

# Exámen ASO PowerShell Carla García Parra

## PARTE 1: MENU

--FOTOS

--apartado 1 junto con un fichero salida.csv

```
function leerFichero(){
    #importamos el fichero y añadimos cada columna con su nombre
    $csv = import-csv -Delimiter ';' -Header CORREO, NOMBRESAM, NOMBRE, APELLIDOS, TITULO, DEPARTAMENTO, COMPANIA, JEFE, CLAVE "C:\users\administrador\documents\Examen2Tiv21.
}

    ForEach ($rows in $csv){
        $CORREO = ($rows.correo)
        $NOMBRESAM = ($rows.nombresam)
        $NOMBRE = $($rows.nombre)
        $APELLIDOS = $($rows.apellidos)
        $TITULO = $($rows.titulo)
        $DEPARTAMENTO = $($rows.departamento)
        $COMPANIA = $($rows.compañia)
        $JEFE = $($rows.jefe)
        $CLAVE = $($rows.clave)

        Write-output 'CORREO' $CORREO " " 'NOMBRESAM' $NOMBRESAM " " 'NOMBRE' $NOMBRE " " 'APELLIDOS' $APELLIDOS " " 'TITULO' $TITULO ";" 'DEPARTAMENTO' $DEPARTAMENTO ";" "
    }
}
```

```

Selección una opción del menú: 2
directiva : El término 'directiva' no se reconoce como nombre de un cmdlet, función, archivo de script o programa ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta es correcta e intente de nuevo.
En C:\Users\Administrador\WIN-VOLVIO80ED\Desktop\examen1.ps1: 89 Carácter: 3
+ ~~~~~
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (directiva:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

Presione Entrar para continuar.... |

```

```

Unidad      : C:
Etiqueta    :
Tamaño (GB) : 49,51
Libre (GB)  : 34,00
% Libre     : 68,68

```

```
Introduce una dirección a traducir: 8.8.8.8

Name                NameHost
----                -
8.8.8.8.in-addr.arpa dns.google

Presione Entrar para continuar...
```

Status	Name	RequiredServices	DependentServices
Stopped	applockerfltr	{AppIDSvc, AppID, FltMgr}	{}
Stopped	AudioSrv	{AudioEndpointBuilder, RpcSs}	{}
Stopped	WdNisSvc	{WdNisDrv}	{}
Stopped	WdNisDrv	{BFE}	{WdNisSvc}
Stopped	SharedAccess	{BFE}	{}
Stopped	RemoteAccess	{Http, RasMan, RpcSS, Bfe}	{}
Running	PolicyAgent	{Tcpip, bfe}	{}
Stopped	NcaSvc	{NSI, dnscache, iphlpsvc, BFE}	{}
Running	MpsSvc	{mpsdrv, bfe}	{}
Running	IKEEXT	{nsi, BFE}	{}
Stopped	embeddedmode	{BrokerInfrastructure}	{}
Stopped	applockerfltr	{AppIDSvc, AppID, FltMgr}	{}
Stopped	AppIDSvc	{RpcSs, CryptSvc, AppID}	{applockerfltr}
Stopped	XblGameSave	{XblAuthManager, UserManager}	{}

--SCRIPT

```
function mostrarMenu(){
    clear-host
    write-host "          MENÚ          "
    write-host "===== "
    write-host "1. Leer un fichero csv"
    write-host "2. Cambiar la directiva de ejecución de PowerShell"
    write-host "3. Mostrar el espacio libre/ocupado de cada disco del equipo"
    write-host "4. Traducir una dirección IP"
    Write-Host "5. Servicios dependientes de otros servicios"
    write-host "6. Salir"
```

}

```
function leerfichero(){
    #importamos el fichero y añadimos cada columna con su nombre
    $csv = import-csv -Delimiter : -Header CORREO, NOMBRESAM, NOMBRE, APELLIDOS,
    TITULO, DEPARTAMENTO, COMPAÑIA, JEFE, CLAVE
    "C:\users\administrador\documents\Examen2T1v21.csv"

    ForEach ($rows in $csv){
        $CORREO = ($rows.correo)
        $NOMBRESAM = ($rows.nombresam)
        $NOMBRE = $($rows.nombre)
        $APELLIDOS = $($rows.apellidos)
        $TITULO = $($rows.titulo)
        $DEPARTAMENTO = $($rows.departamento)
        $COMPAÑIA = $($rows.compañia)
        $JEFE = $($rows.jefe)
        $clave = $($rows.clave)

        Write-output 'CORREO' $CORREO " ," 'NOMBRESAM' $NOMBRESAM " ," 'NOMBRE'
        $NOMBRE " ," 'APELLIDOS' $APELLIDOS " ," 'TITULO' $TITULO " ; " 'DEPARTAMENTO'
        $DEPARTAMENTO " ; " 'COMPAÑIA' $COMPAÑIA " ; " 'JEFE' $JEFE " ; " 'CLAVE' $clave | out-file
        C:\Users\salida.csv -Append
    }
}
```

```
function directiva(){
    #creamos una variable directiva en la que se le pida al usuario que directiva quiere cambiar
    $directiva = Read-Host "Introduce la directiva que deseas cambiar"
    Set-ExecutionPolicy $directiva
    get-ExecutionPolicy
```

```
}
```

```
}
```

```
function espacioDisco(){
```

```
Get-CimInstance -Class Win32_LogicalDisk | Where-Object {$_.DriveType -eq 3} `
    #seleccionamos todos los apartados (name, volumenname, size (poderlo en GB) y el espacio
de disco tambien en GB) | Select-Object @{n="Unidad";e={$_.Name}},
    @{n="Etiqueta";e={$_.VolumeName}},
    @{n='Tamaño (GB)';e="{0:n2}" -f ($_.size/1gb)},
    @{n='Libre (GB)';e="{0:n2}" -f ($_.freespace/1gb)},
    @{n='% Libre';e="{0:n2}" -f ($_.freespace/$_.size*100)}
}
```

```
function servicios(){
```

```
    #buscamos en el cmdlet get-service y filtramos por las características que nos solicitan (stado,
nombre, servicios que requieren y servicios que dependen)
    #Lo ponemos en formato de tabla y para mostrarlo lo he metido en un fichero
    #con get-content muestro el contenido del fichero que se ha creado
    get-service -DependentServices | select Status, Name, RequiredServices, DependentServices
| Format-Table > c:\users\mostrar.txt
    get-content c:\users\mostrar.txt
    #como ya no nos hace falta después de mostrarlo, el fichero se borrará
    Remove-Item C:\Users\mostrar.txt
}
```

```
function traducir(){
```

```
    #pedimos al usuario que nos dé una dirección y esa información la guardamos en una
variable
    $direccion = Read-Host "Introduce una dirección a traducir"
    #utilizamos resolve-dnsname para que nos traduzca la ip que nosotros le hemos dado
    Resolve-DnsName $direccion | Format-Table name, NameHost
}
```

```
mostrarMenu
```

```
while (($opcion = Read-Host -Prompt "Selecciona una opción del menú") -ne "6"){
    switch ($opcion){
        1{
            #Leer un fichero csv
            leerfichero
            pause
            break
        }
    }
}
```

```
2{
#Cambiar la directiva de ejecución en PS
directiva
pause
break
}
```

```
3{
#Mostrar el espacio libre/ocupado
espacioDisco
pause
break
}
```

```
4{
#Traducción de las direcciones IP
traducir
pause
break
}
```

```
5{
#Servicios dependientes de otros servicios, usando get-service y select-object para filtrar
resultado
servicios
pause
break
}
```

```
6{
#Salir del menu
exit
}
```

```
}
}
```

mostrarMenu

--PARTE 2: FACTORIAL DE UN NÚMERO

--FOTOS

```
PS C:\Users\Administrador.WIN-V0LV1BK80EB> C:\Users\Administrador.WIN-V0LV1BK80EB\Desktop\factorial.ps1
introduce un numero para calcular su factorial: 5
5 = 120
PS C:\Users\Administrador.WIN-V0LV1BK80EB>
```

```
PS C:\Users\Administrador.WIN-VOLV1BK80EB> C:\Users\Administrador.WIN-VOLV1BK80EB\Desktop\factorial.ps1
introduce un numero para calcular su factorial: 300
El número introducido es demasiado grande

PS C:\Users\Administrador.WIN-VOLV1BK80EB> |
```

--SCRIPT

```
#pedimos al usuario un número para que se calcule el factorial
$numero = Read-Host "introduce un numero para calcular su factorial"
$factorial=1

#controlamos que el numero no sea mayor de 238
if ($numero -gt "238"){
for ($inicio=1;$inicio -le $numero; $inicio++)
{
#multiplicacion entre todos los números que lo anteceden
    $factorial= $factorial * $inicio

}
    write-host "$numero = $factorial"
#en el caso de que sea mayor escribiremos un error
} else {
    Write-Host "El número introducido es demasiado grande"
}
}
```