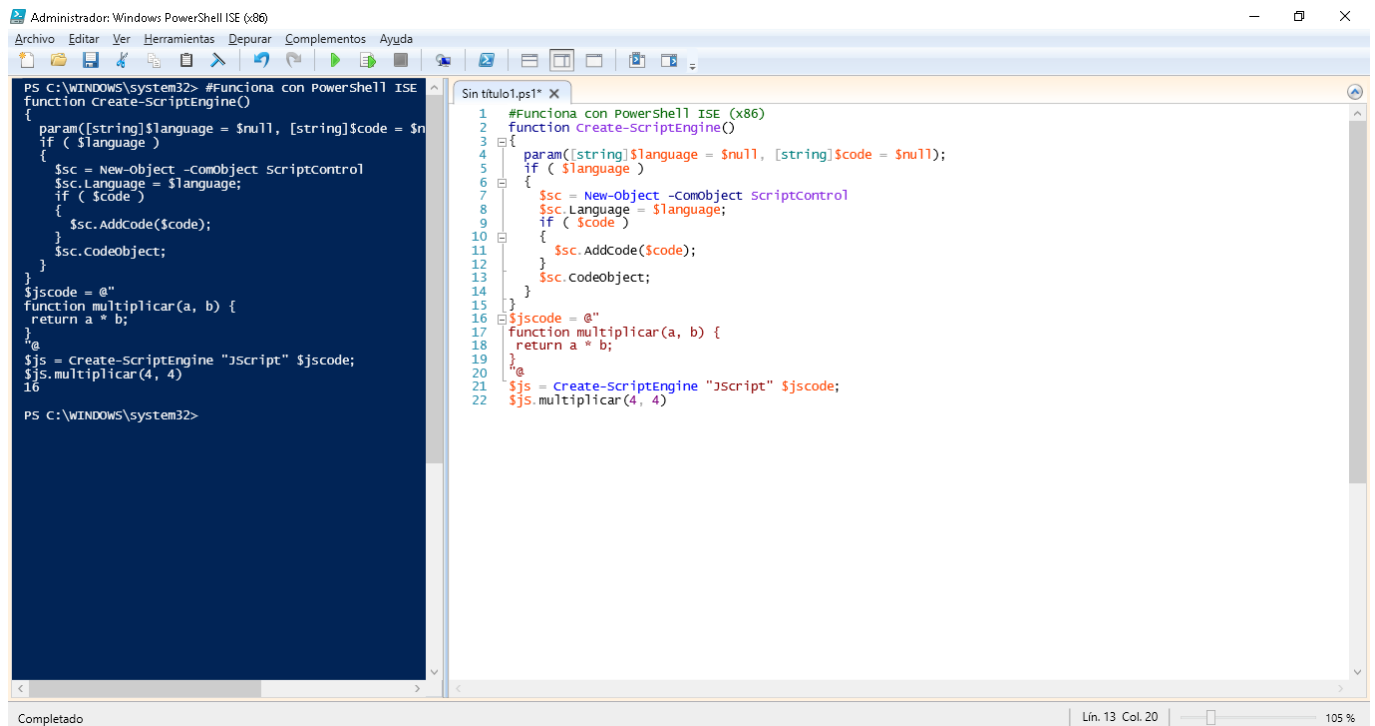


Utilizar funciones de JavaScript en PowerShell (ScriptControl)

[crayon-6a9d5ed35b588139227399/]



The screenshot shows the Windows PowerShell ISE (x86) interface. The left pane displays a PowerShell script, and the right pane displays its JavaScript equivalent. The PowerShell script defines a function `Create-ScriptEngine()` that takes `$language` and `$code` as parameters, creates a `ScriptControl` object, adds the code, and then uses `JavaScript` to call a `multiplicar` function. The JavaScript code defines the same function and uses `ScriptEngine` to call the `multiplicar` function.

```
PS C:\WINDOWS\system32> #Funciona con PowerShell ISE
function Create-ScriptEngine()
{
    param([string]$language = $null, [string]$code = $null);
    if ( $language )
    {
        $sc = New-Object -comobject ScriptControl
        $sc.Language = $language;
        if ( $code )
        {
            $sc.AddCode($code);
        }
        $sc.CodeObject;
    }
}
$jscode = @"
function multiplicar(a, b) {
    return a * b;
}
"@
$js = Create-ScriptEngine "JavaScript" $jscode;
$js.multiplicar(4, 4)
16
PS C:\WINDOWS\system32>
```

```
1 #Funciona con PowerShell ISE (x86)
2 function Create-ScriptEngine()
3 {
4     param([string]$language = $null, [string]$code = $null);
5     if ( $language )
6     {
7         $sc = New-Object -comobject ScriptControl
8         $sc.Language = $language;
9         if ( $code )
10        {
11            $sc.AddCode($code);
12        }
13        $sc.CodeObject;
14    }
15 }
16 $jscode = @"
17 function multiplicar(a, b) {
18     return a * b;
19 }
20 @"
21 $js = Create-ScriptEngine "JavaScript" $jscode;
22 $js.multiplicar(4, 4)
```

Completado | Lin. 13 Col. 20 | 105 %