Hecho por: David Fernández, Jose Garcia, Jaime Molina, Jesus Molina, Gabriel Cupsan y Jesus Molina

1. Algoritmos de planificación de sistemas operativos.

1.1. Generalidades.

Cuando un PC es multiprograma, se suele dar el caso de que varios procesos compiten por la CPU al mismo tiempo. Si solo existe una CPU disponible se tiene que decidir cuál de los procesos listos para ejecutarse será ejecutado a continuación. De esto se encarga una parte del S.O. que se llama planificador de procesos y el algoritmo que utiliza se llama algoritmo de planificación.

1.2.Categorías de algoritmos de planificación.

Distintos entornos necesitan diferentes algoritmos de planificación y los diferentes S.O. tienen sus propios objetivos por lo que el planificador de procesos tiene que optimizar no será lo mismo en todos los sistemas. Tres de los entornos más destacables son:

- Procesamiento por lotes
- Interactivo
- Tiempo real.

1.3.Metas de los algoritmos de planificación

Para poder diseñar un algoritmo de programación, es necesario tener idea de lo que debe hacer un algoritmo. Algunos objetivos dependen del entorno como el procesamiento por lotes, interactivo o de tiempo real), pero hay también algunos otros que son deseables en todos los casos.

1.3.1. Todos los sistemas

- Equidad – Otorgar a cada proceso una parte justa de la CPU
- Aplicación de políticas – Verificar que se lleven a cabo las políticas establecidas
- Balance – Mantener ocupadas todas las partes del sistema.

1.3.2. Sistemas de procesamiento por lotes

- Rendimiento – Maximizar el número de trabajos por hora
- Tiempo de retorno – Minimizar el tiempo entre la entrega y la terminación
- Utilización de la CPU – Mantener ocupada la CPU todo el tiempo

1.3.3. Sistemas interactivos

- Tiempo de respuesta – Responder a las peticiones con rapidez
- Proporcionalidad – Cumplir las expectativas de los usuarios

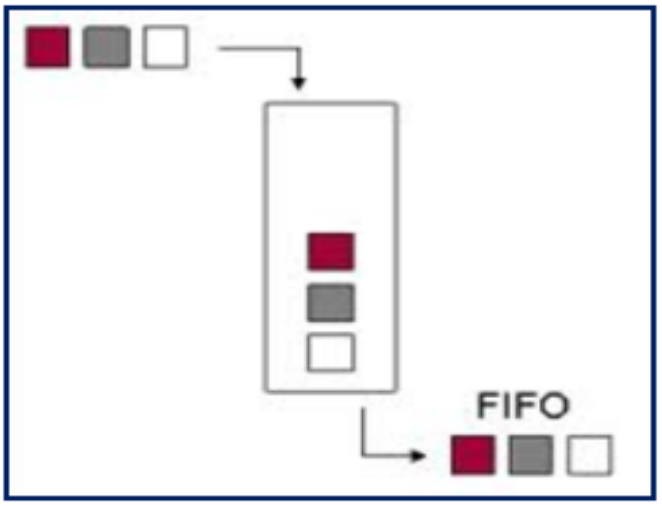
1.3.4. Sistemas de tiempo real

- Cumplir con los plazos – Evitar perder datos
- Predictibilidad – Evitar la degradación de la calidad en los sistemas multimedia

1.4.Algoritmos de Planificación en sistemas de procesamiento por lotes.

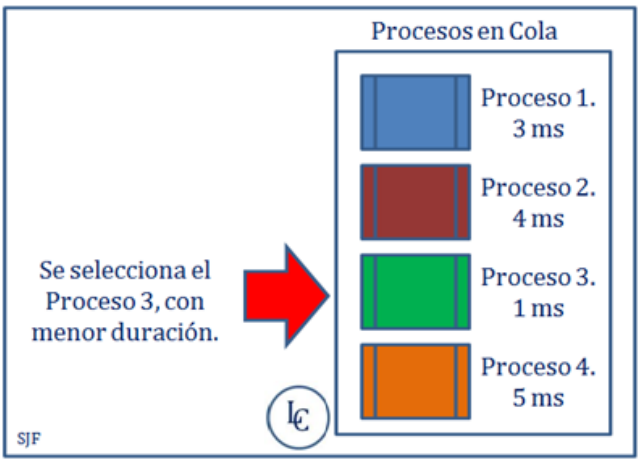
1.4.1. FIFO:

Acrónimo de “First in, first out” (primero que entra, primero que sale). Con este algoritmo la CPU se asigna a los procesos en el orden en el que la solicitan. Solo hay una sola cola de procesos listos. Cuando el primer trabajo entra al sistema, se inicia de inmediato y se le permite ejecutarse. A medida que van entrando otros trabajos, se colocan al final de la cola. Si el proceso en ejecución se bloquea, el primer proceso en la cola se ejecuta a continuación. Cuando un proceso bloqueado pasa al estado listo, al igual que un trabajo recién llegado, se coloca al final de la cola. Este algoritmo es fácil de comprender e igualmente sencillo de programar.



1.4.2. SJF:

Acrónimo de Shortest Job First (trabajo más corto primero) y algoritmo que supone que los tiempos de ejecución se conocen de antemano. Cuando hay varios trabajos de igual importancia esperando a ser iniciados en la cola de entrada, el planificador selecciona el trabajo más corto primero.



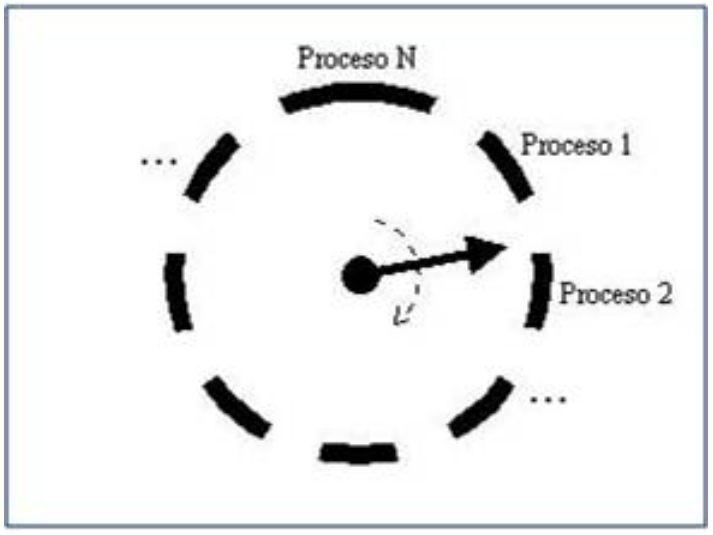
1.4.3.SRTN:

Shortest Remaining Time Next (menor tiempo restante a continuación). Algoritmo donde el planificador siempre selecciona el proceso cuyo tiempo restante de ejecución sea el más corto. De nuevo, se debe conocer el tiempo de ejecución de antemano. Cuando llega un nuevo trabajo, su tiempo total se compara con el tiempo restante del proceso actual. Si el nuevo trabajo necesita menos tiempo para terminar que el proceso actual, éste se suspende y el nuevo trabajo se inicia.

1.5.Algoritmos de Planificación en sistemas interactivos.

1.5.1. Round Robin:

Uno de los algoritmos más antiguos, simples, equitativos y de mayor uso es el de turno circular (round-robin). A cada proceso se le asigna un intervalo de tiempo, conocido como quántum, durante el cual se le permite ejecutarse. Si el proceso se sigue ejecutando al final del quántum, la CPU es apropiada para dársela a otro proceso. Si el proceso se bloquea o termina antes de que haya transcurrido el quántum, la conmutación de la CPU se realiza cuando el proceso se bloquea. Un quántum con un valor entre 20 y 50 mseg es lo más adecuado para un uso óptimo de la CPU por proceso.



1.5.2. Por prioridad:

La idea básica es simple: a cada proceso se le asigna una prioridad y el proceso ejecutable con la prioridad más alta es el que se puede ejecutar. Para evitar que los procesos con alta prioridad se ejecuten de manera indefinida, el planificador puede reducir la prioridad del proceso actual en ejecución en cada pulso del reloj (es decir, en cada interrupción del reloj). Si esta acción hace que su prioridad se reduzca a un valor menor que la del proceso con la siguiente prioridad más alta, ocurre una conmutación de procesos. De manera alternativa, a cada proceso se le puede asignar un quántum de tiempo máximo que tiene permitido ejecutarse.

1.5.3. Planificación garantizada:

Si hay por ejemplo 5 usuarios conectados mientras está trabajando, recibirá aproximadamente 1/5 del poder de la CPU. De manera similar, en un sistema de un solo usuario con 5 procesos en ejecución, mientras no haya diferencias, cada usuario debe obtener 1/5 de los ciclos de la CPU.

Para cumplir esta promesa, el sistema debe llevar la cuenta de cuánta potencia de CPU ha tenido cada proceso desde su creación. Después calcula cuánto poder de la CPU debe asignarse a cada proceso al saber el tiempo desde que se creó dividido entre 5 en este caso.

1.5.4. Planificación por partes equitativas:

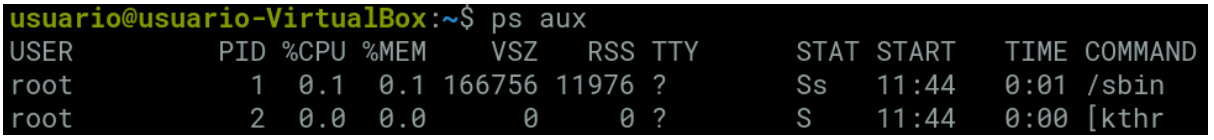
Hasta ahora hemos asumido que cada proceso se planifica por su cuenta, sin importar quién sea su propietario. Como resultado, si el usuario 1 inicia 9 procesos y el usuario 2 inicia 1 proceso, con la planificación por turno circular o por prioridades iguales, el usuario 1 obtendrá 90 por ciento del tiempo de la CPU y el usuario 2 sólo recibirá 10 por ciento. Para evitar esta situación, algunos sistemas toman en consideración quién es el propietario de un proceso antes de planificarlo. En este modelo, a cada usuario se le asigna cierta fracción de la CPU y el planificador selecciona procesos de tal forma que se cumpla con este modelo. Por ende, si a dos usuarios se les prometió 50 por ciento del tiempo de la CPU para cada uno, eso es lo que obtendrán sin importar cuántos procesos tengan en existencia.

1.6.Algoritmos de Planificación en sistemas de tiempo real:

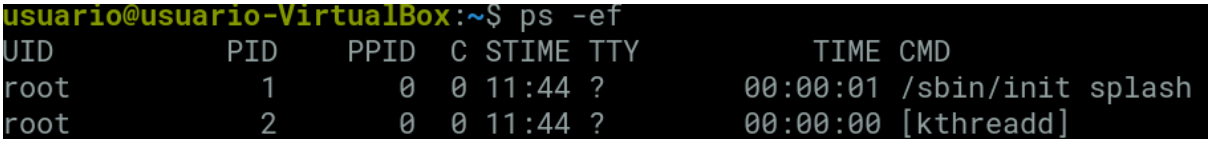
En un sistema de tiempo real, el tiempo desempeña un papel esencial. Por lo general, uno o más dispositivos físicos externos a la computadora generan estímulo y la computadora debe reaccionar de manera apropiada a ellos dentro de cierta cantidad fija de tiempo. En general, los sistemas de tiempo real se categorizan como de tiempo real duro, lo cual significa que hay tiempos límite absolutos que se deben cumplir, y como de tiempo real suave, lo cual significa que no es conveniente fallar en un tiempo límite en ocasiones, pero sin embargo es tolerable. En ambos casos, el comportamiento en tiempo real se logra dividiendo el programa en varios procesos, donde el comportamiento de cada uno de éstos es predecible y se conoce de antemano. Por lo general, estos procesos tienen tiempos de vida cortos y pueden ejecutarse hasta completarse en mucho menos de 1 segundo. Cuando se detecta un evento externo, es responsabilidad del planificador planificar los procesos de tal forma que se cumpla con todos los tiempos límite.

2. Opciones de interés del comando ps.

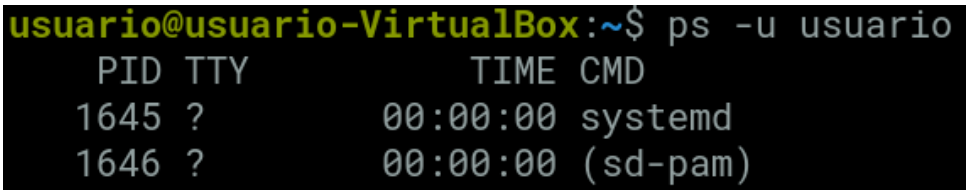
- ps aux: Enseña todos los procesos que se están ejecutando, incluyendo información detallada sobre el usuario, el estado del proceso y el uso de recursos.



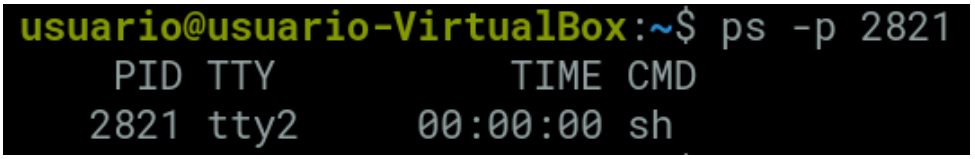
- ps -ef: Enseña todos los procesos en ejecución en el sistema, incluyendo información detallada sobre el usuario, el estado del proceso, el uso de recursos y el comando completo utilizado para iniciar el proceso.



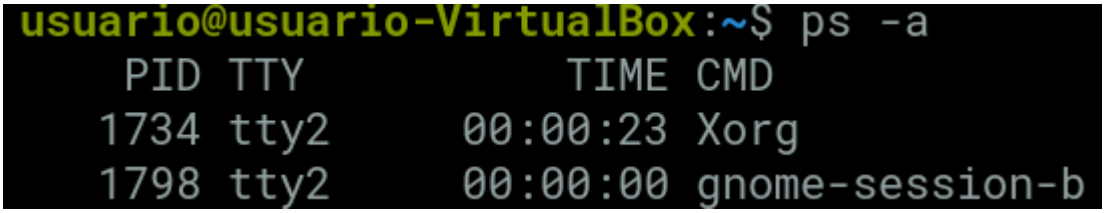
- ps -u [usuario]: Enseña todos los procesos en ejecución para un usuario específico. Reemplaza [usuario] con el nombre de usuario deseado.



- ps -p [pid]: Enseña información detallada sobre un proceso específico, identificado por su identificador de proceso (PID). Reemplaza [pid] con el PID deseado.



- ps -a: Muestra todos los procesos, incluyendo los que no tienen un terminal asociado.



- ps -x: Muestra todos los procesos, incluyendo los que no tienen un terminal asociado y los procesos de sistema.

```
usuario@usuario-VirtualBox:~$ ps -x
    PID TTY      STAT   TIME COMMAND
   1645 ?        Ss     0:00 /lib/systemd/systemd --user
   1646 ?        S      0:00 (sd-pam)
   1653 ?        S<sl   0:00 /usr/bin/pipewire
   1654 ?        Ssl    0:00 /usr/bin/pipewire-media-session
```

- ps -u: Muestra información detallada sobre el usuario, el estado del proceso y el uso de recursos.

```
usuario@usuario-VirtualBox:~$ ps -u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
usuario     1732  0.0  0.0 200608  6520 tty2     Ssl+ 11:45   0:00 /usr/libexec/gdm-x-session --register-session --run-script gnome-session --buil
usuario     1734  1.5  0.9 343428 83160 tty2     Sl+  11:45   0:25 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthority -nolist
usuario     1798  0.0  0.2 697752 19152 tty2     Sl+  11:45   0:00 /usr/libexec/gnome-session-binary --builtin --session=pantheon
usuario     2038  0.0  0.1 504208 10116 tty2     Sl+  11:45   0:00 /usr/libexec/gsd-sharing
usuario     2039  0.0  0.0 350336  7912 tty2     Sl+  11:45   0:00 /usr/libexec/gsd-housekeeping
```

- ps -p: Muestra información detallada sobre un proceso específico, identificado por su PID.

```
usuario@usuario-VirtualBox:~$ ps -p 2613
    PID TTY          TIME CMD
   2613 ?        00:00:00 zeitgeist-fts
```

- ps -e: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos y el comando completo utilizado para iniciar el proceso.

```
usuario@usuario-VirtualBox:~$ ps -e
    PID TTY          TIME CMD
      1 ?        00:00:01 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      5 ?        00:00:00 netns
```

- ps -f: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos, el comando completo utilizado para iniciar el proceso y los procesos hijos del proceso.

```
usuario@usuario-VirtualBox:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
usuario     3030    2999  0 11:48 pts/0    00:00:00 /bin/bash
usuario     4166    3030  0 12:16 pts/0    00:00:00 ps -f
```

- ps -l: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos, el comando completo utilizado para iniciar el proceso y los procesos hijos del proceso.

```
usuario@usuario-VirtualBox:~$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CM
D
0 S  1000    3030    2999  0  80   0 - 12285 do_wai pts/0    00:00:00 ba
sh
0 R  1000    4197    3030  0  80   0 - 12687 -      pts/0    00:00:00 ps
```

- ps -r: Muestra solo los procesos en ejecución.

```
usuario@usuario-VirtualBox:~$ ps -r
    PID TTY      STAT   TIME COMMAND
   4205 pts/0    R+     0:00 ps -r
```

- ps -s: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos y el comando completo utilizado para iniciar el proceso.

```
usuario@usuario-VirtualBox:~$ ps -s
  UID     PID         PENDING         BLOCKED         IGNORED          CAUGHT STAT TTY        TIME COMMAND
 1000    1732 0000000000000000 0000000000000000 0000000000001000 0000000100014000 Ssl+ tty2       0:00 /usr/libexec/gdm-x-session --register-ses
 1000    1734 0000000000000000 0000000000000000 0000000000001000 00000001418066ef Sl+  tty2       0:35 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth
 1000    1798 0000000000000000 0000000000000000 0000000000001000 0000000100014a02 Sl+  tty2       0:00 /usr/libexec/gnome-session-binary --built
 1000    2038 0000000000000000 0000000000000000 0000000000001000 0000000100004000 Sl+  tty2       0:00 /usr/libexec/gsd-sharing
 1000    2039 0000000000000000 0000000000000000 0000000000001000 0000000100004000 Sl+  tty2       0:00 /usr/libexec/gsd-housekeeping
 1000    2040 0000000000000000 0000000000000000 0000000000001000 0000000100004000 Sl+  tty2       0:00 /usr/libexec/gsd-wacom
 1000    2045 0000000000000000 0000000000000000 0000000000001000 0000000100004000 Sl+  tty2       0:00 /usr/libexec/gsd-datetime
 1000    2048 0000000000000000 0000000000000000 0000000000001000 0000000100004000 Sl+  tty2       0:00 /usr/libexec/gsd-rfkill
 1000    2049 0000000000000000 0000000000000000 0000000000001000 0000000100004000 Sl+  tty2       0:00 /usr/libexec/gsd-keyboard
 1000    2052 0000000000000000 0000000000000000 0000000000001000 0000000100004000 Sl+  tty2       0:00 /usr/libexec/gsd-sound
```

3. Opciones de interés del comando at.

El comando at en Linux se utiliza principalmente para la programación única de tareas. Si este comando no se encuentra en el sistema, se podrá obtener mediante el comando sudo apt install at.

```
rubencio@rubencio-VirtualBox:~$ sudo apt install at
[sudo] contraseña para rubencio:
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.
  dctrl-tools dkms libdouble-conversion3 libflashrom1 libftdi1-2
  libgsoap-2.8.117 liblzf1 libmd4c0 libpcre2-16-0 libqt5core5a libqt5dbus5
  libqt5gui5 libqt5network5 libqt5opengl5 libqt5printsupport5 libqt5svg5
  libqt5widgets5 libqt5x11extras5 libsdl1.2debian libxcb-xinerama0
  libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n virtualbox
  virtualbox-dkms virtualbox-qt
Utilice «sudo apt autoremove» para eliminarlos.
Paquetes sugeridos:
  default-mta | mail-transport-agent
Se instalarán los siguientes paquetes NUEVOS:
  at
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 11 no actualizados.
Se necesita descargar 41,1 kB de archivos.
Se utilizarán 166 kB de espacio de disco adicional después de esta operación.
```

Mediante este comando se pueden automatizar tareas para realizar en un momento único especificando la hora, fecha y la tarea que deseamos realizar. La sintaxis de este comando es: at hora [fecha] [-f fichero].

```
rubencio@rubencio-VirtualBox:~$ at 13:00
warning: commands will be executed using /bin/sh
at Thu Jan 19 13:00:00 2023
at> touch at_prueba2.txt
at> cp at_prueba2.txt /home/Rubencio/at
at>
at> <EOT>
job 2 at Thu Jan 19 13:00:00 2023
```

Las principales opciones de interés de este comando son las siguientes:

- Una de las características de este comando es la de recibir un correo electrónico al finalizar el proceso, esto se consigue mediante -m.
- Se pueden indicar tareas que se van a lanzar mediante un fichero con el modificador -f o por el contrario escribir la tarea y guardarla con el comando CTRL + R.
- Esta opción no es altamente destacable dado que muchos comandos cuentan con la misma función: at -l. De esta forma podemos listar los procesos listos para ejecutarse mediante at.

```
rubencio@rubencio-VirtualBox:~$ at -l
2       Thu Jan 19 13:00:00 2023 a rubencio
```

- De la mano de la opción anterior, tenemos los dos siguientes comando:
  - at -d num_tarea: Para eliminar la ejecución de cualquiera de las tareas programadas.
  - at -c num_tarea: Para ver el desglose de subtareas de una tarea programada

```
rubencio@rubencio-VirtualBox:~$ at -c 2
#!/bin/sh
# atrun uid=1000 gid=1000
# mail rubencio 0
umask 2
SESSION_MANAGER=local/rubencio-VirtualBox:@/tmp/.ICE-unix/1720,
rtualBox:/tmp/.ICE-unix/1720; export SESSION_MANAGER
QT_ACCESSIBILITY=1; export QT_ACCESSIBILITY
COLORTERM=truecolor; export COLORTERM
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg; export XDG_CONFIC
SSH_AGENT_LAUNCHER=gnome-keyring; export SSH_AGENT_LAUNCHER
XDG_MENU_PREFIX=gnome-; export XDG_MENU_PREFIX
GNOME_DESKTOP_SESSION_ID=this-is-deprecated; export GNOME_DESKT
GNOME_SHELL_SESSION_MODE=ubuntu; export GNOME_SHELL_SESSION_MOD
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh; export SSH_AUTH_SOCK
XMODIFIERS=@im=ibus; export XMODIFIERS
DESKTOP_SESSION=ubuntu; export DESKTOP_SESSION
GTK_MODULES=gail:atk-bridge; export GTK_MODULES
PWD=/home/rubencio; export PWD
LOGNAME=rubencio; export LOGNAME
XDG_SESSION_DESKTOP=ubuntu; export XDG_SESSION_DESKTOP
XDG_SESSION_TYPE=wayland; export XDG_SESSION_TYPE
SYSTEMD_EXEC_PID=1751; export SYSTEMD_EXEC_PID
XAUTHORITY=/run/user/1000/.mutter-Xwaylandauth.VIS0Y1; export X
HOME=/home/rubencio; export HOME
USERNAME=rubencio; export USERNAME
IM_CONFIG_PHASE=1; export IM_CONFIG_PHASE
LANG=es_ES.UTF-8; export LANG
```

.

- Trasladando la visión a un punto de vista más amplio, se encuentra el comando para listar las tareas programas, pero a diferencia del comando at -l, este comando lista las tareas de todos los usuarios. Este comando es atq.
-

```
rubencio@rubencio-VirtualBox:~$ atq
2       Thu Jan 19 13:00:00 2023 a rubencio
```

-
-
- El siguiente comando sigue las características del comando anterior pero su función es similar a la de at -d num_tarea. El comando atrm se encarga de eliminar las tareas programadas por cada usuario.

```
rubencio@rubencio-VirtualBox:~$ atrm 2
rubencio@rubencio-VirtualBox:~$ atq
rubencio@rubencio-VirtualBox:~$ at 13:00
warning: commands will be executed using /bin/sh
at Thu Jan 19 13:00:00 2023
at> touch at_prueba2
at> cp at_prueba2 /home/Rubencio/at
at> <EOT>
job 3 at Thu Jan 19 13:00:00 2023
```

- Si lo que se busca es comprobar el contenido de la tarea programada se recurre al comando at -c, de esta manera se podrá comprobar los comandos que se encuentran en la tarea.

- Junto con el comando at se puede encontrar un “túnel de tiempo” que sirve para acelerar la tarea: echo “touch at_prueba2.txt” | at now +1 minute.

```
rubencio@rubencio-VirtualBox:~$ at 13:05
warning: commands will be executed using /bin/sh
at Thu Jan 19 13:05:00 2023
at> touch at_prueba2.txt
at> cp at_prueba2.txt /home/Rubencio/at
at> <EOT>
job 4 at Thu Jan 19 13:05:00 2023
rubencio@rubencio-VirtualBox:~$ echo "touch at_prueba2.txt" | at now +1 minute
warning: commands will be executed using /bin/sh
job 5 at Thu Jan 19 13:03:00 2023
```

```
rubencio@rubencio-VirtualBox:~$ ls
at              Descargas   Escritorio  Imágenes  Plantillas  snap
at_prueba2.txt  Documentos  examen3     Música    Público     Vídeos
```

```
rubencio@rubencio-VirtualBox:~/at$ ls
at_prueba2.txt  at_test.txt
```

- Una de las opciones más curiosas y destacadas de este comando es que se puede restringir el acceso al mismo. Esto no se realizará mediante comando sino que se tendrá que acceder al archivo que contiene todos los nombre de usuario que pueden trabajar con este. La ruta para acceder a este archivo es la siguiente: /etc/a.deny.

```
rubencio@rubencio-VirtualBox:/etc$ nano at.deny
```

```
 GNU nano 6.2                                  at.deny *
alias
backup
bin
daemon
ftp
games
gnats
guest
irc
lp
mail
man
nobody
operator
proxy
qmaild
qmaill
qmailp
qmailq
qmailr
qmails
sync
sys
www-data
```

4. Analizar y destacar las herramientas descritas en el apartado 4.7.

```
jesus@jesus-VirtualBox:~$ uptime
 11:56:45 up 12 min,  1 user,  load average: 0,00, 0,02, 0,04
```

El comando *uptime* muestra la hora del sistema, el tiempo del sistema en activo, el número de usuarios y la carga del sistema en intervalos de 1,5 y 15 minutos. Equivale a la primera línea de la orden *top.*

```
jesus@jesus-VirtualBox:~$ top

top - 11:57:21 up 13 min,  1 user,  load average: 0,00, 0,02, 0,04
Tareas: 175 total,   4 ejecutar,  171 hibernar,    0 detener,    0 zombie
%Cpu(s):  0,0 us, 33,3 sy,  0,0 ni, 66,7 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   1976,0 total,    152,7 libre,    748,5 usado,   1074,8 búfer/caché
MiB Intercambio:   3220,0 total,   3218,2 libre,      1,8 usado.   1041,0 dispo

    PID USUARIO    PR  NI    VIRT    RES    SHR S  %CPU  %MEM     HORA+ ORDEN
   1643 jesus      20   0  386216  24500  18932 S   0,0   1,2   0:00.06 gsd-po+
   1644 jesus      20   0  258732  11440   9996 S   0,0   0,6   0:00.00 gsd-pr+
   1649 jesus      20   0  466740   6496   5856 S   0,0   0,3   0:00.00 gsd-rf+
   1658 jesus      20   0  245172   6608   6092 S   0,0   0,3   0:00.00 gsd-sc+
   1659 jesus      20   0  474956   9160   8012 S   0,0   0,5   0:00.00 gsd-sh+
   1675 jesus      20   0  468796   8300   7372 S   0,0   0,4   0:00.01 gsd-sm+
   1677 jesus      20   0  328372   9492   8428 S   0,0   0,5   0:00.00 gsd-so+
   1685 jesus      20   0  350076  22604  17432 S   0,0   1,1   0:00.05 gsd-wa+
   1692 jesus      20   0  815056  64788  49676 S   0,0   3,2   0:00.14 evolut+
   1694 jesus      20   0  246260   7580   6972 S   0,0   0,4   0:00.00 ibus-d+
   1695 jesus      20   0  356664  28936  18108 S   0,0   1,4   0:00.65 ibus-e+
   1700 jesus      20   0  232260   6740   5400 S   0,0   0,3   0:00.00 gsd-di+
   1707 jesus      20   0  246184   7756   7132 S   0,0   0,4   0:00.01 ibus-p+
   1751 jesus      20   0  351240  15116  13152 S   0,0   0,7   0:00.00 gsd-pr+
   1782 jesus      20   0  997652 152644  55088 S   0,0   7,5   0:02.99 snap-s+
   1784 jesus      20   0  172432   7476   6880 S   0,0   0,4   0:00.04 ibus-e+
   1873 jesus      20   0  353304  25360  18196 S   0,0   1,3   0:00.08 xdg-de+
   1897 jesus      20   0 2542920  26936  22068 S   0,0   1,3   0:00.03 gjs
   1919 jesus      20   0 2802640  56844  41796 S   0,0   2,8   0:00.32 gjs
   1958 jesus      20   0  171868   6744   6176 S   0,0   0,3   0:00.00 gvfsd-+
```

```
   1958 jesus      20   0  171868   6744   6176 S   0,0   0,3   0:00.00 gvfsd-+
   1960 root       20   0  474388  29288  24484 S   0,0   1,4   0:00.22 fwupd
   2288 jesus      20   0  571244  51928  39440 R   0,0   2,6   0:01.29 gnome-+
top - 11:59:19 up 15 min,  1 user,  load average: 0,08, 0,05, 0,04
Tareas: 174 total,   3 ejecutar,  171 hibernar,    0 detener,    0 zombie
%Cpu(s):  2,2 us,  0,0 sy,  0,0 ni, 97,8 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   1976,0 total,    152,7 libre,    747,4 usado,   1076,0 búfer/caché
MiB Intercambio:   3220,0 total,   3218,2 libre,      1,8 usado.   1042,0 dispo

   PPID     HORA+  %CPU  %MEM  PR  NI S    VIRT    RES   UID ORDEN          PID
   2306  0:00.58   0,0   0,2  20   0 R   21944   4260  1000 top           3063
      2  0:00.06   0,0   0,0  20   0 I       0      0     0 kworker/+     3061
   1423  0:00.10   0,0   1,5  20   0 S  511124  30404  1000 update-n+     2929
   2288  0:00.00   0,0   0,2  20   0 S   19660   4512  1000 bash          2306
   1300  0:02.08   0,0   2,6  20   0 R  572724  52660  1000 gnome-te+     2288
      1  0:00.23   0,0   1,4  20   0 S  474388  29288     0 fwupd         1960
   1300  0:00.00   0,0   0,3  20   0 S  171868   6744  1000 gvfsd-me+     1958
   1448  0:00.32   0,0   2,8  20   0 S 2802640  56844  1000 gjs           1919
   1300  0:00.03   0,0   1,3  20   0 S 2542920  26936  1000 gjs           1897
   1300  0:00.08   0,0   1,3  20   0 S  353304  25360  1000 xdg-desk+     1873
   1628  0:00.08   1,1   0,4  20   0 S  172432   7476  1000 ibus-eng+     1784
   1300  0:03.00   0,0   7,5  20   0 S  997652 152644  1000 snap-sto+     1782
   1300  0:00.00   0,0   0,7  20   0 S  351240  15116  1000 gsd-prin+     1751
   1300  0:00.01   0,0   0,4  20   0 S  246184   7756  1000 ibus-por+     1707
   1423  0:00.00   0,0   0,3  20   0 S  232260   6740  1000 gsd-disk+     1700
   1628  0:00.65   0,0   1,4  20   0 S  356664  28936  1000 ibus-ext+     1695
   1628  0:00.00   0,0   0,4  20   0 S  246260   7580  1000 ibus-dco+     1694
   1423  0:00.14   0,0   3,2  20   0 S  815056  64788  1000 evolutio+     1692
   1300  0:00.05   0,0   1,1  20   0 S  350076  22604  1000 gsd-wacom     1685
```

```
   1300    0:00.00   0,0   0,5  20   0 S  328372   9492  1000 gsd-sound     1677
   1300    0:00.01   0,0   0,4  20   0 S  468796   8300  1000 gsd-smar+     1675
   1300    0:00.00   0,0   0,5  20   0 S  474956   9160  1000 gsd-shar+     1659
[1]+  Detenido                 top
```

El comando ps se utiliza para mostrar por pantalla un listado de los procesos que se están ejecutando en el sistema

```
jesus@jesus-VirtualBox:~$ ps
    PID TTY          TIME CMD
   2306 pts/0    00:00:00 bash
   3063 pts/0    00:00:00 top
   3394 pts/0    00:00:00 ps
```

El comando *free* obtiene información sobre el espacio libre y usado de la memoria real y física.

```
jesus@jesus-VirtualBox:~$ free
               total        used        free      shared  buff/cache   availabl
e
Memoria:     2023460      765248      156112       34184     1102100     106705
2
Swap:        3297276        1804     3295472
```

La herramienta *vmstat* devuelve la información de la memoria RAM, memoria virtual, intercambios entre memoria RAM y disco, interrupciones y el procesador.

```
jesus@jesus-VirtualBox:~$ vmstat
procs -----------memoria---------- ---swap-- -----io---- -sistema-- ------cpu--
---
 r  b   swpd  libre búfer caché   si   so    bi    bo   in   cs us sy id wa st
 2  0   4900 185752  51968 1003484    0    2   374   193  493  237  1  1 98  0
 0
```

Su sintaxis es: *vmstat /tiempo /actualizaciones//,* donde:

- tiempo: es el tiempo transcurrido entre dos actualizaciones
- actualizaciones: es el número de muestras. Como: *vmstat 3 5.*

```
jesus@jesus-VirtualBox:~$ vmstat 3 5
procs -----------memoria---------- ---swap-- -----io---- -sistema-- ------cpu--
---
 r  b   swpd  libre búfer caché   si   so    bi    bo   in   cs us sy id wa st
 3  0   4900 182992  52016 1008152    0    2   333   172  488  227  1  1 98  0
 0
 0  0   4900 182992  52024 1008152    0    0     0    32  495  180  2  0 98  0
 0
 0  0   4900 182992  52032 1003524    0    0     0     8  502  328  1  0 99  0
 0
 0  0   4900 182992  52032 1003496    0    0     0     0  490  130  1  0 99  0
 0
 0  0   4900 182992  52040 1003488    0    0     0    17  453  154  1  0 99  0
 0
```

El uso del *df,* muestra el porcentaje de uso de la unidades de almacenamiento del sistema

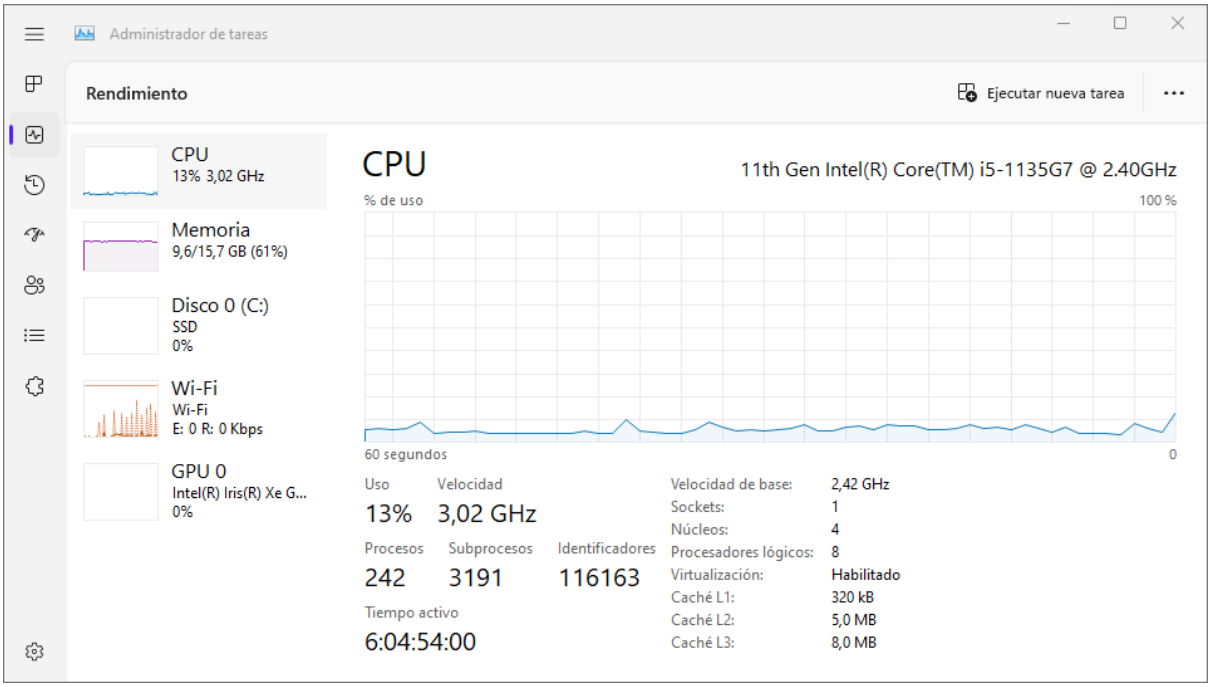
```
jesus@jesus-VirtualBox:~$ df
S.ficheros     bloques de 1K   Usados Disponibles Uso% Montado en
tmpfs             202348      1528      200820   1% /run
/dev/sda3       30267332  13109716    15594788  46% /
tmpfs            1011728         0     1011728   0% /dev/shm
tmpfs               5120         4        5116   1% /run/lock
/dev/sda2         524252      5364      518888   2% /boot/efi
tmpfs             202344      4716      197628   3% /run/user/1000
```

El comando w muestra quien esta conectado y que está haciendo (*parecido al* who)

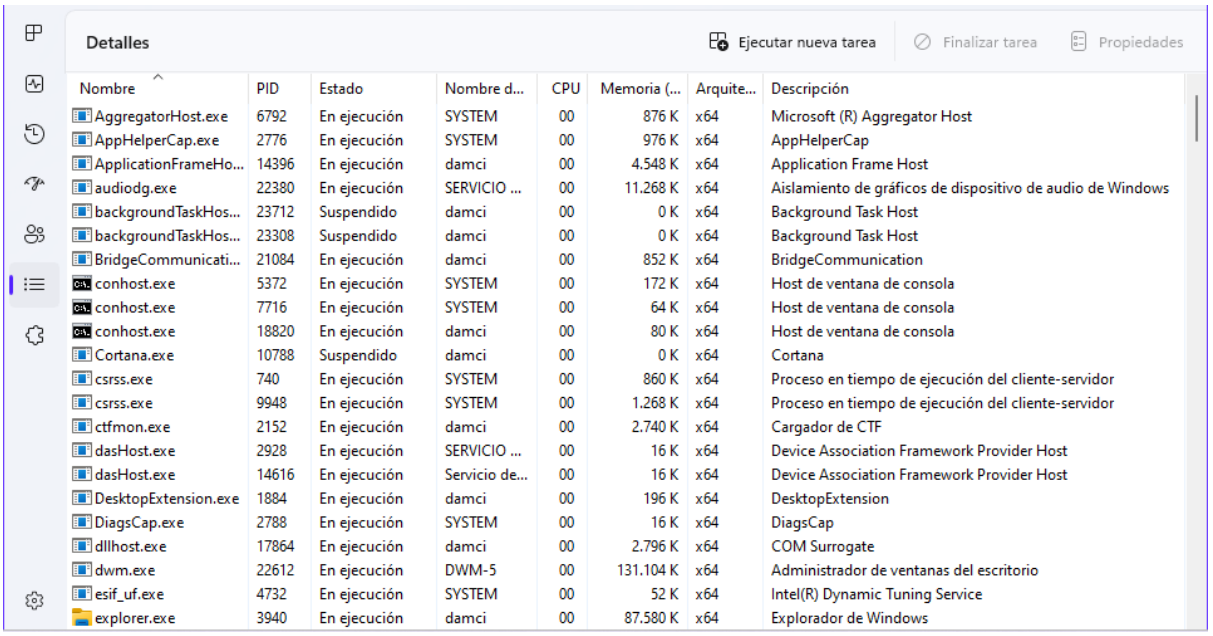
```
jesus@jesus-VirtualBox:~$ w
 12:05:30 up 21 min,  1 user,  load average: 0,00, 0,02, 0,01
USUARIO  TTY      DE               LOGIN@   IDLE   JCPU   PCPU WHAT
jesus    tty2     tty2             11:44   21:17   0.01s  0.00s /usr/libexec/g
jesus@jesus-VirtualBox:~$ who
jesus    tty2         2023-01-19 11:44 (tty2)
```

```
jesus@jesus-VirtualBox:~$ who
jesus    tty2         2023-01-19 11:44 (tty2)
```

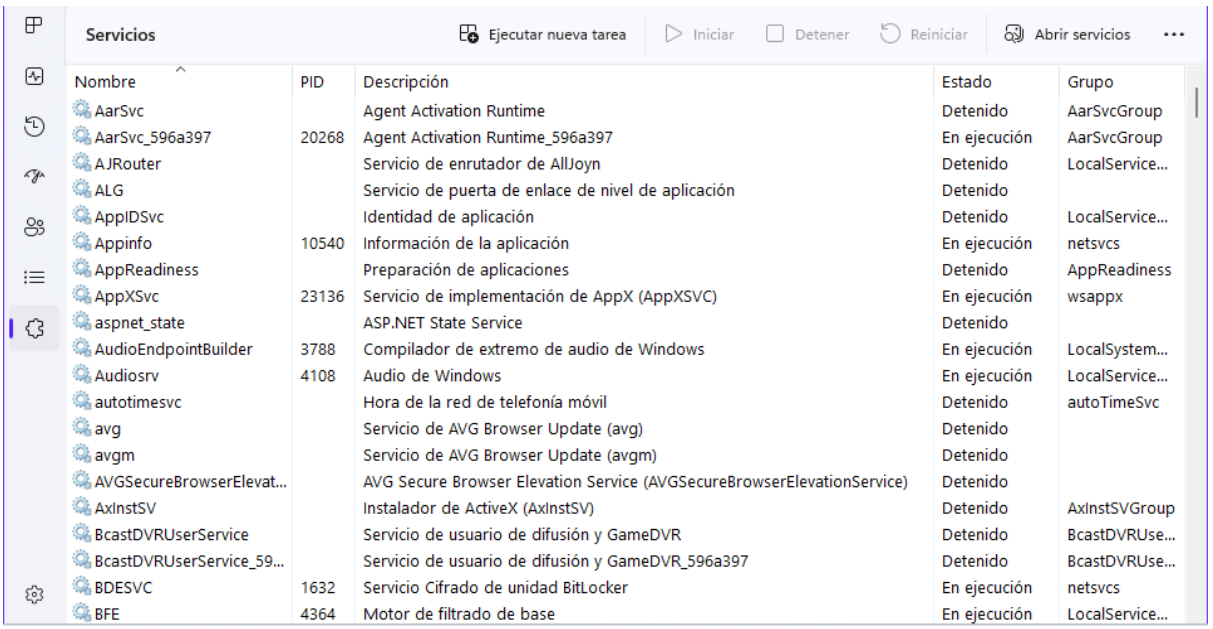
En windows 11, a través del ‘Administrador de tareas’ podemos hacer un estudio preciso del ‘Rendimiento’ donde podemos analizar el uso de CPU, cantidad de procesos y subprocesos activos, estado de la memoria RAM, usos de los discos, tráfico de los adaptadores de red…



Además, podemos estudiar las aplicaciones en segundo plano y los servicios de windows en las pestañas ‘Detalles’ y ‘Servicios’



Nombre	PID	Estado	Nombre d...	CPU	Memoria (...	Arquite...	Descripción
AggregatorHost.exe	6792	En ejecución	SYSTEM	00	876 K	x64	Microsoft (R) Aggregator Host
AppHelperCap.exe	2776	En ejecución	SYSTEM	00	976 K	x64	AppHelperCap
ApplicationFrameHo...	14396	En ejecución	damci	00	4.548 K	x64	Application Frame Host
audiodg.exe	22380	En ejecución	SERVICIO ...	00	11.268 K	x64	Aislamiento de gráficos de dispositivo de audio de Windows
backgroundTaskHos...	23712	Suspendido	damci	00	0 K	x64	Background Task Host
backgroundTaskHos...	23308	Suspendido	damci	00	0 K	x64	Background Task Host
BridgeCommunicati...	21084	En ejecución	damci	00	852 K	x64	BridgeCommunication
conhost.exe	5372	En ejecución	SYSTEM	00	172 K	x64	Host de ventana de consola
conhost.exe	7716	En ejecución	SYSTEM	00	64 K	x64	Host de ventana de consola
conhost.exe	18820	En ejecución	damci	00	80 K	x64	Host de ventana de consola
Cortana.exe	10788	Suspendido	damci	00	0 K	x64	Cortana
csrss.exe	740	En ejecución	SYSTEM	00	860 K	x64	Proceso en tiempo de ejecución del cliente-servidor
csrss.exe	9948	En ejecución	SYSTEM	00	1.268 K	x64	Proceso en tiempo de ejecución del cliente-servidor
ctfmon.exe	2152	En ejecución	damci	00	2.740 K	x64	Cargador de CTF
dasHost.exe	2928	En ejecución	SERVICIO ...	00	16 K	x64	Device Association Framework Provider Host
dasHost.exe	14616	En ejecución	Servicio de...	00	16 K	x64	Device Association Framework Provider Host
DesktopExtension.exe	1884	En ejecución	damci	00	196 K	x64	DesktopExtension
DiagsCap.exe	2788	En ejecución	SYSTEM	00	16 K	x64	DiagsCap
dllhost.exe	17864	En ejecución	damci	00	2.796 K	x64	COM Surrogate
dwm.exe	22612	En ejecución	DWM-5	00	131.104 K	x64	Administrador de ventanas del escritorio
esif_uf.exe	4732	En ejecución	SYSTEM	00	52 K	x64	Intel(R) Dynamic Tuning Service
explorer.exe	3940	En ejecución	damci	00	87.580 K	x64	Explorador de Windows



Nombre	PID	Descripción	Estado	Grupo
AarSvc		Agent Activation Runtime	Detenido	AarSvcGroup
AarSvc_596a397	20268	Agent Activation Runtime_596a397	En ejecución	AarSvcGroup
AJRouter		Servicio de enrutador de AllJoyn	Detenido	LocalService...
ALG		Servicio de puerta de enlace de nivel de aplicación	Detenido	
AppIDSvc		Identidad de aplicación	Detenido	LocalService...
Appinfo	10540	Información de la aplicación	En ejecución	netsvcs
AppReadiness		Preparación de aplicaciones	Detenido	AppReadiness
AppXSvc	23136	Servicio de implementación de AppX (AppXSVC)	En ejecución	wsappx
aspnet_state		ASP.NET State Service	Detenido	
AudioEndpointBuilder	3788	Compilador de extremo de audio de Windows	En ejecución	LocalSystem...
Audiosrv	4108	Audio de Windows	En ejecución	LocalService...
autotimesvc		Hora de la red de telefonía móvil	Detenido	autoTimeSvc
avg		Servicio de AVG Browser Update (avg)	Detenido	
avgm		Servicio de AVG Browser Update (avgm)	Detenido	
AVGSecureBrowserElevat...		AVG Secure Browser Elevation Service (AVGSecureBrowserElevationService)	Detenido	
AxInstSV		Instalador de ActiveX (AxInstSV)	Detenido	AxInstSVGroup
BcastDVRUserService		Servicio de usuario de difusión y GameDVR	Detenido	BcastDVRUse...
BcastDVRUserService_59...		Servicio de usuario de difusión y GameDVR_596a397	Detenido	BcastDVRUse...
BDESVC	1632	Servicio Cifrado de unidad BitLocker	En ejecución	netsvcs
BFE	4364	Motor de filtrado de base	En ejecución	LocalService...

5. Ampliar las aplicaciones descritas en el apartado 4.8.

- **Aplicaciones de actualización y control de drivers.**
    - Drivereasy: Es de las más clásicas, tiene una interfaz sencilla en la cual te encuentras una sección de update(actualizar), otra de scan, otra de la información del hardware del equipo y otra de herramientas. Su uso principal es la actualización de driver aunque el inconveniente que tiene es que a no ser que pagues la suscripción la actualización de esto tienes que hacerla manualmente.

- Driverbooster: Una aplicación muy parecida al drivereasy pero con la diferencia que la interfaz es más moderna y lo bueno de esta es que aunque tenga una versión de pago, con la versión gratis puedes hacer directamente la actualización de los drivers sin la necesidad de que sea manualmente.

- Device doctor: Esta trae una interfaz muy sencilla. Lo malo de esta es que se limita a solo poder descargar un driver por día si no se instala la versión de pago, además con la versión de pago te trae también una opción para realizar un backup a los drivers por si instalaras uno que está mal pueda volver hacia atrás. Además con la versión de pago te trae un desinstalador, un limpiador de caché, un monitor del sistema para poder comprobar como está.

- Slimdriver: Con una interfaz simple, en la cual las opciones son inició, explorar, resultados, opciones, restaurar, copias de seguridad y soporte. En principio no necesitas pagar para poder usar todas las opciones.

- **Aplicaciones de sincronización, copias de seguridad e imágenes del sistema.**

  - EaseUs Todo Backup Free:BackUp Maker almacena automáticamente sus archivos y al mismo tiempo ofrece una funcionalidad intuitiva.Viene con una interfaz de usuario sencilla pero con potentes habilidades para crear copias de seguridad del disco, la partición, el sistema operativo y los archivos y guardar las copias de seguridad en unidades locales, NAS, red y nube.

  - Clonezilla:Clonezilla es un programa de clonación de imágenes de partición y disco. Le ayuda a realizar la implementación del sistema, la copia de seguridad y la recuperación completa. Clonezilla guarda y restaura sólo los bloques usados en el disco duro, esto aumenta la eficiencia de la clonación.

  - FreeFileSync:Se trata de un programa gratuito y de código abierto que se utiliza para la sincronización de archivos. Es multiplataforma por lo que está disponible en Windows, Linux y OS X. Con FreeFyleSync podremos crear y gestionar copias de seguridad de nuestros archivos más importantes. Es un programa muy sencillo de usar ya que solo debemos de indicar las rutas de las dos carpetas que deseamos comparar.

  - Syncthing:Se trata de una aplicación que nos permitirá sincronizar archivos punto por punto entre dispositivos de una red local entre dispositivos remotos conectados a Internet.Syncthing está escrito en Go e implementa su propio protocolo de intercambio de bloques, igualmente libre, 2 que permite generar una nube personal bajo el modelo BYO, donde los usuarios proporcionan el hardware y el software para la misma.

- **Optimización del sistema.**

    - Windows 10:
        - Desinstala aplicaciones que no utilices
        - Mantén limpio el escritorio del ordenador
        - Controla las aplicaciones que se ejecuten al inicio
        - Comprueba que el PC esté libre de malware
        - Libera espacio en tu disco duro
        - Desfragmenta el disco duro
        - Configuración memoria virtual
        - Cambia el plan de energía del ordenador
        - Menos efectos visuales
        - Quítale transparencias a la interfaz
        - Haz que Windows 10 se quede callado
        - Busca actualizaciones de sistema y controladores
        - Reinstala Windows 10 desde cero
        - No te olvides de limpiar tu torre

    - Linux:

    - Hay unas cuantas maneras de optimizar un sistema Linux. Algunas cosas que puedes hacer son:

        - Limpiar el sistema: Eliminar archivos y programas no deseados puede liberar espacio en disco y mejorar el rendimiento.
        - Actualizar el sistema: Asegurarse de tener las últimas actualizaciones de seguridad y software puede ayudar a mejorar el rendimiento.
        - Optimizar el arranque: Ajustar el orden de inicio de los programas y servicios puede mejorar el tiempo de inicio.
        - Ajustar la configuración del kernel: Ajustar la configuración del kernel para adaptarse a las necesidades del sistema puede mejorar el rendimiento.
        - Instalar un gestor de ventanas ligero: Utilizar un gestor de ventanas ligero en lugar de uno más pesado puede mejorar el rendimiento en sistemas con pocos recursos.
        - Utilizar un almacenamiento en caché: Utilizar un almacenamiento en caché puede mejorar el rendimiento al reducir el tiempo de acceso a disco.
        - Usar herramientas de optimización: Herramientas como "sysctl" o "tune2fs" pueden ayudar a optimizar el sistema.