

1. Realizar un programa en Powershell que pida un número e devolva si é primo ou non. Utiliza funcións na medida do posible. O programa tenos que permitir volver a facer ese cálculo unha e outra vez se así o desexamos:

```
Exer1.ps1 x Exer2.ps1 Exer3.ps1
Powershell > Ejercicios2 > Exer1.ps1
1 function EsPrimo {
2     param ([int]$Number)
3     if ($Number -le 1) {
4         return $false
5     }
6     for ($i = 2; $i -le [math]::Sqrt($Number); $i++) {
7         if ($Number % $i -eq 0) {
8             return $false
9         }
10    }
11    return $true
12 }
13
14 function Consultar {
15     do {
16         $num = Read-Host "Introduce un número para comprobar se é
17         primo"
18         if (EsPrimo -Number $num) {
19             Write-Host "$num é un número primo"
20         } else {
21             Write-Host "$num non é un número primo"
22         }
23         $continue = Read-Host "Queres comprobar outro número? (s/n)"
24     } while ($continue -eq 's')
25 }
26
27 clear
28
29 Consultar
```

2. Escribir un xogo de adiviñanza. O programa pedirá ao usuario dous números (o número inferior e o número superior). O programa obterá, a continuación, un número aleatorio entre eses dous números, e o usuario doberá adiviñalo. Cada vez que o usuario introduce un número, o programa lle dice se é maior ou menor. Ao final, o programa indica o número de intentos utilizado.

```
Exer1.ps1 Exer2.ps1 x Exer3.ps1
Powershell > Ejercicios2 > Exer2.ps1
1 function Adivina {
2     $num1 = [int](Read-Host "Introduce o número inferior")
3     $num2 = [int](Read-Host "Introduce o número superior")
4
5     if ($num1 -ge $num2) {
6         Write-Host "O número inferior debe ser menor que o superior"
7         return
8     }
9
10    $numR = Get-Random -Minimum $num1 -Maximum ($num2+ 1)
11    $intentos = 0
12    $guess = -1
13
14    while ($guess -ne $numR) {
15        $guess = [int](Read-Host "Adiviña o número")
16        $intentos++
17        if ($guess -lt $numR) {
18            Write-Host "O número é maior"
19        } elseif ($guess -gt $numR) {
20            Write-Host "O número é menor"
21        }
22    }
23
24    Write-Host "Parabéns! Adiviñaches o número $numR en $intentos
25    intentos."
26 }
27
28 clear
29 Adivina
```

3. Programa que recolla unha lista de números (ata que o usuario escriba un 0) e logo devolva esa lista ordenada.

```
Exer1.ps1  Exer2.ps1  Exer3.ps1  x
Powershell > Ejercicios2 > Exer3.ps1
1  function OrdenarNumeros {
2      $numeros = @()
3
4      while ($true) {
5          $input = Read-Host "Introduce un número (escribe 0 para
6              rematar)"
7          if ([int]$input -eq 0) {
8              break
9          }
10         $numeros += [int]$input
11     }
12
13     if ($numeros.Count -eq 0) {
14         Write-Host "Non introduciches ningún número."
15         return
16     }
17
18     $numerosOrdenados = $numeros | Sort-Object
19     Write-Host "Lista ordenada: $($numerosOrdenados -join ', ')"
20 }
21
22 clear
23 OrdenarNumeros
24
```