

Tracing

Una lista de eventos ordenados a través del tiempo acumulado (timestamps)

Tracing provee en detalle el comportamiento dinámico de una aplicación/software

El objetivo principal del tracing es seguir el flujo de información en el programa

Tracing no es lo mismo que hacer registros (logs)

Si una función tuvo un problema, el tracing te ayuda a encontrar ese problema verificar que parámetros se usaron, el tiempo que duro y como se llama la función

Tracing es complicado sin las herramientas adecuadas

El tracing generalmente se realiza post release

Fuente

<https://www.bmc.com/blogs/monitoring-logging-tracing/>

Debug vs trace

Debug se utiliza con fines puros de depuración. Emite mensajes enriquecidos en la ejecución de depuración (modo de depuración).

Trace ayuda en la depuración de aplicaciones, la corrección de errores y la creación de perfiles (después del lanzamiento).

La clase Debug no sirve para nada en el modo de lanzamiento.

<https://stackoverflow.com/questions/179868/trace-vs-debug-in-net-bcl>

When to tracing? Debugging

Debug mientras la aplicación no fue lanzada porque nos ayuda a encontrar errores en nuestro código, no se puede medir el rendimiento mientras encontramos errores

Tracing nos ayuda a obtener información sobre el rendimiento de nuestra aplicación en producción

<https://www.c-sharpcorner.com/interview-question/what-is-difference-between-debug-and-tracing#:~:text=Debug%20is%20going%20through%20the,execution%20plan%2C%20process%20timing%20details.&text=Debug%20and%20trace%20enables%20you,debugging%20code%20inside%20the%20executable.>

.NET capabilities for tracing / instrumentation

.NET tiene la herramienta dotnet trace para hacer seguimiento

Net Tools es una enorme caja de herramientas de Windows, tiene un número impresionante de herramientas, todas con una interfaz de usuario muy intuitiva que estoy seguro que te gustará.

<https://www.pchardwarepro.com/net-tools-una-herramienta-para-hackers/>

Static code analyzers

Es un tipo de análisis que se realiza al software bajo prueba sin ejecutar el programa

¿Que se revisa?

- Estándares
- Reglas de diseño
- Inspección de código

¿Que detecta?

- Código malicioso
- Faltas a los estándares
- Mal uso de recursos

Algunos para usarlos son:

- Automatizar la búsqueda de errores.
- -Comprobar convenciones de programación y estilo de forma automática.
- -Facilitar el trabajo en equipo (al detectar y evitar errores triviales al modificar código de terceros).

Respecto a los puntos débiles son:

- La dificultad en la comprensión de los mensajes de error (¿Cuál es el problema? ¿Qué tengo que arreglar?).
- El exceso de falsos negativos.
- La dificultad en configurar un analizador para ocultar falsos positivos (p.ej. no mostrar los posibles errores en este fichero).
- El problema de encajar el analizador en el flujo de herramientas y de trabajo de la empresa (¿Cuándo se hace el análisis? ¿En cada nueva versión? ¿Cada cierto tiempo? ¿Lo hace el propio programador?).