TECNOLÓGICA DEL ORIENTE
INSTITUCIÓN DE EDUCACIÓN SUPERIOR

SISTEMAS DISTRIBUTIVOS

SISTEMAS DISTRIBUIDOS BASADOS EN CLIENTE-SERVIDOR

SISTEMAS DISTRIBUIDOS BASADOS EN MIDDLEWARE

Los sistemas distribuidos basados en middleware representan una arquitectura que utiliza una capa intermedia de software encargada de facilitar la comunicación, coordinación y gestión de recursos entre los distintos nodos de un sistema distribuido. Esta capa de middleware oculta la complejidad de la infraestructura distribuida, permitiendo que los desarrolladores se concentren en la lógica de negocio sin preocuparse por detalles como la ubicación de los recursos, las diferencias de hardware o las heterogeneidades de red (Celaya Luna, 2014).

¿Qué es el middleware en sistemas distribuidos?

El middleware actúa como un "pegamento lógico" que conecta aplicaciones y servicios distribuidos, gestionando tareas como la comunicación de mensajes, la invocación de métodos remotos, la gestión de transacciones, la seguridad y la sincronización de procesos (Urbano López, 2015).

Esta capa permite lograr transparencia en la distribución, facilitando el acceso a recursos distribuidos como si se encontraran en el mismo entorno local, lo que reduce la complejidad de las aplicaciones distribuidas.

Características principales

- **Transparencia de ubicación:** los usuarios y aplicaciones interactúan con servicios distribuidos sin conocer su ubicación física.
- **Interoperabilidad:** permite la integración entre sistemas heterogéneos, funcionando entre distintos sistemas operativos y tecnologías.
- **Gestión de comunicación:** simplifica la comunicación entre nodos, gestionando peticiones y respuestas de forma confiable.
- **Servicios adicionales:** provee seguridad, manejo de errores, control de concurrencia y transacciones distribuidas.

Ejemplos de middleware en sistemas distribuidos

- **CORBA (Common Object Request Broker Architecture):** facilita la invocación de métodos en objetos distribuidos escritos en distintos lenguajes de programación, sin preocuparse por su ubicación.
- **RMI (Remote Method Invocation) de Java:** permite a las aplicaciones Java llamar a métodos de objetos que se encuentran en otras máquinas virtuales de Java en la red.
- **Middleware de colas de mensajes (MQ):** como RabbitMQ o Apache Kafka, facilita la comunicación asíncrona entre componentes distribuidos, gestionando grandes volúmenes de mensajes de forma confiable.
- **gRPC:** un sistema moderno de invocación de procedimientos remotos (RPC) que permite la comunicación eficiente entre servicios distribuidos.
- **Enterprise Service Bus (ESB):** como MuleSoft o WSO2, que permiten integrar múltiples servicios y aplicaciones a través de una arquitectura orientada a servicios (SOA).

Ventajas de utilizar middleware

- Facilita el desarrollo de aplicaciones distribuidas al abstraer detalles complejos de comunicación y sincronización.
- Permite escalar servicios de manera organizada en sistemas distribuidos.
- Mejora la portabilidad y reutilización de componentes en diferentes plataformas.
- Permite la integración de sistemas legados con aplicaciones modernas.

Ejemplo práctico

Una empresa de comercio electrónico que distribuye sus servicios entre diferentes servidores, puede utilizar un middleware de colas de mensajes (como RabbitMQ) para procesar pedidos de forma asíncrona: cuando un cliente realiza una compra, el pedido se coloca en una cola, permitiendo que diferentes microservicios lo procesen en paralelo (pago, inventario y envío) sin bloquear el sistema, mejorando así la escalabilidad y eficiencia del servicio.