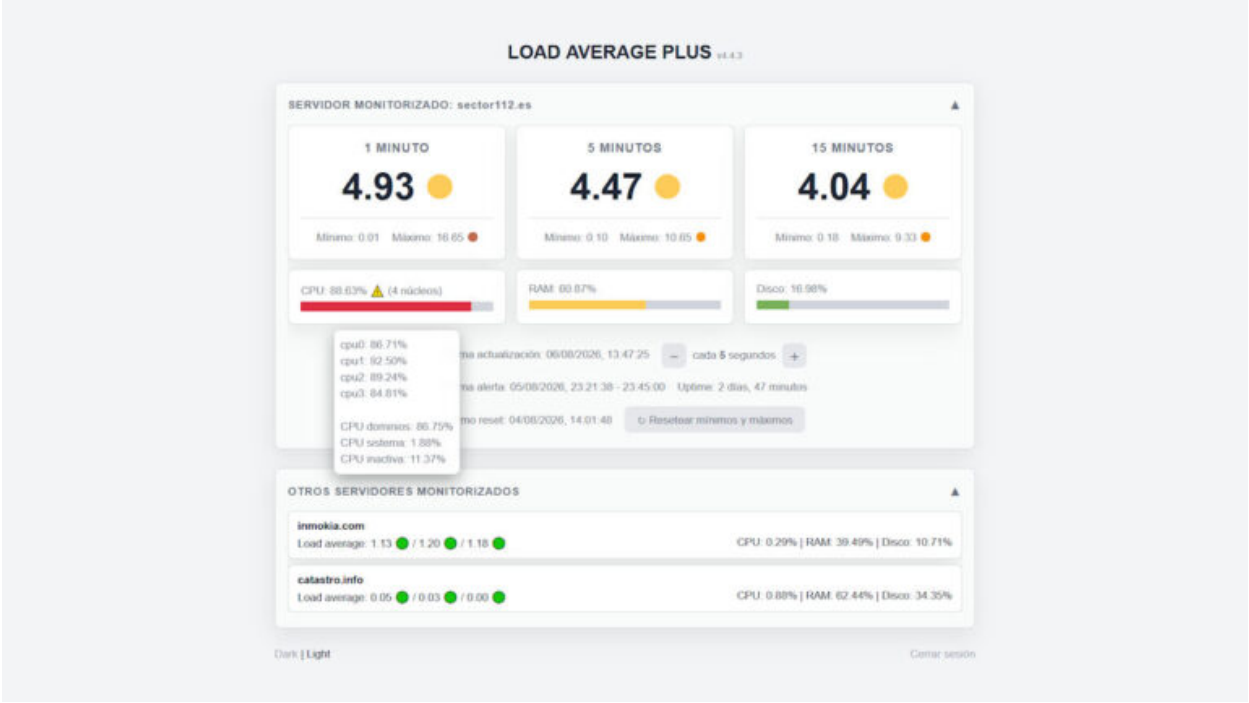
¿Tu servidor web se estresa a menudo?



¿Tu servidor se estresa? Con Load Average Plus monitorizas CPU, RAM, disco y red en tiempo real sin bases de datos y con alertas por email instantáneas.

1. Introducción: El Problema de la «Sobre-Monitorización»

En el ecosistema actual de administración de sistemas, hemos caído en la trampa de la complejidad. Las herramientas de monitorización modernas suelen ser titanes de software que requieren sus propias bases de datos relacionales, agentes pesados en cada nodo y un consumo de recursos que, paradójicamente, a veces supera al de las aplicaciones que intentan vigilar. Para el desarrollador o administrador saturado de herramientas «bloated», **LOAD AVERAGE PLUS** surge como una antítesis elegante. Se trata de un script monolítico

en PHP que recupera la esencia del minimalismo técnico, demostrando que es posible obtener una visibilidad profunda y proactiva del servidor con una huella de recursos prácticamente inexistente.

2. El Poder del Monolito: Un Solo Archivo para Dominarlo Todo

Resulta disruptivo encontrarse con una pieza de software donde un único archivo index.php gestiona con solvencia métricas de bajo nivel, autenticación, alertas y una interfaz visual. Esta arquitectura reduce drásticamente los puntos de fallo: al no depender de una base de datos externa, el sistema es inmune a las corrupciones de tablas que suelen cegar a otras herramientas durante una crisis de I/O.

Como arquitectos, apreciamos los detalles de robustez «bajo el capó». El script no solo persiste los datos en un load.json, sino que utiliza una estrategia de fichero temporal + rename para garantizar una escritura atómica. Además, en su primera ejecución, se auto-protege generando un archivo .htaccess que bloquea el acceso web a sus archivos de datos y logs. Una lección de inteligencia pura es su desglose de procesos: mediante la lectura de /proc, el script distingue entre carga del sistema y de los dominios simplemente filtrando por UID ≥ 1000, el mismo criterio de eficiencia que utiliza la lista de procesos de Plesk.

«No requiere base de datos ni dependencias externas: un único archivo index.php concentra la lógica de monitorización, autenticación, alertas por correo y la interfaz web».

3. El Sistema de Alertas en «Tres Actos»: Más allá del Spam

Una de las decisiones de diseño más brillantes de este sistema es su ciclo de vida de alertas, estructurado para ofrecer contexto en lugar de ruido. La lógica de detección no es estática, sino que se adapta dinámicamente al hardware disponible mediante la variable $nCPUs. El sistema gestiona una narrativa completa en tres fases:

* Acto 1 (Alerta): Se activa si se cumple cualquiera de estas tres

condiciones de saturación: 1) Load 1, 5 y 15 min > $nCPUs × 2; 2) Load 1 y 5 min > $nCPUs × 3; o 3) Load 1 min > $nCPUs × 4.
* Acto 2 (Recuperación): Informa cuando todos los valores de Load Average vuelven a ser ≤ $nCPUs, proporcionando la duración exacta del incidente.
* Acto 3 (Análisis): El sistema espera exactamente 15 minutos tras la recuperación para enviar un informe técnico detallado.

Este retardo de 15 minutos está alineado matemáticamente con la variable $historyWindowSeconds (900s). Esta pausa permite que el histórico de métricas se estabilice para ofrecer un análisis post-mortem con perspectiva, entregando datos útiles cuando el administrador ya tiene la calma necesaria para investigar la causa raíz.

4. Inteligencia Artificial… sin la «IA»: El Post-Mortem Automatizado

LOAD AVERAGE PLUS demuestra que no se necesitan algoritmos complejos para realizar diagnósticos precisos. Mediante el uso de reglas deterministas aplicadas sobre la ventana de 900 segundos, el script es capaz de identificar la naturaleza del problema con una claridad sorprendente.

Tras un incidente, el sistema evalúa los siguientes criterios para «adivinar» la causa:

* Saturación de cómputo: Uso de CPU o RAM ≥ 90%.
* Agotamiento de memoria: Uso de Swap ≥ 50%.
* Estrés de almacenamiento: Ocupación del disco ≥ 90%.
* Latencia de red: Variaciones significativas en el ping hacia el host externo 1.1.1.1.
* Cuellos de botella de I/O: Load Average elevado coexistiendo con un uso de CPU bajo.
* Huecos de monitorización: Detección de silencios en la recolección de datos mediante el cálculo de $gapToleranceFactor × intervalo, lo cual suele ser el síntoma definitivo de un bloqueo del kernel o «freeze» del sistema.

5. La Magia de la Autodetección y el Ecosistema Multi-Servidor

La versión 4.4.3 elimina la fricción en el despliegue. Gracias al uso de $_SERVER['SCRIPT_NAME'], el script detecta automáticamente su ubicación, facilitando su instalación uniforme en cualquier flota de servidores. Para la gestión centralizada, el administrador puede supervisar múltiples nodos desde una sola interfaz mediante una API REST protegida por una $apiKey única.

«Desde la versión actual, el script detecta automáticamente la carpeta donde está instalado… ya no es necesario configurar manualmente las rutas».

Esta capacidad permite realizar consultas en paralelo mediante curl_multi. Es escalabilidad horizontal en su forma más pura: monitorizar una flota entera sin instalar agentes pesados, simplemente saltando de un servidor a otro mediante el login automático que proporciona la clave compartida.

6. Conclusión: El Futuro es Ligero

LOAD AVERAGE PLUS es un recordatorio de que la eficiencia técnica y la claridad visual no tienen por qué ser esclavas de la complejidad. Al centrarse en lo que realmente importa —métricas precisas, alertas con contexto y un consumo de recursos despreciable—, ofrece una robustez que muchas soluciones empresariales envidiarían.

Como profesionales de sistemas, debemos hacernos una pregunta provocadora: para nuestros proyectos actuales, ¿realmente necesitamos desplegar una infraestructura de monitorización masiva que requiere mantenimiento constante, o seríamos más resilientes adoptando un enfoque minimalista y autocontenido de un solo archivo?

Si quieres ver LOAD AVERAGE PLUS en acción o probarlo en tu servidor contacta para solicitarlo sin compromiso. Te ayudará a evitar el stress de tu servidor y los usuarios de tu web lo agradecerán.

LAP ha sido puesto a prueba en un caso real

Semanas después, un servidor con esta herramienta empezó a ir lento sin causa aparente. El diagnóstico llevó cuatro días y el culpable no era el que parecía: la investigación completa, con cifras de antes y después, está en tu servidor no está siendo atacado aunque lo parezca.

Si te ha gustado este PDF, compártelo. Si no te ha gustado, díselo a Andy.