

05/02/2024

Desarrollo De Aplicaciones Distribuidas

Investigación:

Middleware y su evolución

Integrantes:

Perea Valdez Leonardo Alberto



Concepto:

El middleware es un software que actúa como una capa de abstracción entre el software de aplicación y la infraestructura subyacente, como sistemas operativos, hardware, redes y bases de datos.

Su objetivo es ocultar la complejidad de las interacciones entre sistemas, proporcionando servicios comunes y facilitando la comunicación entre aplicaciones distribuidas.

La evolución del middleware ha pasado por varias etapas:

- Década de 1980: Surgieron los primeros tipos de middleware como TCP/IP, SQL y RPC.
- Década de 1990: Se desarrollaron middleware más sofisticados como CORBA, DCOM de Microsoft y Java RMI.
- Década de 2000: Surgieron arquitecturas orientadas a servicios (SOA) y middleware empresarial como J2EE, .NET y middleware de mensajería como IBM WebSphere.
- Década de 2010-actualidad: Ha evolucionado hacia servicios en la nube, microservicios, contenedores, DevOps y middleware de IoT.

El middleware moderno proporciona funcionalidades como administración de transacciones, mensajería, seguridad, integración, orquestación, caché, monitoreo y gestión de API, facilitando el desarrollo de aplicaciones escalables y distribuidas.

Antecedentes:

El middleware nació como una solución para abordar los desafíos de la computación distribuida en la década de 1980. A medida que las redes de computadoras y los sistemas heterogéneos se volvieron más comunes, surgió la necesidad de una capa intermediaria que facilitara la comunicación y la interoperabilidad entre diferentes aplicaciones, sistemas operativos, hardware y bases de datos.

Los primeros tipos de middleware, como TCP/IP y SQL, sentaron las bases para la comunicación a través de redes y el acceso a bases de datos relacionales. Sin embargo, estos protocolos y APIs eran bastante primitivos y no proporcionaban una abstracción suficiente para ocultar la complejidad de la distribución.

En la década de 1990, surgieron middleware más sofisticados, como CORBA (Common Object Request Broker Architecture), que proporcionaba un estándar para la comunicación entre objetos distribuidos en diferentes lenguajes de programación y plataformas. Microsoft también introdujo DCOM (Distributed Component Object Model) y Sun Microsystems lanzó Java RMI (Remote Method Invocation).

Estos middleware utilizaban técnicas como el enrutamiento de solicitudes, la conversión de datos y la transparencia de ubicación para hacer que la distribución pareciera más como una aplicación

monolítica. Sin embargo, todavía eran complejos de configurar y mantener, y a menudo tenían problemas de escalabilidad y rendimiento.

A medida que las aplicaciones empresariales se volvieron más grandes y complejas, surgieron middleware más especializados en la década de 2000, como J2EE (Java Enterprise Edition) y .NET de Microsoft. Estos proporcionaban un conjunto integrado de servicios y componentes para el desarrollo de aplicaciones distribuidas a gran escala.

En la actualidad, el middleware ha evolucionado para admitir arquitecturas modernas como microservicios, contenedores y computación en la nube. También se ha extendido a nuevos dominios, como el Internet de las cosas (IoT), donde se utiliza para facilitar la comunicación entre dispositivos heterogéneos y sistemas back-end.