**Práctica Examen RA2 (10.12.25)**

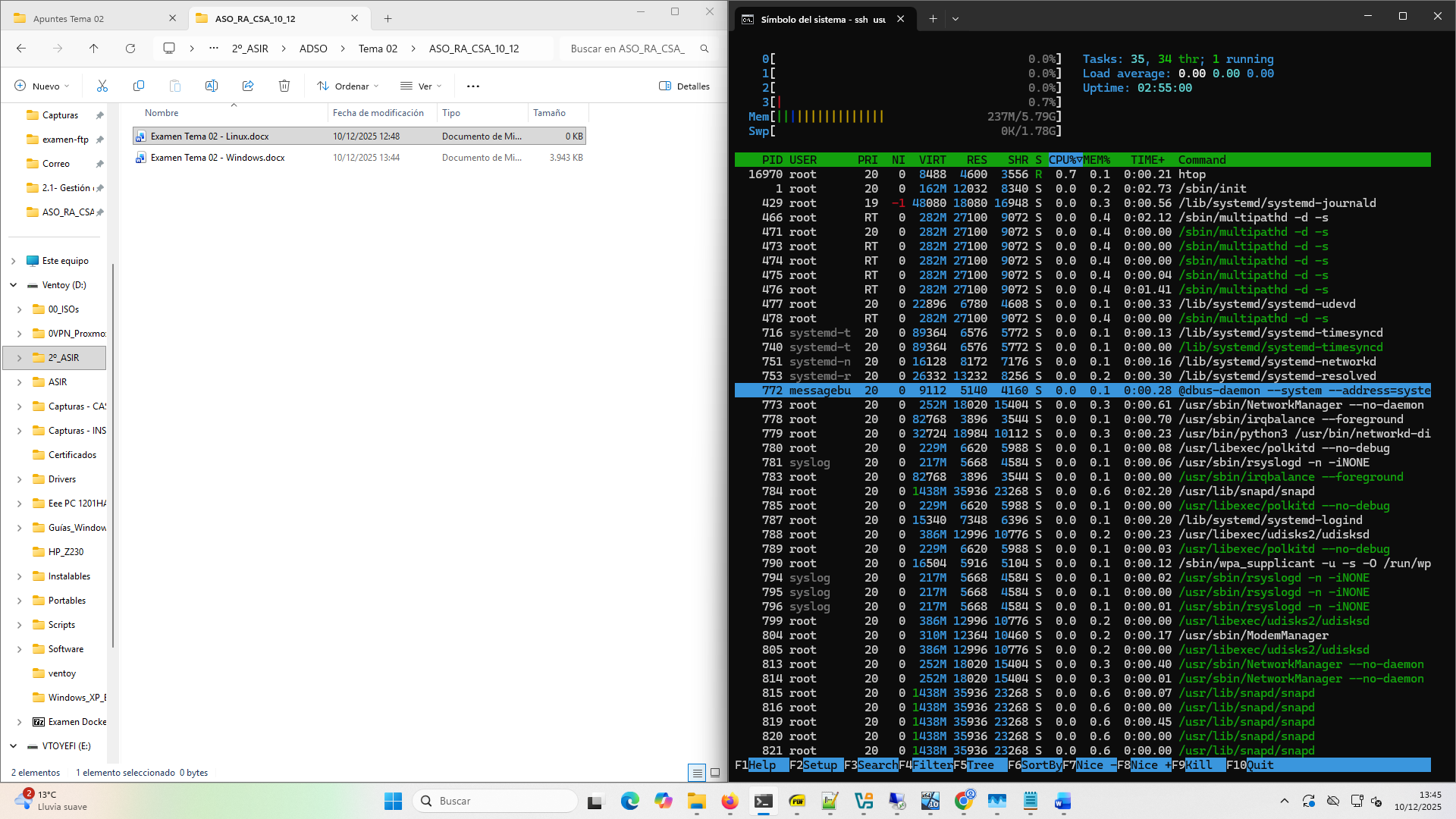
**Linux**

**Cristóbal Suárez Abad**

Un servidor Linux está presentando lentitud y un uso elevado de CPU y memoria. Tu tarea como técnico/a de sistemas es analizar, identificar y resolver las causas del problema, tomando decisiones justificadas sobre qué procesos mantener, modificar o finalizar.

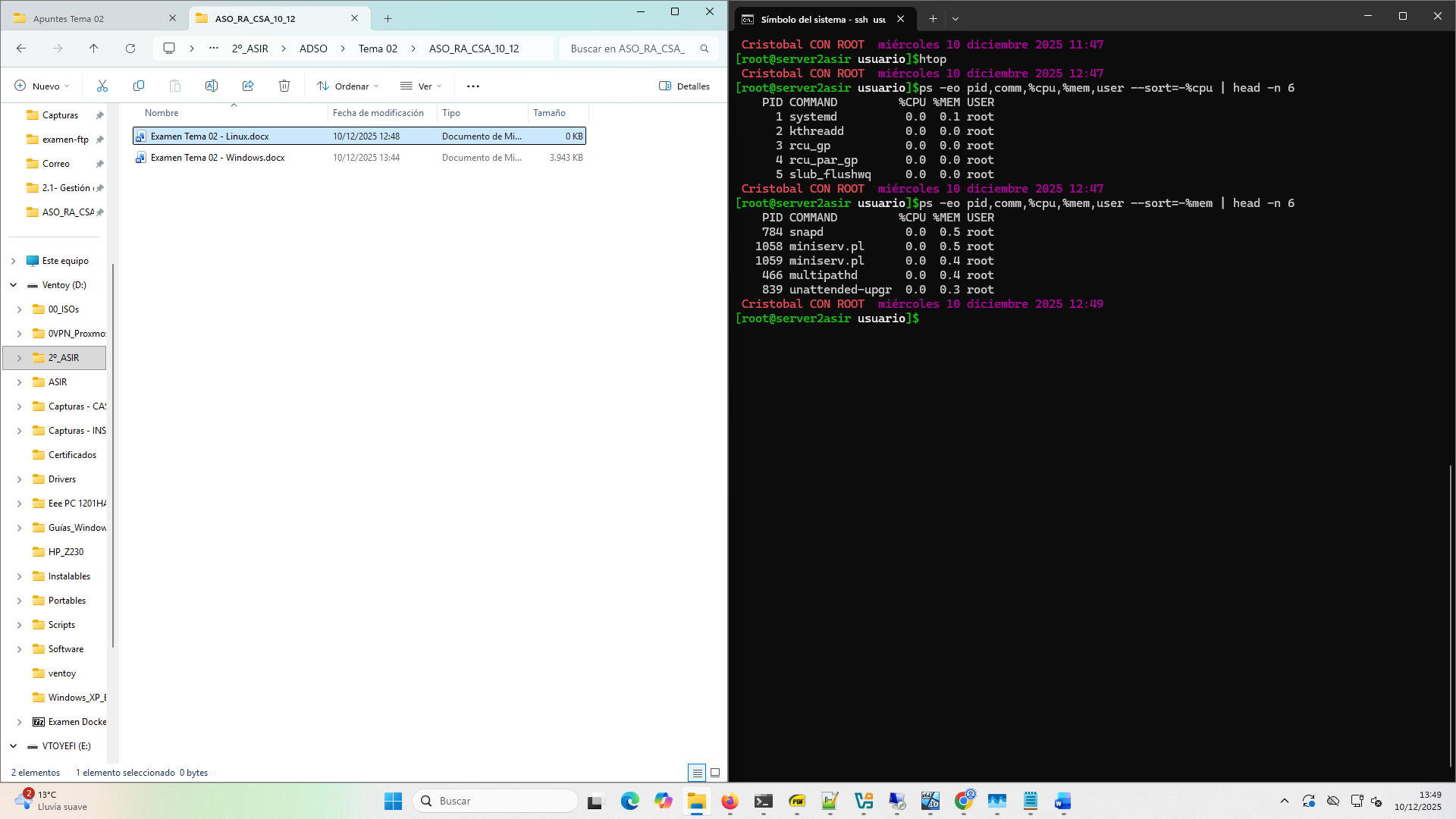
**Análisis inicial del sistema**

* Observa el consumo de CPU, memoria y carga del sistema en el momento actual.
* Identifica los 5 procesos con mayor impacto en el rendimiento (cpu o RAM).
* Determina si pertenecen al sistema, a servicios críticos o a usuarios.
* Observa el consumo de CPU, memoria y carga del sistema en el momento actual.



* Identifica los 5 procesos con mayor impacto en el rendimiento (cpu o RAM).

Podrías usar HTOP o también: Los que más CPU y también los que más RAM consumen:



* Determina si pertenecen al sistema, a servicios críticos o a usuarios.

Como se pueden ver en la imagen de arriba, pertenecen todos al sistema (usan el usuario root).

**Diagnóstico detallado de procesos**

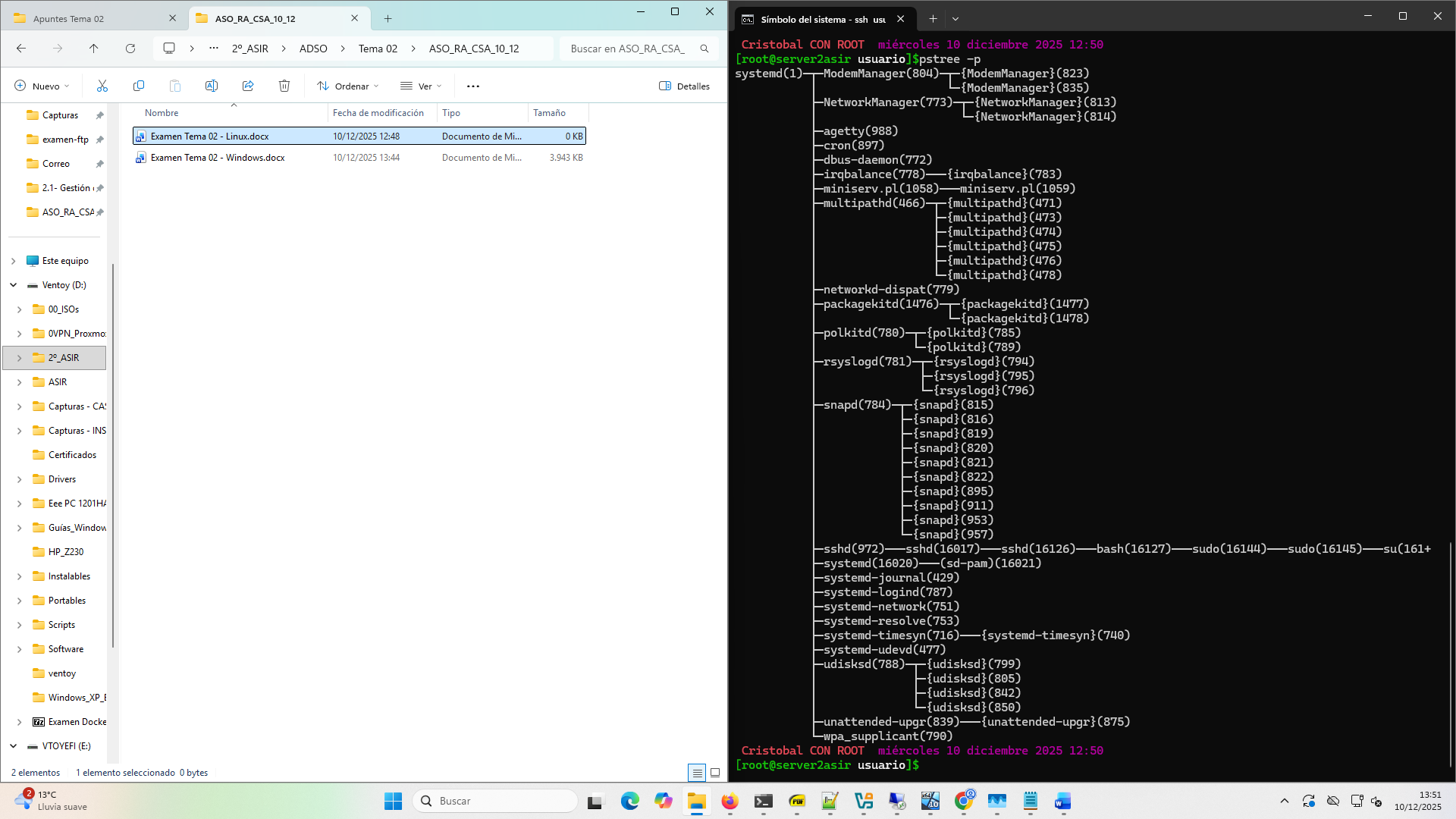
* Inspecciona los procesos sospechosos y sus relaciones jerárquicas.
* Escoge **uno de los procesos detectados como sospechosos** y analiza su árbol de procesos.  
  Indica:

·         Proceso padre (PID y nombre)

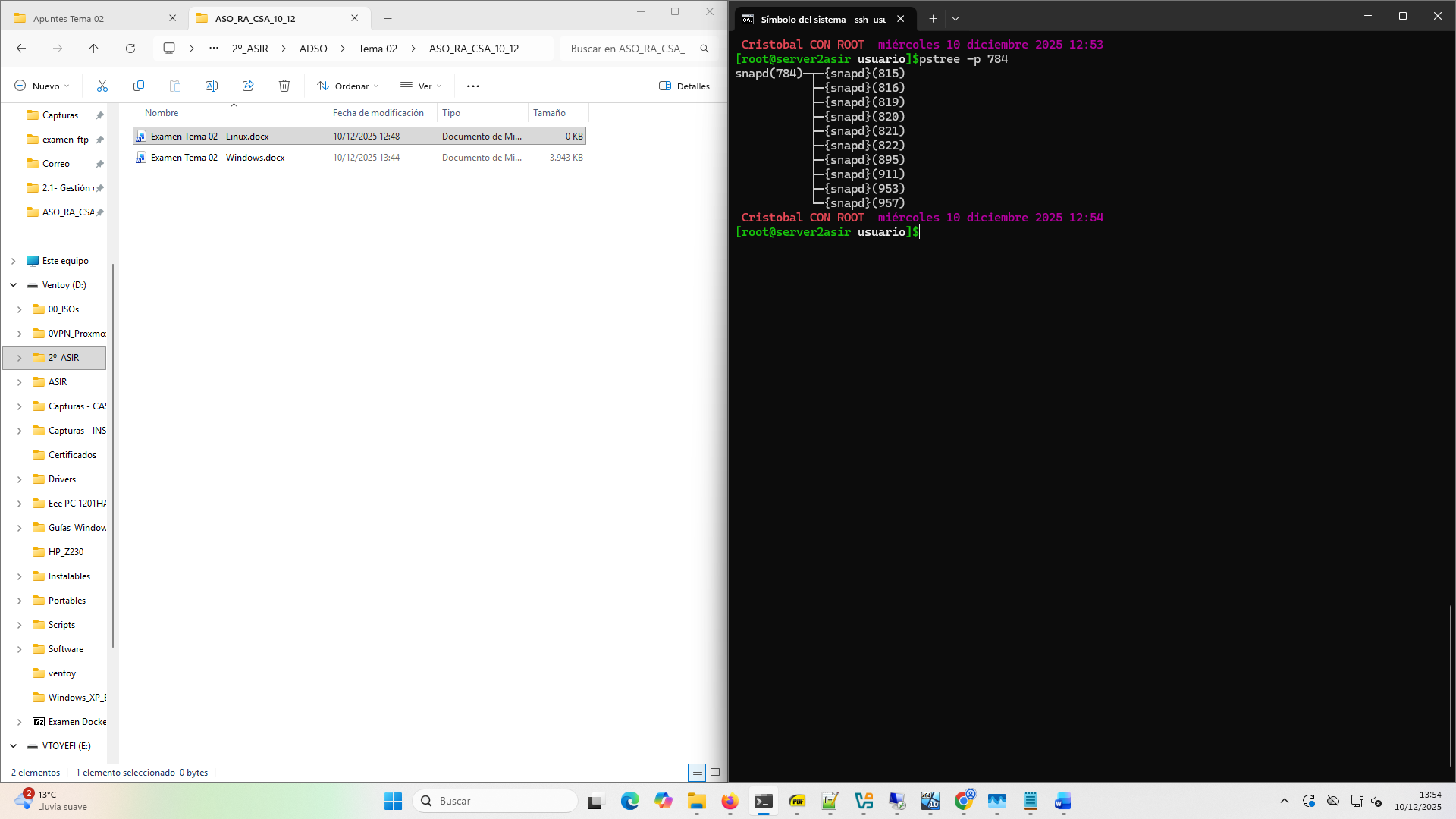
·         Hijos (si los tiene)

·         ¿La relación es normal o anómala?

Vamos a usar de ejemplo a systemd porque no se ven procesos sospechosos a simple vista. Con el comando “**pstree -p + (id del proceso)**” podemos ver el árbol del proceso. La relación parece normal



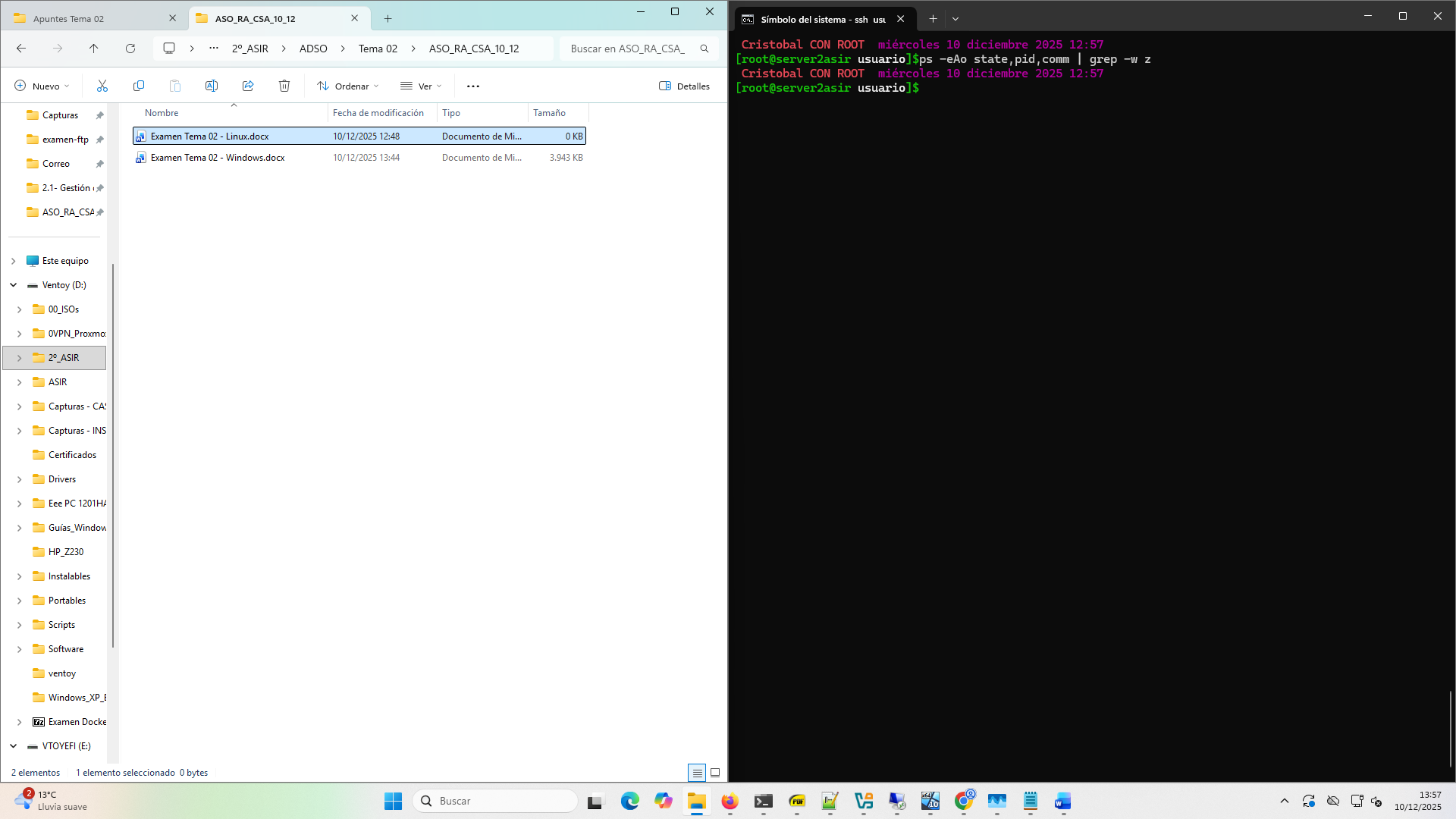
Otro ejemplo sería snapd.



* Verifica si existen procesos huérfanos, zombis o en estado de espera prolongado (más de un minuto).
* <https://www.site24x7.com/learn/linux/how-to-kill-zombie-process-in-linux.html>

**ps -eAo state,pid,comm | grep -w Z**

En Nuestro caso, como se puede ver en la imagen de abajo, no hay ningún proceso Zombie.

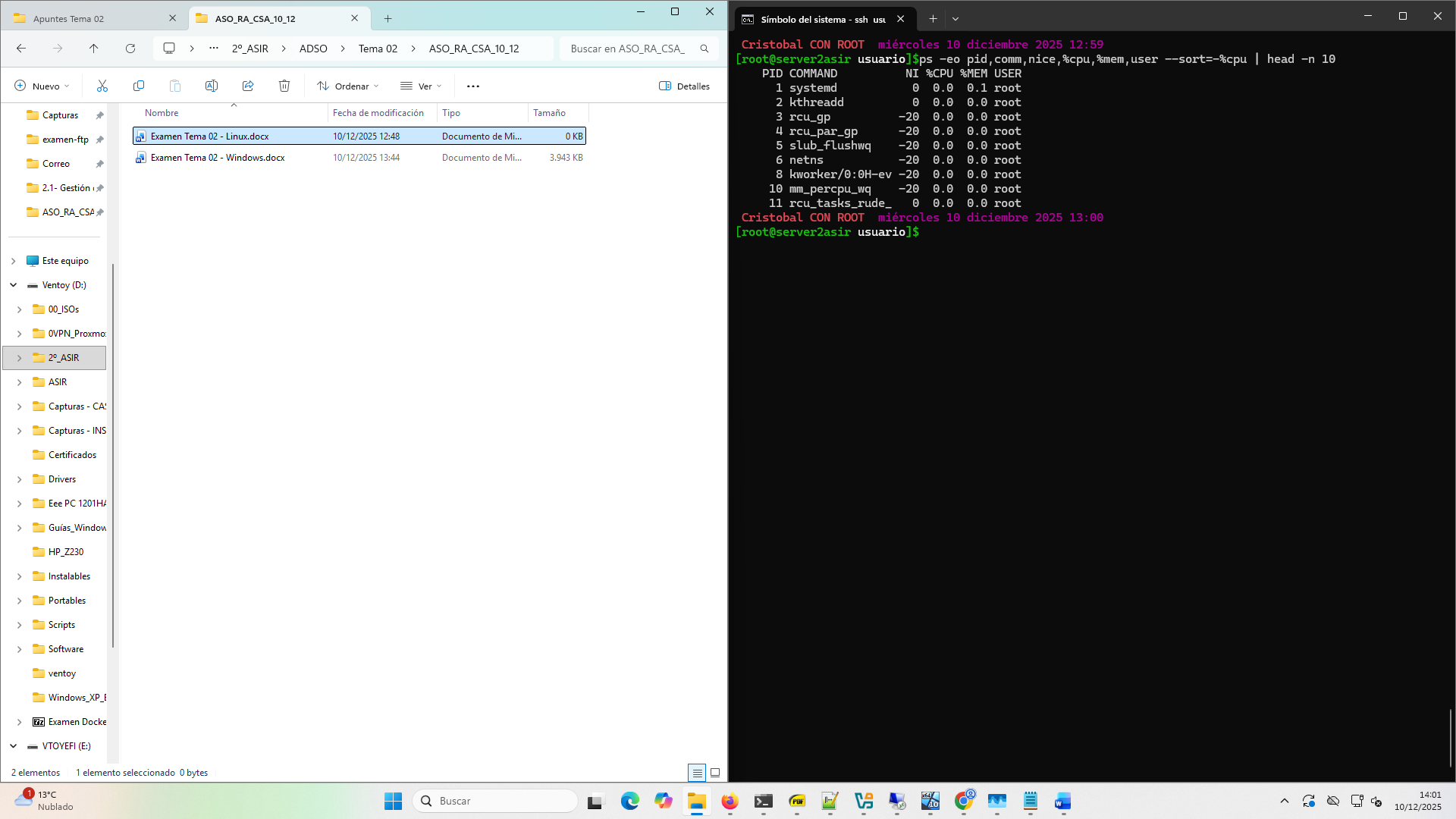


* Comprueba si alguno se ejecuta con permisos de root sin justificación.

**Optimización del sistema**

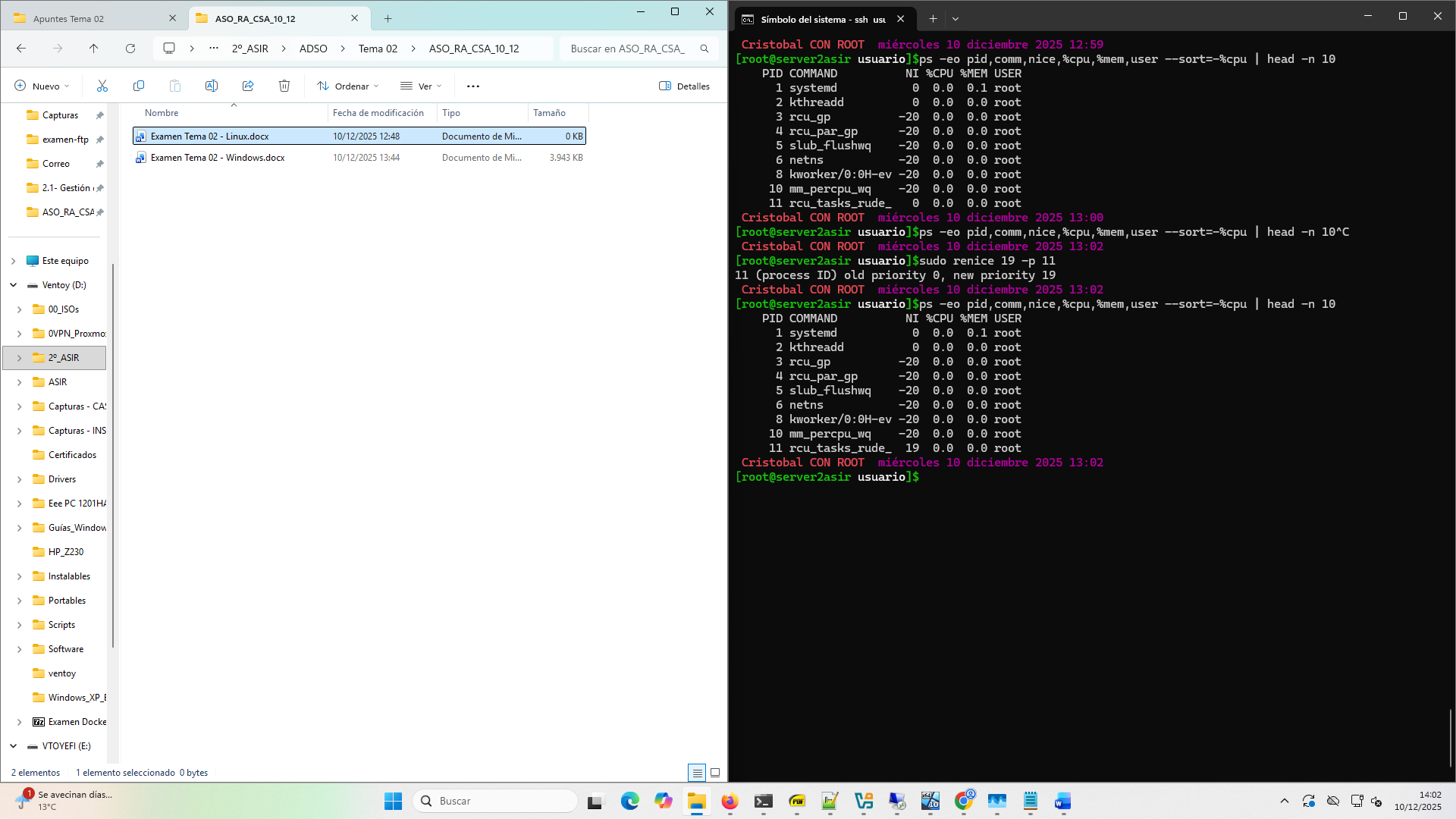
* Comprueba la prioridad de los procesos. Modifica la prioridad de uno asignándole la más baja posible. (renice)
* Finaliza de forma controlada (no violenta) los procesos innecesarios o que consuman recursos en exceso.
* Comprueba si la carga general mejora tras tus acciones.

Comprobamos la prioridad de los procesos (tercera columna “NI”): Mostramos los 11 primeros. Los valores pueden ir desde –20 (máxima prioridad), hasta 19 (mínima prioridad)



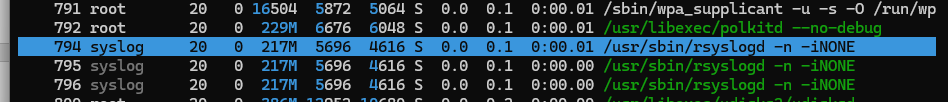
Modifica la prioridad de uno asignándole la más baja posible. (renice)

sudo renice 20 -p 11 🡪 Lo hemos dejado casi en lo más bajo posible (aunque pongas 20 te lo deja en 19).



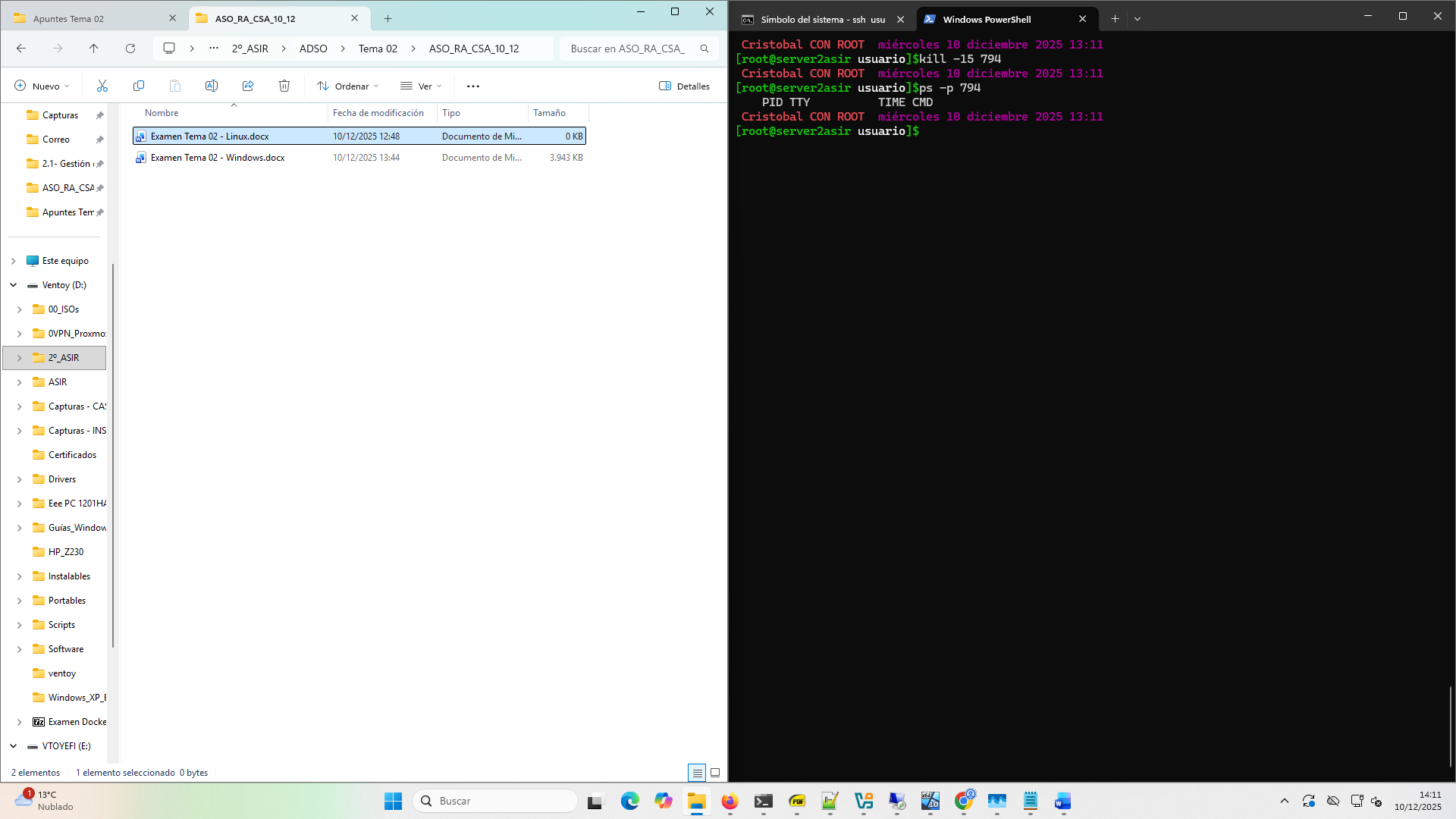
Finaliza de forma controlada (no violenta) los procesos innecesarios o que consuman recursos en exceso.

Como no hay ninguno que consuma en exceso, vamos a eliminar uno de syslog.



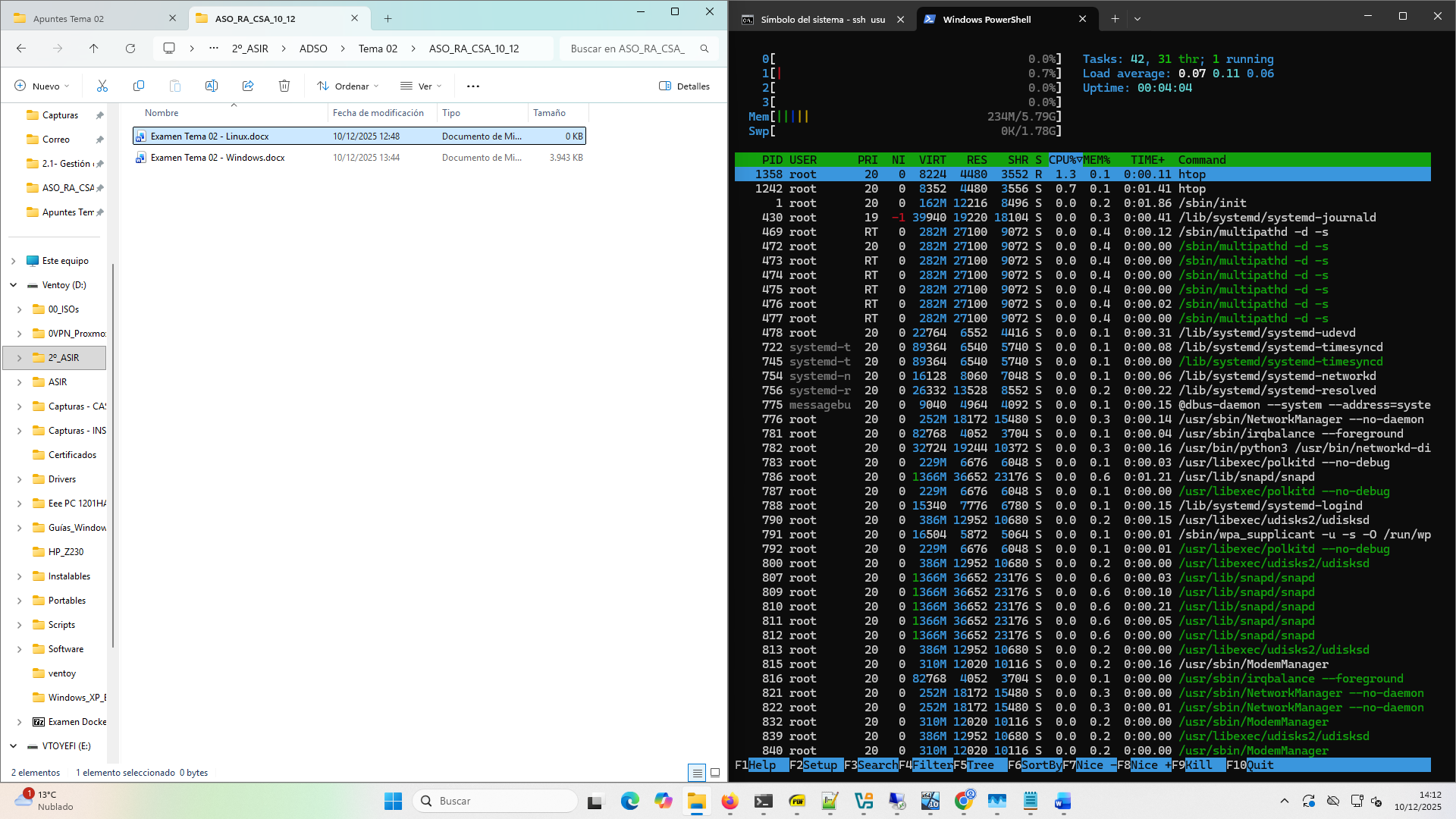
Usamos: kill -15 794

Ya no está.



* Comprueba si la carga general mejora tras tus acciones.

Hay pocos cambios, porque no había ningún proceso que consumiese mucho.



**Seguridad y verificación**

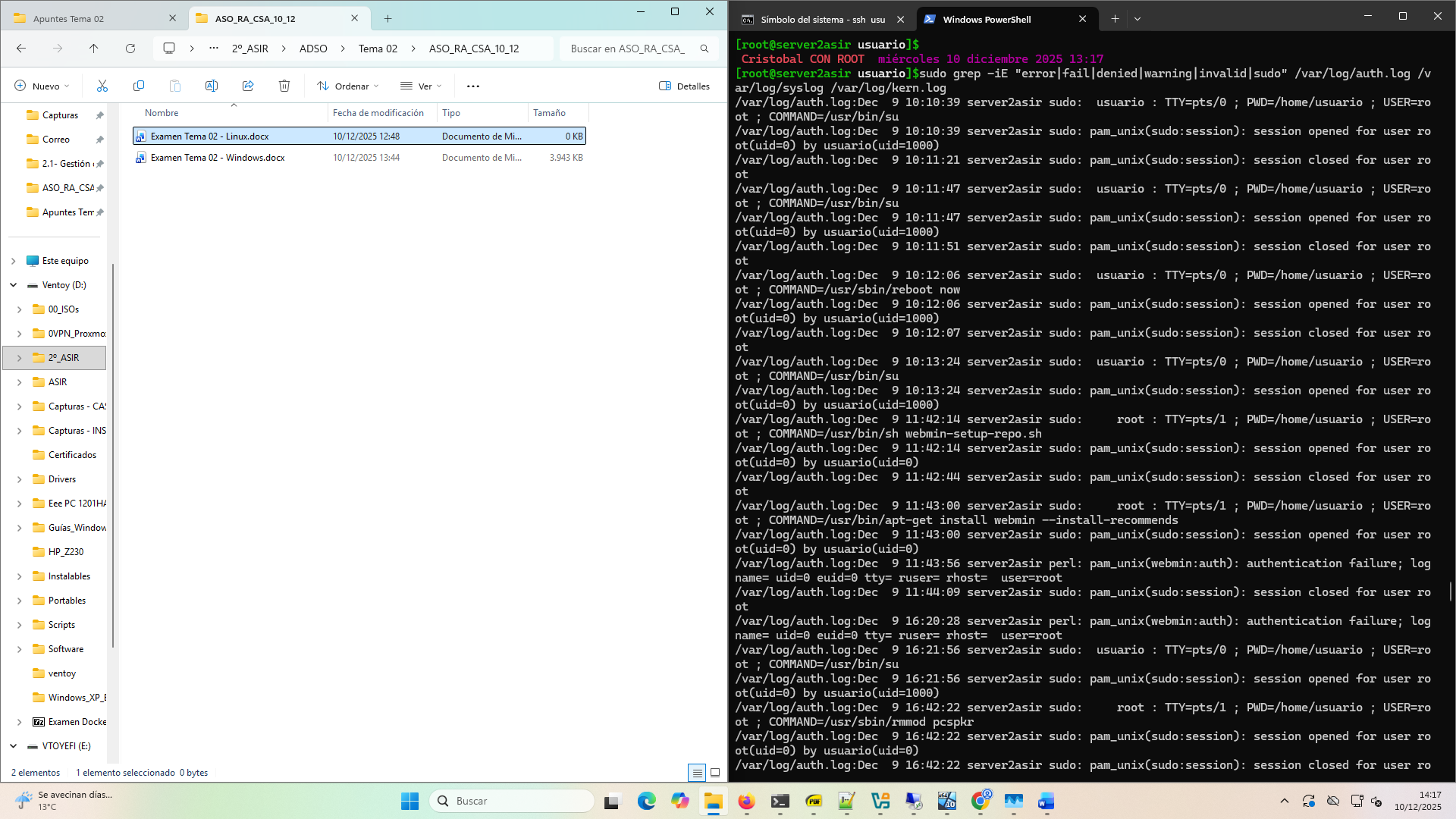
* Revisa los logs del sistema en busca de errores o intentos de ejecución no autorizada. Haz que esos log se guarden en un archivo llamado EXAMEN\_TusIniciales.txt

**Los logs principales están en /var/log. Y los más importantes:**

* **/var/log/syslog → registro general del sistema (Debian/Ubuntu)**
* **/var/log/auth.log → intentos de login y autenticación**
* **/var/log/kern.log → mensajes del kernel**

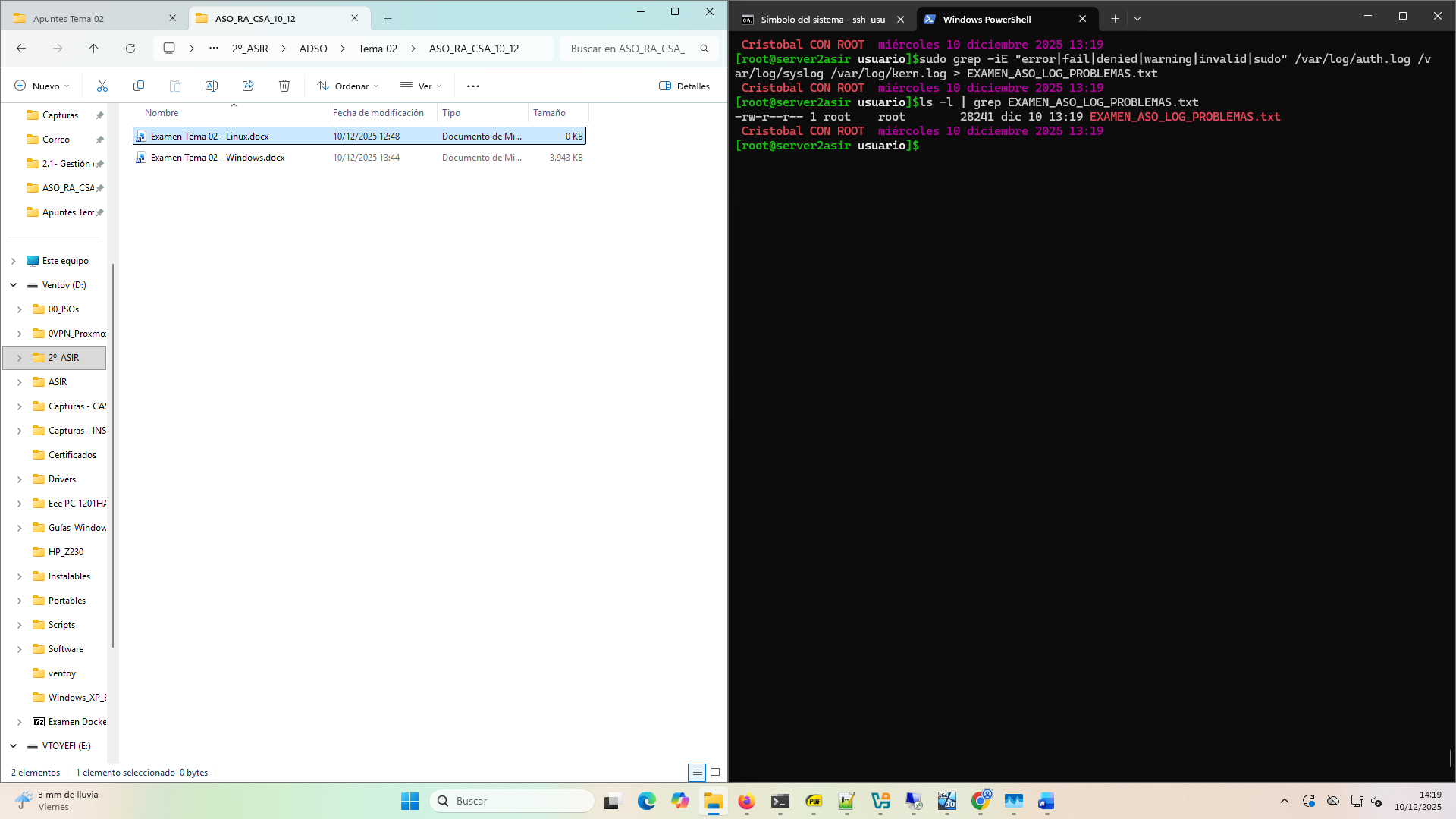
Podemos usar un comando que filtre todos estos archivos y busque palabras claves:

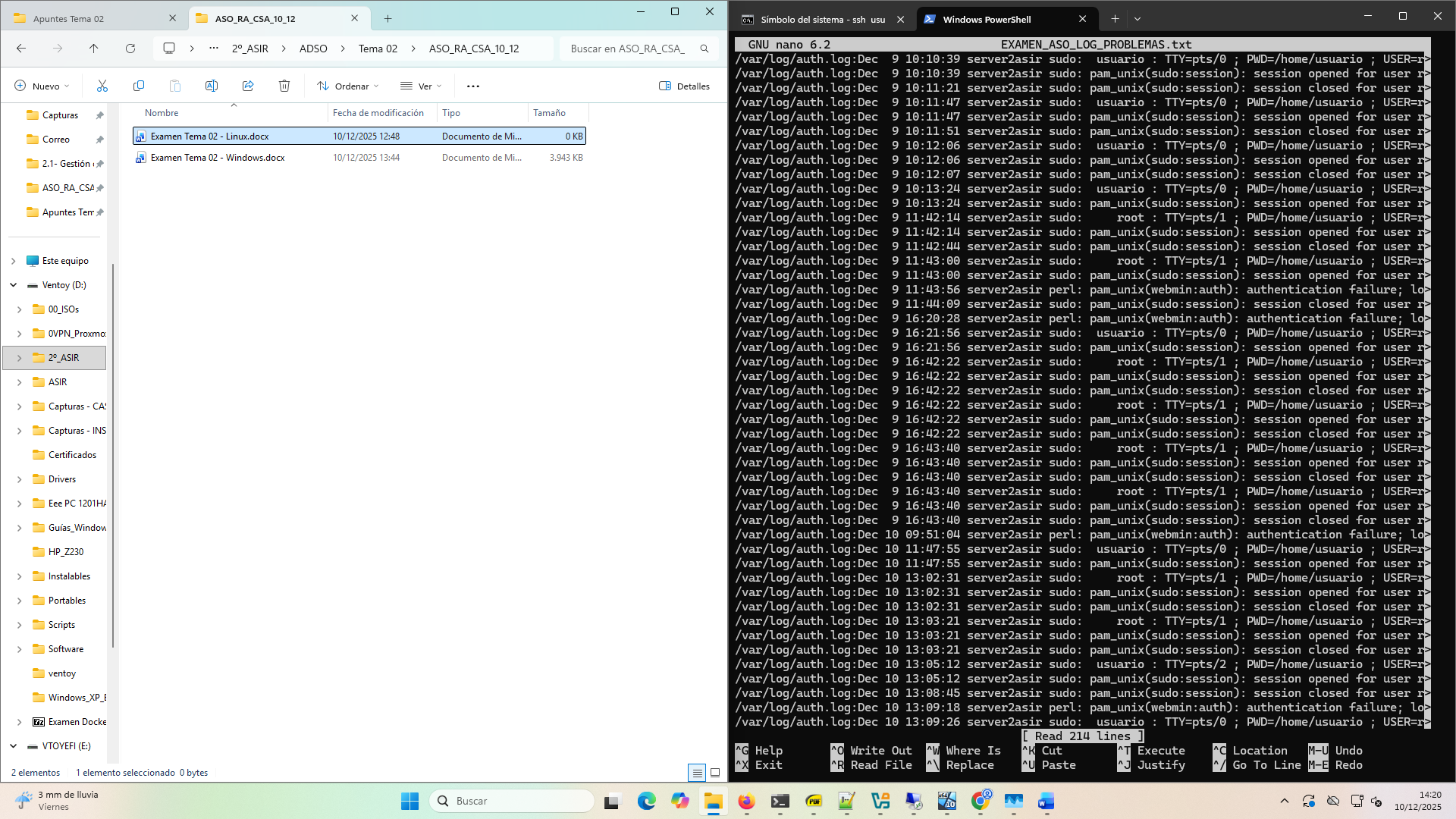
**sudo grep -iE "error|fail|denied|warning|invalid|sudo" /var/log/auth.log /var/log/syslog /var/log/kern.log**



Podemos hacer que el resultado se guarde en un archivo:

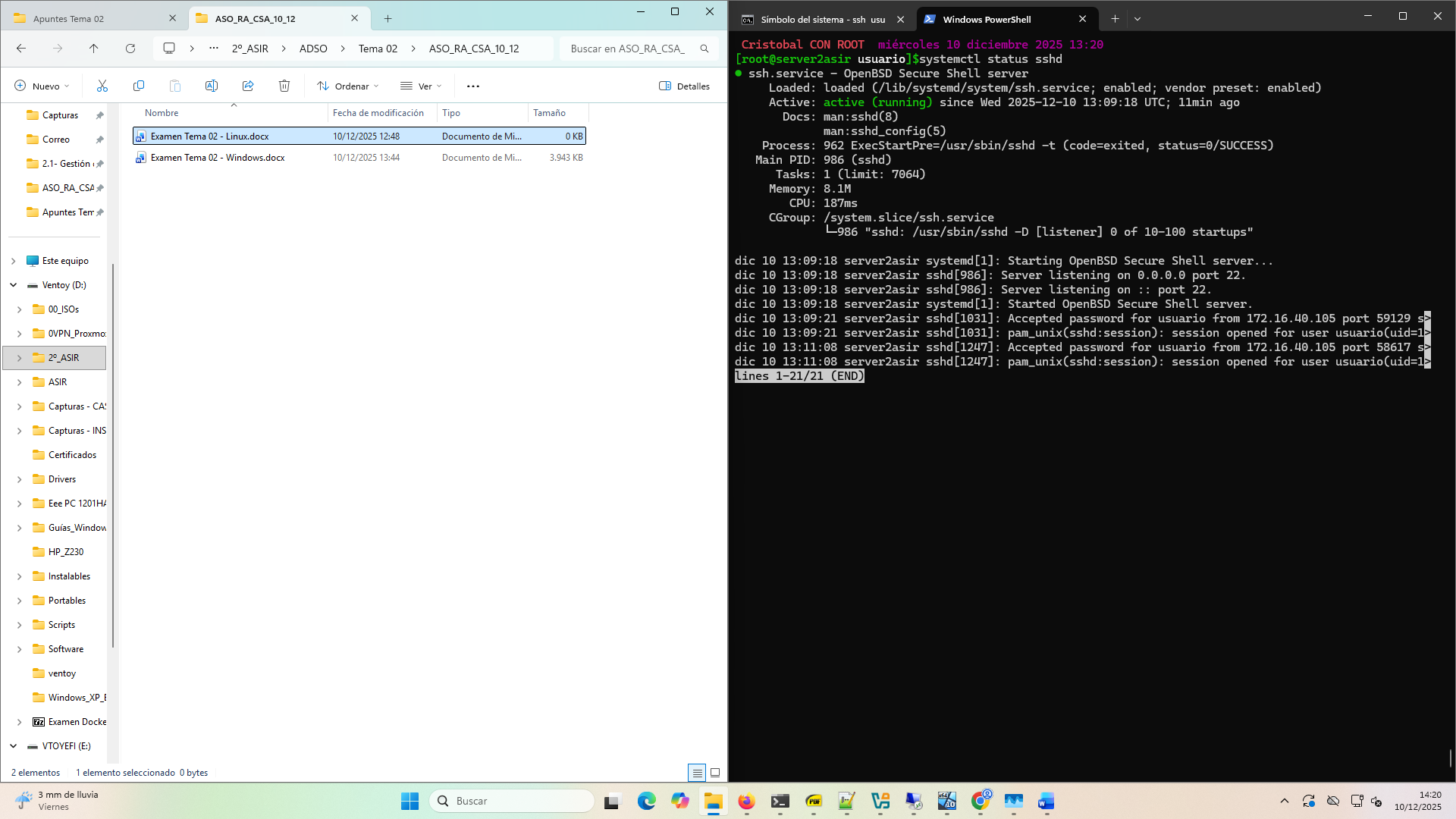
**sudo grep -iE "error|fail|denied|warning|invalid|sudo" /var/log/auth.log /var/log/syslog /var/log/kern.log > EXAMEN\_ASO\_LOG\_PROBLEMAS.txt**



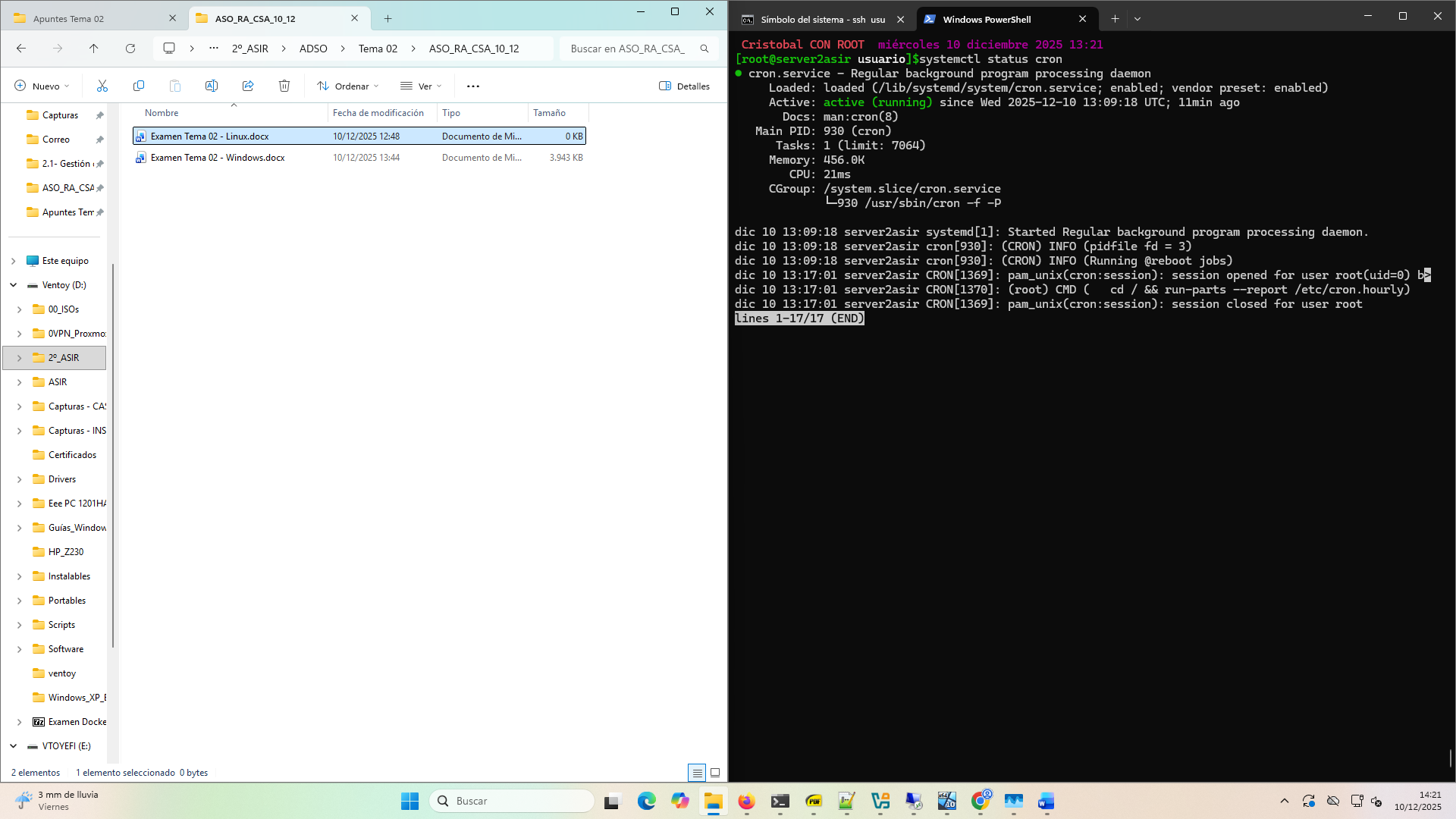


* Comprueba la integridad y el estado de dos servicios esenciales de tu elección). (si están activos o su estado actual)

SSH:

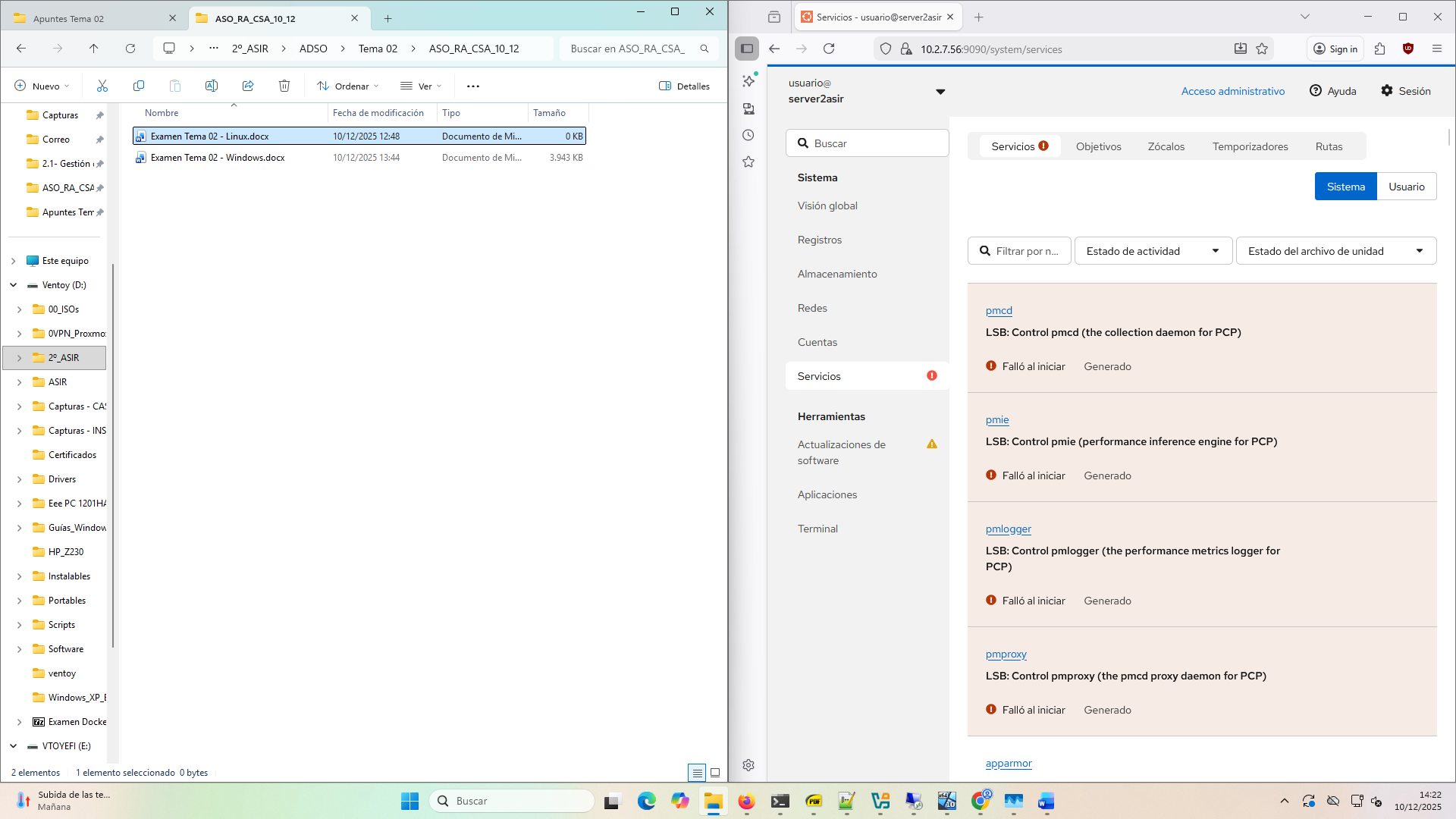


Cron



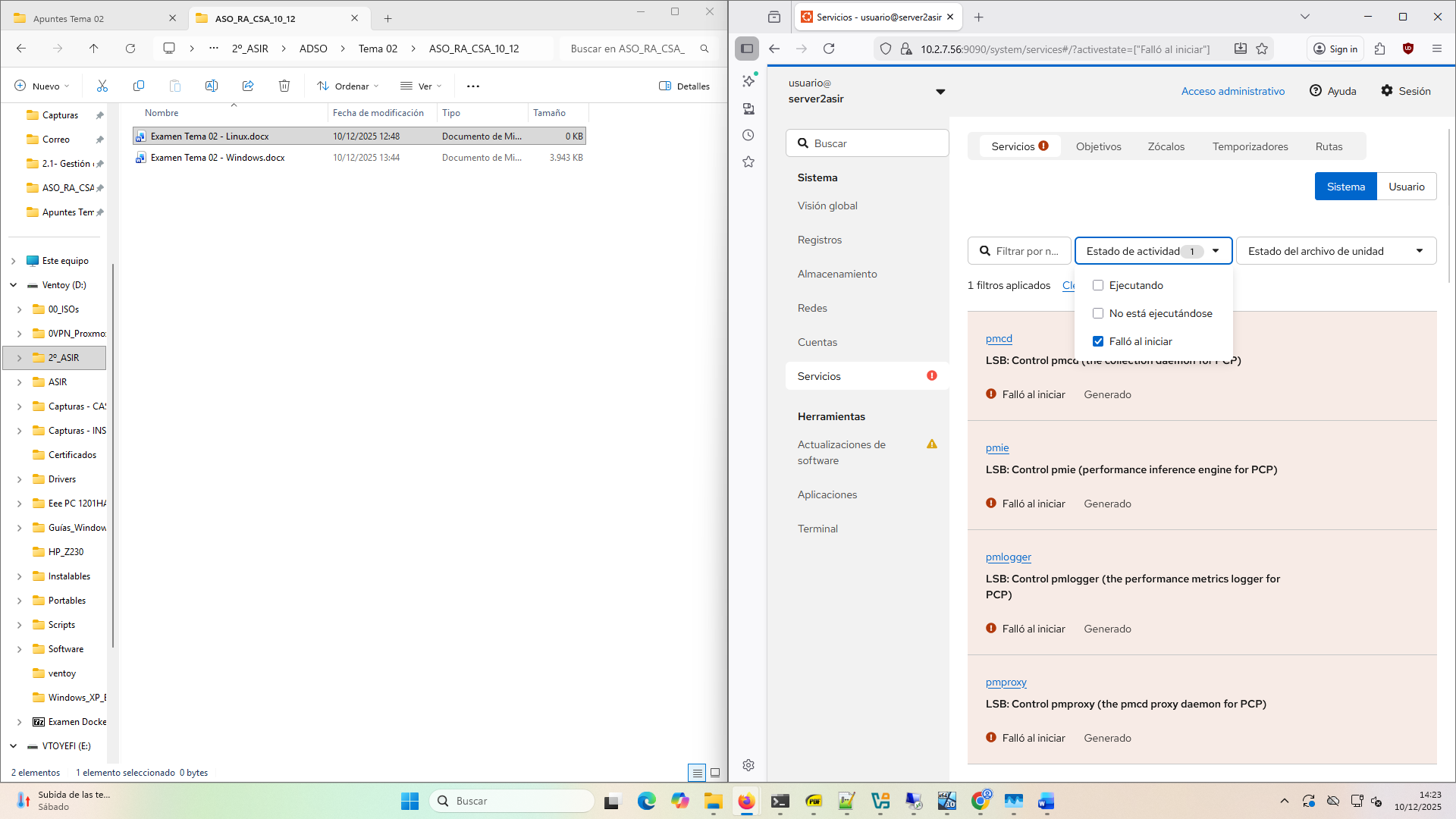
**Análisis con herramientas gráficas I (Cockpit)**

* Accede a Cockpit y entra en la sección de **Servicios**.

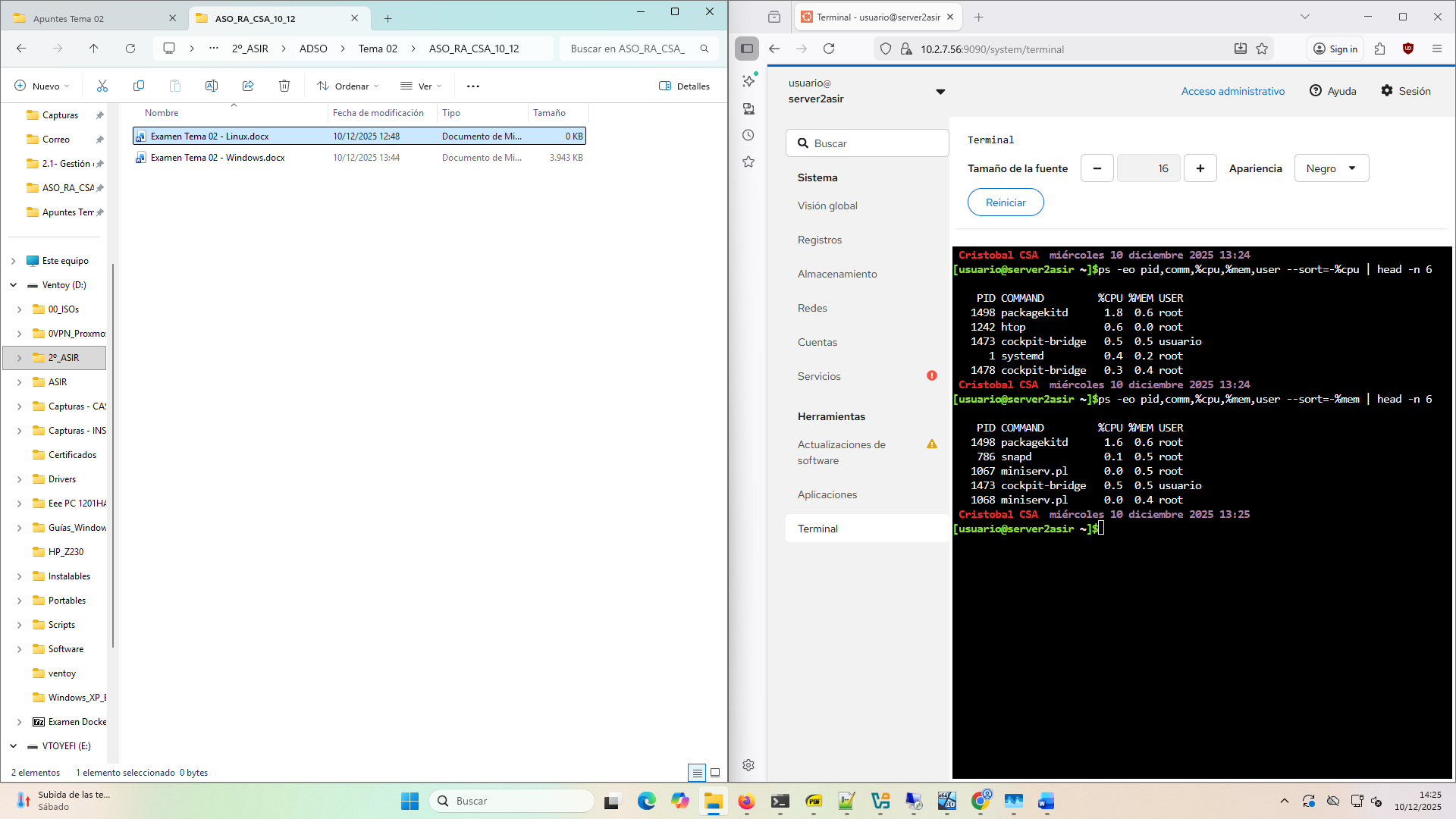


* Comprueba su estado actual y realiza una acción sobre él

Filtramos por aquellos procesos que fallaron al iniciarse.



* Desde la **Terminal** de Cockpit, obtén una lista de procesos ordenados por consumo de CPU o memoria.



Identifica un proceso que consideres sospechoso o llamativo.

Vamos a usar de ejemplo a: **1478 cockpit-bridge 0.3 0.4 root**

 Muestra en tu entrega:

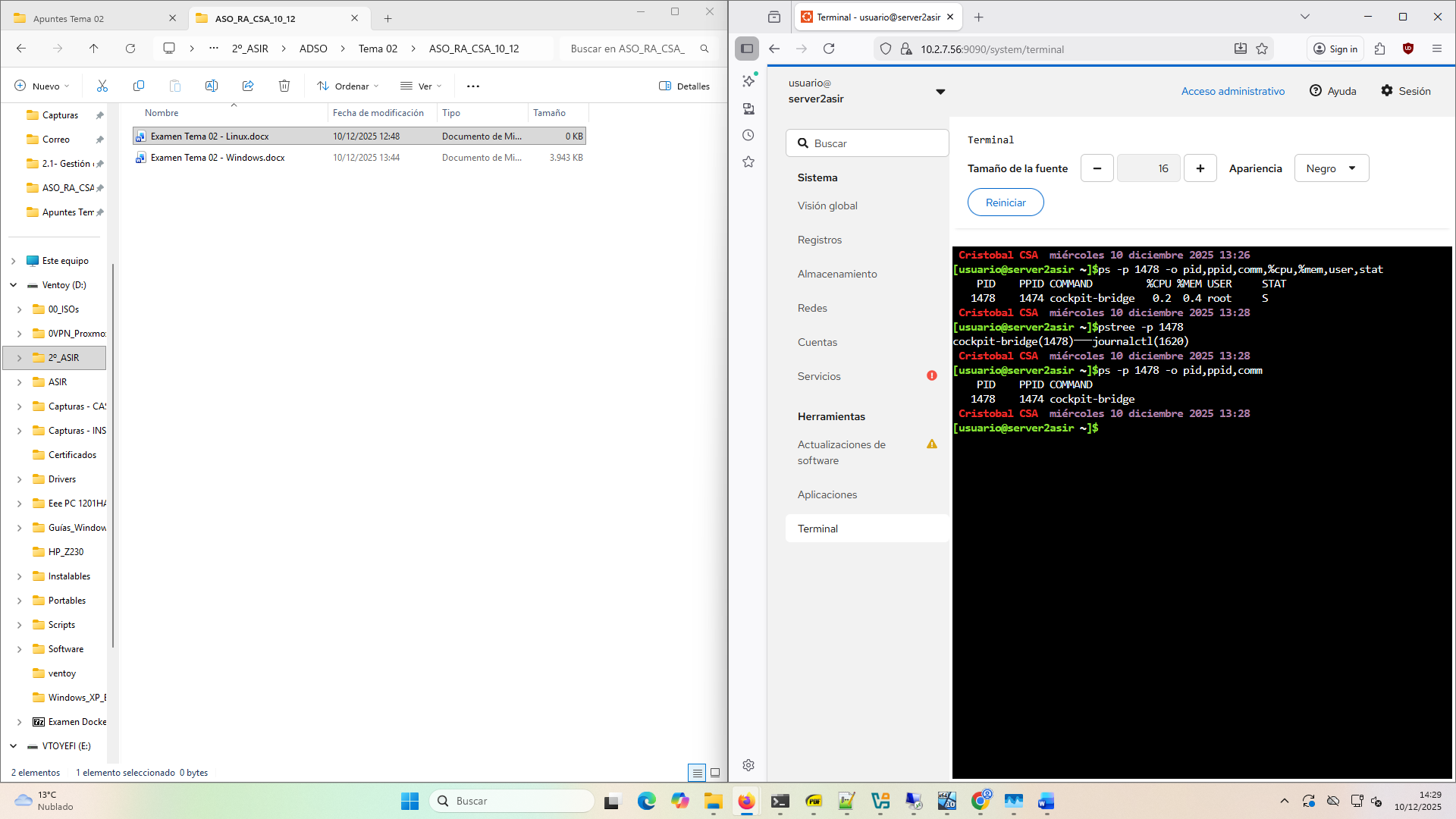
* El proceso resaltado
* Consumo
* Usuario
* Estado
* Jerarquía (PID y PPID)

Usamos los siguientes comandos:

**ps -p 1478 -o pid,ppid,comm,%cpu,%mem,user,stat**

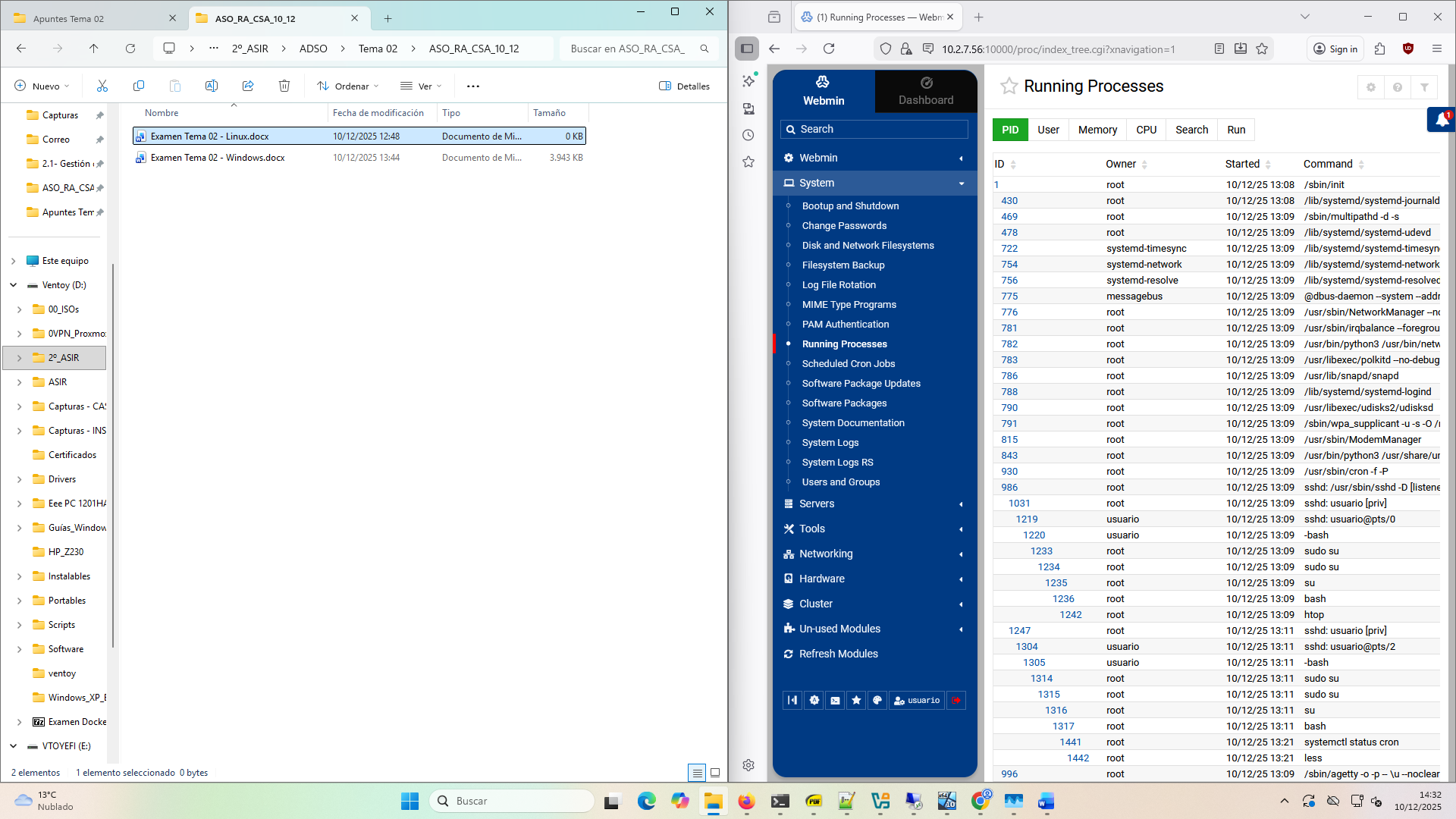
**pstree -p 1478**

**ps -p 1478 -o pid,ppid,comm**



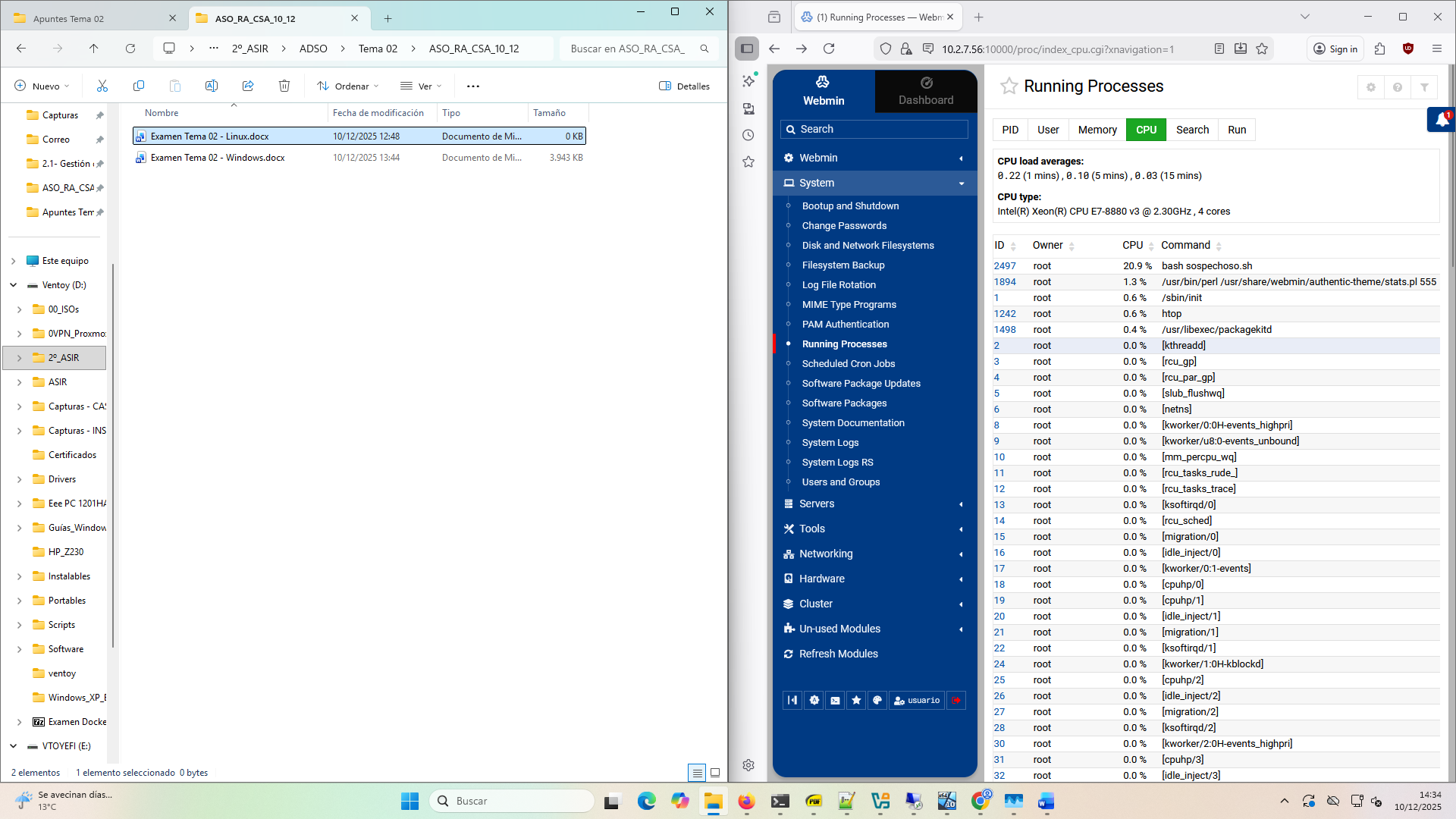
**Análisis con herramientas gráficas I (Webmin)**

* Accede al módulo *System → Running Processes*.

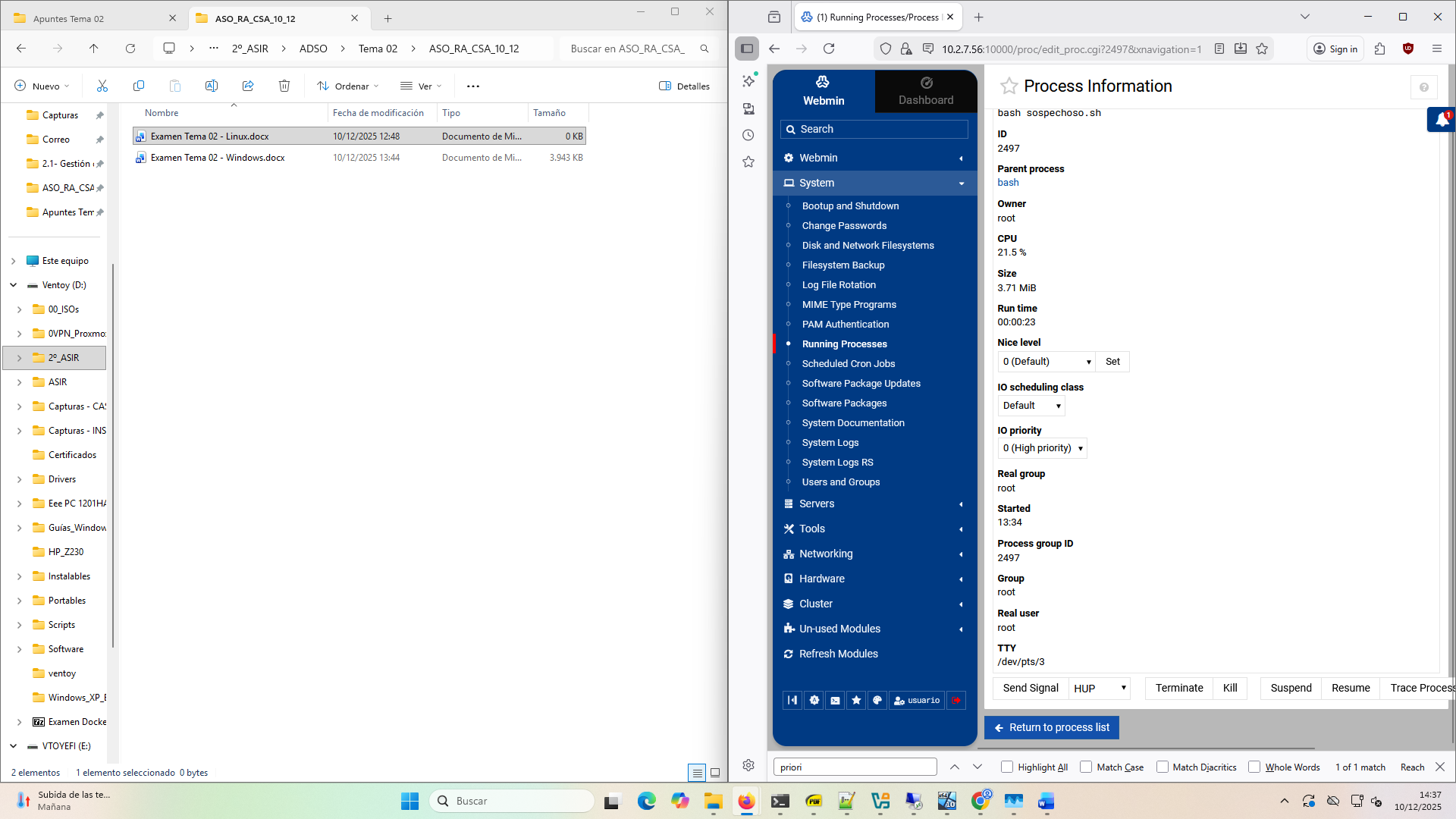


* Encuentra un proceso sospechoso (dime por qué sospechas) y muestra una captura donde se vea:

Hay proceso llamado “bash sospechoso.sh” que está consumiendo bastante CPU.



* + PID: 2497
  + Comando: bash sospechoso.sh
  + Prioridad o nice: su nivel de prioridad es “0”.



* + Si Webmin lo permite matar el proceso: Si. Si apretamos el botón “Kill” lo ha matado sin ningún problema. No era un proceso vital del sistema.

