

LISTADO DE EJERCICIOS A REPARTIR ENTRE EL GRUPO

AYUDA

```
function listar-di{
    Write-Output "listar directorio"
    Write-Output "Introducir el directorio donde queremos Listar"
    $directorio= Read-Host
    if (Test-Path $directorio){
        dir $directorio
    }
    else{
        Write-Output "El directorio no existe"
    }
}

$Salir=0
while ($Salir -eq "0"){
    write-output "Menu"
    Write-Output "-----"
    Write-Output "1 Crear directorio"
    Write-Output "2 Borrar Directorio"
    Write-Output "3 listar directorio"
    Write-Output "4 Salir"
    Write-Output "Introduce una de las opciones: "
    $menu= Read-Host
    switch ($menu) {
        1 { crear-di }
        2 { borrar-di }
        3 { listar-di }
        4 { Write-Output "Gracias por usar este programa"
            $Salir=1
        }
    }
}
}
```

1. PROCESOS: MARCO

MENÚ DE PROCESOS

- Ver listado de procesos
- Ver información detallada de un proceso. Pide nombre
- Gestión de procesos
 - Parar un proceso (pregunta si quiere por nombre o por id y para dicho proceso)
 - Iniciar un proceso
- Salir

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada. El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

2. SERVICIOS Y CONTRASEÑAS:

SERVICIOS: JULIO

Crea un SCRIPT que muestre un menú al usuario con las siguientes operaciones:

- Sacar la lista de los servicios que estén en ejecución
- Sacar la lista de los servicios que estén detenidos ordenados por el id
- Arrancar el servicio Telefonía(TapiSrv)
- Parar el servicio Telefonía

- Ver los servicios que dependen de otros indicando nombre e id

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

CONTRASEÑAS: JOSÉ MARÍA SÁNCHEZ

Desarrolla un script de powershell que genere contraseñas aleatoriamente.

El número de contraseñas y el número de caracteres se debe indicar por parámetro.

Para ello crea un array llamado caracteres inicializado con todas las letras (de la a la z) y números (del 0 al 9). Utiliza un bucle para generar todas las contraseñas y otro bucle para generar aleatoriamente cada carácter. Usa get-random 28 para obtener una posición del array.

```
$caracteres="a","b","c","d","e","f","g","h","i","j","k","l","m",
"n","o","p","q","r","0","1","2","3","4","5","6","7","8","9",
```

```
$nombre=""
for($j=0;$j -lt $Args[1];$j++) {
    $c= Get-Random 28
    $nombre=$nombre+$caracteres[$c]
}
```

3. RED: VÍCTOR

Menú de gestión de redes

1. Mostrar información simple de todos los adaptadores de red
2. Mostrar información detallada de un adaptador (pide nombre de adaptador)
3. Activar/desactivar adaptador (pide nombre)
4. Configuración IP con las siguientes opciones:
 - A) Ver configuración actual
 - C) Comprobar conectividad a Internet
 - E) Cambiar con Conf. estática (pide IP, máscara, pta. enlace y DNS)
 - D) Cambiar con Conf. dinámica (Habilita las opciones oportunas)
 - V) Volver

5. Salir

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

4. PUERTOS Y CONECTIVIDAD A VARIAS IPS: DANIEL

Crea un script que contenga un menú con las siguientes opciones:

- **Puertos:** defina un array con los puertos que queremos que estén en abiertos. Visualiza los puertos están en estado listening y cuáles no.
- **Conectividad:** Pide el último número de una dirección ip inicial y de dirección ip final y comprueba si dicha IP está o no conectada . Ejemplo (para las IPS que hay en clase). (Ayuda: utiliza el parámetro -quiet con la orden de comprobación de conexión correspondiente)
- salir.

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

5. USUARIOS: BRIAYAN

Menú de usuarios

1. Listar usuarios
2. Crear usuarios (pide usuario y contraseña)
3. Elimina usuarios (pide usuario). Si no existe, debe indicarlo.
4. Modifica usuarios (pide usuario y nuevo nombre). Si no existe, debe indicarlo.
5. Salir

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir. Cada operación debe realizarse en una función independiente.

6. GRUPOS: ERICK

Menú de usuarios

1. Listar grupos.
2. Crear un grupo (pide nombre)
3. Añade usuarios al grupo.
4. Elimina usuario del grupo (pide usuario y grupo). Si no existe, debe indicarlo.
5. Elimina grupo (pide nombre). Si no existe, debe indicarlo.
6. Muestra usuarios del grupo (pide nombre). Si no existe, debe indicarlo.
7. Salir

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

7. DIRECTORIOS : ÁNGEL

Crea un SCRIPT que muestre un menú al usuario con las siguientes operaciones:

- Mostrar directorio, ordenando los elementos por tamaño(longitud) y mostrando tamaño y nombre.
- Crear directorio.
- Eliminar directorio.
- Cambiar a un directorio, ver su contenido y volver al directorio anterior.
- Salir.

Una vez elegida una opción correcta el programa debe solicitar un nombre de directorio y hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

8. ARCHIVOS: JESÚS

Crea un SCRIPT que muestre un menú al usuario con las siguientes operaciones:

- Crear archivo.
- Mostrar contenido de archivo. Si no existe debe mostrar mensaje.
- Añadir contenido al archivo. Si no existe debe mostrar mensaje.
- Eliminar archivo. Si no existe debe mostrar mensaje.
- Crea un enlace simbólico, otro duro y otro union.
- Salir.

Una vez elegida una opción correcta el programa debe solicitar un nombre de archivo y hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

9. INFORMACIÓN DEL SISTEMA: KARIM

Crea un SCRIPT que muestre un menú al usuario con las siguientes operaciones:

- Nombre del equipo.
- Sistema operativo.
- Nombre del procesador.
- Rendimiento.
- Todos los eventos.
- Todos los proveedores.
- Todas las unidades.
- Espacio libre del disco.
- Software instalado.
- Salir.

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

10. GESTIÓN DE DISCOS BÁSICOS: ALBA

Crea un SCRIPT que muestre un menú al usuario con las siguientes operaciones:

- Ver el tipo de tabla de particiones.
- Mostrar información sobre los discos.
- Crear una partición sobre uno de los discos con tamaño de $\frac{1}{4}$ del tamaño total.
- Dar formato a la partición anterior.
- Listar particiones del sistema.
- Listar el nombre de las unidades de las particiones.
- Chequee un disco.
- Cree una cuota de disco para un usuario con un valor de 1Gb y límite 0,95Gb

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

11. COPIAS DE SEGURIDAD 1: JOSÉ MARÍA VIÚDEZ

Crea un SCRIPT que muestre un menú al usuario con las siguientes operaciones:

ayuda comprimir y descomprimir una carpeta: <https://kirukiru.es/como-comprimir-y-descomprimir-archivos-con-powershell/>

1. Crear una copia de un directorio

1. La copia se guardará en un fichero comprimido llamado fecha-nombre_directorio. El nombre del directorio se pasará a mayúsculas con nombre_directorio.Toupper()
2. La copia debe almacenarse en un directorio llamado backup y localizado en el directorio raíz C:

2. Restaurar la copia del directorio copiado anteriormente(descomprimiendo el fichero en la ubicación del directorio original)

3. Crear una tarea programada para que realice la copia del directorio de un usuario todos las semanas a las 2:00.

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.

12. COPIAS DE SEGURIDAD 2: PEDRO

Crea un scripts para crear copia de seguridad de un directorio del equipo y posteriormente restaurarla. Para ello consulta este link

<https://docs.microsoft.com/es-es/windows-server/identity/ad-ds/manage/component-updates/ca-backup-and-restore-windows-powershell-cmdlets>

Debe solicitar la operación mediante un menú. Debe contener una opción para salir del menú.

Una vez elegida una opción correcta el programa debe hacer la operación seleccionada.

El menú debe repetirse mientras no se pulse la opción de salir.

Cada operación debe realizarse en una función independiente.