



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO



## **APLICACIONES PARA COMUNICACIONES EN RED**

UNIDAD III: Arquitectura Cliente-Servidor

### **TAREA – TAMAÑO DE LA PILA DE HILO**

Grupo 6CM2 - Ciclo 2026A

Alumnas:

Cruz Rodríguez Arely Amairani

Ortiz Villaseñor Alexandra

Fecha de entrega: Martes 06 de enero de 2026

# INVESTIGACIÓN

## ¿Qué es la Pila de Hilo (Thread Stack)?

En el desarrollo de aplicaciones en red, un hilo (thread) es la unidad más pequeña de ejecución que un sistema operativo puede gestionar. Cuando un programa inicia un hilo para atender a un cliente, el sistema le asigna una porción de memoria exclusiva llamada Stack (Pila).

La pila se utiliza para almacenar:

- **Variables locales:** Datos temporales dentro de una función.
- **Marco de llamadas (Call Stack):** El rastro de qué funciones ha llamado el hilo y a dónde debe regresar al terminar.
- **Argumentos de funciones.**

A diferencia del *Heap* (donde los datos se comparten), la pila es privada. Si el hilo intenta usar más memoria de la asignada, ocurre el famoso error StackOverflow.

## Contextos de Uso

- **En Sistemas Operativos.** El S.O. es quien define el límite máximo y el valor por defecto. Es el "dueño" de la memoria RAM y decide cuánta libertad darle a cada proceso para crear hilos.
- **En Lenguajes de Programación y Runtimes.** Lenguajes como Java (JVM) o .NET gestionan su propia pila sobre la del sistema operativo. Aquí es donde el programador suele ajustar el tamaño mediante parámetros (como -Xss en Java) para optimizar el servidor.
- **En Navegadores Web.** Los navegadores modernos son multi-proceso y multi-hilo. Cada pestaña suele ser un proceso, pero dentro de esa pestaña, el motor de JavaScript (como V8 en Chrome) gestiona hilos para tareas asíncronas, renderizado y peticiones de red. Si la pila de estos hilos es muy pequeña, scripts complejos fallarán; si es muy grande, el navegador consumirá demasiada RAM.

## Tamaños Predeterminados (Benchmarks)

Los valores varían según la arquitectura (32-bit vs 64-bit), pero estos son los estándares actuales:

Sistemas Operativos	
Windows (Generic)	1 MB
Linux (pthreads)	2 MB a 8 MB (varía según la distro)
macOS	512 KB (Hilos secundarios) / 8 MB (Hilo principal)

Lenguajes y Entornos (Runtimes)	
Java (JVM 64-bit)	1,024 KB (1 MB)
.NET (C# / F#)	1 MB
Node.js (V8)	984 KB aproximadamente
Python	Variable (depende del S.O., usualmente 2 MB)

Navegadores (Motores JS)	
Los navegadores suelen ser más restrictivos para evitar que un sitio web malicioso o mal programado congele la computadora:	
Chrome (V8)	984 KB
Firefox (SpiderMonkey)	1 MB
Safari (JavaScriptCore)	512 KB

## La relación entre el tamaño de la pila y la escalabilidad es inversamente proporcional

### El Límite de Conexiones

Si diseñas un servidor socket que crea un hilo por cada conexión entrante (modelo *Thread-per-Connection*), el tamaño de la pila determina cuántos usuarios puedes recibir antes de colapsar la RAM.

**Ejemplo matemático:** Supongamos que tienes un servidor con **16 GB de RAM** disponibles para hilos:

1. Con Stack de **2 MB**:

$$\frac{16,384MB}{2MB/hilo} = 8,192 \text{ conexiones máximas}$$

2. Con Stack de **256 KB** (0.25 MB):

$$\frac{16,384MB}{0.25MB/hilo} = 65,536 \text{ conexiones máximas}$$

### Consecuencias de un valor mal configurado:

- **Si el valor es muy alto:** El servidor se queda sin memoria rápidamente (Out of Memory). Podrías tener un procesador potente ocioso, pero sin RAM para registrar más hilos, el servidor rechazará nuevas conexiones.
- **Si el valor es muy bajo:** El servidor soportará millones de conexiones, pero en cuanto una función realice una operación medianamente compleja o una recursividad, el hilo morirá por `StackOverflowError`, desconectando al usuario de forma abrupta.

### Conclusión

La investigación demuestra que el Thread Stack Size es un parámetro de "tira y afloja". En aplicaciones de red de alto rendimiento (como servidores de chat o juegos en línea), la tendencia es usar modelos no bloqueantes (Event Loop) o reducir el tamaño de la pila al mínimo necesario para maximizar la concurrencia.

### Bibliografía

- [1] Oracle Corporation, "Java Platform, Standard Edition Tools Reference: Java HotSpot VM Options," *Oracle Help Center*, 2024. [En línea]. Disponible en: <https://docs.oracle.com/javase/8/docs/technotes/tools/unix/java.html>. [Accedido: 06-ene-2026].
- [2] Microsoft Learn, "Thread.StackSize Property: Managed Threading in .NET," *Microsoft Documentation*, 2023. [En línea]. Disponible en: <https://learn.microsoft.com/en-us/dotnet/api/system.threading.thread>. [Accedido: 06-ene-2026].
- [3] IBM Corporation, "Default stack sizes for threads on Linux and AIX," *IBM Documentation*, 2022. [En línea]. Disponible en: <https://www.ibm.com/docs/en/sdk-java-technology/8>. [Accedido: 06-ene-2026].
- [4] Google Open Source, "V8 JavaScript Engine: Stack Guard Implementation," *V8.dev*, 2025. [En línea]. Disponible en: <https://v8.dev/docs/embed>. [Accedido: 06-ene-2026].
- [5] Mozilla Developer Network (MDN), "Internal errors: too much recursion," *MDN Web Docs*, 2024. [En línea]. Disponible en: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Errors/Too\\_much\\_recursion](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Errors/Too_much_recursion). [Accedido: 06-ene-2026].
- [6] Gemini 3 Flash, "Análisis sobre la arquitectura de la pila de hilos y su impacto en la concurrencia de servidores de red," *Google AI Assistance*, comunicación privada, ene. 06, 2026.