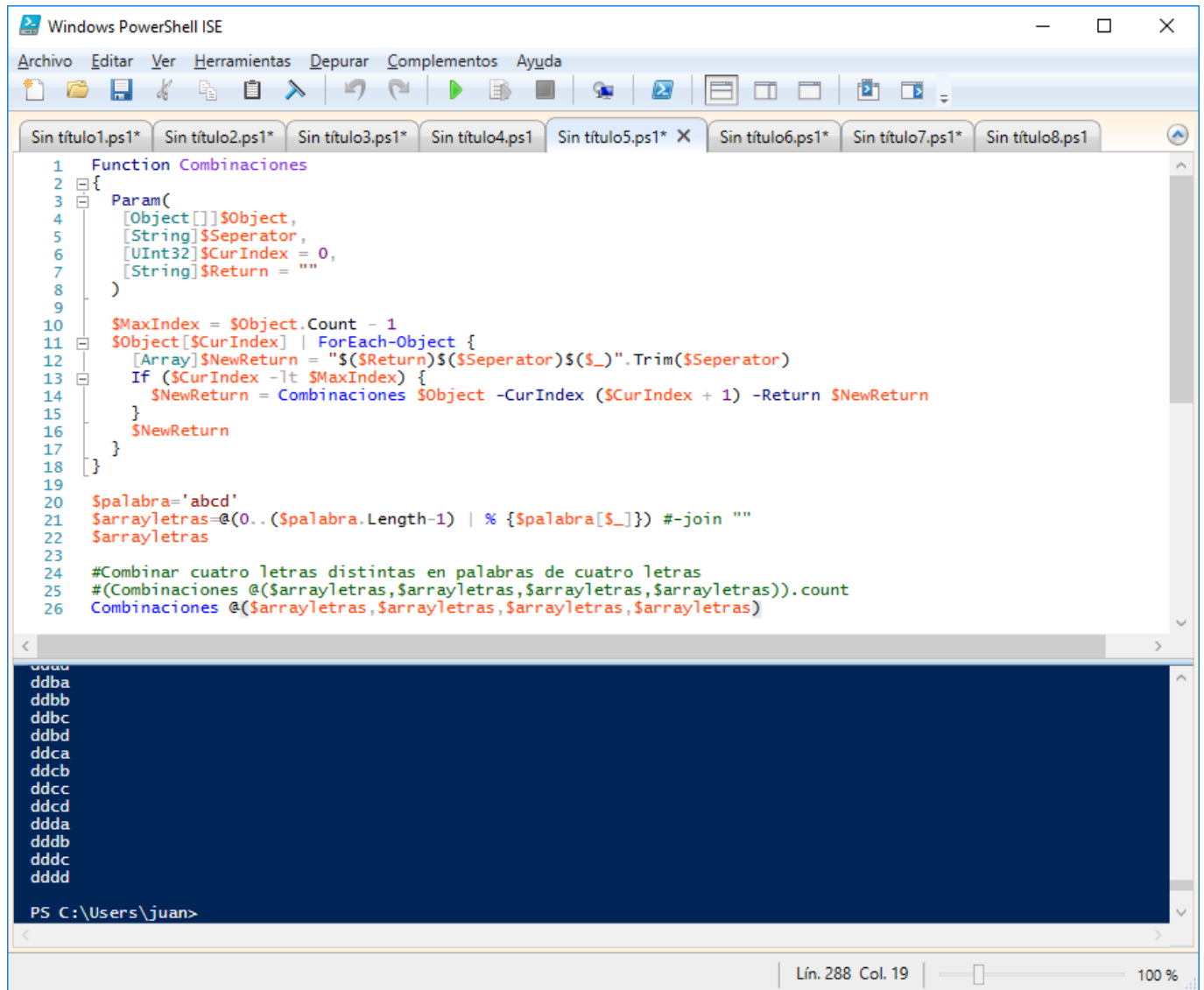


Realizar combinaciones de letras en PowerShell

[crayon-6a9d6769c9b49908307679/]



The screenshot shows the Windows PowerShell ISE interface. The script defines a recursive function named 'Combinaciones' that takes an array of objects, a separator, a current index, and a return string. It iterates through the array, building combinations by appending characters and calling itself recursively. The script uses the word 'ddba' as an example and prints the first 16 combinations of its letters.

```
1 Function Combinaciones
2 {
3     Param(
4         [Object[]]$Object,
5         [String]$Seperator,
6         [UInt32]$CurIndex = 0,
7         [String]$Return = ""
8     )
9
10    $MaxIndex = $Object.Count - 1
11    $Object[$CurIndex] | ForEach-Object {
12        [Array]$NewReturn = "$($Return)$($Seperator)$($_)".Trim($Seperator)
13        If ($CurIndex -lt $MaxIndex) {
14            $NewReturn = Combinaciones $Object -CurIndex ($CurIndex + 1) -Return $NewReturn
15        }
16        $NewReturn
17    }
18 }
19
20 $palabra='ddba'
21 $arrayletras=@(0..($palabra.Length-1) | % {$palabra[$_]} ) #-join ""
22 $arrayletras
23
24 #Combinar cuatro letras distintas en palabras de cuatro letras
25 #(Combinaciones @($arrayletras,$arrayletras,$arrayletras,$arrayletras)).count
26 Combinaciones @($arrayletras,$arrayletras,$arrayletras,$arrayletras)
```

Output:

```
ddba
ddbb
ddbc
ddbd
ddca
ddcb
ddcc
ddcd
ddda
dddb
dddc
dddd
```

PS C:\Users\juan>

Lín. 288 Col. 19 | 100 %