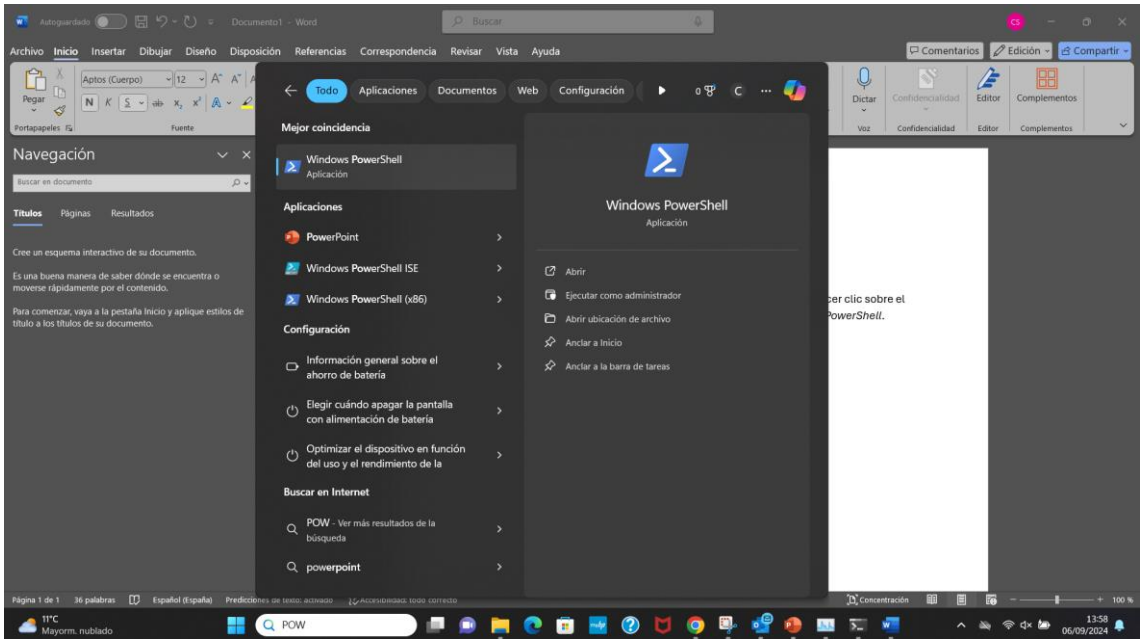
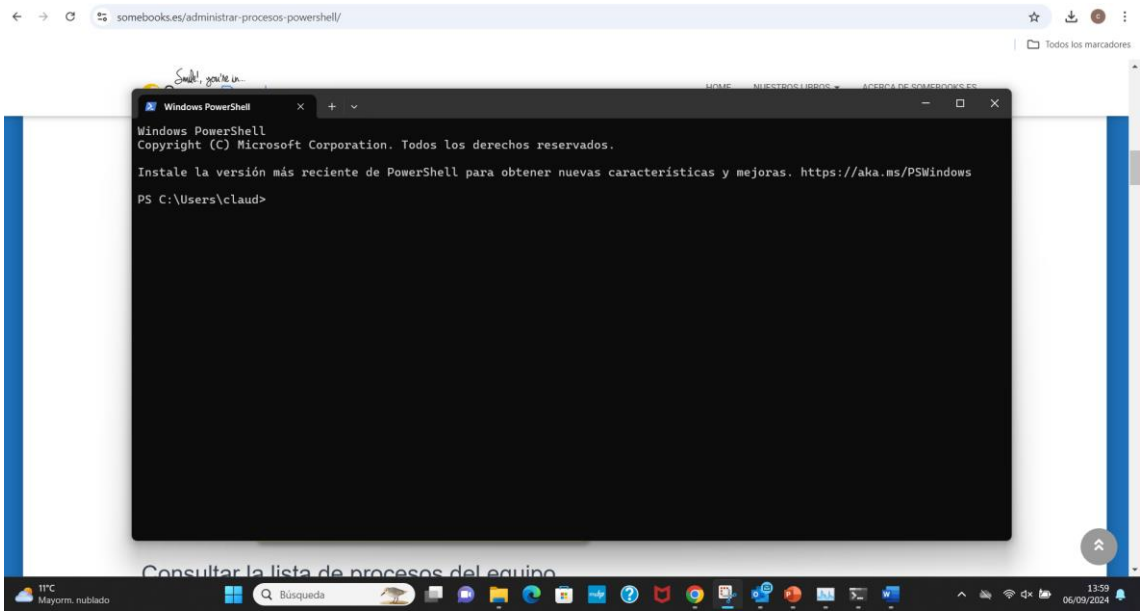
GESTIÓN DE PROCESOS CON POWERSHELL

1. Iniciar la ejecución de PowerShell

Para abrir una nueva ventana de *PowerShell*, comenzaremos por hacer clic sobre el botón *Inicio* y desplazarnos por el menú hasta la carpeta *Windows PowerShell*.

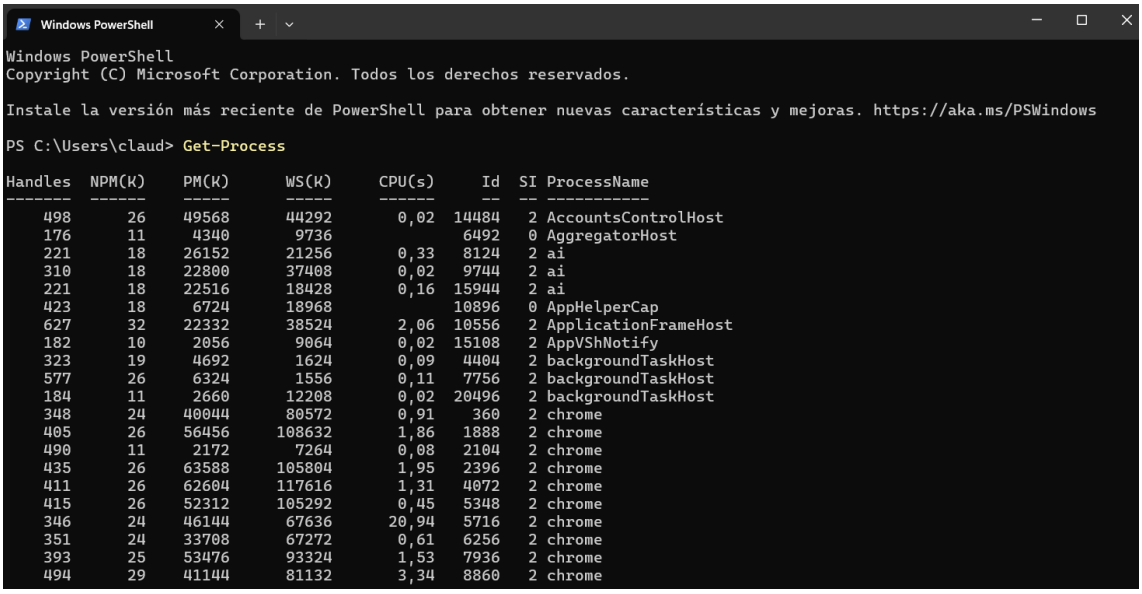


Al poco aparecerá una ventana, lista para comenzar a escribir nuestras órdenes:

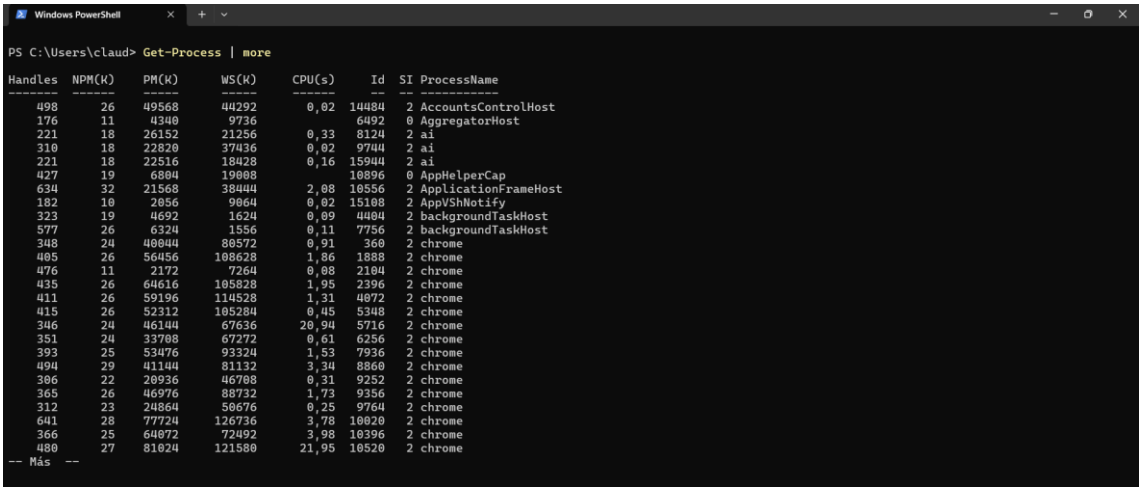


2. Consultar la lista de procesos del equipo

Para obtener la lista con todos los procesos que se están ejecutando en ese momento en el equipo, basta con utilizar el *comando* **Get-Process** sin argumentos:



Aunque, si la salida es demasiado larga y no cabe en la ventana, siempre puedes hacer una pausa procesándola con el comando more:



Y cuando necesitamos sólo algunos de los procesos, siempre podemos establecer un filtro. Por ejemplo, por su nombre:

```
PS C:\Users\claud> Get-Process -Name a*

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    498      26    49568      44292       0,02   14484   2 AccountsControlHost
    176      11     4340       9536              6492   0 AggregatorHost
    221      18    26152      20972       0,33    8124   2 ai
    310      18    22872      37496       0,02    9744   2 ai
    221      18    22516      18176       0,16   15944   2 ai
    423      18     6760      18904              10896   0 AppHelperCap
    640      33    20756      38792       2,11   10556   2 ApplicationFrameHost
    182      10     2056       8716       0,02   15108   2 AppVShNotify
```

Por ejemplo, por ID:

```
PS C:\Users\claud> Get-Process -id 18856

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    722      34    57256      68200       0,31   18856   1 Notepad
```

Por otro lado, si lo que necesitamos es una información detallada sobre un proceso, o un grupo de procesos, podemos enviar la salida de **Get-Process** a **Format-List**, que se encarga de formatear la salida de las propiedades de un objeto de modo que cada una aparezca en una nueva línea.

```
PS C:\Users\claud> Get-Process -Name explorer | Format-List *


Name                       : explorer
Id                         : 9780
PriorityClass              : Normal
FileVersion                : 10.0.22621.4168 (WinBuild.160101.0800)
HandleCount                : 7100
WorkingSet                 : 312496128
PagedMemorySize            : 377860096
PrivateMemorySize          : 377860096
VirtualMemorySize          : 1450520576
TotalProcessorTime         : 00:02:20.8750000
SI                         : 2
Handles                    : 7100
VM                         : 2204768743424
WS                         : 312496128
PM                         : 377860096
NPM                        : 192648
Path                       : C:\WINDOWS\Explorer.EXE
Company                    : Microsoft Corporation
CPU                        : 140,875
ProductVersion             : 10.0.22621.4168
Description                : Explorador de Windows
Product                    : Sistema operativo Microsoft® Windows®
__NounName                 : Process
BasePriority               : 8
ExitCode                   :
HasExited                  : False
ExitTime                   :
Handle                     : 3296
SafeHandle                 : Microsoft.Win32.SafeHandles.SafeProcessHandle
MachineName                : .
```

3. Detener un proceso por nombre

Cuando necesitamos forzar la detención de un proceso que está ejecutándose, podemos recurrir al *cmdlet* **Stop-Process**. Por ejemplo, podríamos escribir la siguiente orden para detener el navegador (necesario abrir primero el navegador para que cargue el proceso):

Stop-Process -name chrome*

Si lo prefieres, podrías utilizar el argumento -id y utilizar el número de proceso. Por ejemplo, si abrimos el Bloc de notas, vemos que el ID asociado es 22032

```
PS C:\Users\claud> Get-Process -Name n*

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    234      15     3508      12048               18168   0 NetworkCap
    627      33    67240     106868       0,03   22032   2 Notepad
```

Si ejecutamos el comando vemos que se ha borrado el proceso:

```
PS C:\Users\claud> Stop-Process -id 22032
PS C:\Users\claud> Get-Process -Name n*

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    234      15     3508      12048               18168   0 NetworkCap
```

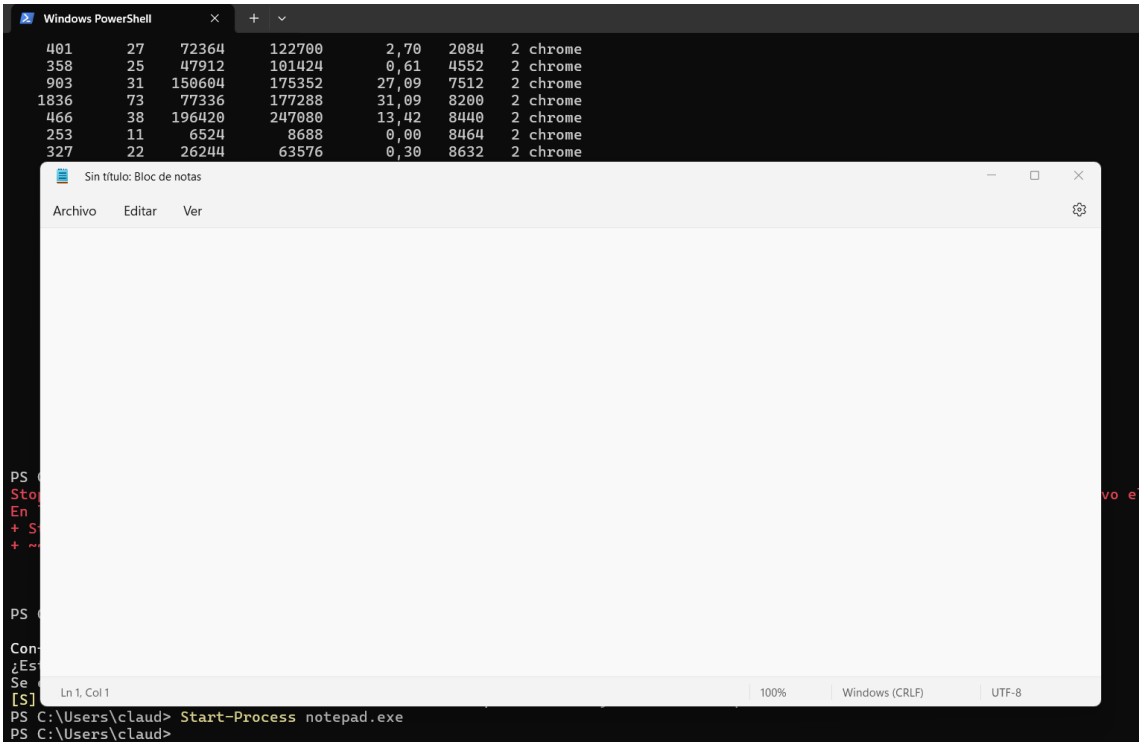
Incluso podemos hacer que el *cmdlet* nos pida confirmación antes de parar el proceso. Para ello, basta con utilizar el argumento -Confirm. Por ejemplo, abrir la calculadora y ver que está activo el proceso asociado:

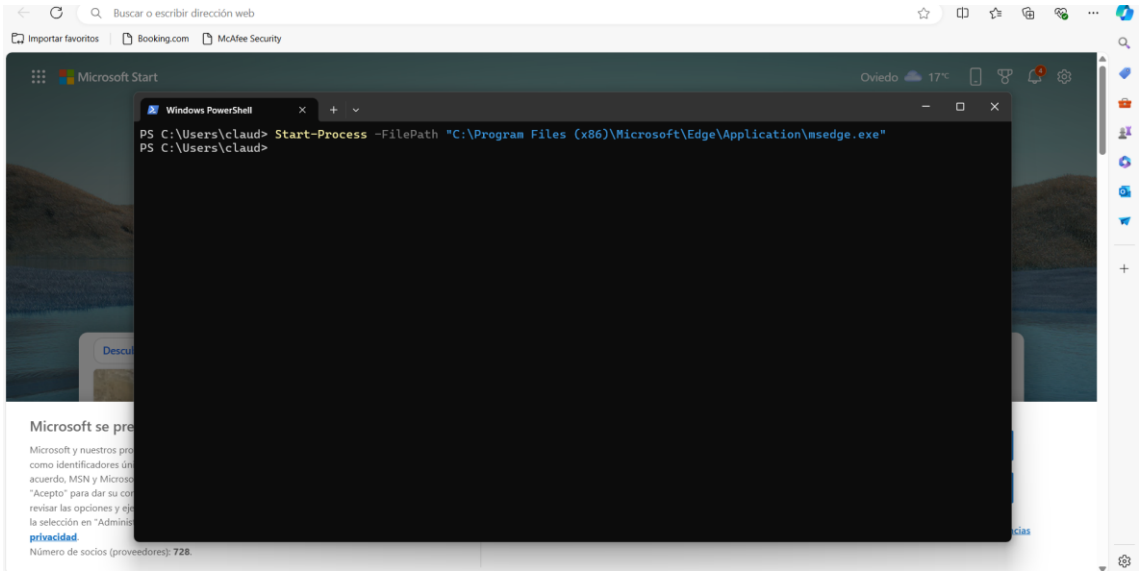
```
PS C:\Users\claud> Get-Process -Name c*

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    646      47    66648     110112       0,11    7004   2 CalculatorApp
    401      27    72364     122700       2,70    2084   2 chrome
    358      25    47912     101424       0,61    4552   2 chrome
    903      31   150604     175352      27,09    7512   2 chrome
   1836      73    77336     177288      31,09    8200   2 chrome
    466      38   196420     247080      13,42    8440   2 chrome
    253      11     6524       8688       0,00    8464   2 chrome
```

```
PS C:\Users\claud> Stop-Process -name CalculatorApp -Confirm

Confirmar
¿Está seguro de que desea realizar esta acción?
Se está realizando la operación "Stop-Process" en el destino "CalculatorApp (7004)".
[S] Sí  [O] Sí a todo  [N] No  [T] No a todo  [U] Suspender  [?] Ayuda (el valor predeterminado es "S"):
```

4. Iniciar un nuevo proceso

Podemos utilizar el comando Start-Process incluyendo a continuación el nombre de un programa o de un script.





5. Detener un proceso por ID

Cuando necesitamos forzar la detención de un proceso que está ejecutándose, podemos recurrir al *cmdlet* **Stop-Process**. Por ejemplo, podríamos escribir la siguiente orden para detener Notepad (necesario abrir primero el navegador para que cargue el proceso):

Stop-Process -id numero

Primero buscamos el ID del proceso que queremos detener. En este caso tenemos dos procesos abiertos de Notepad.

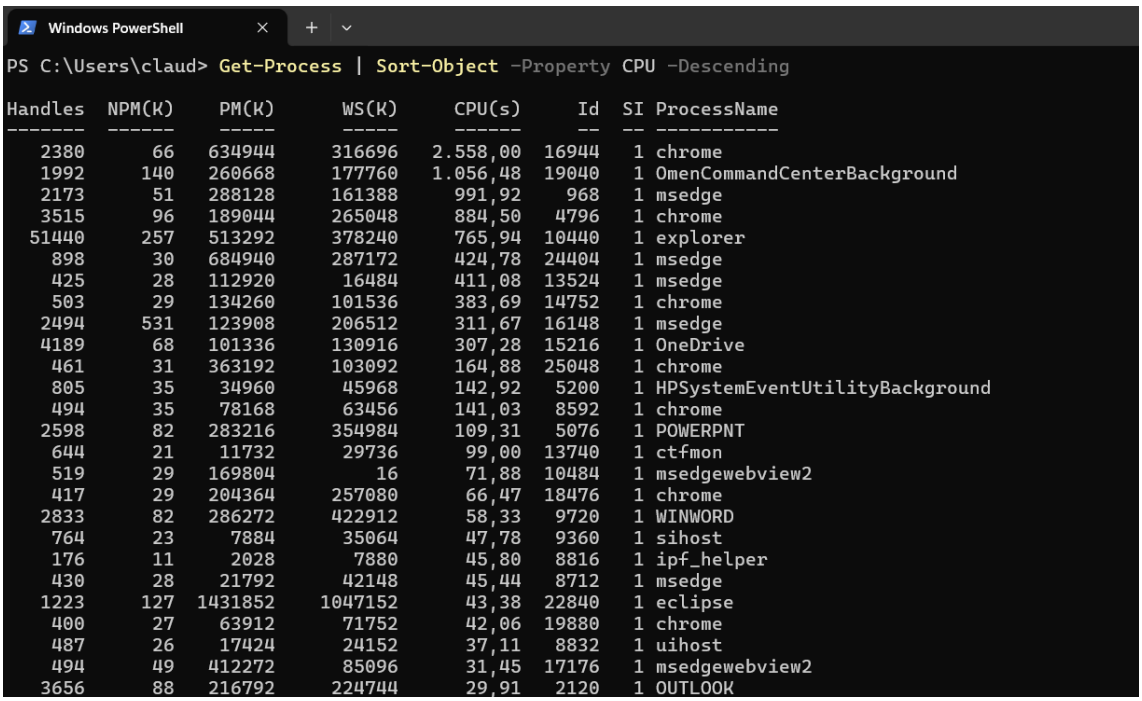
```
PS C:\Users\claud> Get-Process -Name notepad*

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    216      13     2924      13700       0,38    1904   1 Notepad
    722      34    57252     104980       0,31   18856   1 Notepad
```

A continuación, ejecutamos el comando con el ID correspondiente para detener uno de los procesos:

```
PS C:\Users\claud> Stop-Process -id 1904
PS C:\Users\claud> Get-Process -Name notepad*

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    722      34    57252     104980       0,31   18856   1 Notepad
```

6. Obtener Información de Uso de Recursos

Para mostrar el uso de CPU por los diferentes procesos en orden descendente se usa el siguiente comando Get-Process | Sort-Object -Property CPU -Descending



7. Consultar la lista de servicios del equipo

Para obtener la lista con todos los procesos que se están ejecutando en ese momento en el equipo, basta con utilizar el *comando* **Get-Service** sin argumentos:

Aunque, si la salida es demasiado larga y no cabe en la ventana, siempre puedes hacer una pausa procesándola con el comando more:

```
Windows PowerShell

PS C:\Users\claud> Get-Service | more

Status   Name               DisplayName
------   ----               -----------
Stopped  AarSvc_d7dbf       Agent Activation Runtime_d7dbf
Stopped  AJRouter           Servicio de enrutador de AllJoyn
Stopped  ALG                Servicio de puerta de enlace de niv...
Running  AppIDSvc           Identidad de aplicación
Running  Appinfo            Información de la aplicación
Stopped  AppReadiness       Preparación de aplicaciones
Running  AppXSvc            Servicio de implementación de AppX ...
Running  AudioEndpointBu... Compilador de extremo de audio de W...
Running  Audiosrv           Audio de Windows
Stopped  autotimesvc        Hora de la red de telefonía móvil
Stopped  AxInstSV           Instalador de ActiveX (AxInstSV)
Running  BcastDVRUserSer... Servicio de usuario de difusión y G...
Running  BDESVC             Servicio Cifrado de unidad BitLocker
Running  BFE                Motor de filtrado de base
Running  BITS               Servicio de transferencia inteligen...
Stopped  BluetoothUserSe... Servicio de soporte técnico de usua...
Running  BrokerInfrastru... Servicio de infraestructura de tare...
Stopped  BTAGService        Servicio de puerta de enlace de aud...
Stopped  BthAvctpSvc        Servicio AVCTP
Running  bthserv            Servicio de compatibilidad con Blue...
Running  camsvc             Servicio Administrador de funcional...
Stopped  CaptureService_... CaptureService_d7dbf
Running  cbdhsvc_d7dbf      Servicio de usuario del portapapele...
Running  CDPSvc             Servicio de plataforma de dispositi...
Running  CDPUserSvc_d7dbf   Servicio de usuario de plataforma d...
Stopped  CertPropSvc        Propagación de certificados
-- Más  --
```

Y cuando necesitamos sólo algunos de los procesos, siempre podemos establecer un filtro. Por ejemplo, por su nombre:

```
PS C:\Users\claud> Get-Service -name appinfo

Status   Name               DisplayName
------   ----               -----------
Running  appinfo            Información de la aplicación
```

8. Iniciar o detener un servicio

```powershell
Start-Service -Name "wuauserv"
Stop-Service -Name "wuauserv"
```