

PowerShell + SendKeys: el truco “mágico” que puede convertir Windows en un caos

Muchos descubren SendKeys en PowerShell y piensan:

“Increíble... puedo controlar el teclado automáticamente.”

Y sí, técnicamente puedes hacerlo.

Con apenas unas líneas puedes:

- escribir texto solo
- abrir programas
- pulsar atajos
- automatizar ventanas
- simular acciones humanas

El problema es que SendKeys no entiende aplicaciones. Solo lanza teclas a la ventana que esté activa en ese momento.

Y ahí empiezan los problemas.

👉 **Escribir texto automáticamente**

```
Add-Type -AssemblyName System.Windows.Forms
```

```
[System.Windows.Forms.SendKeys]::SendWait("Hola mundo")
```

Parece simple... y lo es.

Pero si cambias de ventana en ese instante:

□ el texto acaba donde no debe.

☐☐ Abrir programas desde teclado

Add-Type -AssemblyName System.Windows.Forms

```
[System.Windows.Forms.SendKeys]::SendWait("^{{ESC}}")  
Start-Sleep -Milliseconds 500  
[System.Windows.Forms.SendKeys]::SendWait("notepad")  
[System.Windows.Forms.SendKeys]::SendWait("{ENTER}")
```

Aquí se abre el menú inicio y lanza el bloc de notas.

☐☐ Automatizar atajos de teclado

Add-Type -AssemblyName System.Windows.Forms

CTRL + S

```
[System.Windows.Forms.SendKeys]::SendWait("^s")
```

ALT + TAB

```
[System.Windows.Forms.SendKeys]::SendWait("%{TAB}")
```

Símbolos útiles:

- ^ → CTRL
 - % → ALT
 - + → SHIFT
-

☐ Buscar cosas automáticamente en

Google

```
Start-Process "https://www.google.com"  
Start-Sleep 3
```

```
Add-Type -AssemblyName System.Windows.Forms
```

```
[System.Windows.Forms.SendKeys]::SendWait("PowerShell  
SendKeys")  
[System.Windows.Forms.SendKeys]::SendWait("{ENTER}")
```

Ideal para demos rápidas.

Terrible para automatización seria.

□ El típico script troll

```
Add-Type -AssemblyName System.Windows.Forms
```

```
while ($true) {  
[System.Windows.Forms.SendKeys]::SendWait("{CAPSLOCK}")  
Start-Sleep -Milliseconds 300  
}
```

Resultado:

□ las mayúsculas empiezan a cambiar sin parar y el teclado se vuelve inutilizable.

□ Efecto “máquina de escribir”

```
$text = "Esto parece inteligencia artificial..."
```

```
Add-Type -AssemblyName System.Windows.Forms
```

```
foreach ($char in $text.ToCharArray()) {
```

```
[System.Windows.Forms.SendKeys]::SendWait($char)  
Start-Sleep -Milliseconds 100  
}
```

Muy usado en vídeos y demostraciones.

⚠️ El verdadero problema

SendKeys depende COMPLETAMENTE del foco de la ventana.

Eso significa que:

- una notificación puede romper el script
- mover el ratón puede cambiar el foco
- un popup inesperado puede provocar acciones peligrosas
- un lag puede enviar comandos equivocados

Y entonces ocurren cosas muy divertidas... hasta que no lo son.

He visto scripts que:

- enviaban mensajes al chat incorrecto
 - pulsaban ENTER donde no debían
 - borraban archivos accidentalmente
 - automatizaban contraseñas “a lo bruto”
 - dejaban PCs bloqueados
-

❌ La falsa sensación de control

Este tipo de automatización impresiona mucho porque parece “hacker”.

Pero realmente es una de las técnicas más frágiles que existen en Windows.

Funciona perfecto...
hasta que alguien toca el teclado.