

Ejemplo real: usando PSScriptAnalyzer para mejorar un script de PowerShell

□ Ejemplo real: usando PSScriptAnalyzer para mejorar un script de PowerShell

PSScriptAnalyzer es una herramienta oficial de Microsoft que analiza scripts de PowerShell en busca de errores, malas prácticas y problemas de estilo o rendimiento. Es una forma rápida y eficaz de garantizar la calidad de tu código.

□ Script original con problemas

```
# backup.ps1
param($source, $destination)

if (!(Test-Path $destination)) {
    New-Item -ItemType Directory -Path $destination
}

Copy-Item $source $destination -Recurse
Write-Host "Copia completada de $source a $destination"
```

A simple vista, parece correcto, pero veamos qué ocurre al analizarlo.

❑ Paso 1: Ejecutar el analizador

```
Install-Module PSScriptAnalyzer -Scope CurrentUser
Invoke-ScriptAnalyzer -Path .\backup.ps1
```

Salida del análisis:

RuleName	Message
-----	-----
PSAvoidUsingWriteHost	Avoid using Write-Host, use Write-Output instead.
PSUseApprovedVerbs	The cmdlet 'Copy-Item' uses an unapproved verb.
PSUseConsistentWhitespace	Use consistent indentation and spacing.

El analizador detecta malas prácticas y sugiere mejoras para aumentar la calidad y coherencia del script.

❑❑ Paso 2: Corregimos el script según las sugerencias

```
# backup_fixed.ps1
[CmdletBinding()]
param(
    [Parameter(Mandatory)]
    [string]$Source,

    [Parameter(Mandatory)]
    [string]$Destination
)

if (-not (Test-Path -Path $Destination)) {
    New-Item -ItemType Directory -Path $Destination | Out-Null
}
```

```
Copy-Item -Path $Source -Destination $Destination -Recurse  
Write-Output "Copia completada de $Source a $Destination"
```

En esta versión mejorada:

- Se usa Write-Output en lugar de Write-Host.
 - Se declaran correctamente los parámetros con [CmdletBinding()].
 - Se mejora la legibilidad, la validación y la coherencia del código.
-

⚙️ Paso 3: Verificamos de nuevo

```
Invoke-ScriptAnalyzer -Path .\backup_fixed.ps1
```

Salida:

No issues found.

✅ ¡Perfecto! El script ahora cumple con todas las reglas recomendadas por PSScriptAnalyzer.

🎁 Bonus: Analizar todos tus scripts automáticamente

Puedes analizar en lote todos los scripts .ps1 de un directorio:

```
Get-ChildItem -Path .\Scripts -Filter *.ps1 -Recurse |  
    ForEach-Object { Invoke-ScriptAnalyzer -Path $_.FullName }
```

Así garantizas la calidad de todo tu código antes de publicarlo o desplegarlo.

□ Conclusión

Con **PSScriptAnalyzer** puedes detectar problemas de estilo, rendimiento y seguridad en tus scripts de PowerShell, mejorar su calidad y automatizar su análisis dentro de tus pipelines DevOps. Es una herramienta imprescindible para cualquier desarrollador o administrador que busque escribir código limpio y profesional.

*#PowerShell #DevOps #Scripting #Automatización
#PSScriptAnalyzer #BuenasPrácticas*