



DESARROLLO WEB

NODE.JS Y SU FUNCIONAMIENTO

NODE.JS Y SU FUNCIONAMIENTO: V8 ENGINE Y EJECUCIÓN EN EL SERVIDOR

Node.js representa una revolución en el desarrollo del lado del servidor, al permitir la ejecución de JavaScript fuera del navegador web. Esta plataforma se construye sobre el motor V8 de Google Chrome, proporcionando un entorno de ejecución altamente optimizado que compila JavaScript directamente a código máquina nativo. La arquitectura de Node.js se fundamenta en un modelo de eventos no bloqueante y de un solo hilo, lo que lo convierte en una solución ideal para aplicaciones que requieren alta concurrencia y manejo intensivo de operaciones de entrada y salida.

El motor V8 opera como el corazón de Node.js, ejecutando el código JavaScript con una eficiencia comparable a lenguajes compilados tradicionales. Según Granados La Paz (2023), las tecnologías del servidor moderno requieren arquitecturas que optimicen el rendimiento y la gestión de recursos, características que Node.js logra mediante su event loop y la gestión asíncrona de operaciones. Este enfoque permite que una sola instancia de Node.js maneje miles de conexiones concurrentes sin la sobrecarga típica asociada con la creación de hilos múltiples, resultando en un uso más eficiente de los recursos del sistema.

En el ámbito de la ingeniería de software, Node.js se ha consolidado como una herramienta fundamental para el desarrollo de microservicios y APIs de alto rendimiento. Por ejemplo, empresas como Netflix y LinkedIn han adoptado Node.js para manejar millones de solicitudes simultáneas en sus plataformas, aprovechando su capacidad para procesar operaciones I/O de manera no bloqueante. Esta característica resulta especialmente valiosa en aplicaciones que requieren tiempo real, como sistemas de chat, notificaciones push o dashboards de monitoreo en tiempo real.

Ejercicio práctico. Crear un servidor básico en Node.js que demuestre el event loop.

Paso 1. Instalar Node.js y crear un archivo server.js.

Paso 2. Implementar el código: `const http = require('http'); const server = http.createServer((req, res) => { console.log('Solicitud recibida:', Date.now()); setTimeout(() => { res.writeHead(200, {'Content-Type': 'text/plain'}); res.end('Respuesta después de 2 segundos'); }, 2000); }); server.listen(3000, () => console.log('Servidor en puerto 3000'));`

Paso 3. Ejecutar con `node server.js` y realizar múltiples solicitudes simultáneas.

Resultado. Se observa que el servidor maneja múltiples solicitudes concurrentemente sin bloquear el hilo principal, demostrando la naturaleza asíncrona de Node.js.