

Introducción a la Tecnología

&

Soporte Técnico

GUIA PASO A PASO:
EJERCICIO 2: DIAGNOSTICO DE PROBLEMAS COMUNES
EN LINUX

Aprende soporte técnico para Windows y Linux con
herramientas prácticas paso a paso.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

Guía de Diagnóstico y Gestión de Dispositivos en Ubuntu Jammy

Objetivo: Diagnosticar y gestionar dispositivos en Ubuntu Jammy, desde una visión general del sistema hasta la identificación de problemas y la gestión específica de cada dispositivo.

Parte 1: Visión General del Sistema

1.1: Listar Todos los Dispositivos Instalados

Usa `lshw` para obtener una lista completa del hardware del sistema:

```
sudo lshw -short
```

1.2: Verificar Controladores Instalados

Para ver los controladores cargados, usa:

```
lsmod
```

1.3: Actualizar y Configurar Controladores

Instala y configura controladores con `ubuntu-drivers`:

```
sudo apt update
sudo apt install ubuntu-drivers-common
ubuntu-drivers list
sudo ubuntu-drivers install
```

Para controladores específicos, como AMD o NVIDIA:

- AMD:

```
sudo apt install firmware-amd-graphics mesa-vulkan-drivers
```
- NVIDIA:

```
sudo apt install nvidia-driver-###
```

Parte 2: Diagnóstico Específico de Dispositivos

2.1: Diagnóstico de Dispositivos Utilizando `lspci`, `lsusb` y `dmesg`

A continuación se detalla cómo obtener información más precisa de dispositivos específicos, similar a la gestión en el Administrador de Dispositivos de Windows.

2.1.1: Adaptadores de Pantalla (VGA)

Para ver la información relacionada con los adaptadores de pantalla:

```
echo -e "Adaptadores de Pantalla\n======"
lspci | grep VGA
echo -e "=====\n"
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

2.1.2: Adaptadores de Red

Para obtener los adaptadores de red:

```
echo -e "Adaptadores de Red\n======"  
lspci | grep -i ethernet  
echo -e "=====\n"
```

2.1.3: Baterías

Si tu dispositivo tiene una batería, puedes obtener la información sobre su estado con:

```
echo -e "Baterías\n======"  
upower -i /org/freedesktop/UPower/devices/battery_BAT0  
echo -e "=====\n"
```

2.1.4: Controladoras ATA/IDE y Dispositivos de Almacenamiento

Para listar controladoras ATA/IDE y dispositivos de almacenamiento conectados:

```
echo -e "Controladoras ATA/IDE\n======"  
lspci | grep -i ata  
echo -e "=====\n"
```

2.1.5: Controladoras Bus Serie Universal (USB)

Para ver los dispositivos USB:

```
echo -e "Controladoras Bus Serie Universal\n======"  
lspci | grep -i usb  
echo -e "=====\n"
```

2.1.6: Controladoras de Sonido

Para los dispositivos de audio:

```
echo -e "Controladoras de Sonido\n======"  
lspci | grep -i audio  
echo -e "=====\n"
```

2.1.7: Dispositivos de Interfaz de Usuario (Teclado, Mouse)

Para listar teclados y ratones conectados:

```
echo -e "Dispositivos de Interfaz de Usuario\n======"  
lsusb | grep -i "keyboard\|mouse"  
echo -e "=====\n"
```

2.1.8: Dispositivos del Sistema (CPU, RAM, Disco)

Para obtener información del procesador:

```
echo -e "Dispositivos del Sistema - CPU\n======"  
lscpu  
echo -e "=====\n"
```

Para ver la memoria RAM disponible:

```
echo -e "Dispositivos del Sistema - RAM\n======"
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí](#)!

Aprende a resolver problemas comunes y mucho más.

```
free -h
echo -e "=====\\n"
```

Para el almacenamiento (discos y particiones):

```
echo -e "Dispositivos del Sistema - Disco\\n=====\\n"
df -h
echo -e "=====\\n"
```

2.1.9: Información General del Sistema

Para ver información general sobre todo el sistema:

```
echo -e "Equipo - Información general\\n=====\\n"
lshw -short
echo -e "=====\\n"
```

2.1.10: Monitores

Para ver los monitores conectados:

```
echo -e "Monitores\\n=====\\n"
xrandr --listmonitors
echo -e "=====\\n"
```

2.1.11: Unidades Ópticas

Si tienes una unidad óptica como un CD/DVD, puedes listar las unidades con:

```
echo -e "Unidades Ópticas\\n=====\\n"
lsblk | grep -i dvd
echo -e "=====\\n"
```

Parte 3: Diagnóstico de Errores Comunes con dmesg y journalctl

3.1: Ver Errores del Kernel con dmesg

dmesg muestra mensajes del kernel, y puedes buscar errores específicos con:

```
dmesg | grep -i error
dmesg | grep -i fail
dmesg | grep -i crit
dmesg | grep -i driver
dmesg | grep -i warning
dmesg | grep -i unclaimed
```

También puedes explorar los registros del kernel directamente con:

```
cat /var/log/kern.log | grep -i error
cat /var/log/kern.log | grep -i warning
cat /var/log/kern.log | grep -i fail
```

Otros logs donde puedes buscar errores relacionados incluyen:

```
cat /var/log/messages | grep -i error
cat /var/log/syslog | grep -i error
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```
cat /var/log/syslog | grep -i firmware
cat /var/log/syslog | grep -i driver
```

Para buscar dispositivos reconocidos pero sin controlador asignado, puedes usar:

```
lspci -v | grep -i "driver:"
lspci -v | grep -i "kernel modules:"
lsusb -v | grep -i "Driver"
dmesg | grep -i "not claimed"
dmesg | grep -i "no driver"
```

Para encontrar el ID de hardware y el ID de proveedor (similares a Windows), puedes usar los siguientes comandos:

```
lspci -nn
```

Este comando muestra una lista de dispositivos PCI junto con su ID de hardware y proveedor en el formato [VVVV:DDDD], donde VVVV es el Vendor ID (ID del fabricante) y DDDD es el Device ID (ID del dispositivo).

Ejemplo de salida:

```
00:1f.2 SATA controller [0106]: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E)
SATA Controller [AHCI mode] [8086:2829]
```

En este caso, el Vendor ID es 8086 (Intel) y el Device ID es 2829.

Para dispositivos USB, utiliza:

```
lsusb -v | grep -i "idVendor"
lsusb -v | grep -i "idProduct"
```

Esto te mostrará los IDs del fabricante y del producto de dispositivos conectados por USB.

3.2: Ver Registros del Sistema con journalctl

Puedes ver los errores del sistema con journalctl utilizando filtros como:

```
journalctl -p err
journalctl -p crit
journalctl -p fail
journalctl -p driver
journalctl -p warning
journalctl -p unclaimed
```

Errores comunes en dmesg y journalctl:

- **"firmware" no encontrado:** Problema con controladores.
- **"I/O error":** Problema con dispositivos de almacenamiento.
- **"No buffer space available":** Problemas de red.
- **"resource temporarily unavailable":** Conflictos de hardware.

La identificación de estos errores te ayudará a solucionar problemas específicos del sistema de manera efectiva.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

Parte 4: Simulación de Habilitar y Deshabilitar Dispositivos

En Ubuntu, puedes deshabilitar y habilitar dispositivos usando módulos del kernel. Este proceso es similar al habilitar y deshabilitar dispositivos en el Administrador de Dispositivos de Windows.

4.1: Deshabilitar un Dispositivo (Ejemplo: tarjeta de red eth0) Para deshabilitar un dispositivo, usa el comando `rmmod`:

```
sudo rmmod igb # Deshabilita la tarjeta de red
```

4.2: Habilitar un Dispositivo Para habilitarlo nuevamente, utiliza `modprobe`:

```
sudo modprobe igb # Habilita la tarjeta de red
```

4.3: Deshabilitar/Activar Dispositivos USB y de Red Deshabilitar y habilitar dispositivos USB de manera similar:

```
sudo rmmod usb_storage # Deshabilita el controlador USB  
sudo modprobe usb_storage # Habilita el controlador USB
```

4.4: Actualizar initramfs para que los cambios tomen efecto Para asegurarte de que los cambios realizados en los módulos del kernel persistan correctamente, actualiza el `initramfs`:

```
sudo update-initramfs -u # Actualiza el initramfs para aplicar los cambios
```

Parte 6: Diagnóstico Específico de Dispositivos

6.2: Verificar Dispositivos No Reconocidos

Si tienes un dispositivo conectado, pero no está siendo reconocido o no funciona correctamente, es importante revisar los logs del sistema para encontrar errores relacionados. En lugar de confiar únicamente en la salida de herramientas como `lspci` o `lsusb`, los comandos de logs te ayudarán a identificar si hay problemas con el dispositivo, como controladores faltantes, fallos de hardware o conflictos.

6.2.1: Comandos para Ver Errores del Sistema

Para encontrar posibles errores que puedan indicar un problema con un dispositivo conectado pero no reconocido, usa los siguientes comandos:

1. **Revisar los errores en los mensajes del kernel con `dmesg`:**

```
sudo dmesg | grep -i error
```

Este comando buscará mensajes de error generados por el kernel, que pueden indicar problemas con dispositivos conectados.

2. **Buscar fallos específicos en los mensajes del kernel:**

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí](#)!

Aprende a resolver problemas comunes y mucho más.

```
dmesg | grep -i fail
```

Aquí se buscan mensajes que contengan "fail" (fallo), que pueden mostrar problemas en el sistema relacionados con la conexión de dispositivos.

3. Revisar los registros del sistema con `journalctl`:

```
journalctl -p err
```

Este comando muestra los errores del sistema registrados, que pueden ser más generales, pero útiles para identificar problemas de hardware o software relacionados con dispositivos.

4. Ver los errores críticos en los registros del sistema con `journalctl`:

```
journalctl -p crit
```

`crit` busca mensajes críticos del sistema, que suelen ser los errores más graves que pueden afectar la funcionalidad de los dispositivos.

6.2.2: Interpretación de Errores Comunes

Al usar comandos como `dmesg`, `journalctl` o revisar logs, puedes encontrar ciertos errores relacionados con dispositivos o controladores. Aquí una lista con los mensajes más comunes:

1. **"firmware not found"**: Falta el controlador o firmware del dispositivo.
2. **"I/O error"**: Problemas de lectura/escritura en discos o dispositivos de almacenamiento.
3. **"No buffer space available"**: Problemas de red, como saturación o conflictos de tarjeta de red.
4. **"resource temporarily unavailable"**: Conflicto de recursos, el dispositivo no puede acceder a memoria/puertos.
5. **"device not found"**: El dispositivo no fue detectado por el sistema.
6. **"failed to load module"**: No se pudo cargar el módulo necesario para un dispositivo.
7. **"unable to allocate memory"**: No hay suficiente memoria para operar el dispositivo.
8. **"no such device"**: El dispositivo no existe o no es reconocido por el sistema.
9. **"operation not supported"**: La operación no es compatible con el dispositivo.
10. **"segmentation fault"**: Acceso a memoria incorrecta, puede ser un controlador defectuoso.
11. **"timeout"**: La operación excedió el tiempo de espera, generalmente por problemas de comunicación.
12. **"kernel panic"**: El núcleo del sistema se detuvo, a menudo por problemas graves de hardware o controladores.
13. **"invalid argument"**: Argumento incorrecto o configuración errónea para la operación.
14. **"UNCLAIMED"** o **"desconocidos"**: lo que indica que no tienen un controlador asignado.

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí](#)!

Aprende a resolver problemas comunes y mucho más.

Conclusión

Si un dispositivo no es reconocido, comandos como **dmesg** y **journalctl** son esenciales para identificar errores. Herramientas como **lshw**, **modprobe** y **rmmod** permiten diagnosticar y gestionar dispositivos en Ubuntu Jammy, de forma similar al Administrador de Dispositivos de Windows.

Herramientas gráficas útiles:

1. **GNOME Device Manager**: Administra dispositivos gráficamente (`sudo apt install gnome-device-manager`) ya no disponible en ubuntu jammy.
2. **HardInfo**: Información detallada del hardware (`sudo apt install hardinfo`).
3. **inxi**: Muestra datos del sistema en terminal (`sudo apt install inxi`).
4. **KInfoCenter** (KDE): Información de hardware para entornos KDE (`sudo apt install kinfocenter`).
5. **Ubuntu "Controladores Adicionales"**: Configura controladores desde *Software y actualizaciones*.

Estas herramientas combinan accesibilidad y diagnóstico efectivo.

Glosario de Clasificación de Drivers en Linux (usando lsmod)

1. Módulos de Red

- Ejemplo: `e1000e`, `mptcp_diag`

2. Módulos de Entrada (Teclado, Mouse, Joypads)

- Ejemplo: `joydev`, `usbhid`, `hid_generic`

3. Módulos de Audio

- Ejemplo: `snd_hda_intel`, `snd_pcm`, `snd_seq`

4. Módulos Gráficos (Pantalla)

- Ejemplo: `i915`, `drm`, `ttm`

5. Módulos de Gestión de Energía

- Ejemplo: `intel_rapl_msr`, `acpi_cpufreq`

6. Módulos de Virtualización

- Ejemplo: `kvm`, `vboxdrv`

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí](#)!

Aprende a resolver problemas comunes y mucho más.

7. Módulos de Controladores de Entrada y Salida (I/O)

- **Ejemplo:** usb_storage, pata_acpi

8. Módulos de Seguridad y Criptografía

- **Ejemplo:** aesni_intel, cryptd, sha256_ssse3

9. Módulos del Sistema

- **Ejemplo:** coretemp, i2c_algo_bit

10. Módulos de Redes y Cortafuegos

- **Ejemplo:** nf_conntrack, ip6t_REJECT

11. Módulos de Almacenamiento y Dispositivos de Disco

- **Ejemplo:** sd_mod, ext4, raid

12. Otros Módulos del Sistema

- **Ejemplo:** parport, binfmt_misc
 - **Descripción:** Esta categoría incluye módulos que no encajan en las categorías anteriores, pero que son necesarios para otras funciones del sistema, como la compatibilidad con impresoras, o la carga de binarios para otros entornos de ejecución.
-

Extras:

```
#!/bin/bash
```

```
# Palabras clave de errores comunes
```

```
errores=(
```

```
    "firmware not found"
```

```
    "I/O error"
```

```
    "No buffer space available"
```

```
    "resource temporarily unavailable"
```

```
    "device not found"
```

```
    "failed to load module"
```

```
    "unable to allocate memory"
```

¿Quieres aprender más sobre soporte técnico y otros temas?

¡Consigue el libro completo [aquí!](#)

Aprende a resolver problemas comunes y mucho más.

```
"no such device"
"operation not supported"
"segmentation fault"
"timeout"
"kernel panic"
"invalid argument"
"error"
"fail"
"crit"
"unclaimed"
"unknown"
)
```

```
# Buscar los errores en dmesg
echo "\nBuscando errores en dmesg..."
for error in "${errores[@]}"; do
    echo -e "\nBuscando: '$error' en dmesg"
    dmesg | grep -i "$error"
done
```

```
# Buscar los errores en journalctl
echo -e "\nBuscando errores en journalctl..."
for error in "${errores[@]}"; do
    echo -e "\nBuscando: '$error' en journalctl"
    journalctl | grep -i "$error"
done
```

¿Quieres aprender más sobre soporte técnico y otros temas?
¡Consigue el libro completo [aquí!](#)
Aprende a resolver problemas comunes y mucho más.