graphical.target` .....	6
6. Impedir que un servicio arranque manualmente.....	7
7. Consultar el nivel de ejecución predeterminado .....	7
8. Demostrar que `rsyslog` arranca automáticamente en el nivel de ejecución predeterminado .....	8
9. Comportamiento de `apt-get install isc-dhcp-server` y `systemctl disable isc- dhcp-server` .....	9
10. Comprobar fallos en el arranque de servicios.....	11
11. Saber si un servicio arranca en un nivel de ejecución determinado .....	12
12. Mostrar la configuración del servicio `cups` .....	13

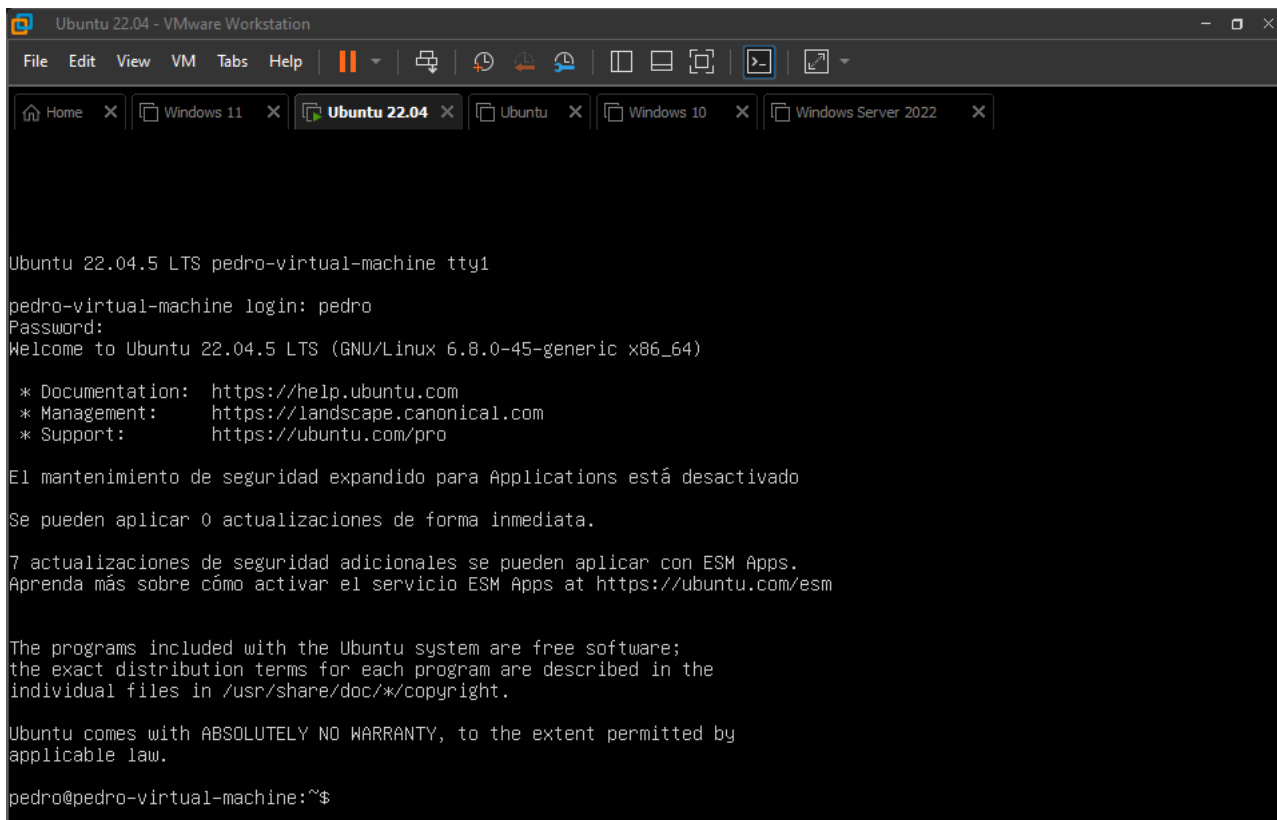
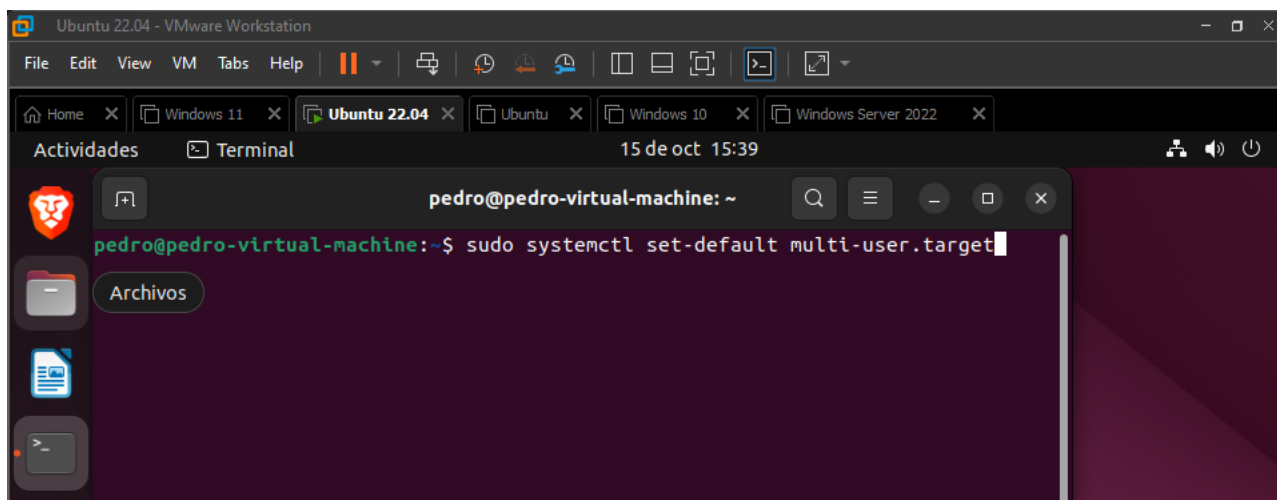
## 1. Arranque sin el servidor gráfico (sin GUI)

Para configurar el sistema para que arranque sin el servidor gráfico, usa:

**sudo systemctl set-default multi-user.target**

El comando hace que inicie el sistema en modo sin entorno gráfico. Si más adelante queremos revertir este cambio y volver al entorno gráfico, ejecutamos:

**sudo systemctl set-default graphical.target**



## 2. Diagnóstico del servicio SSH no arrancado

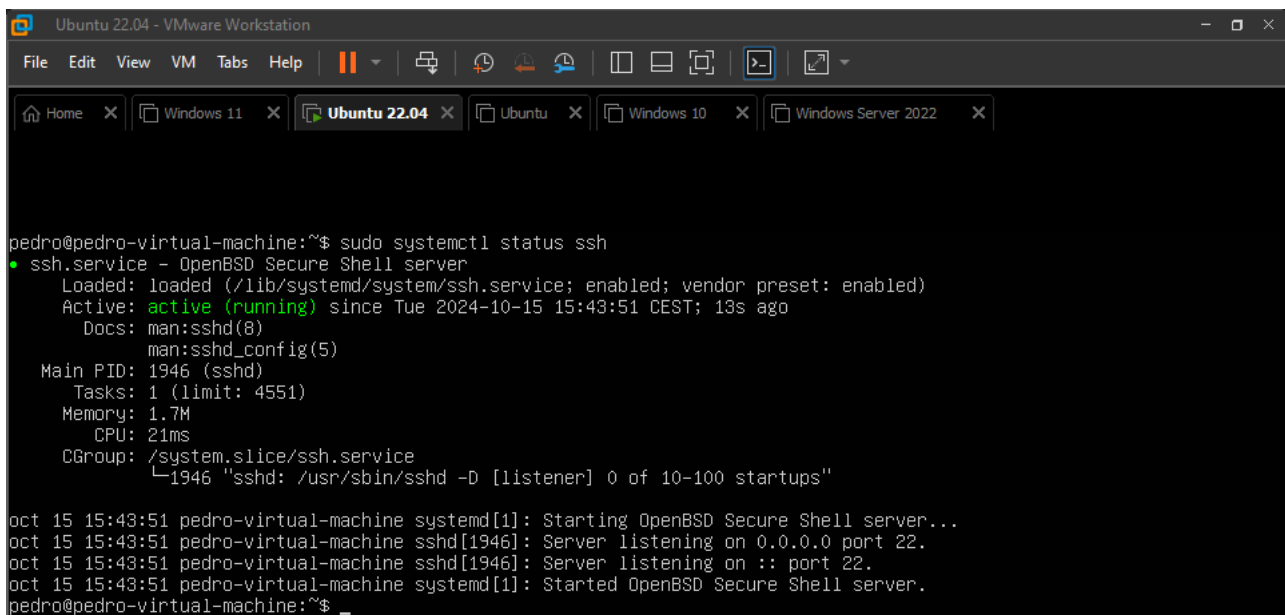
Antes de nada, tenemos que instalar el ssh en caso de no tenerlo instalado ejecutaremos lo siguiente:

```
sudo apt-get update
sudo apt-get install openssh-server
```

a) El servicio intentó arrancar, pero falló:

Verifica el estado del servicio SSH con:

```
sudo systemctl status ssh
```

A screenshot of a terminal window titled 'Ubuntu 22.04 - VMware Workstation'. The terminal shows the command 'sudo systemctl status ssh' and its output. The output indicates that the 'ssh.service' is loaded and enabled, and is currently 'active (running)'. It also shows the main PID (1946) and the tasks running under it. At the bottom, there are log messages from 'systemd' and 'sshd' showing the server starting and listening on port 22.

```
pedro@pedro-virtual-machine:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-10-15 15:43:51 CEST; 13s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 1946 (sshd)
      Tasks: 1 (limit: 4551)
    Memory: 1.7M
       CPU: 21ms
    CGroup: /system.slice/ssh.service
            └─1946 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

oct 15 15:43:51 pedro-virtual-machine systemd[1]: Starting OpenBSD Secure Shell server...
oct 15 15:43:51 pedro-virtual-machine sshd[1946]: Server listening on 0.0.0.0 port 22.
oct 15 15:43:51 pedro-virtual-machine sshd[1946]: Server listening on :: port 22.
oct 15 15:43:51 pedro-virtual-machine systemd[1]: Started OpenBSD Secure Shell server.
pedro@pedro-virtual-machine:~$ _
```

Revisa los logs del servicio SSH para más detalles del fallo:

```
journalctl -u ssh
```

A screenshot of a terminal window showing the output of the command 'journalctl -u ssh'. The output displays the same log messages as the previous screenshot, showing the SSH service starting and listening on port 22.

```
pedro@pedro-virtual-machine:~$ journalctl -u ssh
oct 15 15:43:51 pedro-virtual-machine systemd[1]: Starting OpenBSD Secure Shell server...
oct 15 15:43:51 pedro-virtual-machine sshd[1946]: Server listening on 0.0.0.0 port 22.
oct 15 15:43:51 pedro-virtual-machine sshd[1946]: Server listening on :: port 22.
oct 15 15:43:51 pedro-virtual-machine systemd[1]: Started OpenBSD Secure Shell server.
pedro@pedro-virtual-machine:~$ _
```

## b) El servicio no intentó arrancar:

Comprueba si el servicio está habilitado para arrancar automáticamente

**sudo systemctl is-enabled ssh**

```
pedro@pedro-virtual-machine:~$ sudo systemctl is-enabled ssh
enabled
pedro@pedro-virtual-machine:~$
```

Si no está habilitado, puedes hacerlo con:

**sudo systemctl enable ssh**

## 3. Aplicar cambios en la configuración del servicio `rsyslog`

Para parar un servicio es:

**systemctl stop ssh.service**

Para arrancar un servicio es:

**systemctl start ssh.service**

Es habitual que lo que se necesite sea detener y volver a iniciar un servicio para que éste adopte la nueva configuración aplicada. Para esto, existen dos opciones: reiniciar el servicio, lo que implica una detención seguida de un reinicio:

**systemctl restart ssh.service**

para recargar el servicio:

**systemctl reload ssh.service**

```
pedro@pedro-virtual-machine:~$ systemctl stop ssh.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to stop 'ssh.service'.
Authenticating as: pedro,,, (pedro)
Password:
==== AUTHENTICATION COMPLETE ====
pedro@pedro-virtual-machine:~$ systemctl start ssh.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: pedro,,, (pedro)
Password:
==== AUTHENTICATION COMPLETE ====
pedro@pedro-virtual-machine:~$ systemctl restart ssh.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'ssh.service'.
Authenticating as: pedro,,, (pedro)
Password:
==== AUTHENTICATION COMPLETE ====
pedro@pedro-virtual-machine:~$ systemctl reload ssh.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'ssh.service'.
Authenticating as: pedro,,, (pedro)
Password:
==== AUTHENTICATION COMPLETE ====
pedro@pedro-virtual-machine:~$ systemctl reload-or-restart ssh.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'ssh.service'.
Authenticating as: pedro,,, (pedro)
Password:
==== AUTHENTICATION COMPLETE ====
pedro@pedro-virtual-machine:~$
```

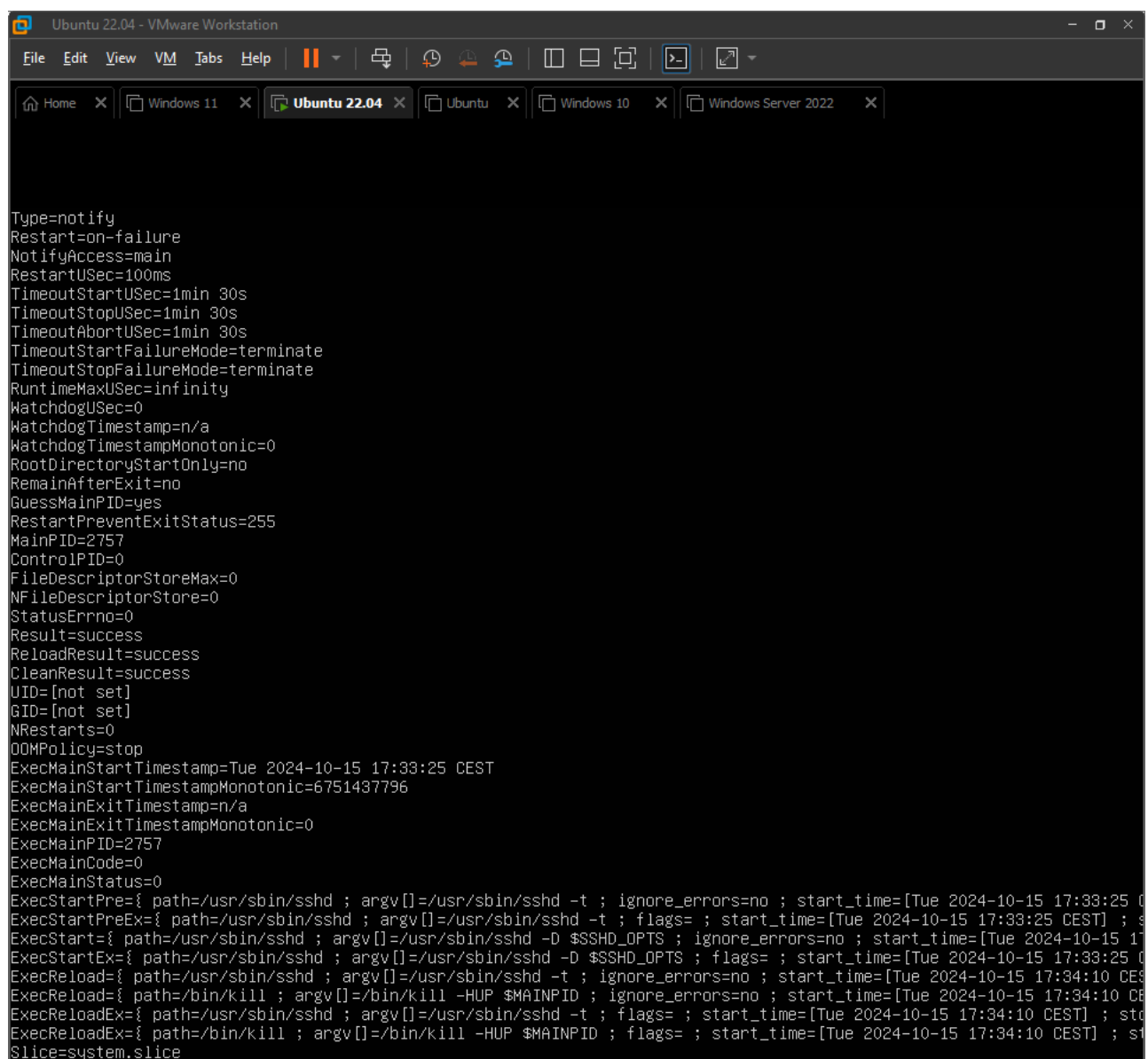
## 4. Por qué SSL intenta arrancar automáticamente

Cuando instalas un servicio como SSL, el sistema crea una unidad de servicio para SystemD en `/etc/systemd/system/` que lo configura para arrancar automáticamente. Esto ocurre porque el servicio se considera crítico para la gestión de tráfico seguro en la red.

## 5. Servicio en `multi-user.target` sin arrancar en `graphical.target`

Un servicio configurado para arrancar en `multi-user.target` puede no arrancar en `graphical.target` si su archivo de unidad tiene restricciones. Para verificar esto, usa:

**sudo systemctl show <nombre\_del\_servicio>**

A screenshot of a terminal window titled 'Ubuntu 22.04 - VMware Workstation'. The terminal displays the output of the command 'sudo systemctl show sshd'. The output lists various systemd unit properties for the 'sshd' service, such as 'Type=notify', 'Restart=on-failure', and 'ExecStart=/usr/sbin/sshd -t'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'VM', 'Tabs', and 'Help'. Below the menu bar is a toolbar with icons for file operations and window management. The terminal tabs at the top show 'Home', 'Windows 11', 'Ubuntu 22.04' (selected), 'Ubuntu', 'Windows 10', and 'Windows Server 2022'.

```
Type=notify
Restart=on-failure
NotifyAccess=main
RestartUsec=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
TimeoutAbortUsec=1min 30s
TimeoutStartFailureMode=terminate
TimeoutStopFailureMode=terminate
RuntimeMaxUsec=infinity
WatchdogUsec=0
WatchdogTimestamp=n/a
WatchdogTimestampMonotonic=0
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
RestartPreventExitStatus=255
MainPID=2757
ControlPID=0
FileDescriptorStoreMax=0
NFileDescriptorStore=0
StatusErrno=0
Result=success
ReloadResult=success
CleanResult=success
UID=[not set]
GID=[not set]
NRestarts=0
OOMPolicy=stop
ExecMainStartTimestamp=Tue 2024-10-15 17:33:25 CEST
ExecMainStartTimestampMonotonic=6751437796
ExecMainExitTimestamp=n/a
ExecMainExitTimestampMonotonic=0
ExecMainPID=2757
ExecMainCode=0
ExecMainStatus=0
ExecStartPre=[ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; ignore_errors=no ; start_time=[Tue 2024-10-15 17:33:25 CEST] ; st
ExecStartPreEx=[ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; flags= ; start_time=[Tue 2024-10-15 17:33:25 CEST] ; st
ExecStart=[ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -D $SSHDOPTS ; ignore_errors=no ; start_time=[Tue 2024-10-15 17:33:25 CEST] ; st
ExecStartEx=[ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -D $SSHDOPTS ; flags= ; start_time=[Tue 2024-10-15 17:33:25 CEST] ; st
ExecReload=[ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; ignore_errors=no ; start_time=[Tue 2024-10-15 17:34:10 CEST] ; st
ExecReloadEx=[ path=/usr/sbin/sshd ; argv[]=/usr/sbin/sshd -t ; flags= ; start_time=[Tue 2024-10-15 17:34:10 CEST] ; st
ExecReload=[ path=/bin/kill ; argv[]=/bin/kill -HUP $MAINPID ; ignore_errors=no ; start_time=[Tue 2024-10-15 17:34:10 CEST] ; st
ExecReloadEx=[ path=/bin/kill ; argv[]=/bin/kill -HUP $MAINPID ; flags= ; start_time=[Tue 2024-10-15 17:34:10 CEST] ; st
Slice=system.slice
```

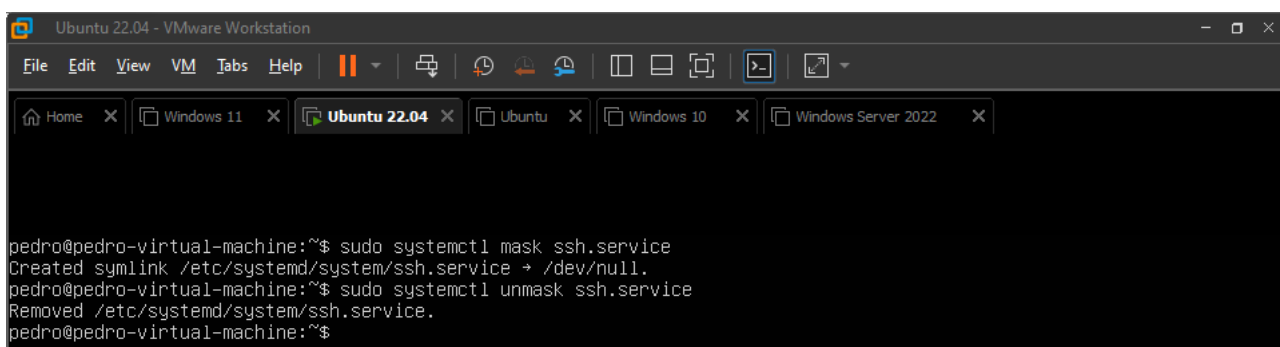
## 6. Impedir que un servicio arranque manualmente

Para impedir que un servicio arranque, incluso de forma manual, en mi caso usaré el `ssh.service`:

**`sudo systemctl mask ssh.servicie`**

Para revertir esta acción y permitir que el servicio vuelva a arrancar, ejecuta:

**`sudo systemctl unmask ssh.service`**

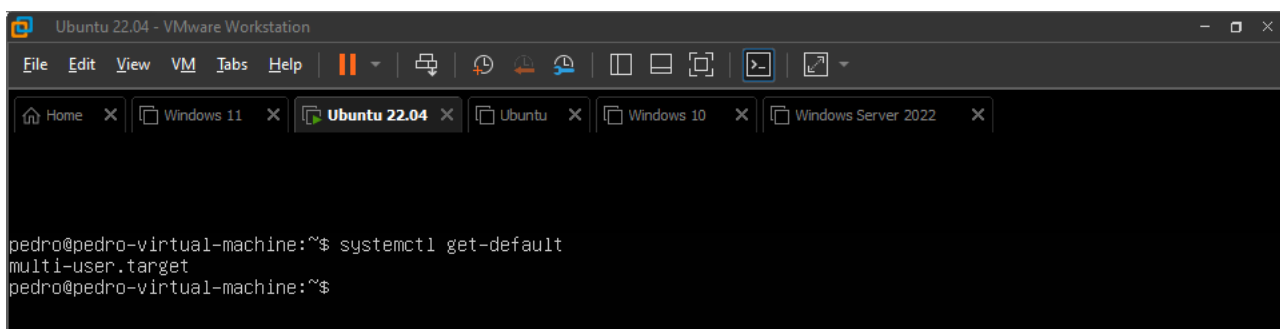


```
Ubuntu 22.04 - VMware Workstation
File Edit View VM Tabs Help
pedro@pedro-virtual-machine:~$ sudo systemctl mask ssh.service
Created symlink /etc/systemd/system/ssh.service → /dev/null.
pedro@pedro-virtual-machine:~$ sudo systemctl unmask ssh.service
Removed /etc/systemd/system/ssh.service.
pedro@pedro-virtual-machine:~$
```

## 7. Consultar el nivel de ejecución predeterminado

Para conocer el nivel de ejecución predeterminado del sistema, ejecuta:

**`systemctl get-default`**

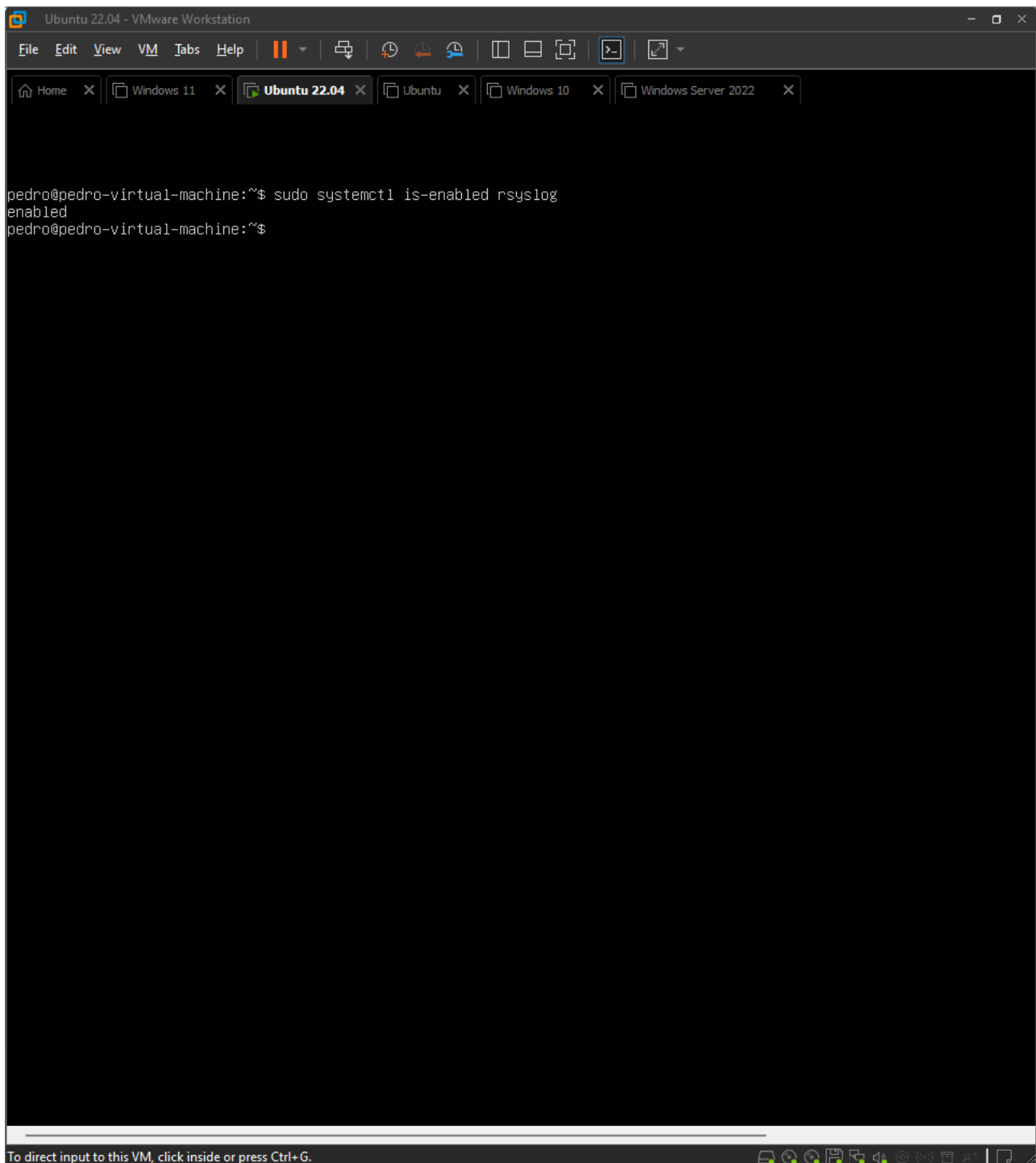


```
Ubuntu 22.04 - VMware Workstation
File Edit View VM Tabs Help
pedro@pedro-virtual-machine:~$ systemctl get-default
multi-user.target
pedro@pedro-virtual-machine:~$
```

## 8. Demostrar que `rsyslog` arranca automáticamente en el nivel de ejecución predeterminado

Comprueba si `rsyslog` está habilitado para arrancar automáticamente con el siguiente comando:

```
sudo systemctl is-enabled rsyslog
```



The screenshot shows a terminal window titled "Ubuntu 22.04 - VMware Workstation". The terminal has a dark background with white text. The command prompt is "pedro@pedro-virtual-machine:~\$". The command entered is "sudo systemctl is-enabled rsyslog". The output is "enabled". The prompt then returns to "pedro@pedro-virtual-machine:~\$". The terminal window has a menu bar with "File", "Edit", "View", "VM", "Tabs", and "Help". Below the menu bar is a toolbar with various icons. At the bottom of the window, there is a status bar that says "To direct input to this VM, click inside or press Ctrl+G." and a row of system icons.

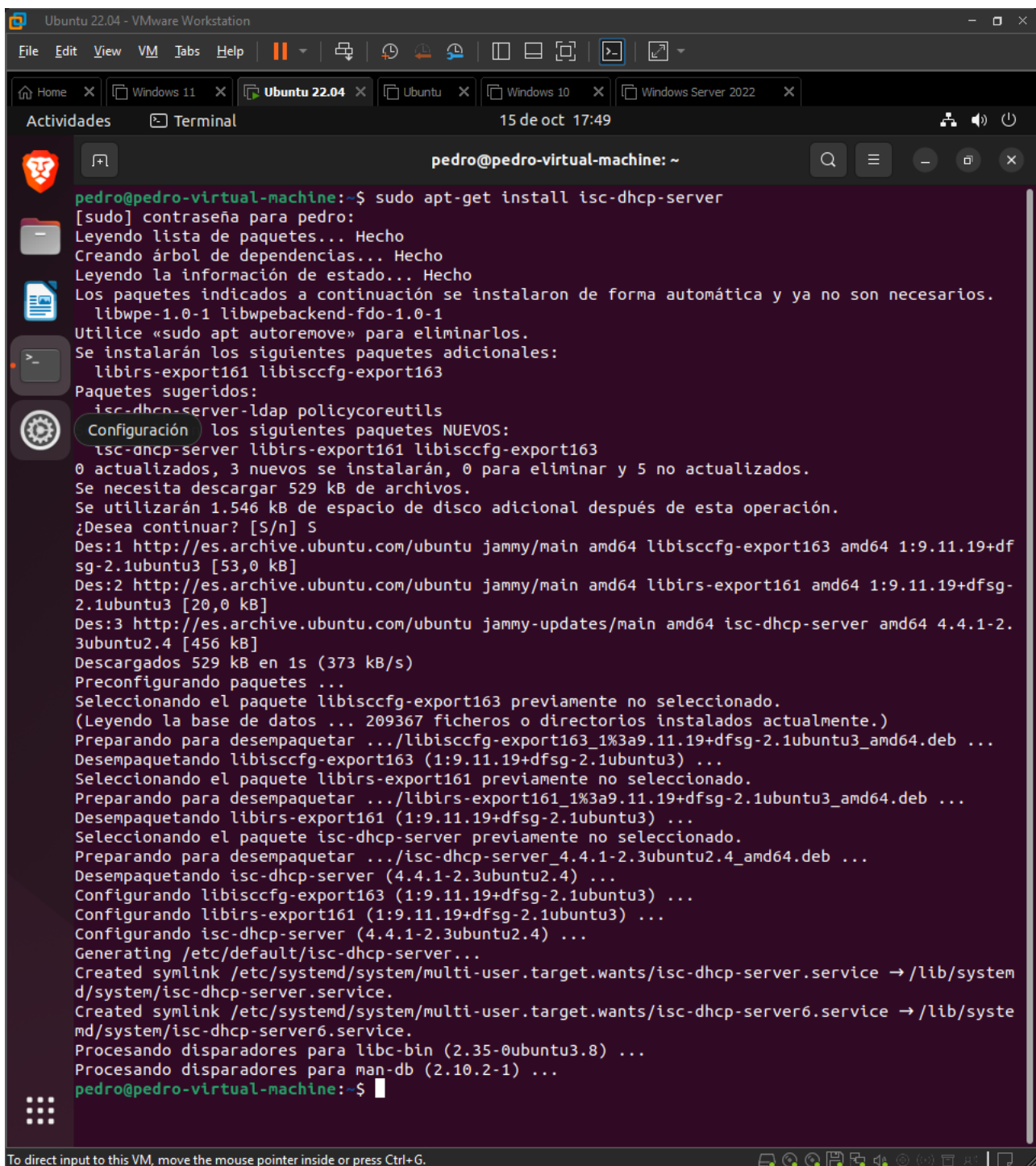
```
pedro@pedro-virtual-machine:~$ sudo systemctl is-enabled rsyslog
enabled
pedro@pedro-virtual-machine:~$
```



## 9. Comportamiento de `apt-get install isc-dhcp-server` y `systemctl disable isc-dhcp-server`

Al instalar el paquete `isc-dhcp-server`, se intentará iniciar el servicio automáticamente. Si lo deshabilitas con:

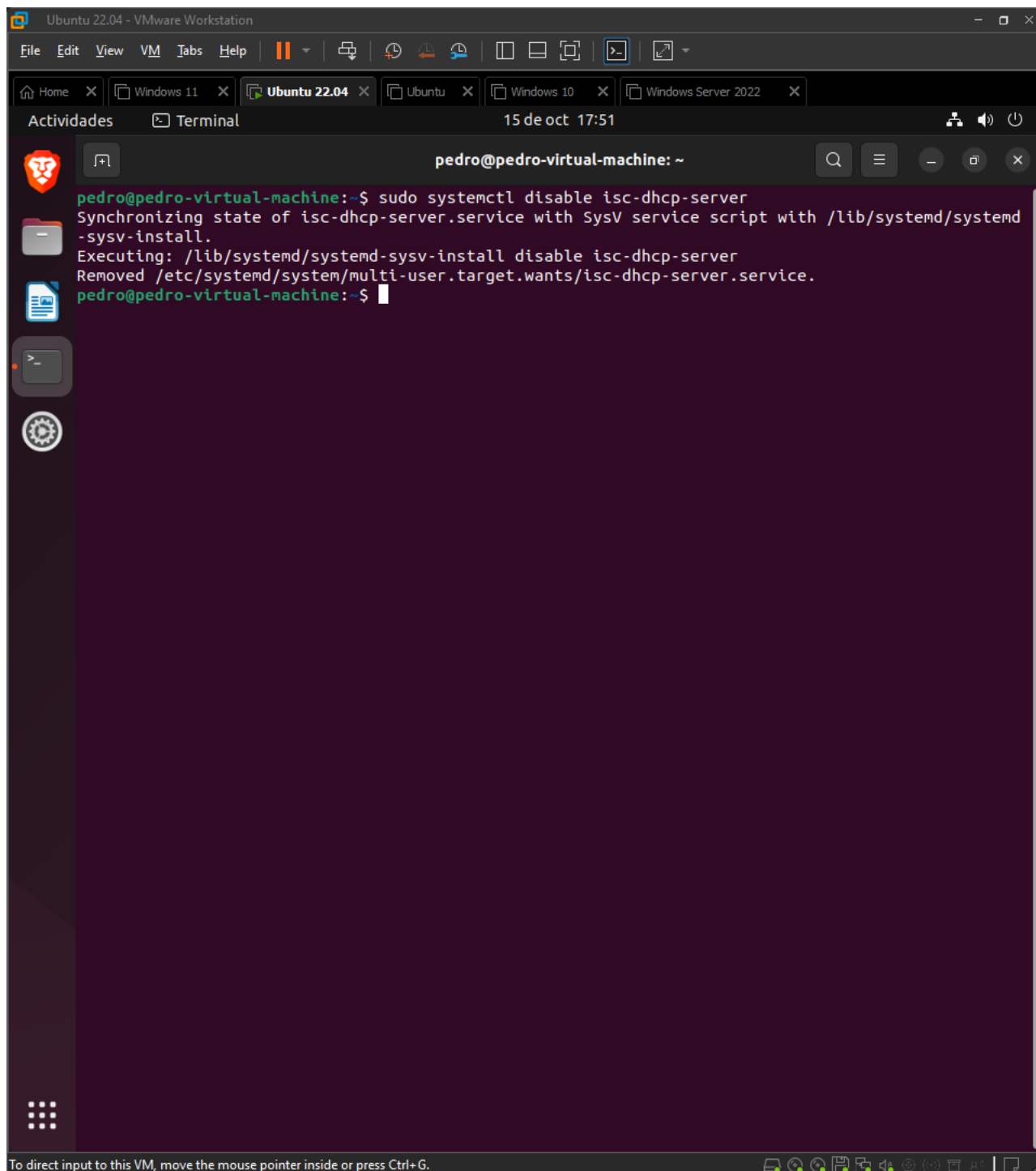
**sudo systemctl disable isc-dhcp-server**



```
pedro@pedro-virtual-machine:~$ sudo apt-get install isc-dhcp-server
[sudo] contraseña para pedro:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libirs-export161 libiscfg-export163
Paquetes sugeridos:
  isc-dhcp-server-ldap polycoreutils
Configuración los siguientes paquetes NUEVOS:
  isc-dhcp-server libirs-export161 libiscfg-export163
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
Se necesita descargar 529 kB de archivos.
Se utilizarán 1.546 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 libiscfg-export163 amd64 1:9.11.19+dfsg-2.1ubuntu3 [53,0 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 libirs-export161 amd64 1:9.11.19+dfsg-2.1ubuntu3 [20,0 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 isc-dhcp-server amd64 4.4.1-2.3ubuntu2.4 [456 kB]
Descargados 529 kB en 1s (373 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete libiscfg-export163 previamente no seleccionado.
(Leyendo la base de datos ... 209367 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libiscfg-export163_1%3a9.11.19+dfsg-2.1ubuntu3_amd64.deb ...
Desempaquetando libiscfg-export163 (1:9.11.19+dfsg-2.1ubuntu3) ...
Seleccionando el paquete libirs-export161 previamente no seleccionado.
Preparando para desempaquetar .../libirs-export161_1%3a9.11.19+dfsg-2.1ubuntu3_amd64.deb ...
Desempaquetando libirs-export161 (1:9.11.19+dfsg-2.1ubuntu3) ...
Seleccionando el paquete isc-dhcp-server previamente no seleccionado.
Preparando para desempaquetar .../isc-dhcp-server_4.4.1-2.3ubuntu2.4_amd64.deb ...
Desempaquetando isc-dhcp-server (4.4.1-2.3ubuntu2.4) ...
Configurando libiscfg-export163 (1:9.11.19+dfsg-2.1ubuntu3) ...
Configurando libirs-export161 (1:9.11.19+dfsg-2.1ubuntu3) ...
Configurando isc-dhcp-server (4.4.1-2.3ubuntu2.4) ...
Generating /etc/default/isc-dhcp-server...
Created symlink /etc/systemd/system/multi-user.target.wants/isc-dhcp-server.service → /lib/systemd/system/isc-dhcp-server.service.
Created symlink /etc/systemd/system/multi-user.target.wants/isc-dhcp-server6.service → /lib/systemd/system/isc-dhcp-server6.service.
Procesando disparadores para libc-bin (2.35-0ubuntu3.8) ...
Procesando disparadores para man-db (2.10.2-1) ...
pedro@pedro-virtual-machine:~$
```

El servicio no arrancará automáticamente en el próximo reinicio, pero puedes iniciarlo manualmente con:

**sudo systemctl start isc-dhcp-server**



The screenshot shows a terminal window titled "Ubuntu 22.04 - VMware Workstation". The terminal prompt is "pedro@pedro-virtual-machine: ~". The user has entered the command "sudo systemctl disable isc-dhcp-server". The output of the command is as follows:

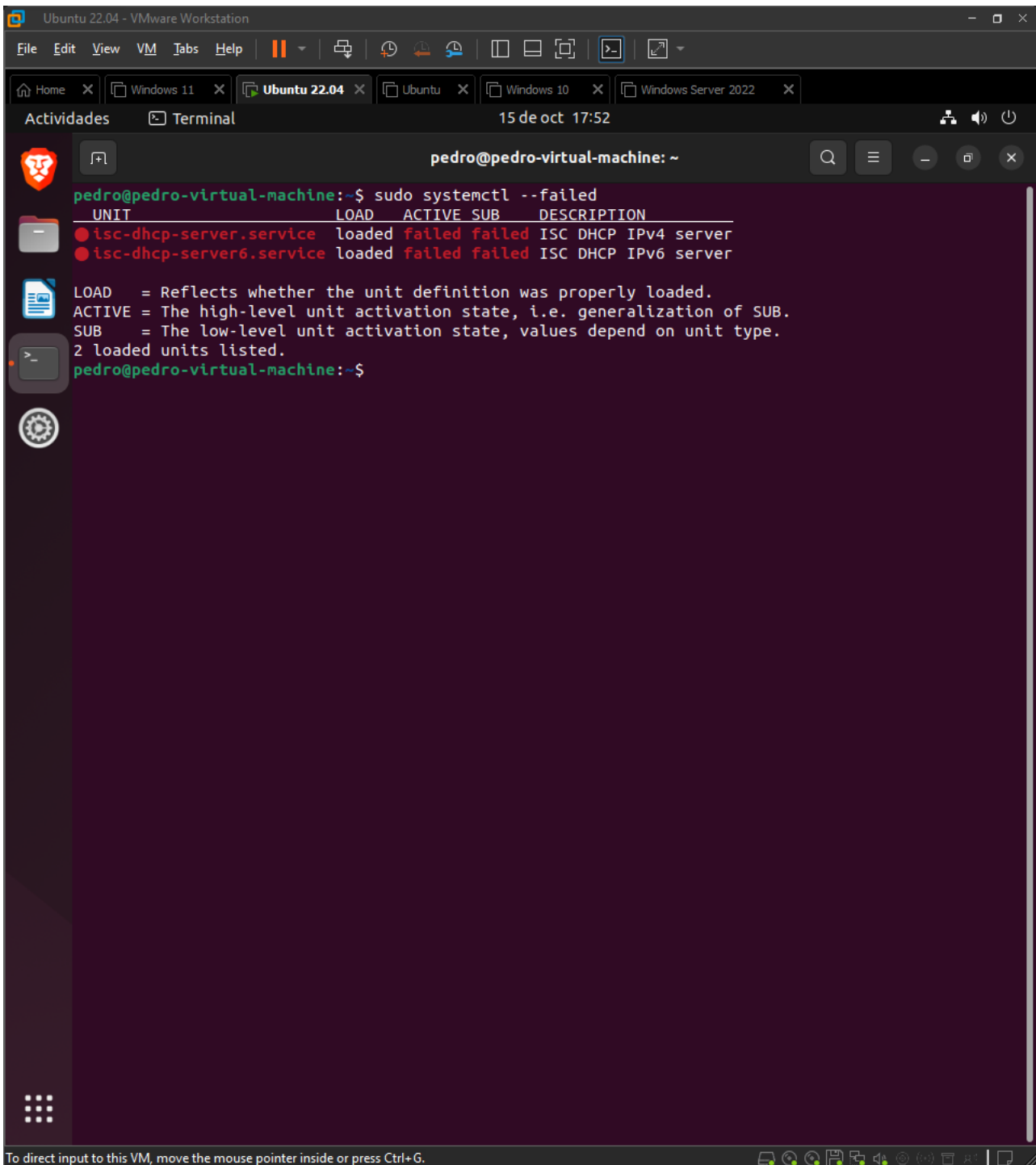
```
pedro@pedro-virtual-machine:~$ sudo systemctl disable isc-dhcp-server
Synchronizing state of isc-dhcp-server.service with SysV service script with /lib/systemd/systemd
-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable isc-dhcp-server
Removed /etc/systemd/system/multi-user.target.wants/isc-dhcp-server.service.
pedro@pedro-virtual-machine:~$
```

The terminal window also shows a sidebar with icons for Home, Files, Applications, and Settings. The top of the window displays the VMware Workstation interface with various tabs and a menu bar.

## 10. Comprobar fallos en el arranque de servicios

Para verificar si algún servicio falló durante el arranque del sistema:

**sudo systemctl --failed**



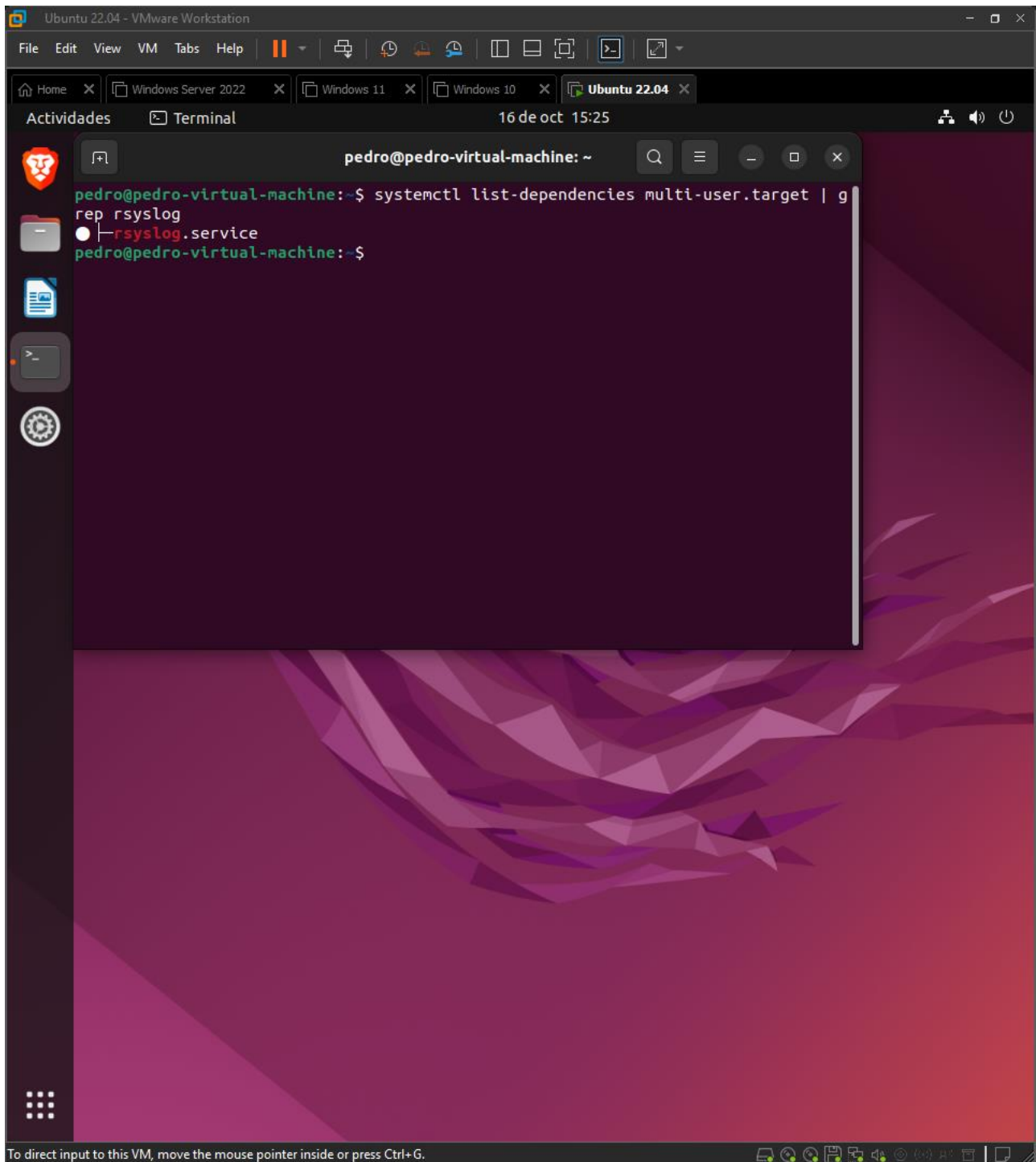
```
pedro@pedro-virtual-machine:~$ sudo systemctl --failed
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
●isc-dhcp-server.service           loaded failed failed  ISC DHCP IPv4 server
●isc-dhcp-server6.service          loaded failed failed  ISC DHCP IPv6 server

LOAD    = Reflects whether the unit definition was properly loaded.
ACTIVE  = The high-level unit activation state, i.e. generalization of SUB.
SUB     = The low-level unit activation state, values depend on unit type.
2 loaded units listed.
pedro@pedro-virtual-machine:~$
```

## 11. Saber si un servicio arranca en un nivel de ejecución determinado

Para verificar si un servicio se arranca en un *\*target\** sin reiniciar, usa:

**systemctl list-dependencies multi-user.target | grep rsyslog**



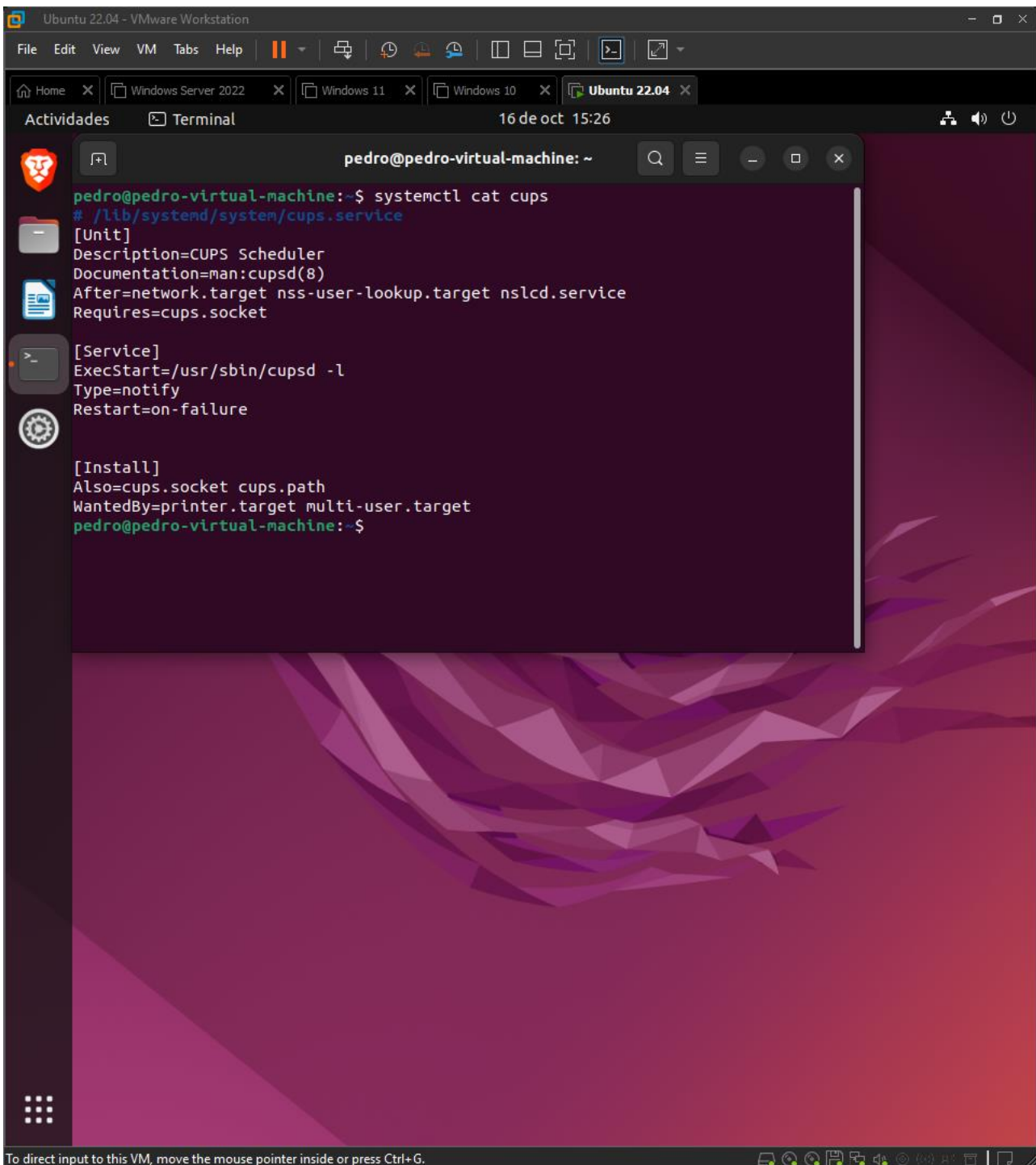
The screenshot shows a terminal window titled 'pedro@pedro-virtual-machine: ~' within a VMware Workstation environment. The terminal displays the command `systemctl list-dependencies multi-user.target | grep rsyslog` and its output, which is `rsyslog.service`. The terminal window is open on a desktop with a purple and red geometric background. The VMware interface includes a menu bar (File, Edit, View, VM, Tabs, Help) and a toolbar with various icons. The bottom status bar indicates 'To direct input to this VM, move the mouse pointer inside or press Ctrl+G.'

```
pedro@pedro-virtual-machine: ~  
pedro@pedro-virtual-machine:~$ systemctl list-dependencies multi-user.target | g  
rep rsyslog  
●├─rsyslog.service  
pedro@pedro-virtual-machine:~$
```

## 12. Mostrar la configuración del servicio `cups`

Para ver la configuración del servicio `cups` en tu sistema, ejecuta:

**systemctl cat cups**



The screenshot shows a terminal window titled 'pedro@pedro-virtual-machine: ~' within a VMware Workstation environment. The terminal displays the output of the command 'systemctl cat cups'. The output is as follows:

```
pedro@pedro-virtual-machine:~$ systemctl cat cups
# /lib/systemd/system/cups.service
[Unit]
Description=CUPS Scheduler
Documentation=man:cupsd(8)
After=network.target nss-user-lookup.target nsld.service
Requires=cups.socket

[Service]
ExecStart=/usr/sbin/cupsd -l
Type=notify
Restart=on-failure

[Install]
Also=cups.socket cups.path
WantedBy=printer.target multi-user.target
pedro@pedro-virtual-machine:~$
```

The terminal window is part of a desktop environment with a purple and red geometric background. The VMware Workstation interface is visible at the top, showing the 'Ubuntu 22.04' virtual machine. The bottom status bar indicates 'To direct input to this VM, move the mouse pointer inside or press Ctrl+G.'

