1. Termina el script matar\_proceso.ps1 completando los cmdlets que están en negrita, así como los comentarios.

#matar\_proceso.ps1

#Autor:

#Borramos pantalla

clear-host

#**Creamos la variable $proceso**

$proceso=read-host "`nIntroduce el nombre de un proceso"

#Sacamos por pantalla un mensaje al usuario

"`n`nBuscando si el proceso $proceso esta en ejecución....`n`n"

#Busqueda del proceso entre los que hay en el sistema .

#Guardamos el resultado de la búsqueda en la variable $esta

# que es un objeto de tipo proceso

#**La variable $\_ contiene ...** **el objeto de tipo proceso que pasa por el “pipe”**

**$esta=get-process|where-object {$proceso.name <completa la comparación>}**

**#Con este if comprobamos … el objeto**

if ($esta -eq $null)

**{write-host -foreground red "`tXXXXXXXXXXXX"}**

else {

#

foreach ($p in $esta)

{

write-host -foreground green "Matando el proceso" $p.name.toupper(), "Id:", $p.Id

#Matamos el proceso utilizando el método kill del objeto proceso

#La sintaxis para utilizar un método es: objeto.metodo()

**XXXXX**

}

}

1. Explica lo que realiza el siguiente script:

$ficheros = Get-ChildItem . -Recurse

$total = 0

For ($i = 1; $i -le $ficheros.Count-1; $i++)

{

    Write-Progress -Activity "Calculando tamaño total..." -status $ficheros[$i].Name -PercentComplete ($i / $ficheros.Count \* 100)

    $total += $ficheros[$i].Length

    Start-Sleep -Milliseconds 50

}

Write-Host "Tamaño total: $($total / 1MB)"

**Cuenta en porcentaje los ficheros que hay para determinar el tamaño total de todos.**

1. Realiza un script llamado servicios.ps1, que lista todos los servicios del sistema, indicando su nombre y estado, en color amarillo.