

Es frecuente que, la mayoría de los usuarios de Windows, utilicen exclusivamente su interfaz gráfica. Algunos, ni siquiera saben que hay otra alternativa... Y *Windows* no tiene una, sino dos: La *línea de comandos* clásica (*CMD.EXE*) y otra, más avanzada, que ofrece *Microsoft* de forma gratuita desde 2006 y que en la actualidad se incluye de forma predeterminada en todos sus sistemas operativos. Me refiero a *PowerShell*.

En el artículo de hoy demostraremos las posibilidades de este segundo método para administrar los procesos que se estén ejecutando en nuestro sistema, prescindiendo de los artificios de la interfaz gráfica.

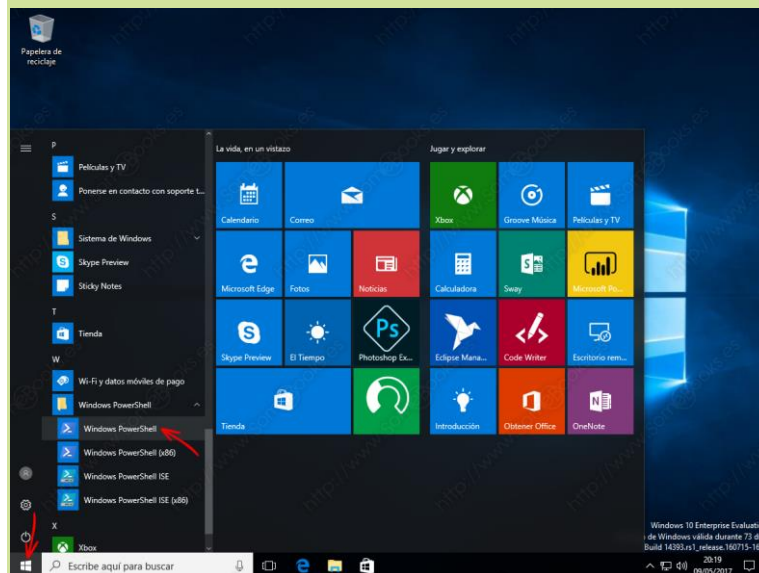
Principalmente, usaremos los siguientes *cmdlets* (que es como se llaman las órdenes en *PowerShell*):

- Get-Process
- Stop-Process
- Start-Process

Iniciar la ejecución de PowerShell

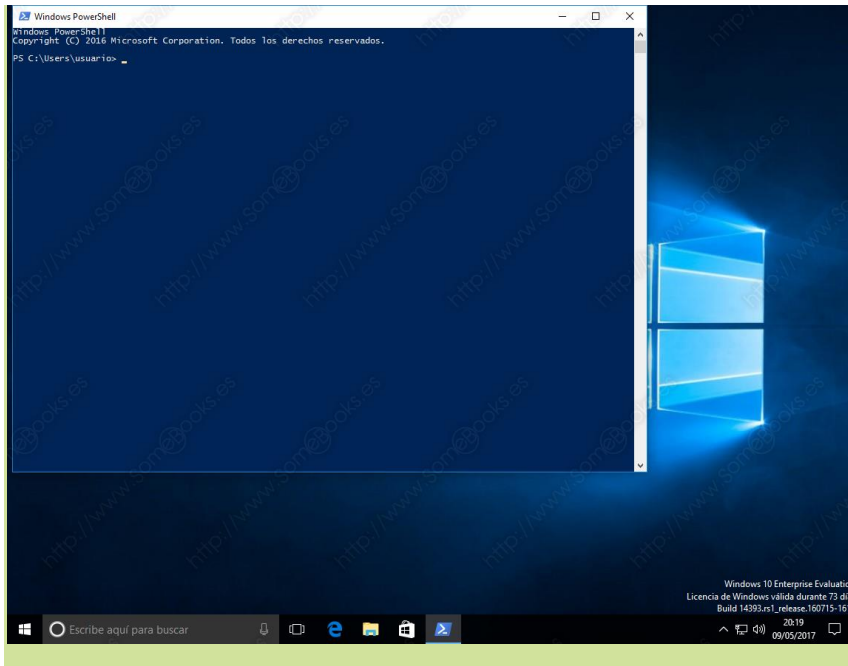
Para abrir una nueva ventana de *PowerShell*, comenzaremos por hacer clic sobre el botón *Inicio* y desplazarnos por el menú hasta la carpeta *Windows PowerShell*. También disponemos de una opción titulada *Windows PowerShell (x86)* para equipos con arquitectura de 32 bits.

Una vez allí, haremos clic sobre el elemento *Windows PowerShell*.



Al poco aparecerá una ventana, lista para comenzar a escribir nuestras órdenes. Como vemos en la imagen, se trata de una consola que se parece a la línea de comandos tradicional, pero con fondo de color azul

Administrar procesos con PowerShell



Consultar la lista de procesos del equipo

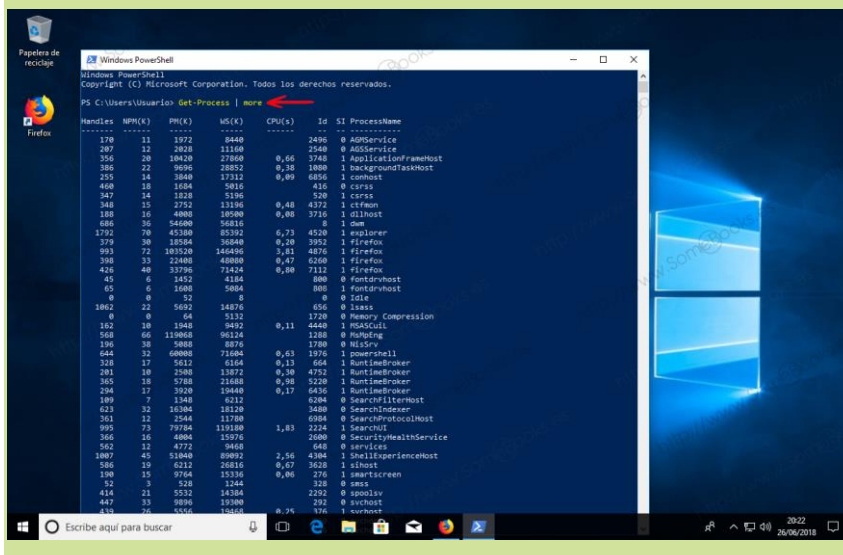
Para obtener la lista con todos los procesos que se están ejecutando en ese momento en el equipo, basta con utilizar el *cmdlet* **Get-Process** sin argumentos:

Get-Process

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

Get-Process | more

El resultado será como en la imagen

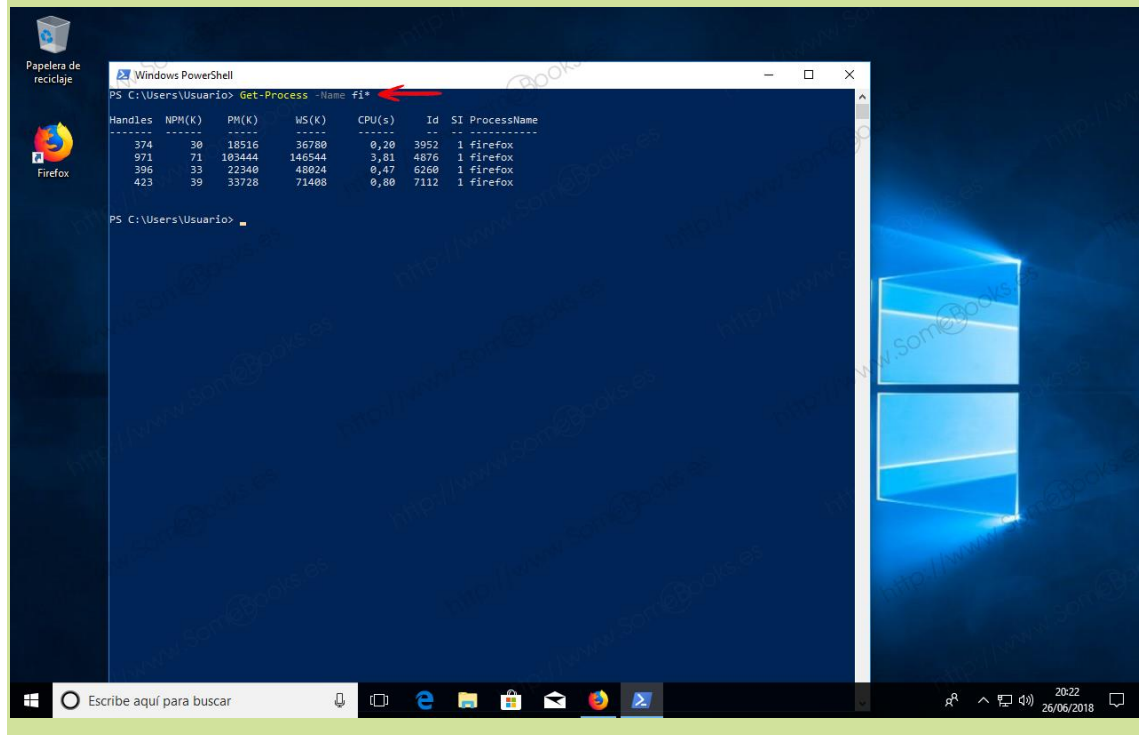


Administrar procesos con PowerShell

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

Get-Process -Name fi*

En este caso, obtenemos únicamente los procesos que comienzan con el texto *fi*



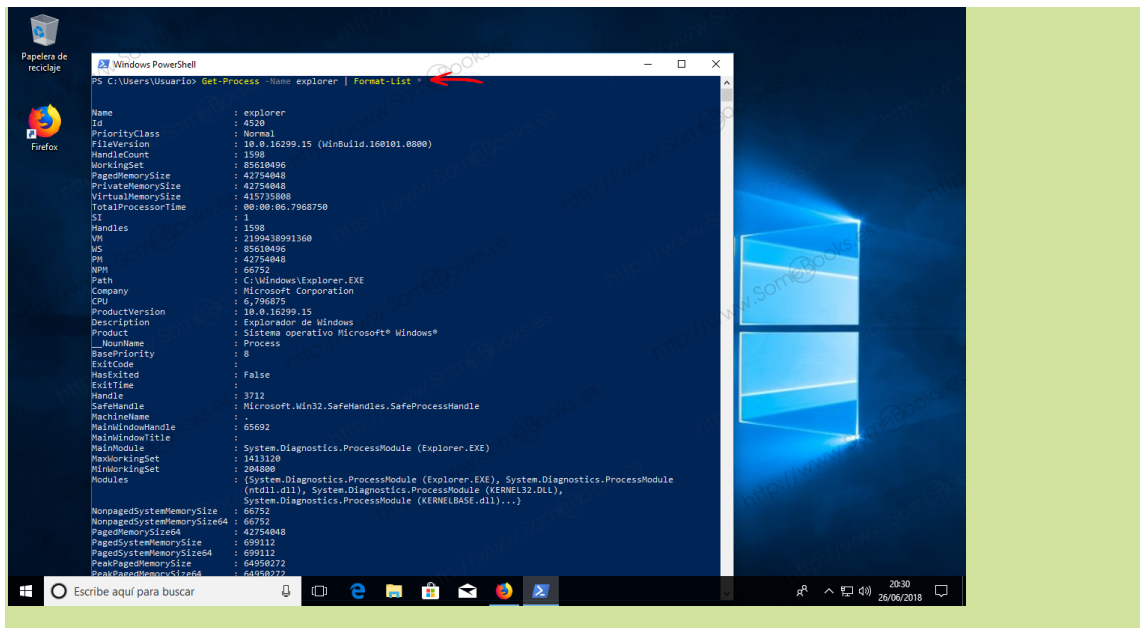
Por otro lado, si lo que necesitamos es una información detallada sobre un proceso, o un grupo de procesos, podemos enviar la salida de **Get-Process** a **Format-List**, que se encarga de formatear la salida de las propiedades de un objeto de modo que cada una aparezca en una nueva línea.

En nuestro caso, si queremos ver todas las propiedades del proceso *explorer*, escribiremos algo como esto:

Get-Process -Name explorer | Format-List *

Y el resultado será como en la imagen

Administrar procesos con PowerShell



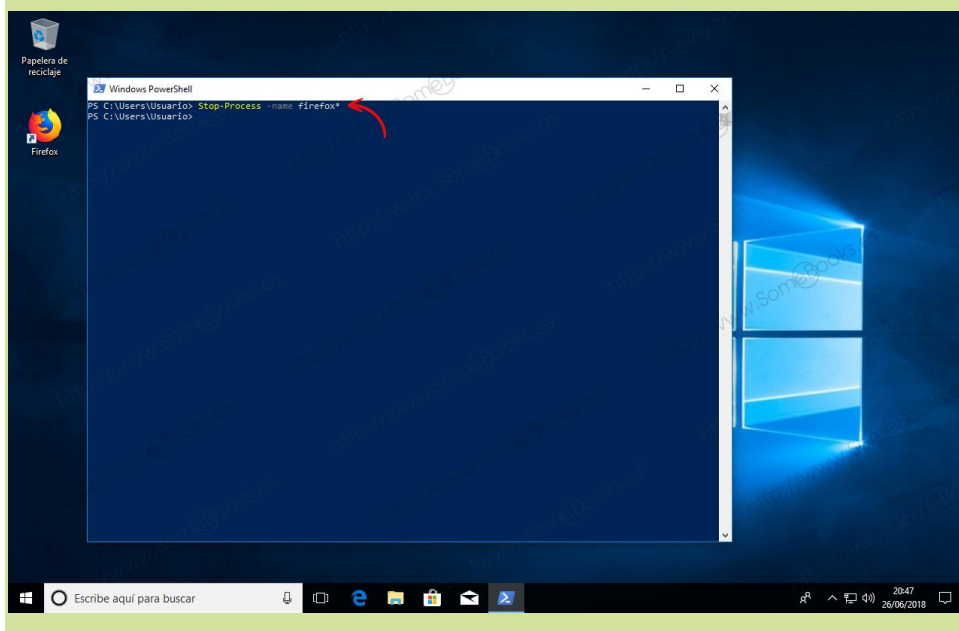
Detener un proceso

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:

`Stop-Process -name firefox*`

Esto detendrá todos los procesos cuyo nombre comience por el texto *firefox*.

Como puedes ver en la barra de tareas, el navegador ha dejado de ejecutarse

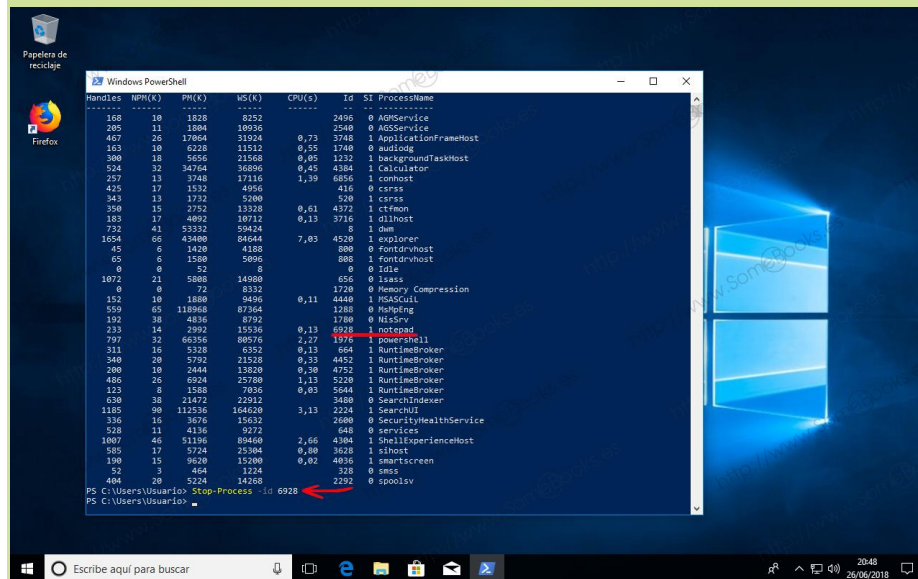


Administrar procesos con PowerShell

Si lo prefieres, podrías utilizar el argumento **-id** y utilizar el número de proceso. Por ejemplo, en el momento de tomar la imagen anterior, el *Bloc de notas* se está ejecutando con el *ID* 6928. Para detenerlo, podríamos escribir lo siguiente:

Stop-Process -id 6928

Como antes, el proceso dejará de ejecutarse de forma inmediata.

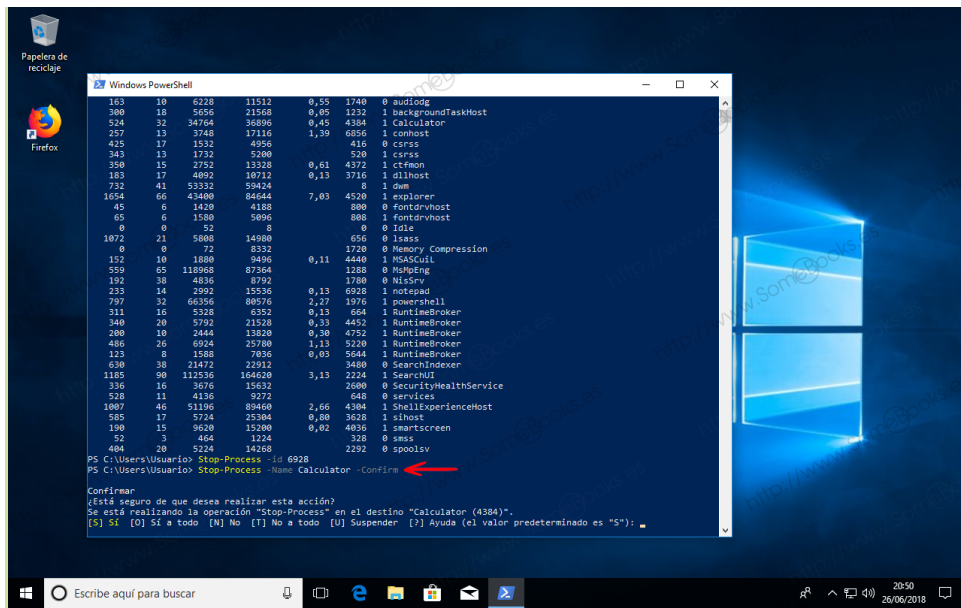


Incluso podemos hacer que el *cmdlet* nos pida confirmación antes de parar el proceso. Para ello, basta con utilizar el argumento **-Confirm**:

Stop-Process -name Calculator -Confirm

Antes de parar el proceso, *PowerShell* nos permite decidir qué hacer.

Administrar procesos con PowerShell



La ventaja de esta opción es que podemos utilizar un filtro que defina varios valores y, después, tomar la decisión de pararlos uno a uno. Por ejemplo:

Stop-Process -name f* -Confirm

Debemos tener en cuenta que algunos procesos no pueden detenerse y para otros necesitaremos permisos administrativos. En ambos casos, el *cmdlet* puede devolver un error al tratar de ejecutarlo.

Iniciar un nuevo proceso

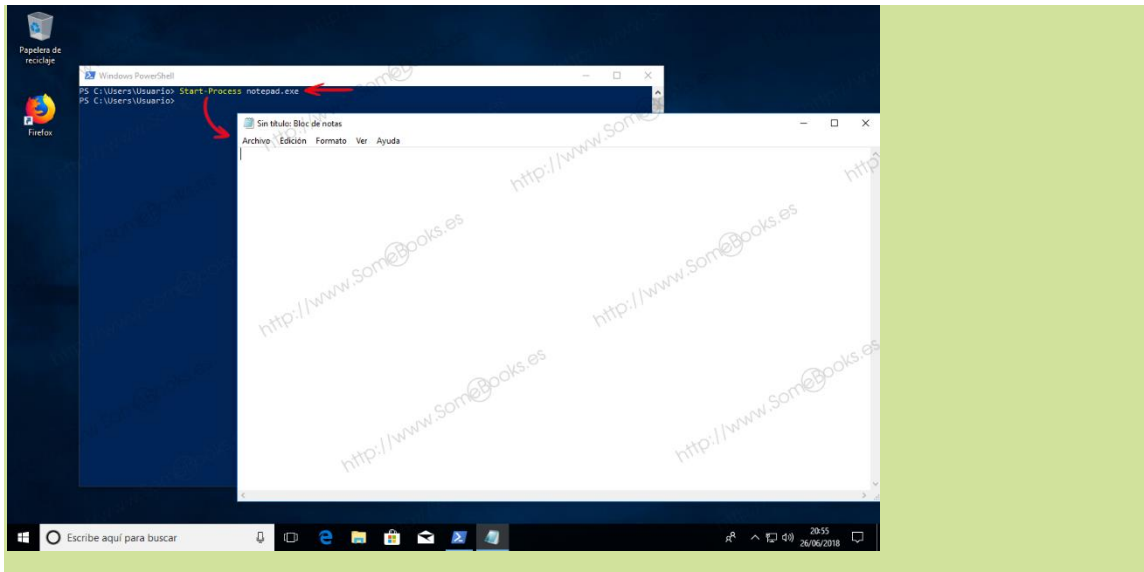
Para iniciar uno o mas procesos, podemos utilizar el *cmdlet* **Start-Process** incluyendo a continuación el nombre de un programa o de un script. Incluso podemos indicar un archivo que se encuentre asociado a un determinado programa del sistema.

De este modo, podríamos ejecutar el *Bloc de notas* escribiendo una orden como esta:

Start-Process notepad.exe

Como cabe esperar, lo siguiente que veremos será la ventana del programa en la pantalla.

Administrar procesos con PowerShell



Como es lógico, si el ejecutable no se encuentra en la ruta predeterminada del sistema, deberemos incluir también su ruta absoluta.

Fuente: <http://somebooks.es/administrar-procesos-powershell/>