**Profiler en visual basic**

**Opciones:**

* **.NET Async**

Herramienta para investigar el uso de async/await en las aplicaciones.NET

* **Contadores de.NET**

Herramienta para visualizar los contadores de rendimiento en las aplicaciones .NET

* **Instrumentación**

Ver recuentos precisos de llamadas e intervalos de funciones en el código

* **Uso de CPU**

Observar a qué dedica tiempo la CPU cuando se ejecuta el código. Resulta útil cuando el cuello de botella que afecta al rendimiento está en la CPU.

* **Uso de memoria**

Realizar un análisis de la memoria de la aplicación para detectar problemas tales como fugas de memoria.

* **Base de datos**

Examinar cuándo se ejecutaron las consultas y medir cuánto tardan

* **E/S de archivo**

Ver qué operaciones de E/S de archivos se están realizando, cuánto tardan y cuántos datos están procesando

* **Seguimiento de asignación de objetos de .NET**

Vea dónde se asignan los objetos .NET y cuando se recuperan por el GC.

* **Uso de GPU**

Examinar el uso de GPU de la aplicación de DirectX. Resulta útil para determinar si el cuello de botella que afecta al rendimiento se encuentra en la CPU o la GPU.

* **Visor de eventos**

Vea los eventos (ETW o NetTrace) que se han producido durante la sesión, como mensajes de registro, excepciones y solicitudes HTTP.

**Posibles herramientas útiles:**

* Uso de CPU
* Bases de Datos
* E/S de archivo3
* Uso de memoria
* Instrumentación

**Procesos a seguir:**

1. **Uso de CPU**

**¿Por qué?** Es un problema de lentitud, por lo que el CPU cuenta con un uso excesivo dentro de alguna o algunas consultas.

**¿Cómo usarlo?**

1. Alt + F2 - para abrir el panel de opciones.

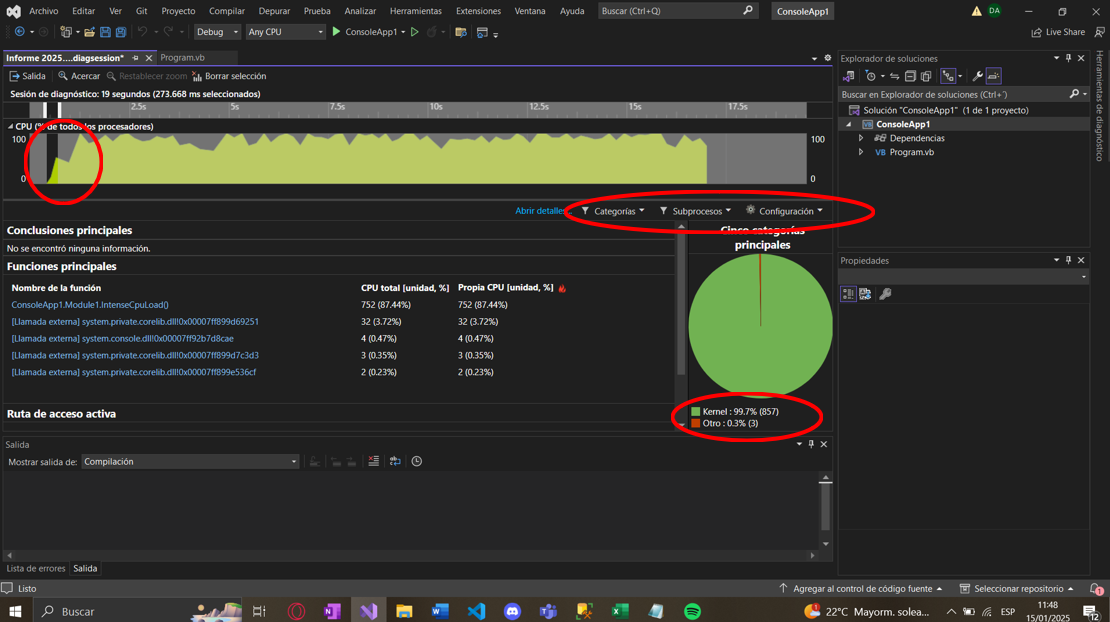
2. Activación de uso

3. Simulación del proceso de guardado de consultas.

4. Revisión de funciones más extensas para encontrar posibles cuellos de botella.

**¿Qué se puede hacer con la información obtenida?**

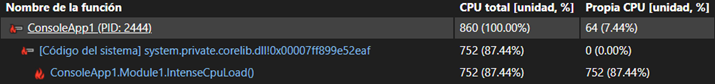
o Filtrar por tiempo en la ejecución, por ejemplo:



Se puede eliminar el ruido tanto del inicio como el final para únicamente ver lo que están ocasionando los “picos”.

Se pueden filtrar subprocesos, códigos (nativo, propio) y categorías(kernel, networking, I/O).

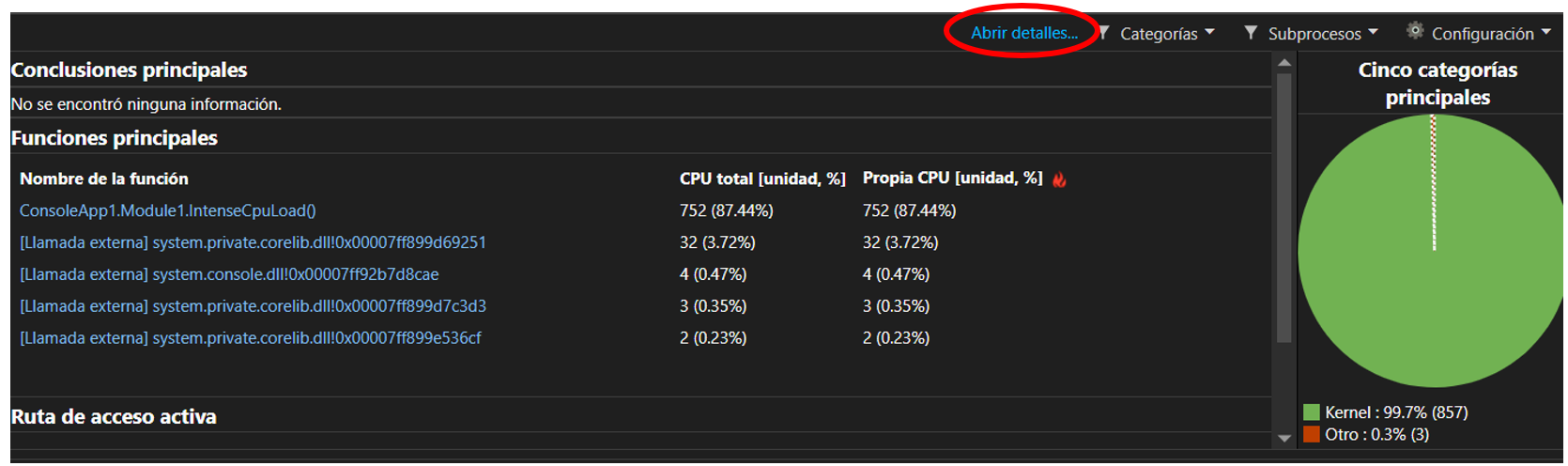
o Funciones específicas:



Provee las funciones y la cantidad de memoria que están utilizando.

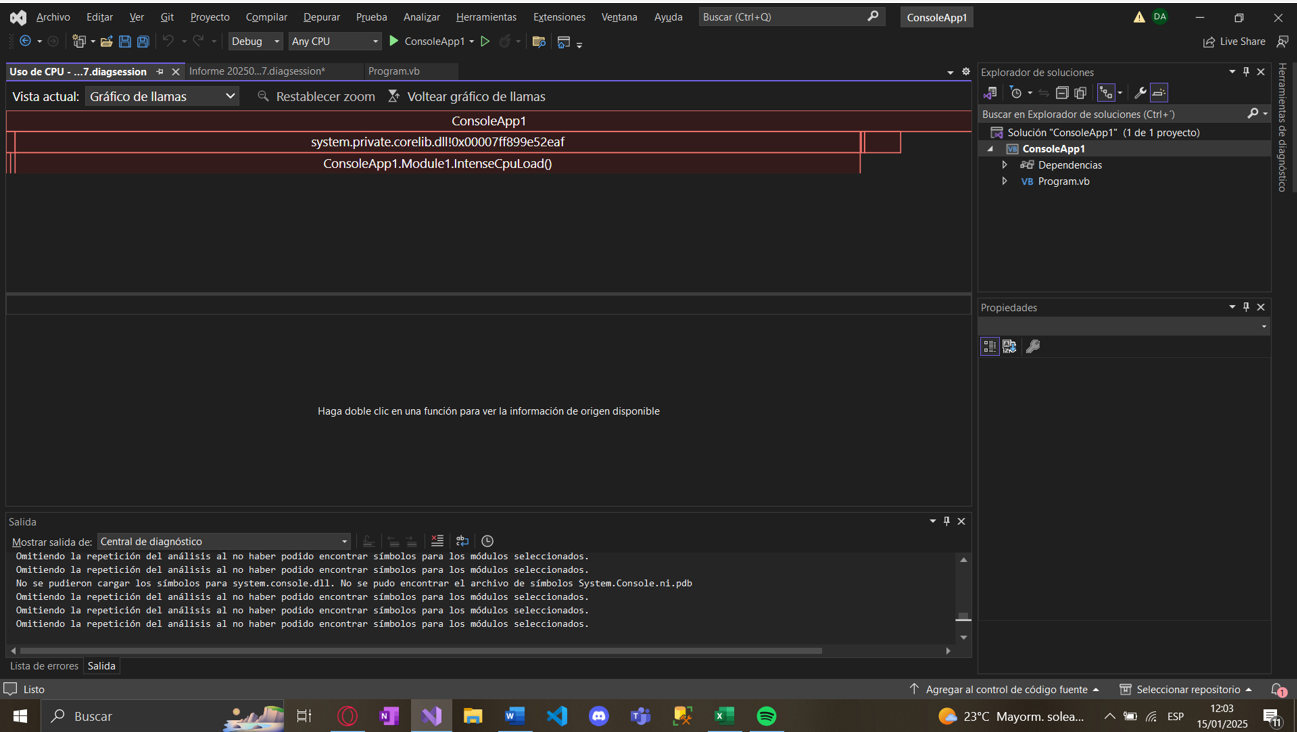
o Búsqueda de detalles

Se cuenta con una pestaña que dice “abrir detalles”, la cual permite una investigación más profunda del performance que esta teniendo el programa.

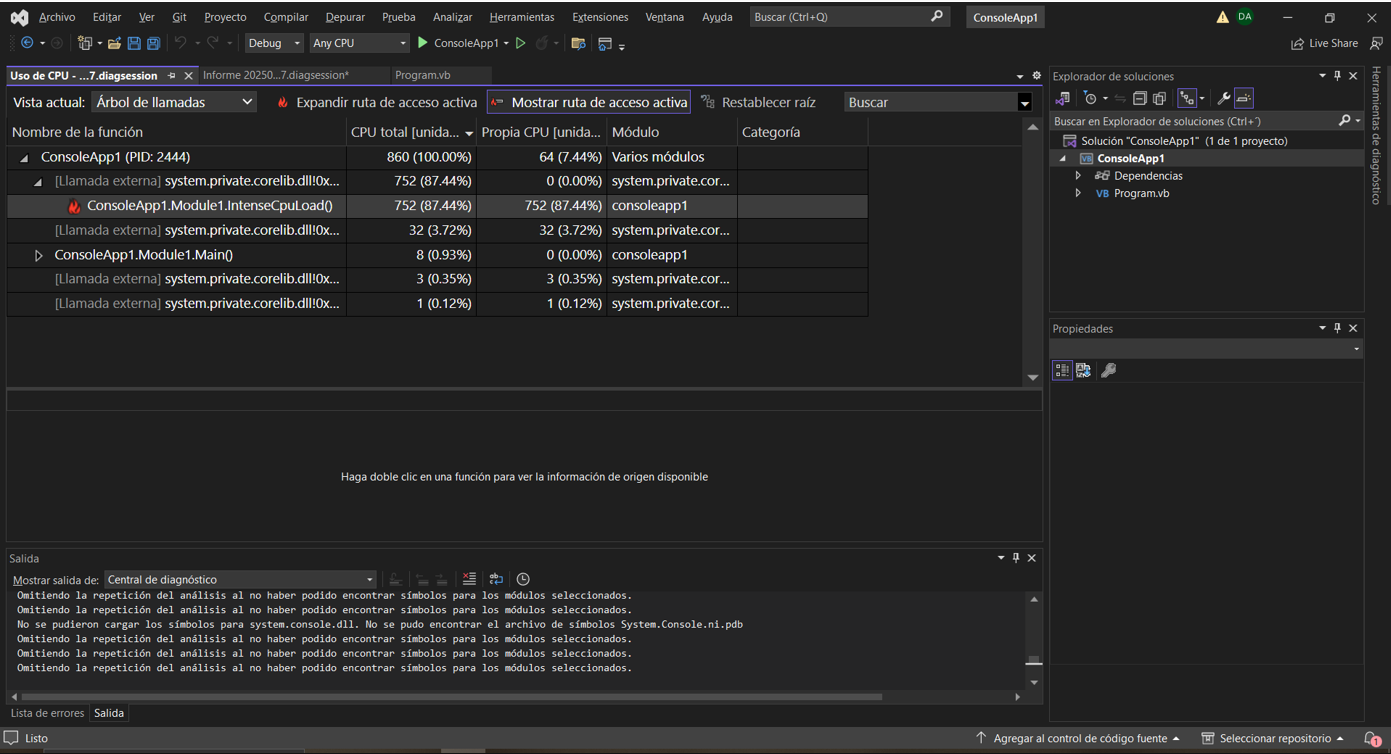


**Detalles de uso:**

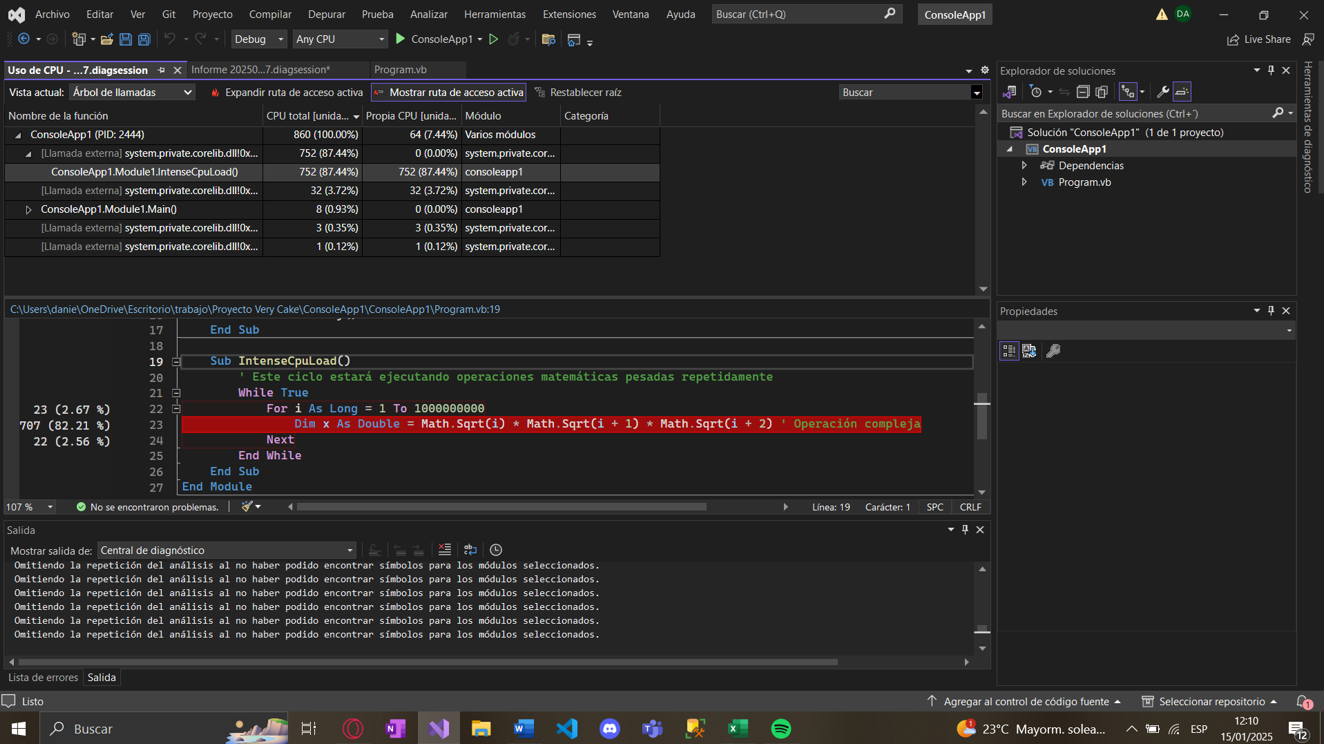
* Flame Graph/Gráfico de llamas: visualización de las llamadas a stack.



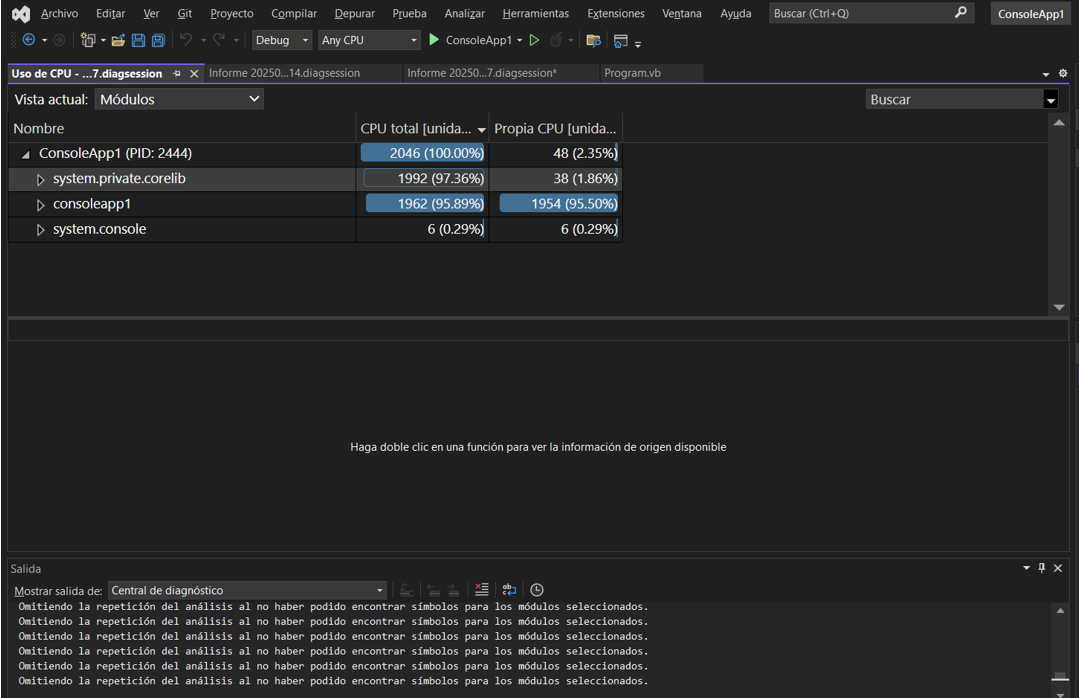
* + Da una idea de dónde el código pasa más tiempo o qué es lo que más se llama.
* Call Tree/ árbol de llamadas: muestra las distintas rutas de código que consumen CPU significativamente.



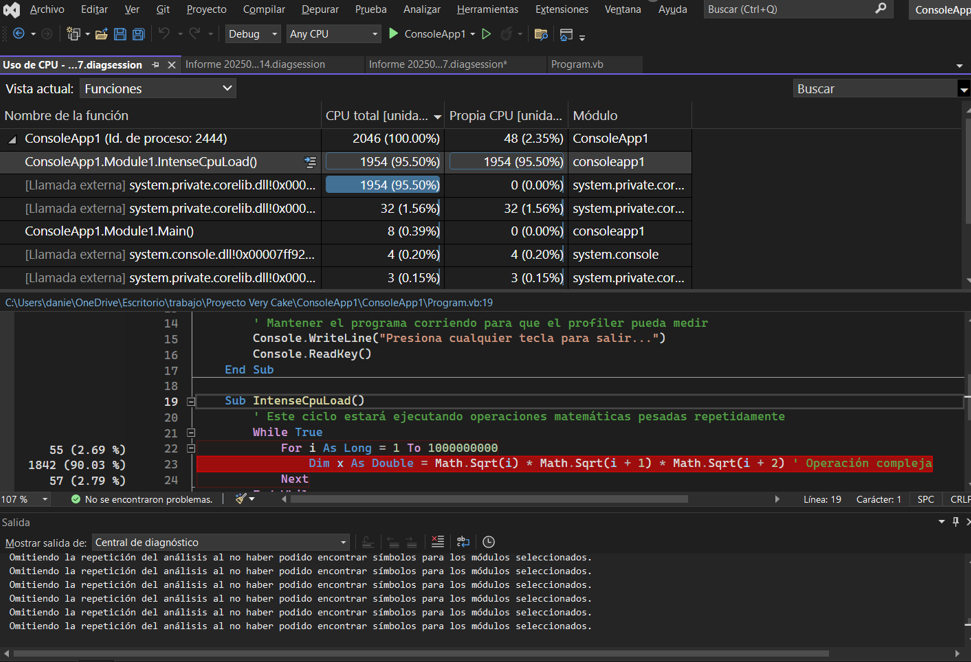
Las líneas con una llama son las que suelen estar ocasionando el problema o tienen un consumo muy alto. Con doble click vemos las líneas específicas.



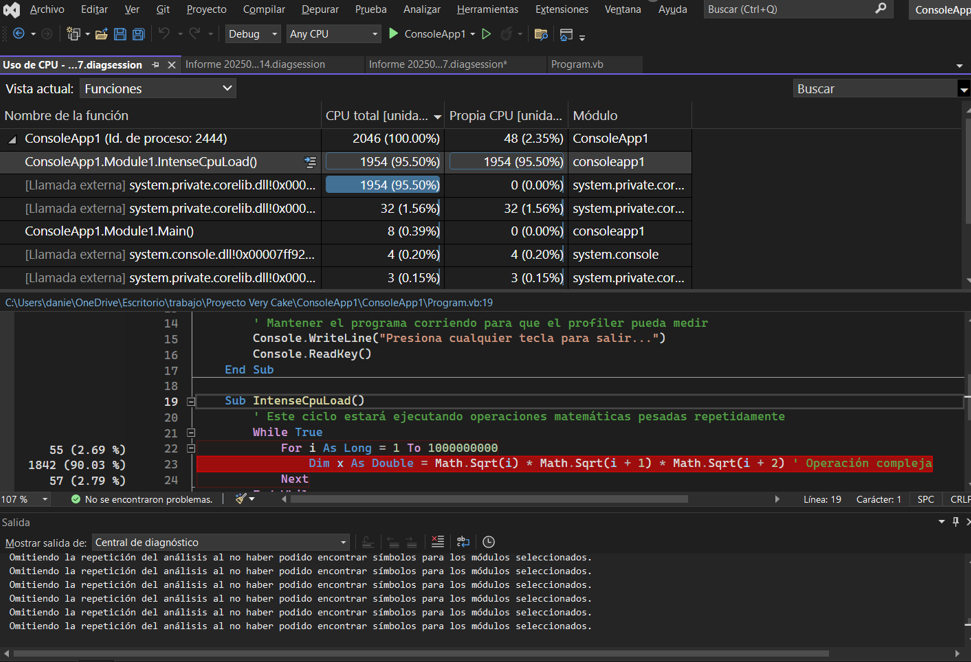
* Módulos/modules: muestra todas las llamadas de pila por módulo



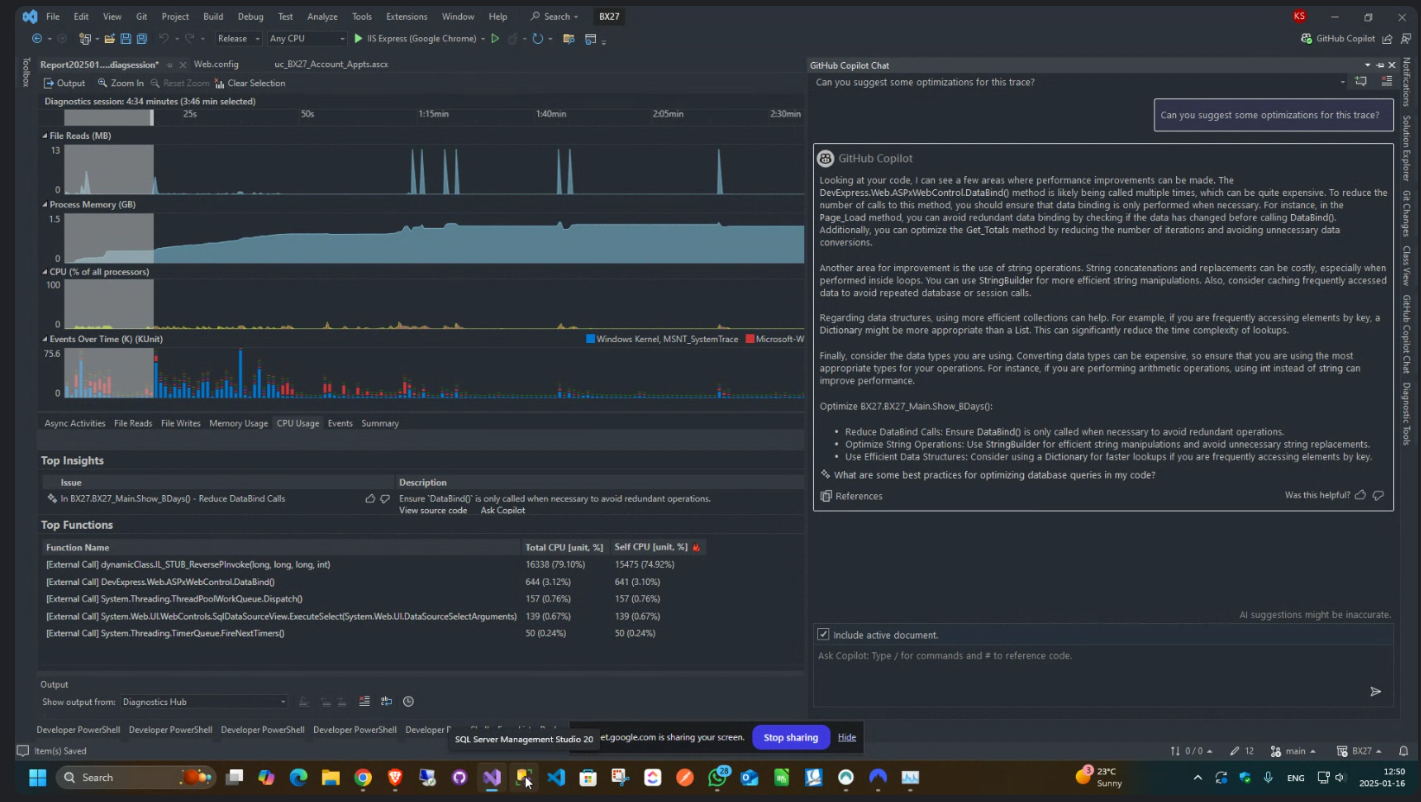
* Funciones/Functions: muestra todas las llamadas de pila por función



* Callee/callee / llamador y destinatario:



* Uso de copilot posterior al diagnóstico de profiler

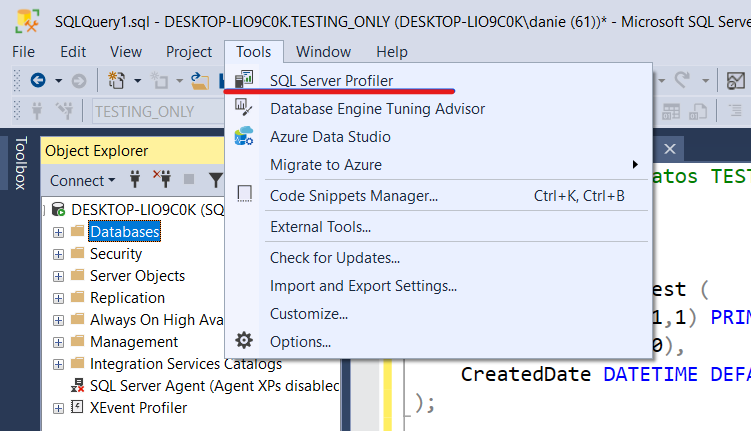


**2. Base de Datos – para Web Forms y ASP.NET - SQL PROFILER**

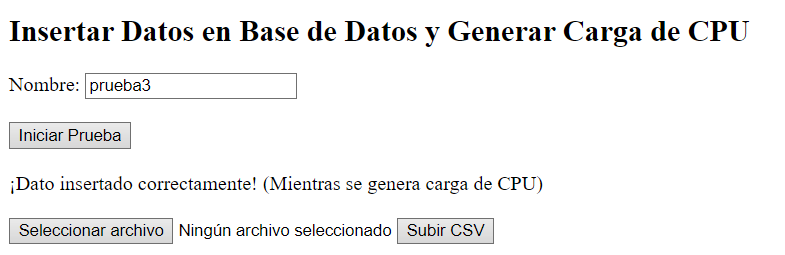
**¿Por qué?** Las aplicaciones web suelen depender de bases de datos, con esta herramienta se analiza el tiempo que tardan las consultas que se ejecutan desde la aplicación.

**¿Cómo usarlo?**

1. Activar el servidor en SQL Server

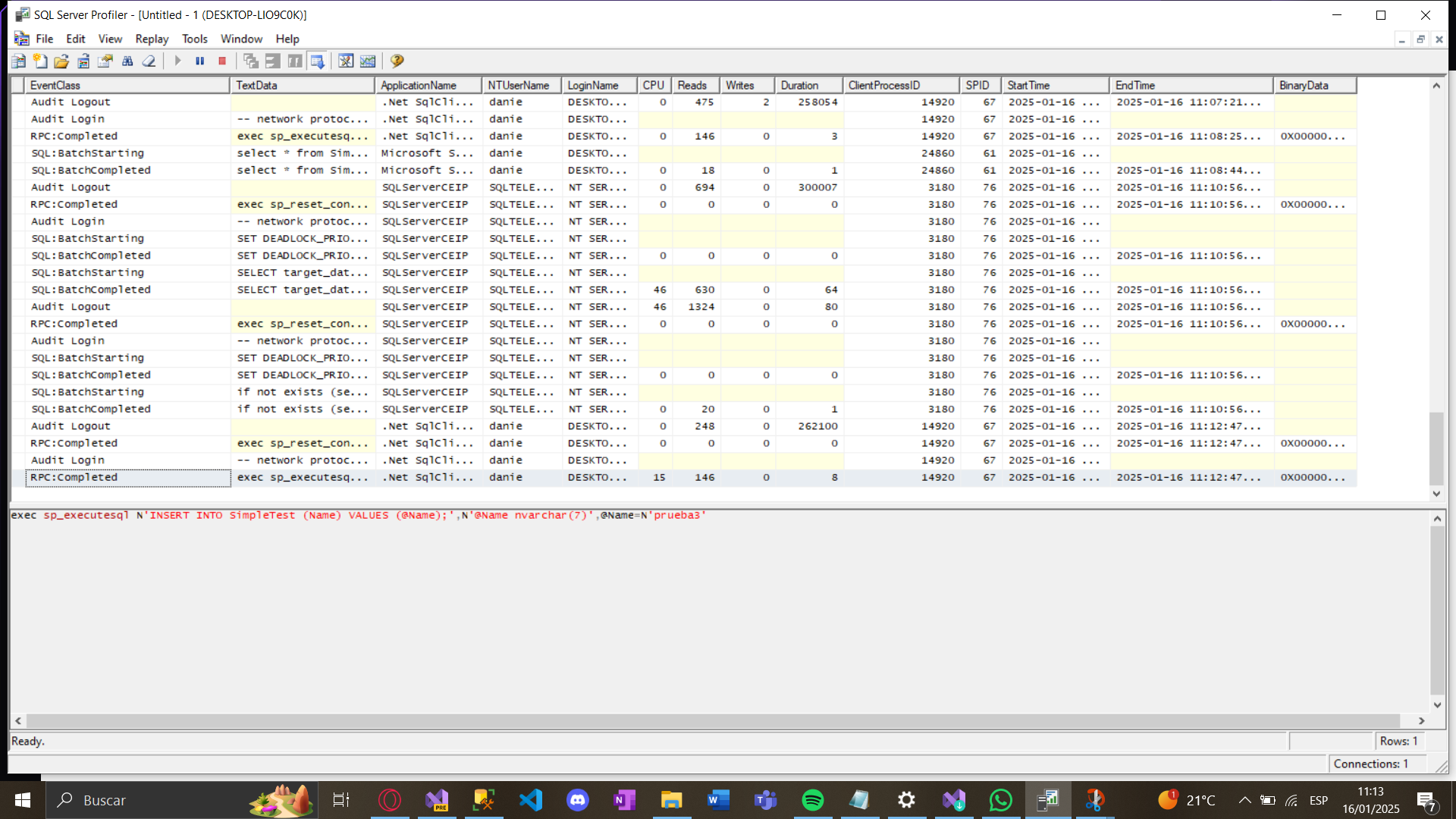


2. Realizar la acción que está ocasionando el retraso.



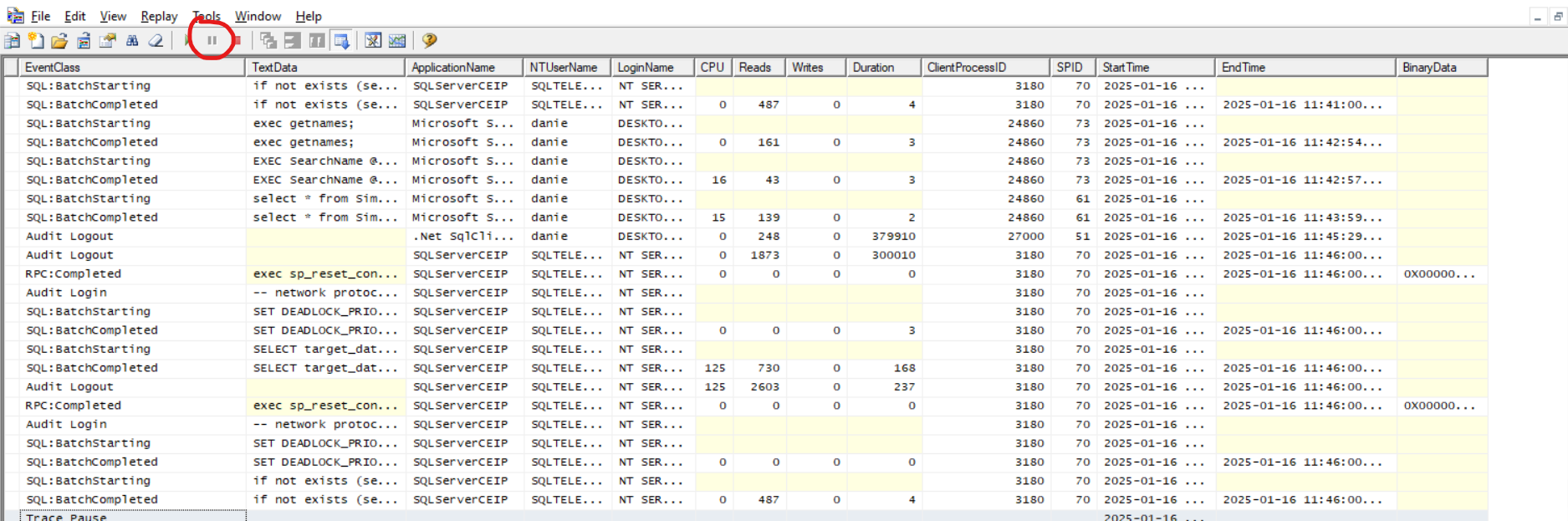
En este caso se están cargados desde datos individuales hasta múltiples para poder poner a prueba el servidor y sus conexiones.

3. Revisar los reportes.



Cada consulta o inserción se registra en el profiler

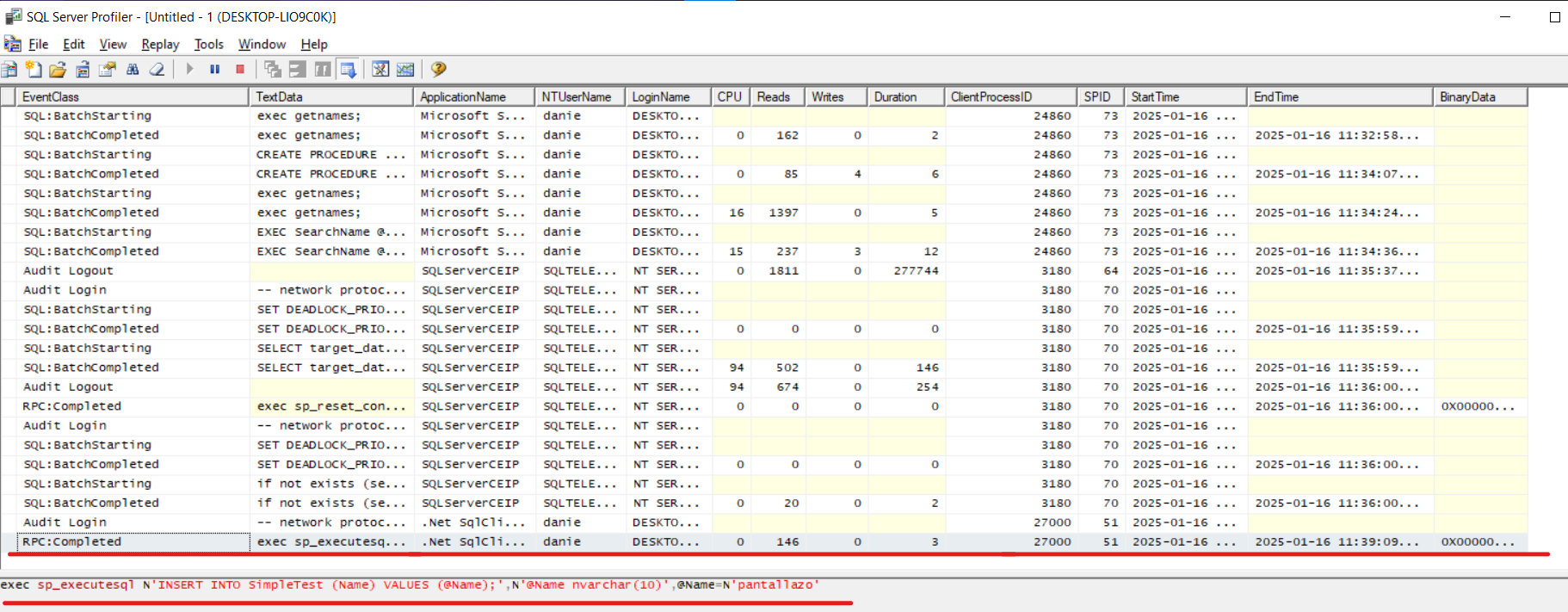
4. Pausar para ver únicamente los eventos de interés:



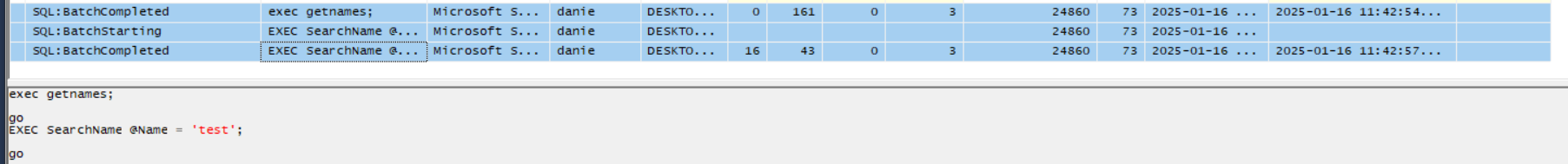
**¿Qué hacer con la información obtenida?**

Obtenemos información detallada de cada acción realizada como:

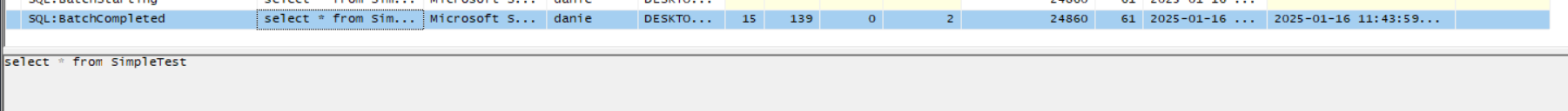
1. Datos ingresados a la base de datos:

****

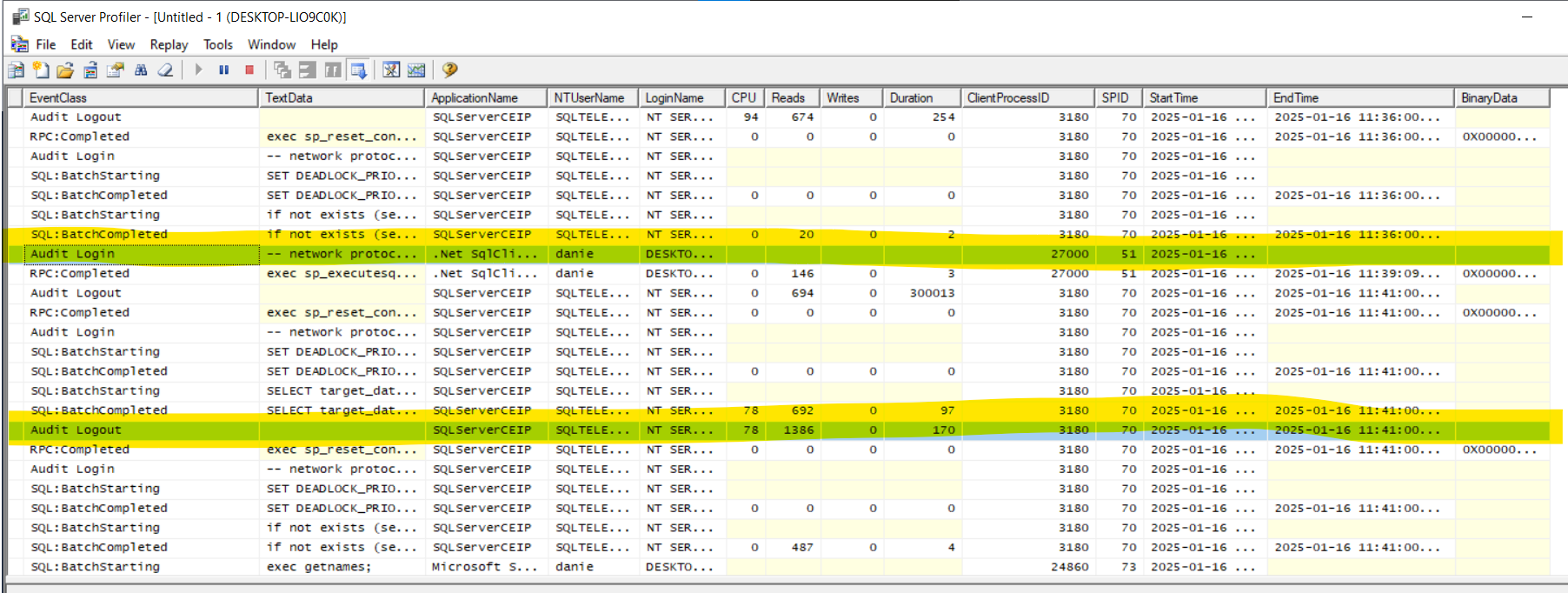
1. Procedimientos ejecutados:



1. Consultas realizadas:



1. Conexiones y salidas del servidor:



1. Seleccionar líneas de interés para ver pasos de ejecución y analizarlos posteriormente en SQL Management:

