ASIR Semipresencial
Administración de Sistemas Informáticos en Red

TAREA QUINCENA 4

De los ejercicios planteados en esta tarea, copia y pega el comando utilizado, así como una captura de pantalla con el resultado obtenido.

Ejercicio 1
¿Cómo se podrían visualizar todos los cmdlets que utilizan el verbo GET?

```
Get-Command -Verb Get
```

Ejercicio 2
Escribe cómo usarías el cmdlet **Out-Host** para paginar la siguiente instrucción:

```
PS> Get-Service | Out-Host -paging
```

Ejercicio 3
¿Qué comando utilizarías para ver qué comandos están disponibles para administrar servicios?

```
Get-Help *-Service
```

Ejercicio 4
Busca todos los archivos del directorio de tu usuario que tengan la extensión ".odt" y que comiencen por la letra "a" o "c".
Nota: Usa para ello el comando "Get-ChildItem"

```
Get-Childitem –force C:\Users\llorens\ –name –recurse –include [ac]*.odt
```

Ejercicio 5
Suponga que desea buscar el archivo DLL del servicio de hora de Windows en la carpeta System32 y todo lo que recuerda del nombre del archivo DLL es que empiza con la letra "w" y que contiene la palabra "time". Escriba el comando adecuado.

```
Get-childitem –force c:\windows\system32\ -name –recurse –include w*time*.dll
```

O también

```
Get-childitem –force c:\windows\system32\w*time*.dll
```

Ejercicio 6
Realiza los siguientes pasos usando cmdlets, realizando una captura de pantalla de cada paso:
a) Crea un directorio llamado "NewDir" en el directorio de tu usuario.

```
new-item –path c:\users\llorens\NewDir –itemtype directory
```

b) Crea dentro de éste un fichero llamado "prueba.txt".

```
new-item –path c:\users\llorens\NewnDir\prueba.txt –itemtype file
```

c) Renombra el fichero “prueba.txt” como “ultimaPrueba.txt”

```
rename-item –path c:\users\llorens\NewDir\prueba.txt ultimaprueba.txt
```

d) Mueve el directorio “NewDir” desde el directorio de tu usuario, hasta la raíz de C.

```
move-item –Path c:\users\llorens\NewDir\ –destination c:\
```

e) Copia el directorio “C:\NewDir”, y todo su contenido, al directorio de tu usuario.

```
copy-item –path c:\NewDir\ -Destination c:\users\llorens\ -recurse
```

f) Elimina el directorio “C:\NewDir” pidiendo confirmación.

```
remove-item c:\NewDir\
```

g) Elimina el directorio “NewDir” del directorio de tu usuario, sin pedir confirmación.

```
remove-item C:\Users\llorens\NewDir –Recurse
```

Ejercicio 7
Sigue los siguientes pasos sobre las tareas de administración de Windows con PowerShell:
a) Abre el navegador Chrome.
Usa el cmdlet “Start-Process”

```
Start-process Chrome
```

b) Lista todos los procesos que empiecen por “c”.
Usa el cmdlet “get-help where-object -examples” para averiguar cómo hacerlo.

```
Get-Process –Name c*
```

c) Detén el proceso Chrome pidiendo confirmación.
Usa el cmdlet “get-help stop-process -examples” para averiguar cómo hacerlo.

```
Stop-process –name Chrome –Confirm –PassThru
```

d) Comprueba ahora que el navegador se ha detenido.

```
Get-Process –name Chrome
```

Ejercicio 8
Crear un script que muestre por pantalla "Di amigo y entra", permitiendo al usuario que escriba por teclado, y que mientras no reciba la palabra "amigo", siga mostrando la frase anterior, y pida al usuario escribir. Una vez se haya introducido la palabra correcta debe mostrar "La contraseña es correcta."
Usa para ello el bucle "do..while", y los cmdlet "Read-Host" y "Write-Host"

PSEUDOCÓDIGO

```
HACER
     <mostrar la frase 'Di amigo y entra' y guardar lo leido de teclado en una
variable>
MIENTRAS <la variable no sea 'amigo'>

<mostrar por pantalla 'La contrasenya es correcta'.
```

```
do
{
Write-Host "Di amigo y entra:"
$pass=Read-Host
}
While($pass -ne "amigo")
Write-Host "La contraseña es correcta!"
```

Ejercicio 9
Con la estructura del bucle "For", implementar la tabla de multiplicar de un número preguntado al usuario, y leído desde teclado.
La salida en pantalla debería quedar así:

```
Introduce un numero: 5
5 * 0 = 0
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

PSEUDOCÓDIGO

```
<mostramos por pantalla 'introduce un número' guardando lo leido de teclado en
var1>

PARA <desde var_I = 0 hasta que valga 10, vamos incrementando de 1 en 1>
    <realizamos la multiplicacion, guardando el resultado en var_R>
    <Mostramos por pantalla la multiplicacion de la tabla de multiplicar>
FIN_PARA
```

```
Write-Host "introduce un número:"
$Var_I=Read-Host
$array=0..10

foreach($item in $array)
{
$var_R=[int]$item*[int]$Var_I
Write-host "$num x $item = $var_R"
}
```

Ejercicio 10
Crea un script que lea un fichero CSV con la siguiente estructura,

```
Nombre,Departamento,Puesto
Carmen,Informatica,Analista
Manuel,Redes,Analista
Amparo,Redes Sociales,Jefa
Eloy,Robotica,Diseñador
Diego,Informatica,Programador
Alberto,Robotica,Electricista
Vicent,Robotica,Mantenimiento
```

El script, tras la lectura del fichero, deberá sacar por pantalla el número de usuarios que contiene el fichero. Además, también deberá sacar el contenido del fichero por pantalla en formato tabla, ordenado por Departamento.

El resultado debe salir de esta forma:

```
En el fichero CSV contiene 7 usuarios.

Nombre  Departamento   Puesto
------  ------------   ------
Diego   Informatica    Programador
Carmen  Informatica    Analista
Manuel  Redes          Analista
Amparo  Redes Sociales Jefa
Vicent  Robotica       Mantenimiento
Alberto Robotica       Electricista
Eloy    Robotica       Diseñador
```

Nota: Para importar los datos de un fichero CSV podéis usar el cmdlet "Import-Csv" y el cmdlet "sort" para guardarlo en una variable con la que trabajar.

PSEUDOCÓDIGO

```
<Leemos el contenido del fichero, ordenándolo con Sort-Object, guardándolo en
la variable var_datos>

<Inicializamos la varible que cuenta lineas>

POR_CADA var_usuario EN var_datos
    <incrementamos el contador de lineas>
FIN_POR_CADA

<Mostramos por pantalla el numero de lineas>

<ponemos directamente la variable var_datos con una tuberia al cmdlet Format-
Table>
```

```
$tabla=import-csv fichero.csv | Sort-Object departamento
$vuelta=0
$array= $tabla | Select-object Nombre
foreach($Item in $array)
{
$vuelta=([int]$vuelta+1)
}
write-host "El fichero contiene $vuelta lineas"
$tabla | Format-table
```