

# Herramientas de Gestión de procesos

Administrador de tareas; Monitor del sistema; kill; ...

# Herramientas que vamos a revisar

Vamos a lanzar la ejecución de varias aplicaciones desde línea de comandos, o con nuestro proyecto CrearProcesos, para disponer de procesos en el sistema con los que poder probar las funcionalidades de las distintas herramientas de gestión de procesos.

Las herramientas que vamos a revisar son:

1. (Windows) Administrador de tareas.
2. (Windows) Comandos tasklist y taskkill.
3. (GNU/Linux) Monitor del sistema.
4. (GNU/Linux) Comandos ps y kill.
5. (Windows) SysInternals: Process Explorer y Process Monitor.

# Lanzar procesos

Antes de comenzar a ver las herramientas de gestión de procesos, recordemos cómo podemos lanzar la ejecución de procesos en el sistema:

1. Utilizando nuestro proyecto CrearProcesos.
2. Ejecutando los comandos que inician la ejecución de las aplicaciones en un intérprete de comandos. Para ello, lo que nunca falla, es siempre escribir la ruta completa de acceso al ejecutable de la aplicación.

Windows `C:\Users\usuario>"c:\Archivos de programa\OpenOffice.org 3\program\swriter"`

GNU/Linux `usuario@Pearl:~$ /usr/bin/oowriter`

`usuario@Pearl:~$ /usr/lib/openoffice/program/swriter`

3. Haciendo clic o doble clic sobre los iconos de acceso directo a las aplicaciones, creados en el menú Programas o Aplicaciones del Sistema, en el Inicio rápido, o en el Escritorio.



Captura de la barra de inicio de un Windows 7.



Captura del panel superior de una distribución GNU/Linux como es Ubuntu.

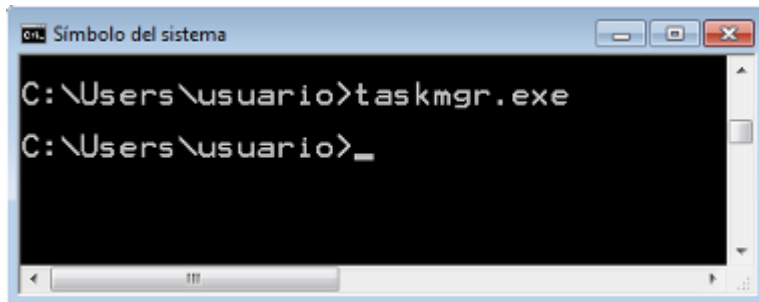
4. Haciendo doble clic sobre el icono de un archivo. El sistema automáticamente lanza por nosotros la aplicación asociada para abrir ese archivo.



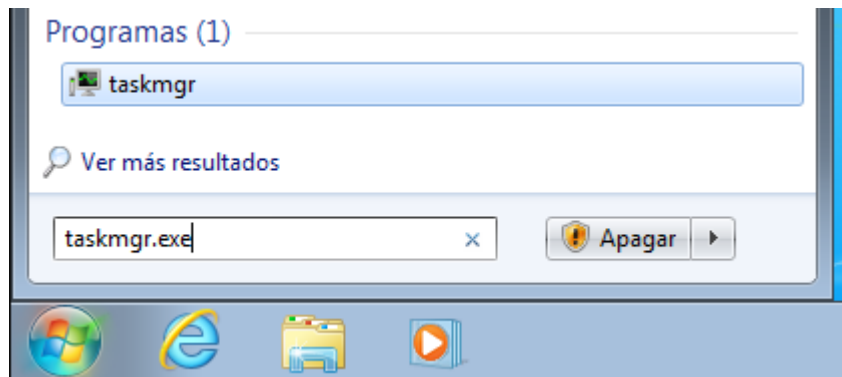
# Administrador de tareas

Respecto al Administrador de tareas de Windows, nos interesa conocer:

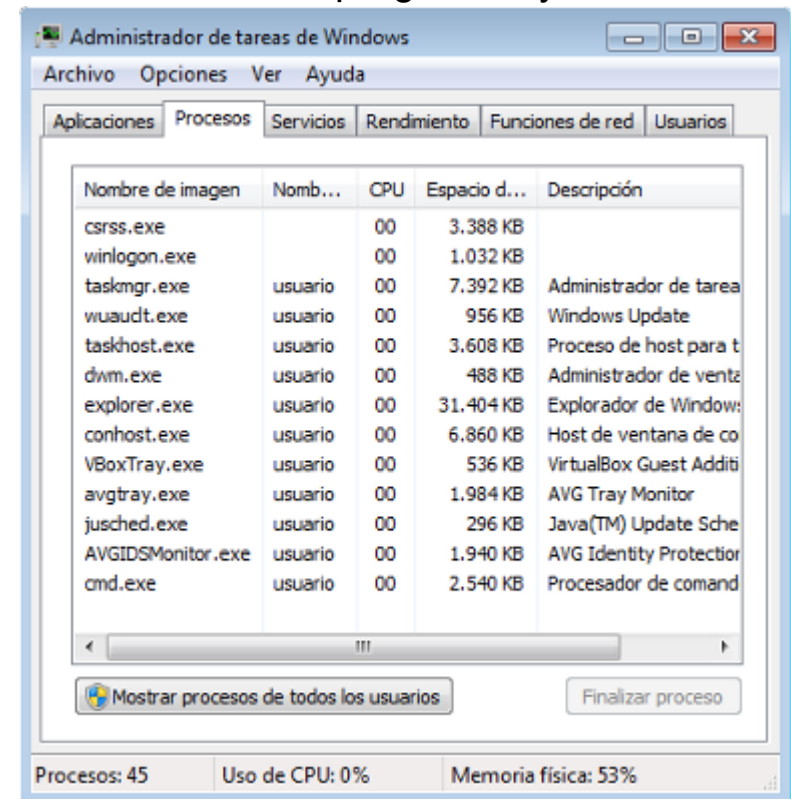
1. Para mostrar el Administrador de tareas, pulsaremos simultáneamente **Ctrl+Alt+Supr.** Windows nos mostrará un menú con las opciones: Bloquear equipo, ...; y, por último, **Iniciar el Administrador de tareas.**
2. También podemos Iniciar el Administrador de tareas haciendo clic con el **botón secundario del ratón** sobre la **barra de tareas** de Windows y seleccionando Iniciar el Administrador de tareas.
3. Otra forma de Iniciar el Administrador de tareas, es a través del **comando: taskmgr.exe.** Lo podemos introducir como un comando en el intérprete de comandos, o buscarlo entre los programas y archivos del sistema con ayuda del Menú Programas de Windows.



Lanzar taskmgr.exe desde el intérprete de comandos.



Lanzar taskmgr.exe desde el Menú Programas.



Aspecto usual del Administrador de tareas.  
Pestaña Procesos.

# Administrador de tareas

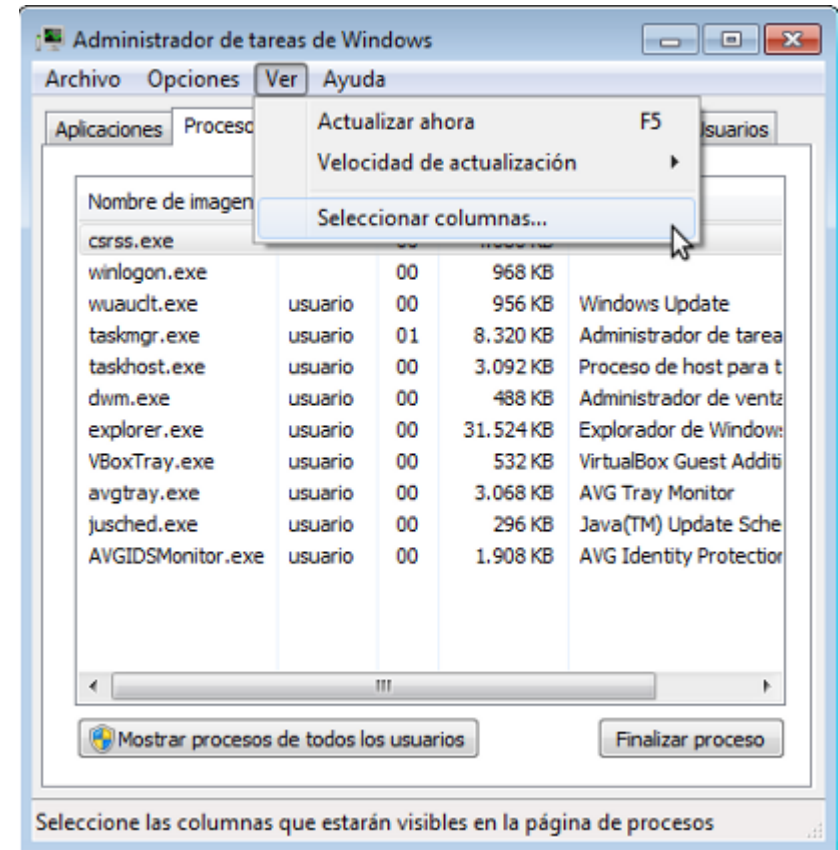
Nos interesa conocer (continuación):

3. Haz clic en las distintas pestañas del Administrador de tareas:

- Aplicaciones. Mostrará un listado de las aplicaciones lanzadas en el sistema.
- Procesos. Mostrará un listado de los procesos del sistema.
- Servicios. Mostrará un listado de los servicios del sistema y el estado en el que se encuentran.
- Redimiento. Muestra información y estadísticas sobre el uso de la CPU y la memoria del sistema.
- Funciones de red. Uso global que está haciendo el sistema de las conexiones de red.
- Usuarios. Muestra un listado de los usuarios que hayan iniciado sesión en el sistema.

Nosotros nos centraremos aquí en la pestaña **Procesos**.

4. Podemos mostrar más información en la pestaña **Procesos**, con la opción del menú **Ver** → **Seleccionar columnas...**



Administrador de tareas. Pestaña Procesos.  
Menú Ver → Seleccionar columnas...

# Administrador de tareas

Nos interesa conocer (continuación):

5. Dentro de las columnas que nos pueden resultar interesantes, resaltamos:

- **Identificador de proceso (PID)**. Número identificador del proceso.
- **Línea de comandos**. Comando que lanza el ejecutable en un terminal.
- **Subprocesos**. Número de subprocesos (hilos) que tiene activos cada proceso.

6. Podemos **mostrar los procesos de todos los usuarios**. Por defecto muestra sólo los procesos lanzados por nuestro usuario. Así veremos, realmente todos los procesos activos en el sistema.

7. Haciendo clic con el botón secundario del ratón sobre la entrada de cualquier proceso, podremos:

- **Abrir la ubicación del archivo**. Abrirá el explorador de Windows la carpeta que contiene el ejecutable.
- **Terminar el proceso**. Finaliza la ejecución del proceso. (También tenemos el botón "Terminar proceso").
- **Finalizar el árbol de procesos**. Finaliza el proceso y todos los procesos que hay sido creados por él.

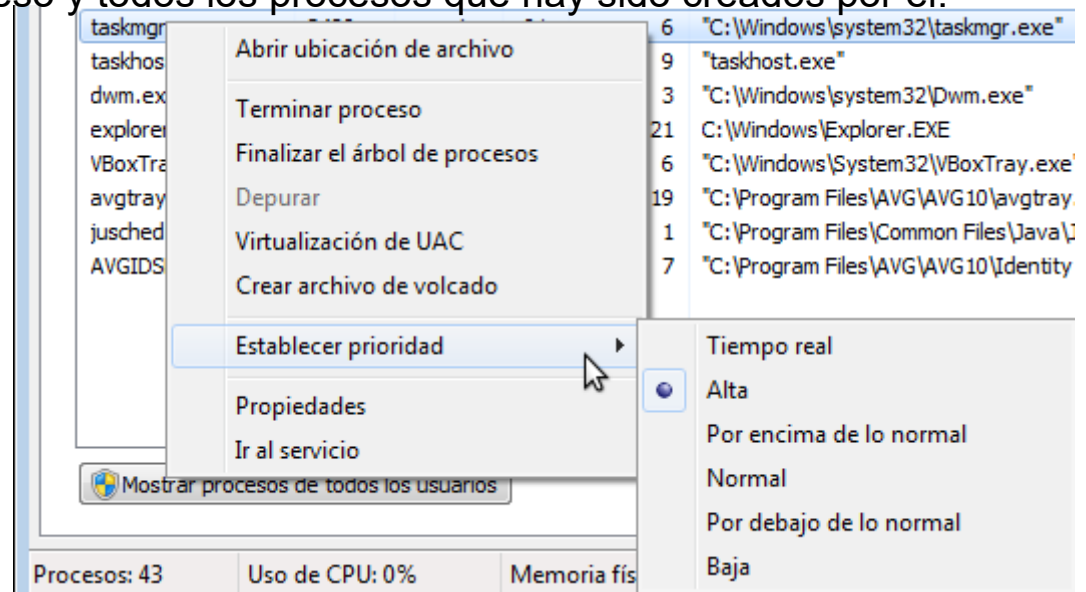
• **Virtualización de UAC**. La virtualización de UAC redirige los errores de escritura de archivos y registro a ubicaciones definidas por cada usuario.

• **Crear un archivo de volcado**. Crea un archivo con el contenido actual del bloque de memoria del proceso.

• **Establecer prioridad**. Muestra y modifica la prioridad de ejecución del proceso.

• **Propiedades**. Abre un cuadro de diálogo con información más detallada sobre las propiedades del proceso.

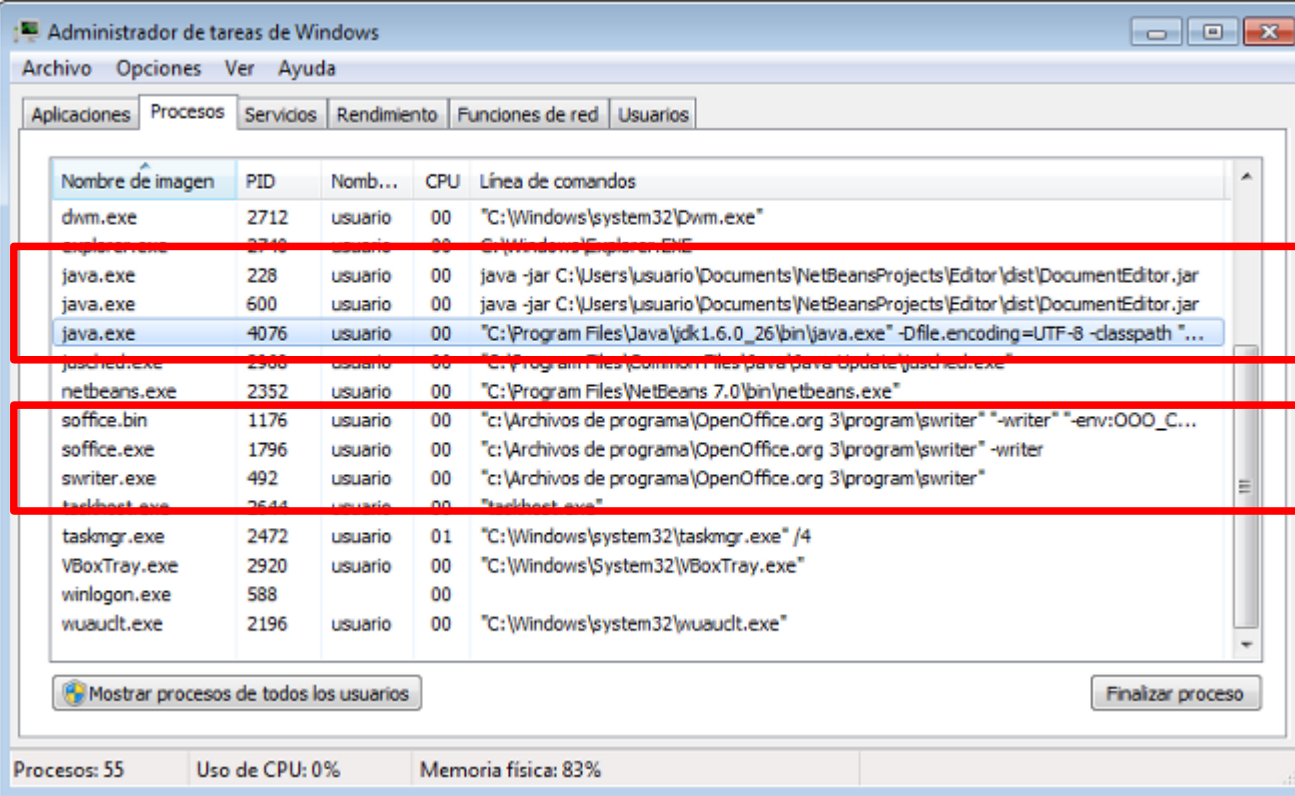
• **Ir al servicio**. Nos lleva a la pestaña Servicios del Administrador de tareas.



Administrador de tareas. Pestaña Procesos.  
Opciones de un proceso.

# Administrador de tareas

Por ejemplo, lancemos varias aplicaciones y procesos: **Writer** de OpenOffice y nuestro **CrearProcesos** con 2 de sus **Editores**. Los procesos que podremos ver en el Administrador de tareas son:



The screenshot shows the Windows Task Manager window with the 'Procesos' tab selected. The table below represents the data visible in the process list:

| Nombre de imagen | PID  | Nomb... | CPU | Línea de comandos   |
|------------------|------|---------|-----|---|
| dwm.exe          | 2712 | usuario | 00  | "C:\Windows\system32\Dwm.exe"   |
| explorer.exe     | 2740 | usuario | 00  | "C:\Windows\Explorer.exe"   |
| java.exe         | 228  | usuario | 00  | java -jar C:\Users\usuario\Documents\NetBeansProjects\Editor\dist\DocumentEditor.jar    |
| java.exe         | 600  | usuario | 00  | java -jar C:\Users\usuario\Documents\NetBeansProjects\Editor\dist\DocumentEditor.jar    |
| java.exe         | 4076 | usuario | 00  | "C:\Program Files\Java\jdk1.6.0_26\bin\java.exe" -Dfile.encoding=UTF-8 -classpath "..." |
| java.exe         | 2308 | usuario | 00  | "C:\Program Files\Java\jdk1.6.0_26\bin\java.exe" -Dfile.encoding=UTF-8 -classpath "..." |
| netbeans.exe     | 2352 | usuario | 00  | "C:\Program Files\NetBeans 7.0\bin\netbeans.exe"  |
| soffice.bin      | 1176 | usuario | 00  | "C:\Archivos de programa\OpenOffice.org 3\program\swriter" *-writer" *-env:OOO_C...     |
| soffice.exe      | 1796 | usuario | 00  | "C:\Archivos de programa\OpenOffice.org 3\program\swriter" *-writer" *-env:OOO_C...     |
| swriter.exe      | 492  | usuario | 00  | "C:\Archivos de programa\OpenOffice.org 3\program\swriter" *-writer" *-env:OOO_C...     |
| taskhost.exe     | 2544 | usuario | 00  | "taskhost.exe"  |
| taskmgr.exe      | 2472 | usuario | 01  | "C:\Windows\system32\taskmgr.exe" /4  |
| VBoxTray.exe     | 2920 | usuario | 00  | "C:\Windows\System32\VBoxTray.exe"  |
| winlogon.exe     | 588  | usuario | 00  | "C:\Windows\system32\winlogon.exe"  |
| wuauclt.exe      | 2196 | usuario | 00  | "C:\Windows\system32\wuauclt.exe"   |

En **1**, podemos ver, tres instancias de la máquina virtual java, una de ellas es la de nuestro proceso CrearProcesos y las otras dos, como se puede apreciar, son los procesos Editor que hemos creado.

En **2**, podemos ver que aunque sólo hayamos lanzado una sola instancia de Writer, automáticamente se han lanzado varias instancias de soffice.

Vemos que por cada aplicación java que lancemos se crea una instancia de la máquina virtual.

# Comandos tasklist y taskkill

1. Comenzamos ejecutando el comando tasklist, este es el resultado:

```
C:\Users\usuario>tasklist

Nombre de imagen                PID  Nombre de sesión  Núm. de ses
=====
System Idle Process             0    Services          0
System                          4    Services          0
smss.exe                        280  Services          0
avgchsvx.exe                    332  Services          0
csrss.exe                       500  Services          0
wininit.exe                     536  Services          0
wuauclt.exe                     2196 Console          1
avgrsx.exe                      1152 Services          0
avgcsrvc.exe                   3224 Services          0
taskmgr.exe                     2472 Console          1
cmd.exe                         2340 Console          1
conhost.exe                     1168 Console          1
netbeans.exe                    2352 Console          1
→ java.exe                      4076 Console          1
→ conhost.exe                   2296 Console          1
→ java.exe                       228 Console          1
→ conhost.exe                   2064 Console          1
→ java.exe                       600 Console          1
→ conhost.exe                   2524 Console          1
swriter.exe                     492  Console          1
soffice.exe                     1796 Console          1
soffice.bin                     1176 Console          1
tasklist.exe                    528  Console          1
WmiPrvSE.exe                   3728 Services          0
```

2. Del listado anterior obtenemos el PID para terminar, uno de los procesos, por ejemplo, uno de los procesos java.

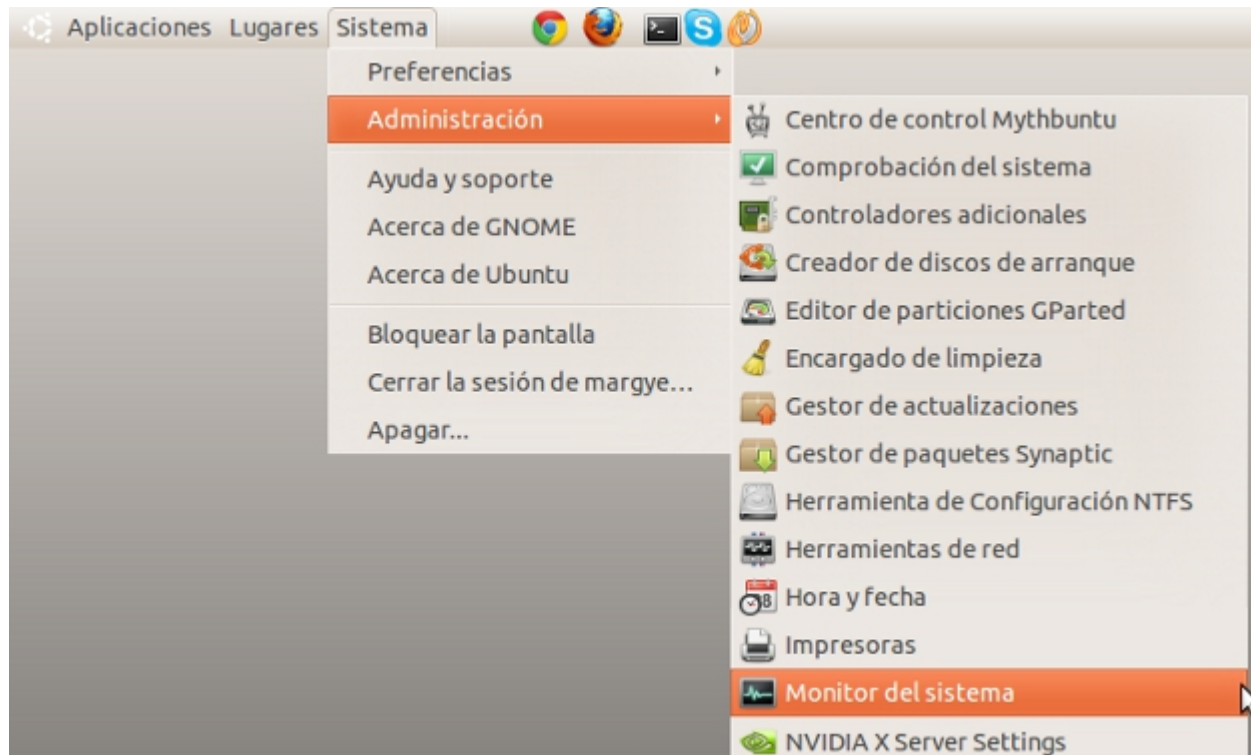
```
C:\Users\usuario>taskkill /PID 228
Correcto: se envió la señal de término al proceso con PID 228.
```



# Monitor del sistema

Respecto al Monitor del sistema de GNU/Linux, nos interesa conocer:

1. Para mostrar el Monitor del sistema, accederemos al menú **Sistema** → **Administración** → **Monitor del sistema**.



Lanzar el Monitor del sistema desde Sistema → Administración.

2. También podemos Iniciar el Monitor del sistema a través del **comando: gnome-system-monitor**. Lo podemos ejecutar como un comando en el intérprete de comandos.

```
usuario@Pearl:~$ gnome-system-monitor
```

Lanzar el Monitor del sistema desde el terminal.

# Monitor del sistema

Información que nos muestra:

3. Sus pestañas nos informan de:

- **Sistema.** Información sobre la versión del sistema operativo que se está ejecutando, el tamaño de la memoria principal del sistema e información sobre el microprocesador.
- **Procesos.** Listado de los procesos del sistema.
- **Recursos.** Muestra información y estadísticas sobre el uso de la CPU, la memoria del sistema y el uso de la interfaz de red.
- **Sistemas de archivos.** Para cada una de los sistemas de ficheros (unidades o particiones de disco duro) existente, su porcentaje de espacio utilizado y libre.

Monitor del sistema

Monitor Editar Ver Ayuda

Sistema Procesos Recursos Sistemas de archivos

Carga media para los últimos 1, 5 y 15 minutos: 0,01, 0,08, 0,21

| Nombre del proceso   | Estado       | % CPU | Prioridad | ID   | Memoria   | Canal en espera       |
|----------------------|--------------|-------|-----------|------|-----------|-----------------------|
| gnome-system-monitor | Ejecutándose | 10    | 0         | 6373 | 7,1 MiB   | 0                     |
| shutter              | Durmiendo    | 0     | 0         | 3423 | 77,1 MiB  | poll_schedule_timeout |
| Xorg                 | Durmiendo    | 0     | 0         | 1851 | 41,7 MiB  | poll_schedule_timeout |
| soffice.bin          | Durmiendo    | 0     | 0         | 2289 | 161,2 MiB | poll_schedule_timeout |
| gnome-panel          | Durmiendo    | 0     | 0         | 2136 | 12,6 MiB  | poll_schedule_timeout |
| nautilus             | Durmiendo    | 0     | 0         | 2135 | 24,4 MiB  | poll_schedule_timeout |
| bash                 | Durmiendo    | 0     | 0         | 6274 | 2,9 MiB   | n_tty_read            |
| kondemand/0          | Durmiendo    | 0     | 0         | 63   | N/D       | worker_thread         |
| kondemand/1          | Durmiendo    | 0     | 0         | 64   | N/D       | worker_thread         |
| gnome-pty-helper     | Durmiendo    | 0     | 0         | 6273 | 792,0 KiB | unix_stream_data_wait |

Finalizar proceso

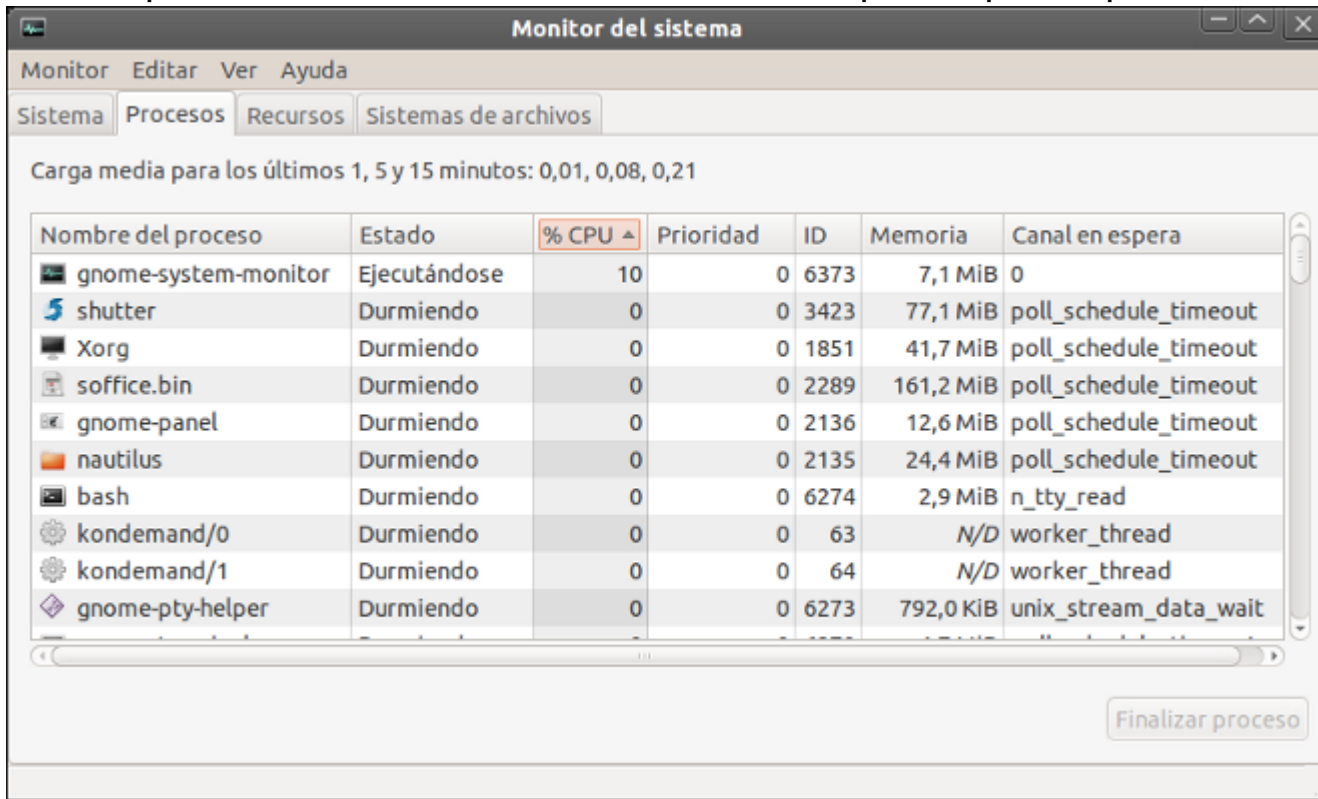
Información por defecto en el Monitor del Sistema. Pestaña Procesos.

# Monitor del sistema

Nos centramos en la pestaña Procesos, información que nos aporta:

4. Las columnas de la Pestaña Procesos:

- **Nombre del proceso.** Nombre del ejecutable del proceso.
- **Estado.** Estado en el que se encuentra el proceso, por ejemplo: ejecutándose, durmiendo, parado, ...
- **%CPU.** Porcentaje de uso de la CPU.
- **Prioridad.** Prioridad de ejecución del proceso.
- **ID.** PID del proceso.
- **Memoria.** Tamaño ocupado en memoria por el proceso.
- **Canal en espera.** Nos permite hacernos una idea de el motivo por el que un proceso está durmiendo.



| Nombre del proceso   | Estado       | % CPU | Prioridad | ID   | Memoria   | Canal en espera       |
|----------------------|--------------|-------|-----------|------|-----------|-----------------------|
| gnome-system-monitor | Ejecutándose | 10    | 0         | 6373 | 7,1 MiB   | 0                     |
| shutter              | Durmiendo    | 0     | 0         | 3423 | 77,1 MiB  | poll_schedule_timeout |
| Xorg                 | Durmiendo    | 0     | 0         | 1851 | 41,7 MiB  | poll_schedule_timeout |
| soffice.bin          | Durmiendo    | 0     | 0         | 2289 | 161,2 MiB | poll_schedule_timeout |
| gnome-panel          | Durmiendo    | 0     | 0         | 2136 | 12,6 MiB  | poll_schedule_timeout |
| nautilus             | Durmiendo    | 0     | 0         | 2135 | 24,4 MiB  | poll_schedule_timeout |
| bash                 | Durmiendo    | 0     | 0         | 6274 | 2,9 MiB   | n_tty_read            |
| kondemand/0          | Durmiendo    | 0     | 0         | 63   | N/D       | worker_thread         |
| kondemand/1          | Durmiendo    | 0     | 0         | 64   | N/D       | worker_thread         |
| gnome-pty-helper     | Durmiendo    | 0     | 0         | 6273 | 792,0 KIB | unix_stream_data_wait |

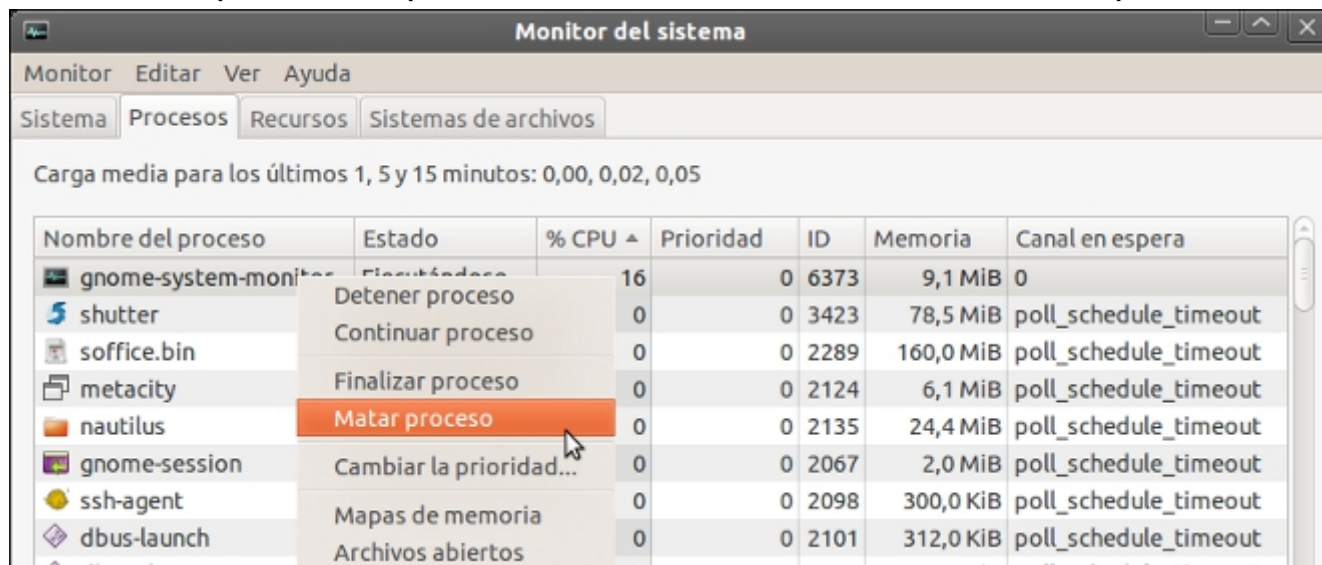
Información por defecto en el Monitor del Sistema. Pestaña Procesos.

# Monitor del sistema

Opciones del menú contextual para cada proceso de la lista.

5. Las opciones disponibles para cada proceso:

- **Detener y continuar proceso.** Para y reanuda la ejecución de un proceso. Un proceso parado, no competirá por el uso de la CPU.
- **Finalizar proceso.** Envía una señal de cierre al proceso. Esta señal, permite que el proceso se finalice correctamente. Es como si seleccionamos la opción Salir de esa aplicación. El proceso, puede ignorar esta señal.
- **Matar proceso.** Envía una señal que fuerza la finalización del proceso. No podrá ignorarla y puede provocar el cierre incorrecto de la aplicación, y por lo tanto pérdida de datos.
- **Cambiar la prioridad.** Nos mostrará una ventana desde la que podremos cambiar la prioridad del proceso. Para asignar prioridades de ejecución negativas (las más altas) necesitaremos poseer privilegios de administrador del sistema.
- **Mapas de memoria.** Muestra para un determinado proceso un volcado de su mapa de memoria en uso.
- **Archivos abiertos.** Muestra para cada proceso la información de los archivos que tiene abiertos.



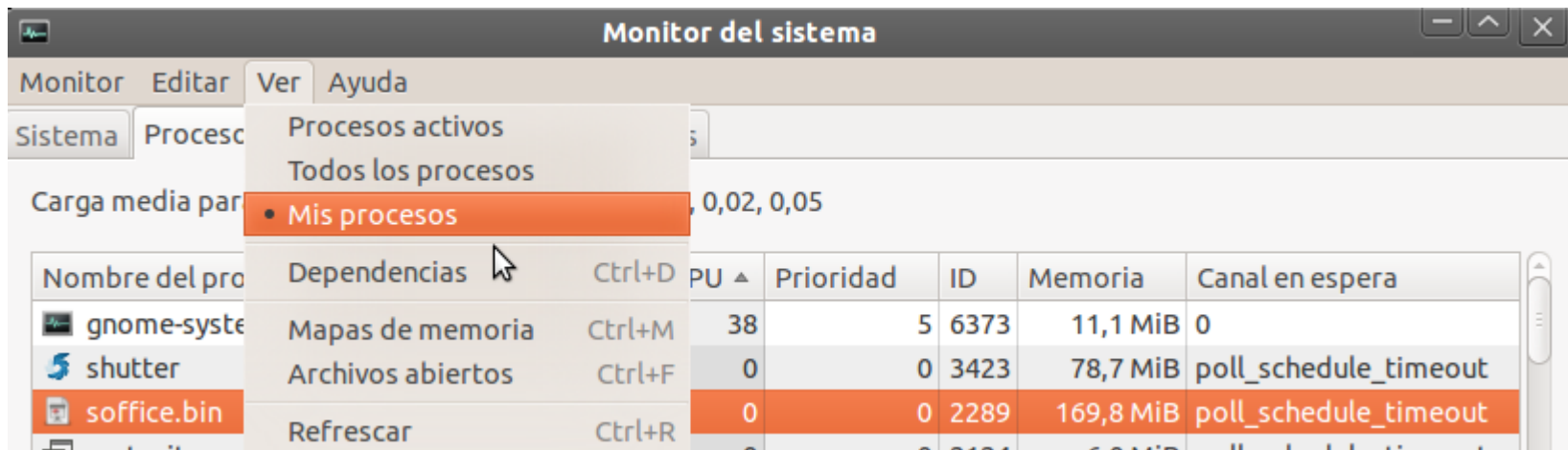
Opciones del menú contextual para cada proceso de la lista.

# Monitor del sistema

## Opciones del menú Ver.

6. Las opciones disponibles son:

- Ver → **Procesos activos**. En el listado de procesos se mostrarán sólo los procesos en ejecución.
- Ver → **Todos los procesos**. Muestra todos los procesos del sistema, también los de otros usuarios.
- Ver → **Mis procesos**. Muestra sólo los procesos del usuario actual.
- Ver → **Dependencias**. Mostrará de forma anidada, las dependencias entre unos procesos y otros. Quien ha creado a quién.
- Ver → **Mapas de memoria**. Muestra para un determinado proceso un volcado de su mapa de memoria en uso. El proceso deberá estar seleccionado en la lista.
- Ver → **Archivos abiertos**. Muestra para cada proceso la información de los archivos que tiene abiertos. El proceso deberá estar seleccionado en la lista.
- Ver → **Refrescar**. Fuerza la actualización de la información del Monitor del sistema.



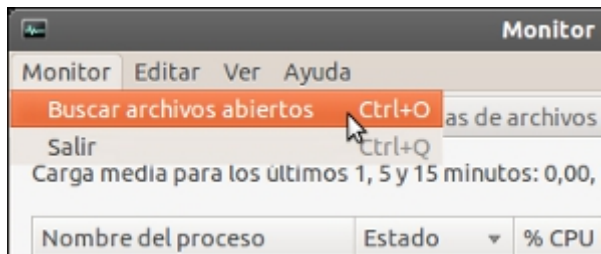
Opciones del menú Ver del Monitor del sistema.



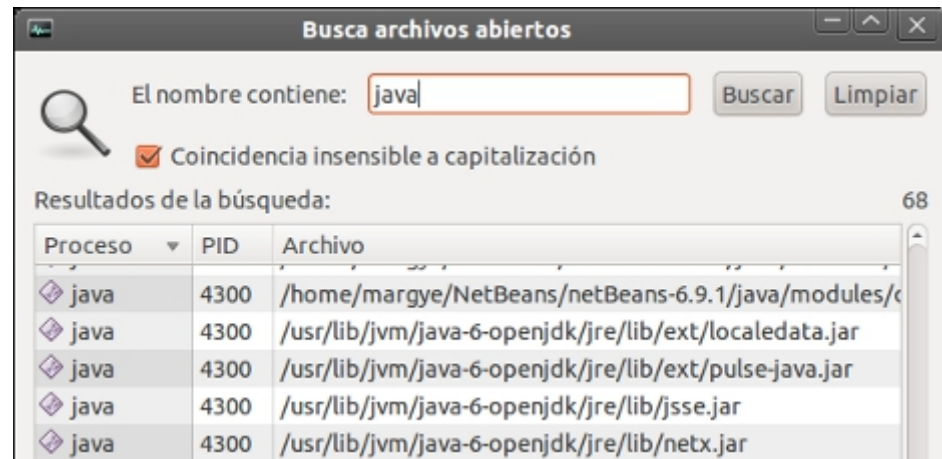
# Monitor del sistema

Buscar procesos que tienen archivos abiertos con la opción “**Buscar archivos abiertos**” del **menú Monitor**.

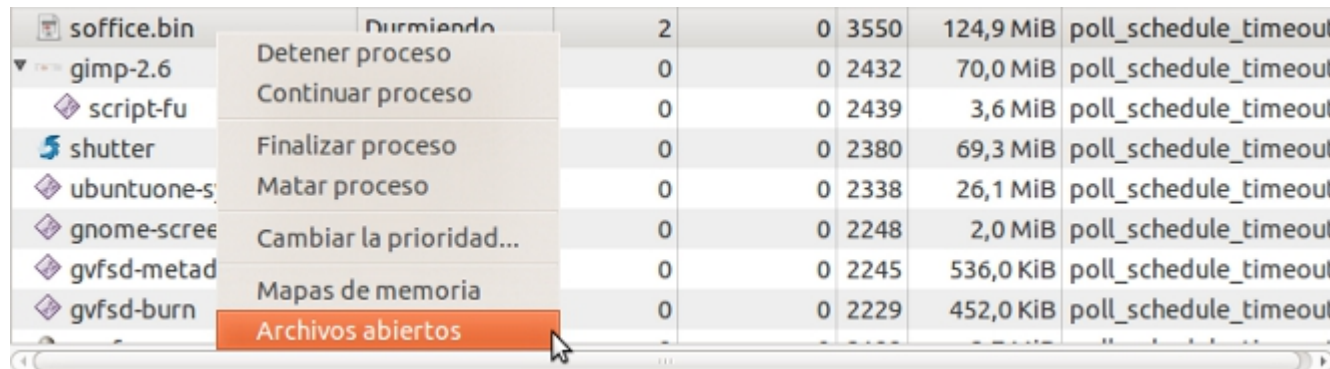
7. Una funcionalidad que nos resultará muy útil es conocer todos los procesos que están accediendo a un cierto archivo. En menú Monitor → Buscar archivos abiertos, podremos obtener esa información. Aquí podemos ver un ejemplo en el que hemos buscado todos los archivos que contengan en su nombre o ruta el texto “java”, teniendo NetBeans iniciado en el sistema.



Menú Monitor → Buscar archivos abiertos.



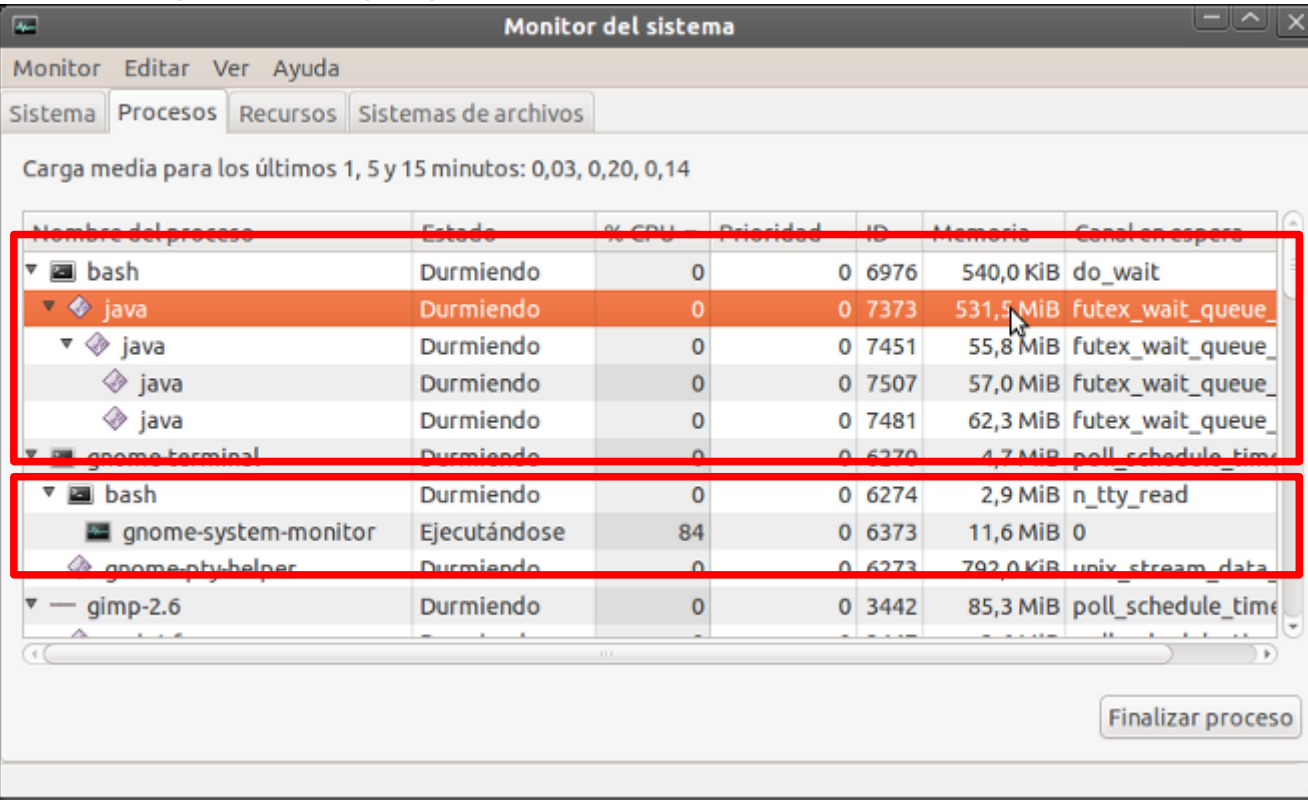
Otra forma de saber los archivos abiertos por un proceso, es la opción del **menú contextual** de un proceso → **Archivos abiertos**. Que nos muestra la información de los archivos y recursos que tenga abiertos ese proceso.



Opciones del menú contextual para cada proceso de la lista.

# Monitor del sistema

Por ejemplo, lancemos varias aplicaciones y procesos: **Writer** de OpenOffice y nuestro **CrearProcesos** con 2 de sus **Editores**. Los procesos que podremos ver en el Monitor del sistema son:



Monitor del sistema

Monitor Editar Ver Ayuda

Sistema Procesos Recursos Sistemas de archivos

Carga media para los últimos 1, 5 y 15 minutos: 0,03, 0,20, 0,14

| Nombre del proceso   | Estado       | % CPU | Prioridad | ID   | Memoria   | Const en espera    |
|----------------------|--------------|-------|-----------|------|-----------|--------------------|
| bash                 | Durmiendo    | 0     | 0         | 6976 | 540,0 KiB | do_wait            |
| java                 | Durmiendo    | 0     | 0         | 7373 | 531,5 MiB | futex_wait_queue   |
| java                 | Durmiendo    | 0     | 0         | 7451 | 55,8 MiB  | futex_wait_queue   |
| java                 | Durmiendo    | 0     | 0         | 7507 | 57,0 MiB  | futex_wait_queue   |
| java                 | Durmiendo    | 0     | 0         | 7481 | 62,3 MiB  | futex_wait_queue   |
| gnome-terminal       | Durmiendo    | 0     | 0         | 6370 | 4,7 MiB   | poll_schedule_tim  |
| bash                 | Durmiendo    | 0     | 0         | 6274 | 2,9 MiB   | n_tty_read         |
| gnome-system-monitor | Ejecutándose | 84    | 0         | 6373 | 11,6 MiB  | 0                  |
| gnome-phy-helper     | Durmiendo    | 0     | 0         | 6273 | 792,0 KiB | unix_stream_data   |
| gimp-2.6             | Durmiendo    | 0     | 0         | 3442 | 85,3 MiB  | poll_schedule_time |

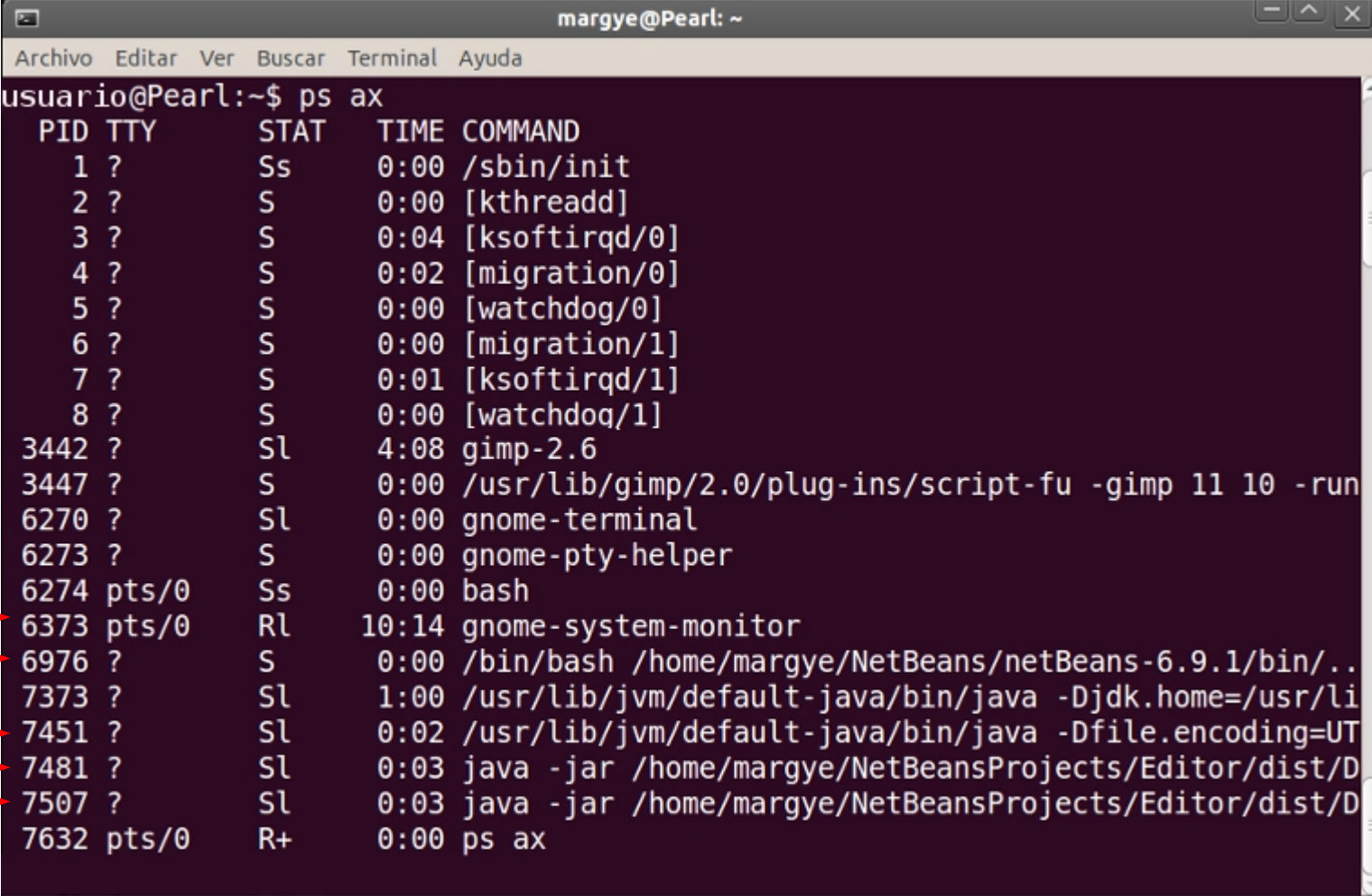
Finalizar proceso

En **1**, podemos ver, cuatro instancias de la máquina virtual java, una de ellas es la que ha lanzado el IDE NetBeans, que ha creado nuestro proceso CrearProcesos y las otras dos, son los procesos Editor que hemos creado.

En **2**, podemos ver el proceso del propio Monitor de sistema. Vemos que por cada aplicación java que lancemos se crea una instancia de la máquina virtual. Por lo tanto, el que gestiona los procesos java sigue siendo el SO y no la máquina virtual.

# Comandos ps y kill

1. Comenzamos ejecutando el comando ps, este es el resultado:



```
margye@Pearl: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
usuario@Pearl:~$ ps ax  
  PID TTY          STAT       TIME COMMAND  
    1 ?           Ss          0:00 /sbin/init  
    2 ?           S            0:00 [kthreadd]  
    3 ?           S            0:04 [ksoftirqd/0]  
    4 ?           S            0:02 [migration/0]  
    5 ?           S            0:00 [watchdog/0]  
    6 ?           S            0:00 [migration/1]  
    7 ?           S            0:01 [ksoftirqd/1]  
    8 ?           S            0:00 [watchdog/1]  
 3442 ?           Sl           4:08 gimp-2.6  
 3447 ?           S            0:00 /usr/lib/gimp/2.0/plugin-script-fu -gimp 11 10 -run  
 6270 ?           Sl           0:00 gnome-terminal  
 6273 ?           S            0:00 gnome-pty-helper  
 6274 pts/0        Ss          0:00 bash  
 6373 pts/0        Rl          10:14 gnome-system-monitor  
 6976 ?           S            0:00 /bin/bash /home/margye/NetBeans/netBeans-6.9.1/bin/..  
 7373 ?           Sl           1:00 /usr/lib/jvm/default-java/bin/java -Djdk.home=/usr/li  
 7451 ?           Sl           0:02 /usr/lib/jvm/default-java/bin/java -Dfile.encoding=UT  
 7481 ?           Sl           0:03 java -jar /home/margye/NetBeansProjects/Editor/dist/D  
 7507 ?           Sl           0:03 java -jar /home/margye/NetBeansProjects/Editor/dist/D  
 7632 pts/0        R+          0:00 ps ax
```

2. Del listado anterior obtenemos el PID para terminar, uno de los procesos, por ejemplo, uno de los procesos java.

```
usuario@Pearl:~$ kill -9 7481
```



# SysInternals: Process Explorer y Process Monitor

Es interesante conocer las utilidades avanzadas de SysInternals, porque nos ofrecen información más amplia sobre los procesos del sistema; permitiéndonos así comprobar el correcto funcionamiento de las aplicaciones que implementemos.

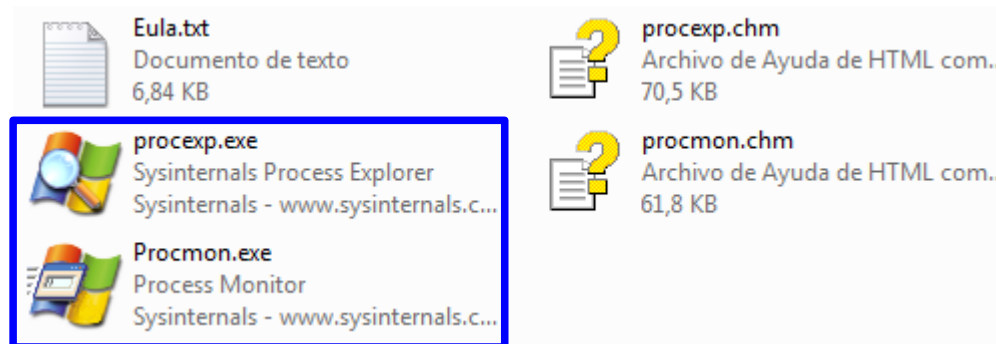
- **Process Explorer**, nos dará información más detallada que el Administrador de tareas sobre los procesos del sistema. Podemos descargar Process Explorer desde esta url:

<http://download.sysinternals.com/Files/ProcessExplorer.zip> (1,5MB)

- **Process Monitor**, nos informará de la actividad (de E/S) de los procesos e hilos activos en el sistema: ficheros a los que están accediendo, actividad en red, creación de hilos, etc... Descargaremos Process Monitor desde esta otra url:

<http://download.sysinternals.com/Files/ProcessMonitor.zip> (1,26MB)

- Sólo tendremos que descargarlos, descomprimirlos en una carpeta y hacer doble clic sobre su ejecutable, para lanzar su ejecución.

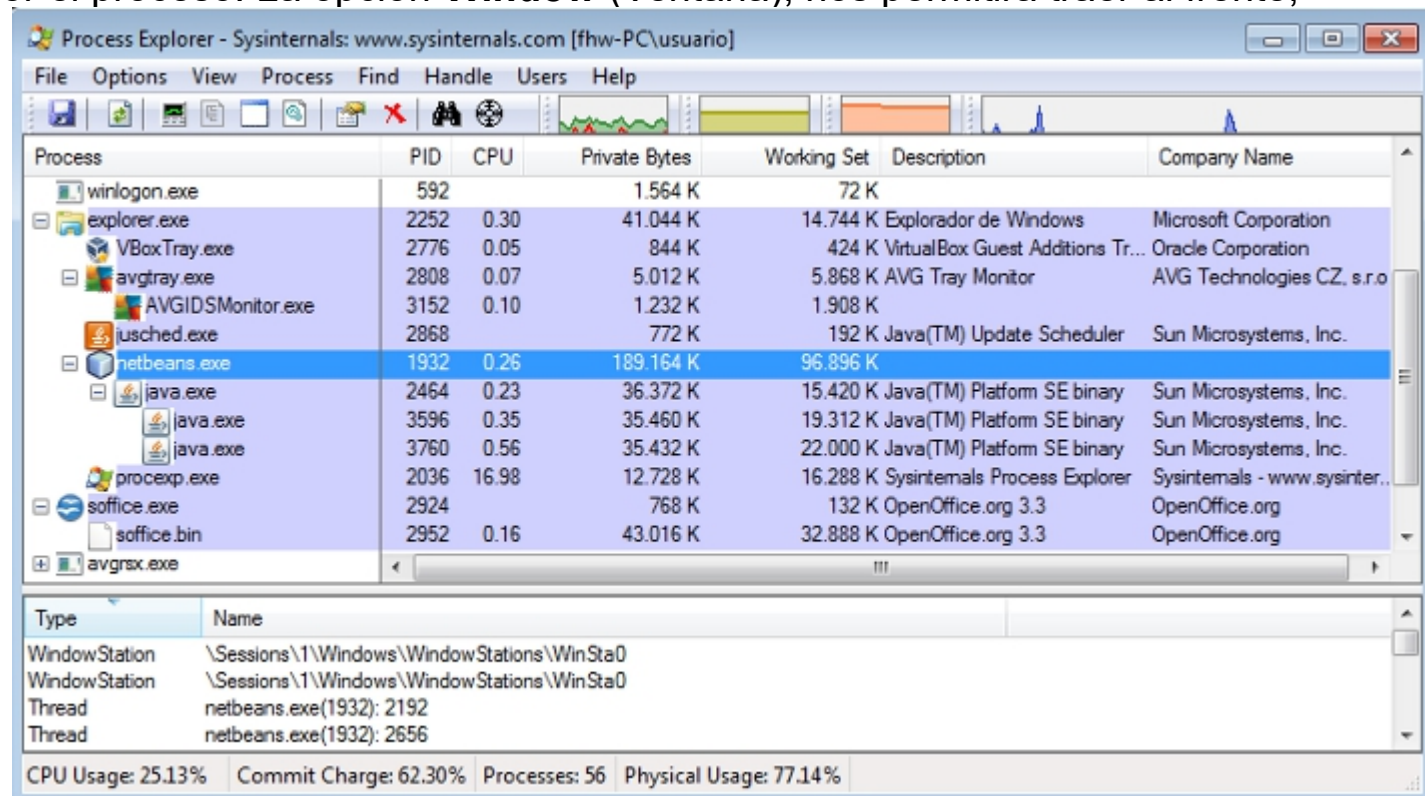


Utilidades avanzadas SysInternals: Process Explorer y Process Monitor.  
Ficheros resultado de descomprimir las descargas.

# SysInternals: Process Explorer y Process Monitor

Comencemos con Process Explorer:

- Muestra los **procesos en forma de árbol**. Así podemos **ver la dependencia** de procesos padre (ha creado a otro proceso) e hijo.
- Para cada proceso muestra el **nombre** de su fichero ejecutable, su **icono** (nos facilita la búsqueda de los procesos de cada aplicación), **PID**, porcentaje de uso de la CPU, tamaño del conjunto de trabajo, ...
- Seleccionando un proceso, en el **panel inferior**, nos mostrará los **recursos que está utilizando**: ficheros, librerías, hilos, claves de registro, etc...
- En el **menú contextual de cada proceso**, las opciones que tendremos: Cambiar su **prioridad**; **Matar** el proceso; Matar el árbol de procesos; Reiniciar, parar y reanudar el proceso; Crear un fichero de volcado del espacio de memoria utilizado por el proceso. La opción **Window** (Ventana), nos permitirá traer al frente, maximizar, minimizar, restaurar y cerrar la **ventana de la aplicación asociada con ese proceso**.
- El menú **File** (Archivo), nos permitirá ejecutar nuevos procesos, también con privilegios de Administrador. Nos permite mostrar los detalles de todos los procesos del sistema (no sólo los nuestros). Y, además de cerrar la aplicación, nos permite Cerrar sesión, Apagar, Reiniciar y Bloquear el equipo.

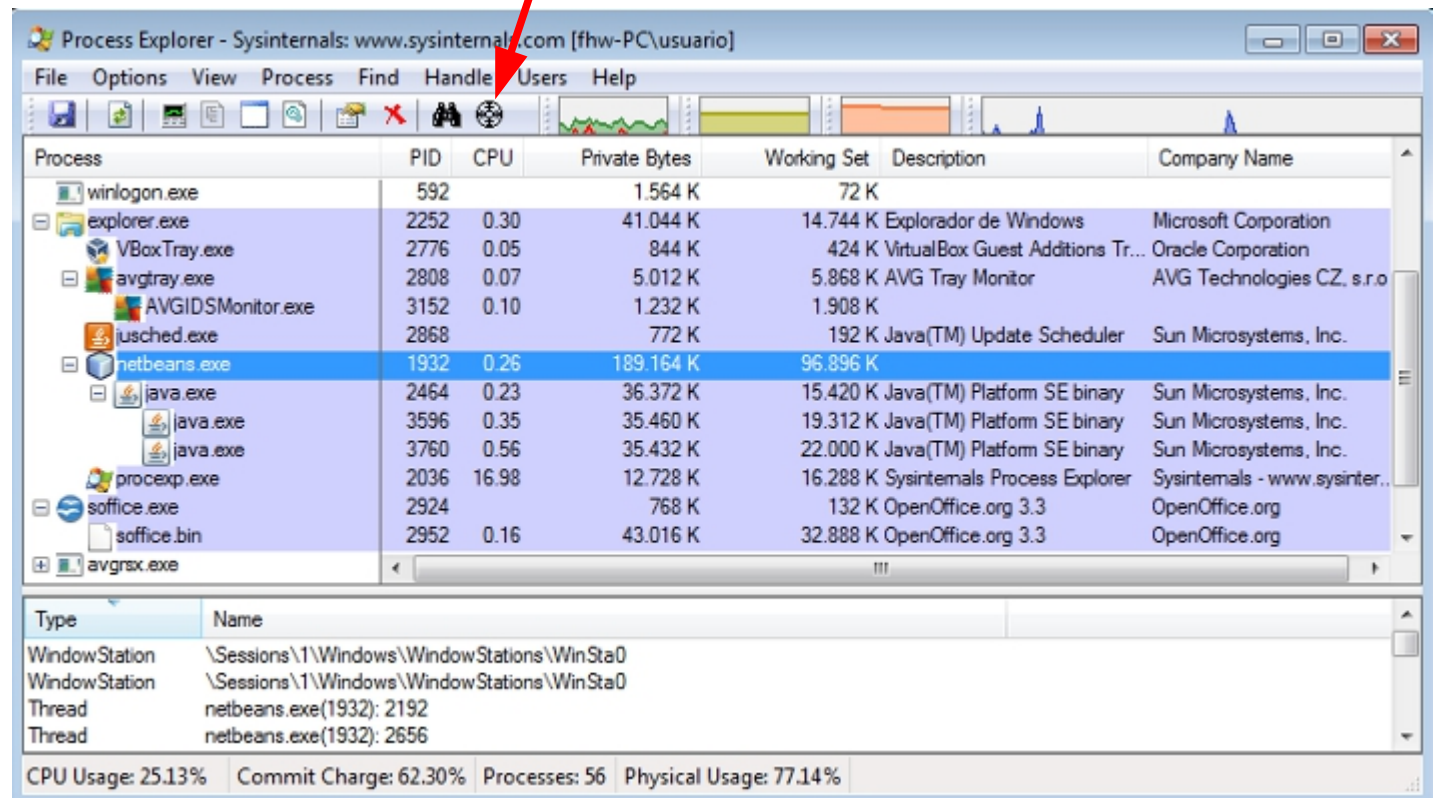


Aspecto usual de Process Explorer.

# SysInternals: Process Explorer y Process Monitor

Continuamos con algunas curiosidades más de Process Explorer:

- En el menú **Options**, podemos hacer que Process Explorer sustituya al Administrador de tareas.
- Como información del sistema, además del uso de la CPU, la memoria y el uso de la interfaz de red, podemos ver estadísticas sobre las operaciones de E/S en general y los accesos a disco.
- Para finalizar, pulsando y soltando sobre una ventana el icono marcado en la captura, Process Explorer nos **buscará en su listado de procesos cuál es el proceso que se corresponde con esa ventana**. Esto es muy útil para identificar cada uno de nuestros procesos. Sino, ¿cómo determinamos qué ventana de nuestro Editor es cada proceso?



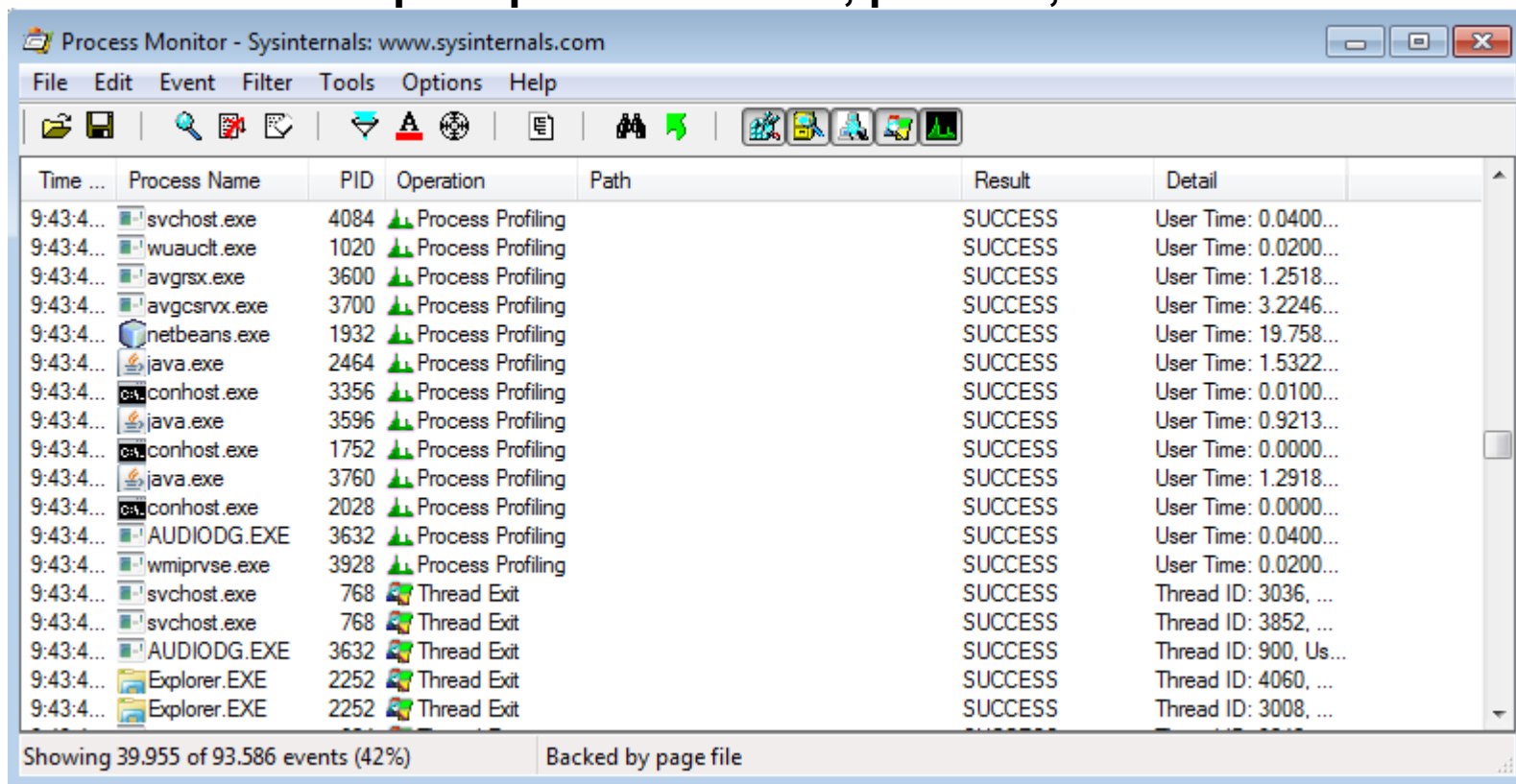
Aspecto usual de Process Explorer.

Como ves **Process Explorer** va a ser **muy útil** en nuestras pruebas de aplicaciones con varios procesos (después, hilos y servicios).

# SysInternals: Process Explorer y Process Monitor

La utilidad de Process Monitor está en:

- Muestra un **registro de todas las actividades que realizan los procesos.**
- Para cada uno de esos registros de actividad, nos permite revisar las **propiedades** de ese evento, su tiempo de duración y si se realizó con éxito o no.
- Permite realizar **filtrados por tipo de actividad, proceso, ...**



Aspecto usual de Process Monitor.

**Process Explorer** va a ser otra de nuestras herramientas en la depuración de la actividad de nuestros procesos (después, hilos y servicios).

# Credenciales

| Imagen  | Datos de licencia |
|---|-------------------|
| <p>Todas las capturas de pantalla de esta presentación, tienen como datos de licencia:</p> <p>Autoría: Margarita I. Nieto Castillejo<br/>Licencia: Uso educativo-no comercial.<br/>Procedencia: Capturas de pantalla de los sistemas operativos Windows 7 y Ubuntu 10.10.; sus terminales de comandos; del Administrador de tareas de Windows 7, del Monitor de sistema de Ubuntu 10.10.; y las utilidades SysInternals Process Explorer y Process Monitor.</p> |                   |