

# Ejercicios VIII – Tema 4

## Ejercicios POWERSHELL

- Realiza los distintos apartados indicados en el ejercicio. Ten en cuenta que cada uno de ellos va ampliando la funcionalidad del anterior.
  - Obtener todos los servicios disponibles en nuestro equipo.
  - Filtrar los servicios obtenidos anteriormente (objetos), seleccionando únicamente aquéllos que estén en funcionamiento.
  - Ordenamos el resultado por el nombre completo (el que se muestra).
  - Seleccionamos únicamente las propiedades *Displayname*, *Name* y *ServicesDependedOn* de cada servicio, indicando además que sólo queremos los 10 primeros objetos del resultado.
  - Finalmente, almacenamos el resultado en un archivo de tipo «.csv».
- Indica brevemente qué realizan los siguientes cmdlets y pon un ejemplo de dos de ellos (excepto del primero).
  - Out-Default.
  - Out-File.
  - Out-GridView.
  - Out-Printer.
  - Tee-Object.
- Realiza un pequeño script en Powershell llamado ***ejercicio3.ps1*** que solicite al usuario tres números entre 0 y 10 y que finalmente se muestra tanto la media como la suma de los tres. Si no se introducen correctamente los números, que se muestre un mensaje de error indicando qué tipo de valor se ha introducido y se vuelva a solicitar.
- Partiendo del fichero adjunto ***servidores.csv***, muestra la siguiente salida por pantalla.

```

TypeName: System.Management.Automation.PSCustomObject

Name      MemberType Definition
-----
Equals    Method    bool Equals(System.Object obj)
GetHashCode Method    int GetHashCode()
GetType   Method    type GetType()
ToString  Method    string ToString()
Direccion IP NoteProperty System.String Direccion IP=192.168.0.11
Servidor   NoteProperty System.String Servidor=SRU01
Sistema Operativo NoteProperty System.String Sistema Operativo=Windows 2008 R2

```

5. Realiza en PowerShell los distintos apartados indicados a continuación en el ejercicio.
  - a) Lista todos los archivos en el directorio actual.
  - b) Cuenta el número de archivos en un directorio específico.
  - c) Muestra únicamente los archivos con la extensión «.gif» del directorio «Imágenes».
  - d) Lista los procesos en ejecución ordenados alfabéticamente por nombre.
  - e) Busca los archivos con un tamaño superior a 150 KB y redirígelos a un archivo de texto.
  - f) Muestra los 10 procesos con mayor uso de memoria.
  - g) Muestra la fecha y hora actuales.
  - h) Lista los servicios ordenados por su estado (en ejecución o detenidos).
  - i) Encuentra todos los archivos que contengan la cadena de texto «ejec».
  - j) Muestra únicamente los procesos que están utilizando más del 50% de la CPU.
  - k) Busca los archivos modificados en los últimos 7 días y almacénalos en un fichero de texto.
  - l) Muestra una lista de procesos agrupados por el usuario que los inició.
  - m) Obtén el tamaño total de todos los archivos en un directorio y muéstralo en formato legible para el usuario.
  - n) Busca todos los archivos «.log» en un directorio y sus subdirectorios y extrae las líneas que contengan la cadena de texto «error».
  - o) Muestra una lista de procesos con su identificador, nombre y uso de CPU y expórtala a un archivo de tipo CSV.
  - p) Muestra el espacio libre y usado en una unidad de disco específica y en formato legible para el usuario.
  - q) Muestra los servicios que se están ejecutando y expórtalos a un archivo XML.
  - r) Obtén información detallada sobre el hardware del sistema y guárdalo en un archivo de texto.
  - s) Muestra una lista de archivos ordenados por tamaño, con los 10 archivos más grandes al principio y los 10 más pequeños al final.
6. Escribe un pequeño *script* en PowerShell llamado ***ejercicio6.ps1*** que cuente el número de archivos en un directorio y sus subdirectorios agrupados por extensión.
7. Crea un *script* sencillo en PowerShell que cuente de forma descendente desde un número introducido por el usuario hasta cero de tres formas distintas. Llama al ejercicio ***ejercicio7.ps1***.

8. Crea un *script* en PowerShell de nombre ***ejercicio8.ps1*** que genere contraseñas aleatorias con una longitud específica y que cumplan con ciertos criterios de complejidad, como incluir al menos una letra mayúscula, una letra minúscula, un número y un carácter especial.
9. Escribe un *script* llamado ***ejercicio9.ps1*** en PowerShell que renombre todos los archivos en un directorio agregando la fecha de creación al principio del nombre del archivo.
10. Crea un *script* en PowerShell de nombre ***ejercicio10.ps1*** que busque archivos en un directorio y en todos sus subdirectorios cuyo nombre comience por la letra A y muestre la ruta completa de cada archivo encontrado.
11. Escribe un *script* en PowerShell que genere los primeros ***n*** números de la secuencia de Fibonacci, donde ***n*** es un número ingresado por el usuario. El nombre del archivo será ***ejercicio11.ps1***.
12. Crea un *script* en PowerShell con nombre ***ejercicio12.ps1*** que realice una copia de todos los archivos modificados en las últimas 24 horas de un directorio a otro.
13. Escribe un *script* en PowerShell que compruebe si hay conectividad de red con un servidor remoto, como «*google.com*», y muestre un mensaje indicando si la conexión se ha realizado de forma correcta o no. El comando para realizar la comprobación es *Test-Connection* y el nombre del archivo será ***ejercicio13.ps1***.