

Idempotencia

La idempotencia se refiere a una propiedad de algunas operaciones que indica que el resultado es el mismo, independientemente del número de veces que se ejecute la operación o de su estado anterior.

En otras palabras, si una operación es idempotente, ejecutarla una vez tendrá el mismo resultado que ejecutarla varias veces seguidas, o que ejecutarla después de haberla ejecutado previamente. Esto permite realizar operaciones de manera segura, sin preocuparse por duplicados o efectos secundarios no deseados.

En el contexto de la automatización de tareas y la programación, la idempotencia es un concepto importante para asegurar que las operaciones se realicen de manera predecible y confiable, y para evitar errores y redundancias en el código.

Ejemplo de cmdlets Idempotente

El cmdlet "New-LocalUser" de PowerShell es idempotente, ya que cada vez que se ejecuta, verifica si el usuario existe y devuelve un error si es el caso, no generando usuarios duplicados ni efectos no deseados.

Así, este código:

```
$user = "Invitado"

$localuser = Get-LocalUser -Name $user -ErrorAction SilentlyContinue

if ($null -eq $localuser){

    Write-Host "El usuario $user no existe: se creará"

    Try {

        New-LocalUser -Name $localuser -Password (ConvertTo-SecureString "CLASE" -AsPlainText -Force)

        Write-Host "Usuario creado con éxito"

    }

    Catch {

        #Do this if a terminating exception happens#

        Write-Host "error al crear usuario: $_.Message"

    }

}

else {

    Write-Host "El usuario $user ya existe: no se creará"

}
```

Es equivalente a crear el usuario directamente

```
New-LocalUser -Name "Invitado" -Password (ConvertTo-SecureString "CLASE" -AsPlainText -Force)
```

Ejemplo de cmdlets No Idempotente

```
New-SmbShare -Name MyShare -Path C:\Demo\Temp -FullAccess Alice -ReadAccess Bob
```

Este script es simple, entendible y sencillo. Sin embargo, si ponemos este script en el entorno de producción, tendremos varios problemas. ¿Qué sucede si el script se ejecuta dos veces seguidas? ¿Qué pasa si Antonio tenía anteriormente acceso completo al recurso compartido?

Una versión "real" del script se verá un poco más similar a lo siguiente.

But actually creating a share in an idempotent way would be

```
$shareExists = $false
```

```
$smbShare = Get-SmbShare -Name $Name -ErrorAction SilentlyContinue
```

```
if($smbShare -ne $null)
```

```
{
```

```
    Write-Verbose -Message "Share with name $Name exists"
```

```
    $shareExists = $true
```

```
}
```

```
if ($shareExists -eq $false)
```

```
{
```

```
    Write-Verbose "Creating share $Name to ensure it is Present"
```

```
    New-SmbShare @PSBoundParameters
```

```
}
```

```
else
```

```
{
```

```
    # Need to call either Set-SmbShare or *ShareAccess cmdlets
```

```
    if ($PSBoundParameters.ContainsKey("ChangeAccess"))
```

```
{  
    #...etc, etc, etc  
}  
}
```