

## Conclusión

Como ha podido ver, a lo largo de la lectura de este libro, no existe una solución única para cada problema. En efecto, existen muchas opciones que difieren según los conocimientos de cada uno. Además, ha observado que las tecnologías COM, WMI, ADSI y .NET a menudo se superponen, lo que hace que a veces sea difícil responder a una cuestión tan sencilla como ésta: «¿con qué tecnología voy a conseguir automatizar tal o cual tarea?».

Le recomendamos buscar en primer lugar en el conjunto de comandos PowerShell, y si no encuentra lo que busca, diríjase a .NET (los accesos avanzados a WMI y ADSI se realizan a través de .NET), y como último recurso diríjase a COM. Es preferible dar prioridad a las tecnologías más recientes que a las más antiguas.

También habrá observado que el acceso al Framework .NET amplía considerablemente el campo de acción de PowerShell. Por consiguiente, según el grado de complejidad de los scripts, la frontera entre el scripting y el desarrollo es cada vez más fina. Conocemos además ciertos desarrolladores C# que recurren constantemente a PowerShell para probar algunas clases del Framework o para realizar maquetas. Hacen esto porque con PowerShell ganan tiempo ya que es posible, gracias a su intérprete de comandos, probar rápidamente trozos de código sin tener que recompilarlos. También hay que decir que el lenguaje de Windows PowerShell es bastante parecido al C#; exagerando un poco se podría casi considerar como una versión simplificada de éste.

De este modo un administrador del sistema curtido en las técnicas de scripting de PowerShell y .NET podrá, también, sin mucho esfuerzo intentar una incursión en el mundo del desarrollo de aplicación en C#. ¿Quizás para añadir funcionalidades adicionales a PowerShell...?