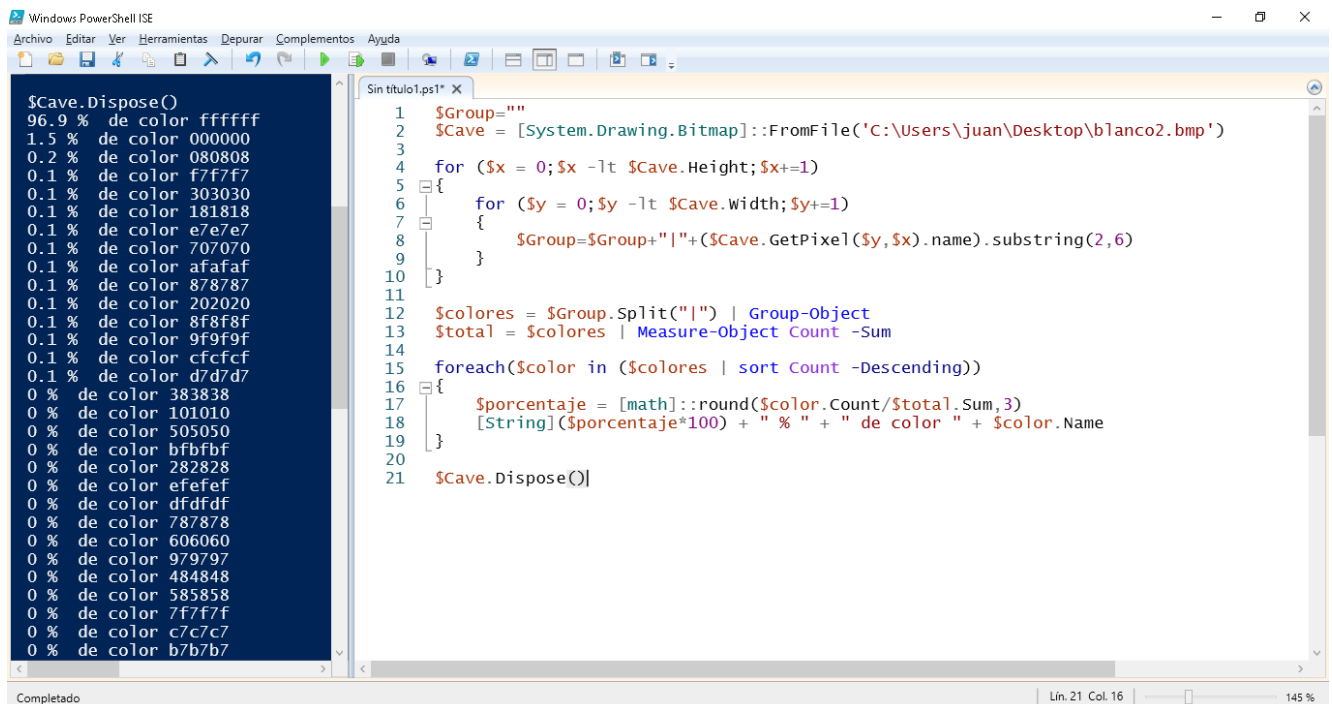


Calcular con PowerShell el porcentaje de colores que hay en una imagen

[crayon-6a9d57f72e597955751682/]



The screenshot shows the Windows PowerShell ISE interface. The left pane displays the output of a script, and the right pane shows the script code.

Output (Left Pane):

```
$Cave.Dispose()
96.9 % de color fffffff
1.5 % de color 000000
0.2 % de color 080808
0.1 % de color f7f7f7
0.1 % de color 303030
0.1 % de color 181818
0.1 % de color e7e7e7
0.1 % de color 707070
0.1 % de color afafaf
0.1 % de color 878787
0.1 % de color 202020
0.1 % de color 8f8f8f
0.1 % de color 9f9f9f
0.1 % de color cfcfcf
0.1 % de color d7d7d7
0 % de color 383838
0 % de color 101010
0 % de color 505050
0 % de color bfbfbf
0 % de color 282828
0 % de color efefef
0 % de color dfdfdf
0 % de color 787878
0 % de color 606060
0 % de color 979797
0 % de color 484848
0 % de color 585858
0 % de color 7f7f7f
0 % de color c7c7c7
0 % de color b7b7b7
```

Script Code (Right Pane):

```
1 $Group=""
2 $Cave = [System.Drawing.Bitmap]::FromFile('C:\Users\juan\Desktop\blanco2.bmp')
3
4 for ($x = 0; $x -lt $Cave.Height; $x+=1)
5 {
6     for ($y = 0; $y -lt $Cave.Width; $y+=1)
7     {
8         $Group=$Group+"|"+($Cave.GetPixel($y,$x).name).substring(2,6)
9     }
10 }
11
12 $colores = $Group.Split("|") | Group-Object
13 $total = $colores | Measure-Object Count -Sum
14
15 foreach($color in ($colores | sort Count -Descending))
16 {
17     $porcentaje = [math]::round($color.Count/$total.Sum,3)
18     [String]($porcentaje*100) + " % " + " de color " + $color.Name
19 }
20
21 $Cave.Dispose()
```

The status bar at the bottom indicates "Completado" (Completed) and "Lin. 21 Col. 16" (Line 21, Column 16).