



CFGs ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

IMPLANTACIÓN DE SISTEMAS OPERATIVOS





6.-Estructuras repetitivas.

Índice

- 1.- Estructuras de control.
- 2.- Estructuras repetitivas.
- 3.- Menú repetitivo.



1.- Estructuras de control

Como cualquier lenguaje de script, PowerShell también tiene **estructuras de control**, que **permiten modificar el flujo de ejecución de las instrucciones** de un programa.

PowerShell ISE incorpora una funcionalidad muy útil, los **snippets (CTRL+J)**. Son **estructuras de código listas para ser utilizadas**.



2.- Estructuras repetitivas

A veces necesitamos **repetir un bloque de instrucciones**.

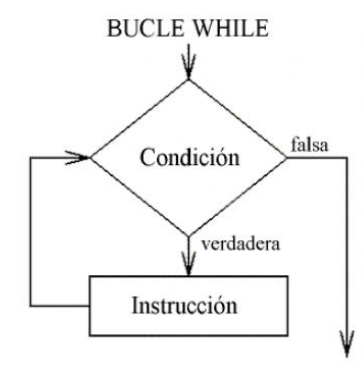
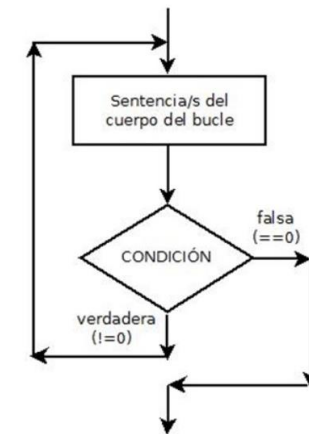
Estructuras posibles:

do while: Repite un bloque de instrucciones mientras la condición sea cierta. La condición se evalúa **al final del bloque**, lo que significa que al menos el bloque se ejecuta una vez.

while: Repite un bloque de instrucciones mientras la condición sea cierta. La condición se evalúa **al principio del bloque**.

do until: Repite un bloque de instrucciones hasta que la condición sea falsa. La condición se evalúa **al final del bloque**, lo que significa que al menos el bloque se ejecuta una vez.

Bucle *do while* (repetir mientras):





2.- Estructuras repetitivas

1.-Bucle do while

```
do  
{  
    Bloque  
}  
while (condición)
```

Ejemplo: DoWhile.ps1

```
1  #Ejemplo do while  
2  clear  
3  $i = 1  
4  do {  
5      Write-Host "Hola $i"  
6      $i++  
7  }While ($i -lt 10)
```

```
Hola 1  
Hola 2  
Hola 3  
Hola 4  
Hola 5  
Hola 6  
Hola 7  
Hola 8  
Hola 9  
PS C:\Users\Windows>
```



2.- Estructuras repetitivas

2.-Bucle while

```
while (condición)
{
    Bloque
}
```

Ejemplo: while.ps1

```
2 clear
3 $i = 7
4 while ($i -lt 10){
5     Write-Host "Hola $i"
6     $i++
7 }
```

```
<
Hola 7
Hola 8
Hola 9
PS C:\Users\Windows>
```



2.- Estructuras repetitivas

3.-Bucle do until

```
do
{
    Bloque
}
until (condición)
```

Ejemplo: doUntil.ps1

```
2  Clear-Host
3  $i = 1
4  do {
5      Write-Host " Hola $i"
6      $i++
7  }until ($i -lt 10)
8
```

```
Hola 1
PS C:\Users\Windows> |
```



2.- Estructuras repetitivas

4.-Estructuras posibles:

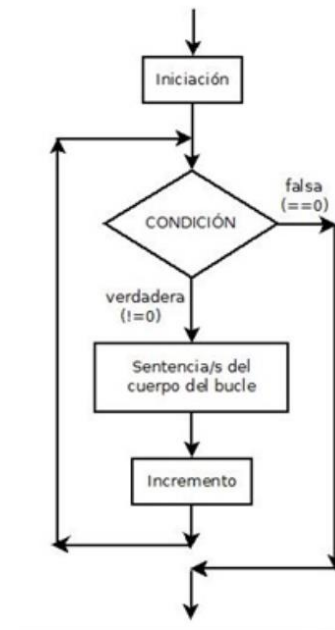
for: Repite un bloque de instrucciones un número determinado de veces.

A priori sabemos el número de veces que se va a ejecutar el bloque de instrucciones.

foreach: Repite un bloque de instrucciones para cada elemento de una lista.

Va ser el ideal para **leer datos de un fichero**.

Bucle *for* (desde):





2.- Estructuras repetitivas

4.1.- Bucle for

```
for (valor inicial; condición,  
incremento)  
{  
    Bloque  
}
```

Ejemplo: for.ps1

```
2 Clear-Host  
3 $i = 1  
4 for ($i=0;$i -lt 5;$i++) {  
5     Write-Host "Hola $i"  
6 }  
7
```

```
Hola 0  
Hola 1  
Hola 2  
Hola 3  
Hola 4  
PS C:\Users\windows>
```



2.- Estructuras repetitivas

4.2.-Bucle foreach

```
foreach ($i in colección de datos)
{
```

Nota:

```
}
```

\$i: es la variable que captura cada uno de los elementos de la colección.

La colección de datos se puede dar de muchas formas:

(\$i in “Manolo”, “María Dolores”, “Mariló”) → Los elementos separados por ,

(\$i in ls d:\) → Cada línea se considera un elemento.

(\$i in cat d:\usuarios.txt) → A través de un fichero. Cada línea de ese fichero se considera un elemento.

Nota: por defecto, el espacio, se considera un separador de elementos.




2.- Estructuras repetitivas

4.2.- Bucle foreach

Ejemplo: foreach.ps1

Los datos están guardados en el fichero datos.txt

```
1 #Prueba de foreach
2 foreach ($i in Get-Content F:\datos.txt){
3     Write-Host "Hola $i"
4 }
```

 datos: Bloc de notas

Archivo Edición Formato

Marilo|
María Dolores
Manolo

```
PS C:\Windows\system32> F:\prueba.ps1
Hola Marilo
Hola María Dolores
Hola Manolo
```



2.- Estructuras repetitivas

4.2.- Bucle foreach

Ejemplo: foreachCsv.ps1

Ahora los datos tienen varios campos y están separados por comas.

Es lo que llamamos un **archivo csv**.

F:\datos.csv

En la primera fila se pone el nombre de los campos, y en las siguientes filas, los datos separados por coma.

```
Ejemplo7b.ps1 X
1 #Ejemplo7: Estructuras repetitivas: foreach
2 $usuarios = Import-Csv -Path d:\datos.csv
3 foreach ($i in $usuarios) {
4     Write-Host "Hola" $i.nombre
5     Write-Host "Vives" en $i.localidad
6 }
```

```
PS C:\Windows\system32> F:\prueba.ps1
Hola Marilo
Vives en Sevilla
Hola Maria Dolores
Vives en Cádiz
Hola Manolo
Vives en Huelva
```

datos.csv: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
nombre,localidad
Marilo,Sevilla
Maria Dolores,Cádiz
Manolo,Huelva
```



2.- Estructuras repetitivas

5.- break y continue

Break → Se sale del bucle y se ejecuta la siguiente instrucción .

```
Ejemplo7.ps1 X
1  #Ejemplo7: Estructuras repetitivas: break
2  Clear-Host
3  $i = 1
4  for ($i=0;$i -lt 10;$i++) {
5  if ($i -eq 5){ break
6  }
7  Write-Host " Hola $i"
8  }
9  Write-Host "Adios"
10
```

```
<
Hola 0
Hola 1
Hola 2
Hola 3
Hola 4
Adios
```



2.- Estructuras repetitivas

5.- break y continue

Continue → Se fuerza la siguiente iteración y se deja de ejecutar las instrucciones que faltaban.

```
Ejemplo7.ps1 x
1 #Ejemplo7: Estructuras repetitivas: break
2 Clear-Host
3 $i = 1
4 for ($i=0;$i -lt 10;$i++) {
5     if ($i -eq 5){ continue
6     }
7     Write-Host " Hola $i"
8 }
9 Write-Host "Adios"
```

```
Hola 0
Hola 1
Hola 2
Hola 3
Hola 4
Hola 6
Hola 7
Hola 8
Hola 9
Adios
```



3.-Menú repetitivo.

Menú repetitivo

Ejemplo: MenuRepetitivo.ps1

```
#Definición de funciones
function Mostrar-Menu{
    Clear-Host
    Write-Host "Menu"
    Write-Host "1.- Desayunar"
    Write-Host "2.- Almorzar"
    Write-Host "3.- Cenar"
    Write-Host "4.- Salir"
}
#Inicio
do{
    Mostrar-Menu
    [int] $opcion = Read-Host "Introduce una opción [1,2,3,4]"
    switch ($opcion){
        1{Write-Host "Nos vamos a Desayunar"}
        2{Write-Host "Nos vamos a Almorzar"}
        3{Write-Host "Nos vamos a Cenar"}
        4{Write-Host "Adios"}
        exit}
    default{Write-Host "Has elegido una opción incorrecta"}
    } # Fin del switch
    $continuar = Read-Host "Pulsa intro para continuar"
}While($true)
```

Do{

}while (\$true)

Hace que el
bucle se repita
Continuamente,
hasta que se
pulse 4.



3.-Menú repetitivo.

Menú repetitivo

Ejemplo: MenuRepetitivo2.ps1

```
#Declaración de funciones
Function Mostrar-Menu{
    Clear-Host
    Write-Host "Menu"
    Write-Host "1.- Desayunar"
    Write-Host "2.- Almorzar"
    Write-Host "3.- Cenar"
    Write-Host "4.- Salir"
}
#Bloque principal
Mostrar-Menu
[int]$opcion=Read-Host "introduce una opción [1,2,3,4]"
do {
    switch ($opcion){
        1{Write-Host "Nos vamos a Desayunar"}
        2{Write-Host "Nos vamos a almorzar"}
        3{Write-Host "Nos vamos a cenar"}
        4{Write-Host "Adios"}
        exit}
    default {Write-Host "Has elegido una opción incorrecta"}
    }
    Read-Host "Pulse intro para continuar" #Sleep-Start -s 10
    Mostrar-Menu
    [int]$opcion=Read-Host "introduce una opción [1,2,3,4]"
}while($opcion -ne 4)
```

Do{

}while (\$true)

Hace que el
bucle se repita
Continuamente,
hasta que se
pulse 4.



Sugerencias/mejoras del tema



Sugerencias /mejoras del tema