

Examen práctico PowerShell. Enunciados y soluciones

p>Este examen práctico está diseñado para evaluar tus conocimientos y habilidades en el uso de PowerShell. A continuación, se presentan varios enunciados con sus respectivas soluciones.

Enunciado 1

Descripción: Crea un script de PowerShell que muestre una lista de todos los procesos en ejecución en el sistema, ordenados por uso de memoria en orden descendente.

Solución:

```
# Obtener todos los procesos y ordenarlos por uso de memoria  
en orden descendente  
Get-Process | Sort-Object -Property WS -Descending
```

Enunciado 2

Descripción: Escribe un script de PowerShell que acepte un nombre de archivo como parámetro y muestre el número de líneas, palabras y caracteres en el archivo.

Solución:

```
param (  
    [string]$filePath  
)  
  
# Verificar si el archivo existe  
if (Test-Path $filePath) {
```

```
# Leer el contenido del archivo
$content = Get-Content $filePath

# Contar el número de líneas
$lineCount = $content.Count

# Contar el número de palabras
$wordCount = ($content -join " " -split "\s+").Count

# Contar el número de caracteres
$charCount = ($content -join "").Length

# Mostrar los resultados
Write-Output "Líneas: $lineCount"
Write-Output "Palabras: $wordCount"
Write-Output "Caracteres: $charCount"
} else {
    Write-Output "El archivo no existe."
}
```

Enunciado 3

Descripción: Escribe un script de PowerShell que busque archivos en un directorio específico (y sus subdirectorios) que no hayan sido modificados en los últimos 30 días y los mueva a una carpeta de respaldo.

Solución:

```
param (
    [string]$sourceDir,
    [string]$backupDir
)

# Verificar si los directorios existen
if (-Not (Test-Path $sourceDir)) {
    Write-Output "El directorio de origen no existe."
    exit
}
```

```

if (-Not (Test-Path $backupDir)) {
    # Crear el directorio de respaldo si no existe
    New-Item -ItemType Directory -Path $backupDir
}

# Obtener la fecha límite (hace 30 días)
$limitDate = (Get-Date).AddDays(-30)

# Buscar archivos no modificados en los últimos 30 días
$files = Get-ChildItem -Path $sourceDir -Recurse | Where-Object { $_.LastWriteTime -lt $limitDate }

# Mover los archivos a la carpeta de respaldo
foreach ($file in $files) {
    $destPath = Join-Path $backupDir $file.Name
    Move-Item -Path $file.FullName -Destination $destPath
}

```

Enunciado 4

Descripción: Crea un script de PowerShell que verifique si un servicio específico está en ejecución, y si no lo está, intente iniciarlo. Si el servicio no existe, debe mostrar un mensaje apropiado.

Solución:

```

param (
    [string]$serviceName
)

# Obtener el estado del servicio
$service = Get-Service -Name $serviceName -ErrorAction SilentlyContinue

if ($null -eq $service) {
    Write-Output "El servicio '$serviceName' no existe."
} elseif ($service.Status -ne 'Running') {
    # Intentar iniciar el servicio

```

```
    Start-Service -Name $serviceName
    Write-Output "El servicio '$serviceName' se ha iniciado."
} else {
    Write-Output "El servicio '$serviceName' ya está en
ejecución."
}
```

Enunciado 5

Descripción: Escribe un script de PowerShell que genere un informe HTML con una lista de todos los usuarios locales del sistema, incluyendo su nombre completo y la fecha del último inicio de sesión.

Solución:

```
# Obtener todos los usuarios locales
$users = Get-LocalUser

# Crear una lista para almacenar la información del informe
$userInfo = @()

foreach ($user in $users) {
    $userInfo += [PSCustomObject]@{
        'Nombre de Usuario' = $user.Name
        'Nombre Completo'   = $user.FullName
        'Último Inicio de Sesión' = $user.LastLogon
    }
}

# Convertir la lista a HTML
$userInfo | ConvertTo-Html -Property 'Nombre de Usuario',
'Nombre Completo', 'Último Inicio de Sesión' -Title 'Informe
de Usuarios Locales' -PreContent "
```

Informe de Usuarios Locales

" | Out-File 'InformeUsuarios.html'

Write-Output "Informe generado: InformeUsuarios.html"