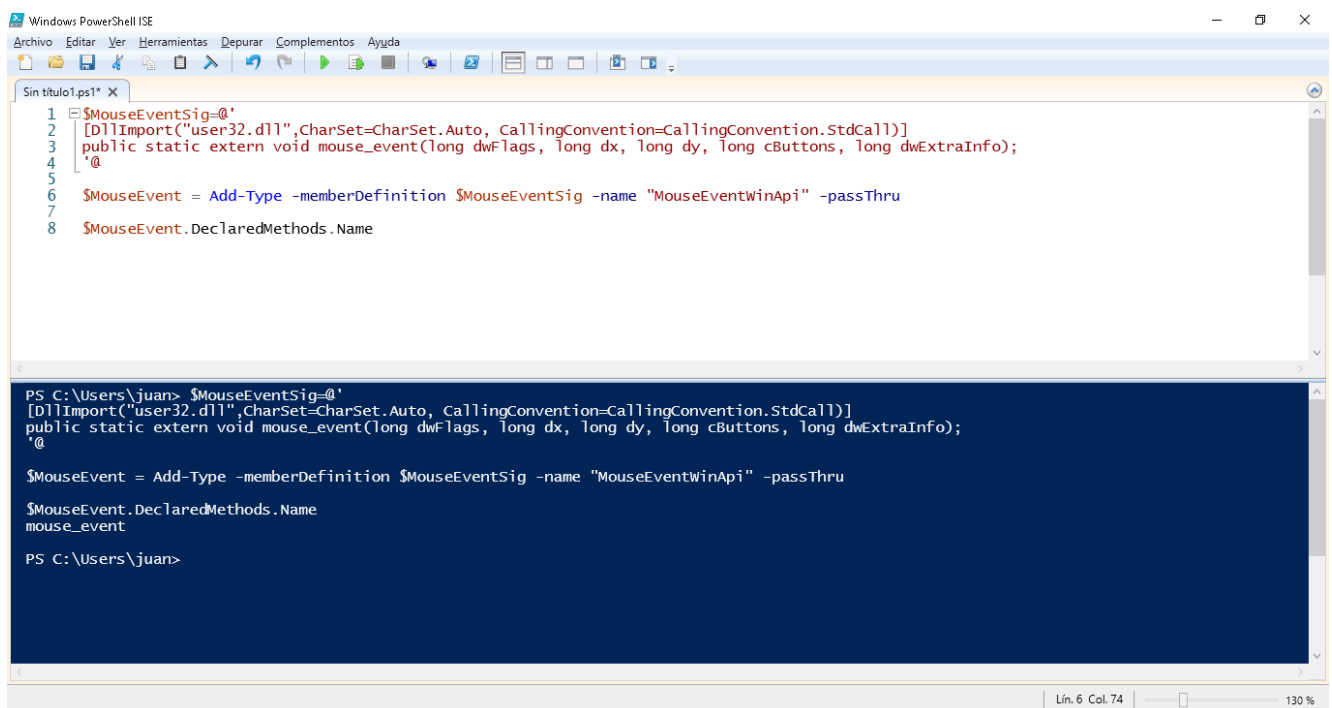


Obtener información sobre los métodos declarados en una librería DLL que hace uso de API nativas de Windows desde PowerShell

```
$MouseEventSig=@'  
[DllImport("user32.dll", CharSet=CharSet.Auto,  
CallingConvention=CallingConvention.StdCall)]  
public static extern void mouse_event(long dwFlags, long dx,  
long dy, long cButtons, long dwExtraInfo);  
'@
```

```
$MouseEvent = Add-Type -memberDefinition $MouseEventSig -name  
"MouseEventWinApi" -passThru
```

```
$MouseEvent.DeclaredMethods.Name
```



The screenshot shows the Windows PowerShell ISE interface. The script editor contains the following code:

```
1 $MouseEventSig=@'  
2 [DllImport("user32.dll",CharSet=CharSet.Auto, CallingConvention=CallingConvention.StdCall)]  
3 public static extern void mouse_event(long dwFlags, long dx, long dy, long cButtons, long dwExtraInfo);  
4 '@  
5  
6 $MouseEvent = Add-Type -memberDefinition $MouseEventSig -name "MouseEventWinApi" -passThru  
7  
8 $MouseEvent.DeclaredMethods.Name
```

The console output shows the execution of the script:

```
PS C:\Users\juan> $MouseEventSig=@'  
[DllImport("user32.dll",CharSet=CharSet.Auto, CallingConvention=CallingConvention.StdCall)]  
public static extern void mouse_event(long dwFlags, long dx, long dy, long cButtons, long dwExtraInfo);  
'@  
  
$MouseEvent = Add-Type -memberDefinition $MouseEventSig -name "MouseEventWinApi" -passThru  
  
$MouseEvent.DeclaredMethods.Name  
mouse_event  
  
PS C:\Users\juan>
```

The status bar at the bottom indicates "Lin. 6 Col. 74" and "130 %".