



**UTT**

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

**GOBIERNO DE BAJA CALIFORNIA**

**TEMA**

Software Monitoring

**PRESENTADO POR**

Perez Bello Vanory Esperanza

**GRUPO**

10° B

**MATERIA**

Software Development Process

**PROFESOR**

Ray Brunett Parra Galaviz

Tijuana, Baja California, 24 Enero del 2025

## Software Monitoring

El monitoreo de software implica recopilar, procesar y analizar datos generados por un sistema en funcionamiento. Estos datos incluyen métricas de rendimiento, registros de eventos (logs), errores, transacciones y otros indicadores clave de rendimiento (KPI). Su propósito principal es ofrecer visibilidad sobre cómo está operando el sistema, identificar problemas antes de que afecten a los usuarios finales y optimizar el rendimiento del software.

### Componentes principales

#### 1. Métricas:

Datos cuantificables sobre el rendimiento, como:

- Uso de CPU, memoria y ancho de banda.
- Latencia y tiempo de respuesta.
- Número de solicitudes por segundo o tasa de errores.

#### 2. Logs (registros):

- Detalles de eventos que ocurren dentro de la aplicación o sistema.
- Útiles para identificar la causa raíz de errores y analizar tendencias.

#### 3. Alertas:

- Notificaciones configuradas para detectar eventos o métricas fuera de los valores esperados.
- Permiten respuestas rápidas a incidentes críticos.

#### 4. Dashboards:

- Interfaces visuales que consolidan métricas y logs en tiempo real.
- Facilitan la comprensión del estado general del sistema y el análisis de datos históricos.

## **Tipos de monitoreo**

1. **Monitoreo de infraestructura:** Supervisión de servidores, bases de datos y redes para garantizar la estabilidad.
2. **Monitoreo de aplicaciones (APM):** Enfoque en el rendimiento y la funcionalidad de aplicaciones específicas.
3. **Monitoreo de experiencia del usuario (UEM):** Análisis de cómo los usuarios interactúan con el software para identificar problemas desde su perspectiva.
4. **Tracing distribuido:** Seguimiento de solicitudes a través de múltiples servicios en arquitecturas de microservicios.

## **Técnicas de monitoreo**

- **Monitoreo con agentes:** Software instalado en los sistemas que recopila datos detallados.
- **Monitoreo sin agentes:** Usa protocolos estándar como SNMP para supervisar de manera remota.
- **Monitoreo basado en eventos:** Se enfoca en los cambios importantes dentro del sistema.

## **Beneficios clave**

- **Prevención de problemas:** Detecta anomalías antes de que afecten a los usuarios finales.
- **Optimización del rendimiento:** Identifica cuellos de botella y optimiza recursos.
- **Mayor confiabilidad:** Mejora la disponibilidad del software mediante la resolución proactiva de fallas.
- **Soporte en la toma de decisiones:** Proporciona datos en tiempo real y tendencias históricas para mejorar procesos y planeaciones.

## Desafíos del monitoreo

- **Manejo de grandes volúmenes de datos:** En sistemas modernos, la cantidad de datos generados puede ser inmensa.
- **Alertas excesivas:** Una configuración inadecuada puede generar "fatiga de alertas".
- **Integración compleja:** Especialmente en sistemas heterogéneos o arquitecturas distribuidas.

## Herramientas de monitoreo populares

- **Prometheus y Grafana:** Open-source, ideales para métricas y visualización.
- **Datadog y New Relic:** Soluciones comerciales que integran múltiples funciones como monitoreo, trazabilidad y análisis de logs.
- **ELK Stack** (Elasticsearch, Logstash, Kibana): Ideal para el análisis de logs y visualización de datos.

## Importancia del Software Monitoring

El monitoreo de software no solo garantiza la operatividad del sistema, sino que también permite a los equipos de desarrollo e infraestructura:

- Mejorar la experiencia del usuario final.
- Cumplir acuerdos de nivel de servicio (SLA).
- Reducir el tiempo de inactividad.
- Aumentar la eficiencia operativa al optimizar recursos tecnológicos.

## Referencias Bibliograficas

Brazil, B. (2020). *Monitoring and observability with Prometheus*. O'Reilly Media.  
ISBN: 978-1098119183

Beyer, B., Murphy, N. R., Rensin, D. K., Kawahara, K., & Thorne, S.  
(2018). *The site reliability workbook: Practical ways to implement SRE*. O'Reilly  
Media.

ISBN: 978-1492029646

Banon, S., Harris, C., & Tong, Z. (2015). *Elasticsearch: The definitive guide: A  
distributed real-time search and analytics engine*. O'Reilly Media.  
ISBN: 978-1449358546

Sridharan, C. (2020). *Distributed systems observability: A guide to building  
robust systems*. O'Reilly Media.