

Watch:  [Soluciones en la industria: StatsD, Datadog, InfluxDB, Prometheus...](#)

08:31

StatsD

[StatsD](#) es un estándar y, por extensión, un conjunto de herramientas que se pueden usar para enviar, recopilar y agregar métricas personalizadas desde cualquier aplicación. Originalmente, StatsD se refería a un demonio escrito por Etsy en Node.js.

Un sistema StatsD requiere tres componentes: un agente, un backend y una UI .

- El **cliente** es una biblioteca que se invoca dentro del código de su aplicación para enviar métricas. Estas métricas son recopiladas por el agente StatsD (a veces también llamado demonio).
- El **servidor** agrega estas métricas, luego envía los agregados a uno o más backends a intervalos regulares.

- **Graphite** es un backend de uso común que le permite ver gráficos en tiempo real de sus métricas. Los componentes de StatsD son modulares, por lo que se pueden agregar, eliminar o reemplazar diferentes implementaciones sin afectar el resto del sistema.

InfluxDB

[InfluxDB](#) es una base de datos de *timeseries* (TSDB) desarrollada por InfluxData. Está escrito en Go y optimizado para el almacenamiento rápido y de alta disponibilidad y la recuperación de métricas de aplicaciones. También tiene soporte para procesar datos de **Graphite**.

La plataforma **InfluxData 1.x** incluye los siguientes componentes de código abierto (se llama **TICK** stack):

- **Telegraf**: recopilar datos
- **InfluxDB**: almacenar datos
- **Chronograf**: visualizar datos
- **Kapacitor**: procesar datos y alertas

La plataforma **InfluxDB 2.0** consolida los componentes de la plataforma InfluxData 1.x en una única solución empaquetada, con características y flexibilidad adicionales:

- **InfluxDB 2.0 beta**: solución de plataforma de código abierto en un solo binario

Datadog

[Datadog](#) es un servicio de monitorización para aplicaciones en la nube, que proporciona monitorización de servidores, bases de datos, herramientas y servicios, a través de una plataforma de análisis de datos SaaS.

Datadog utiliza un agente basado en lenguaje Go, reescrito completamente desde la versión 6. Anteriormente estaba escrito en lenguaje Python (era un fork de StatsD).

Características del Servicio (SaaS):

- Dashboard personalizables.
- Alertas basadas en problemas críticos.
- Soporte para más de 250 integraciones de productos.
- Puede recopilar y analizar automáticamente registros, latencia y tasas de error.
- Permite el acceso a la API.
- Admite aplicaciones escritas en lenguajes como Java, Python, PHP, .NET, Go, Node y Ruby.

Prometheus

[Prometheus](#) es un Sistema de monitorización y alerta open-source construido originalmente por SoundCloud en 2012. Ha sido adoptado por muchas organizaciones y tiene una comunidad activa de desarrolladores y usuarios.

Es un proyecto standalone, mantenido independientemente por cada compañía. Funciona bien para registrar cualquier serie de tiempo puramente numérica (time series data), tanto monitorización centrada en la máquina como monitorización de arquitecturas dinámicas basadas en servicios. En el mundo de los microservicios, soporta colecciones de datos multidimensionales y la consulta de esta información.

Provee un modelo de datos multidimensional y un poderoso lenguaje de consulta (PromQL). Soporta autodetección para cualquier dispositivo de la infraestructura. Recolecta información de más de 5000 métricas automáticamente, con cero configuración, cero dependencias y cero mantenimiento. Las librerías de Prometheus ofrecen cuatro tipos de métricas: contador (counter), calibrador (gauge), histograma (histogram) y resumen (summary).

Características Principales:

- **Modelo de datos multidimensional:** con datos de series de tiempo identificados por nombre métrico y claves/ valores pares.
- **Lenguaje flexible de consultas:** permite aprovechar esta multidimensionalidad para consultar los datos de manera simple y flexible
- **Nodos autónomos de servidor único:** debido a la poca confianza en el almacenamiento distribuido, la colección de series de tiempo ocurre a través de un modelo de extracción, por medio de HTTP.
- **Configuración estática:** los objetivos se determinan mediante configuración estática o descubrimiento de servicios.