

Middleware

¿Qué es un middleware y para qué sirve?

Un *middleware* es un software que permite que las aplicaciones de un sistema se comuniquen entre sí y, también, con otros paquetes de software, con el sistema operativo y con elementos hardware. Es, por tanto, un software que se encarga de conectar entre sí otro software para que éstos interactúen cuando se necesite, produciéndose un traspaso de datos de uno a otro.

Uno de los modelos más conocidos de middleware es el que se encuentra entre un sistema operativo y las aplicaciones que operan sobre él.

Algunas de las gestiones de las que se encarga un middleware son: gestión de datos, servicios de aplicaciones, mensajes, procesamiento de transacciones, autenticación y gestión de API's.

¿Cómo funciona?

Un middleware funciona como una red de comunicaciones internas entre el sistema central y las aplicaciones distribuidas. De esta forma, un middleware hace de traductor automático de instrucciones garantizando que los diferentes elementos del sistema se entiendan y puedan operar entre sí.

El proceso consiste en una solicitud que se envía de un sistema a otro a la que sigue una respuesta gracias a la traducción realizada por el middleware.



Pensemos en una persona que quiere iniciar sesión en su cuenta dentro de una tienda online (como Amazon o Zara). En este caso el middleware reacciona cuando la persona introduce su usuario y contraseña y traspasa estos datos hasta la base de datos de clientes de la tienda. De esta forma se coteja y se autentica el acceso a su espacio de cliente.

Para que pueda cumplir su función de traductor y permita la comunicación entre distintas aplicaciones, el middleware hace uso de distintos marcos de comunicación (*frameworks*) como:

- *Representational State Transfer* (REST, Transferencia de Estado Representacional);
- *Web Services* (Servicio web);
- *JavaScript Object Notation* (JSON, notación de objeto de JavaScript);
- *Simple Object Access Protocol* (SOAP, protocolo de acceso a objetos simples);
- *Enterprise Service Bus* (ESB, bus de servicios de empresa);
- *Application Programming Interface* (API, interfaz de programación de aplicaciones).

Cuanto más compleja sea la conexión entre los sistemas, mayor número de middlewares harán falta entre un sistema y el otro.

Atención: muchas veces se confunde lo que es un middleware con lo que es una API. Una API es una de las herramientas de las que se sirve un middleware para conseguir que 2 plataformas interactúen entre sí.

Tipos de Middleware

El tipo de middleware que elija cada empresa dependerá del servicio que se va a usar y del tipo de información que deba comunicarse. Algunas de las situaciones y entornos más comunes son: administración de transacciones, autenticación de seguridad, servidores de aplicaciones, colas de mensajes, servidores web y directorios.

Los dos tipos principales de middleware son los de integración y los de aplicación. Dentro de cada tipo hay, a su vez, varios subtipos.

Middleware de integración

Proporcionan servicios para las comunicaciones y la integración entre aplicaciones. Dentro de los middlewares de integración se encuentran:

- **Orientados a procedimientos o procesos:** este tipo de middleware se comunica de forma sincronizada y sus elementos principales son el *stub* y el *skeleton*. El *stub* manda una petición al servidor de forma remota (ya sea preguntar por un dato u ordenar que haga una tarea). El *skeleton* recibe el mensaje y lo convierte en una petición y se comunica con la aplicación del servidor para que se procese dicha petición. Una vez procesada la petición, el *skeleton* convierte el mensaje en una respuesta y contesta al *stub*.
- **Orientados a objetos:** este tipo de middleware permite que haya pedidos de objetos distribuidos. Los objetos distribuidos son unidades dentro de un programa informático que tienen un estado y un comportamiento, y se encuentran distribuidos en diferentes espacios de direcciones. Una estructura común de base de datos es:
 - Clase 1 (por ejemplo, gato)

- Objeto 1 (por ejemplo, "Garfield")
 - Atributo 1 (por ejemplo, color naranja)
 - Atributo 2 (por ejemplo, tamaño grande)
 - Atributo 3 (por ejemplo, peso elevado)

Estos objetos distribuidos son gestionados por un servidor (servidor de base de datos) y por los sistemas conectados a ese servidor que recibe una petición remota. Debido a esta petición remota, este tipo de middleware permite que la comunicación pueda ser: sincronizada (en tiempo real), sincronizada diferida (se sincroniza la comunicación en un momento posterior a aquel en el que se está programando) o asíncrona. Además, los middlewares orientados a objetos son capaces de soportar varios pedidos similares realizados por diferentes clientes durante una misma transacción. Este middleware funciona de la siguiente forma: primero el stub reúne la información y se la comunica al agente (*broker*). El agente es un mediador y se comunica con distintas fuentes de información para conseguir sus IDs (*identifications* – códigos únicos de las fuentes), recopilar la información y reorganizarla para mandársela al skeleton. El skeleton, a su vez, se la envía al objeto servidor. Una vez que el objeto servidor proporciona una respuesta, se le hace llegar la misma al cliente siguiendo los pasos inversos.

- **Orientados a mensajes** (MOM, *Message-oriented middleware*): este middleware hace posible que las aplicaciones distribuidas logren comunicarse a través de mensajes. El middleware MOM es el responsable de que los mensajes lleguen a su destino. No obstante, esta comunicación se realiza de forma asíncrona ya que las aplicaciones no están conectadas directamente. El MOM puede ser de punto a punto (solo hay un receptor del mensaje), o de publicación/suscripción (en el que hay 2 actores importantes: el emisor que envía la información y el suscriptor o publicador que guarda el mensaje en la memoria).

- **Orientados a componentes:** este tipo de middleware actúa como un configurador de componentes. Un componente es una pieza de software ejecutable de forma independiente, con una interfaz propia y que puede interoperar con otros componentes. El middleware orientado a componentes puede configurarse incluso mientras se está ejecutando. Esta particularidad lo convierte en un middleware muy flexible que puede ser programado para interactuar con diferentes funciones y aplicaciones de un componente.
- **Agentes:** el middleware agente puede realizar distintas funciones y estar compuesto por distintos elementos, como objetos o procesos; canales para comunicarse y leyes que identifican los distintos tipos de agentes. Esto se debe a que un agente es un programa que actúa realizando una acción que le ha sido solicitada por un usuario o por otro programa. Por ejemplo, la transferencia de cierta información de un programa a otro. Por tanto, el middleware agente debe estar compuesto de elementos que le permita llevar a cabo la acción que le demanda la persona o programa.

Middleware de aplicación

Proporcionan servicios entre el sistema operativo y las diferentes aplicaciones que se ejecutan en dicho sistema. Los middlewares de aplicación son especialmente diseñados según la tipología de la aplicación que necesita comunicarse con el sistema operativo.

Dentro de los middlewares de integración se encuentran:

- **Middleware para acceso a información** (DAM, *Data Access middleware*): este tipo de middleware trabaja con distintas fuentes de datos al mismo tiempo. Por eso, este middleware es el que se dedica a procesar las transacciones, y a gestionar las puertas de entradas (*gateways*) de las bases de datos y de los diferentes softwares distribuidos según el procedimiento o la transacción establecidos.
- **Middleware de escritorio:** este tipo de middleware puede realizar las siguientes funciones:

- variar las formas de presentar la información solicitada al usuario;
- controlar los servicios de transporte de datos;
- generar y recuperar copias de seguridad;
- gestionar la información de la base de datos;
- gestionar los procesos;
- gestionar las instalaciones de software;
- llevar a cabo servicios de cifrado;
- control de acceso;
- entre otras funciones. Por ejemplo, el middleware puede recuperar una copia de seguridad de datos importantes en caso de que se hayan perdido (durante una migración u otro proceso).

• **Middlewares basados en la web:** este middleware se encarga de asistir al usuario mientras navega por la web. Hace uso de interfaces para localizar cuáles son los posibles sitios de interés para el usuario, así como detectar si éste ha cambiado de gustos (basándose en su historial de búsqueda). Este middleware se basa en los historiales web, en las aplicaciones y en las comunicaciones de procesos del sistema operativo. Algunos servicios importantes del middleware basado en la web son los de correo electrónico, los accesos remotos a información y la descarga de archivos, entre otros.

• **Middleware a tiempo real:** este middleware puede planificar las solicitudes del usuario para que sean contestadas en tiempo real o en otro momento. Este tipo de middleware es sensible al tiempo y a las necesidades del momento. Esto significa que la información que se proporciona puede ser correcta en un instante, pero no en otro. Por ejemplo, si un sistema pide saber la cantidad de productos vendidos en un día, no pondrá la misma cantidad a las 9 que a las 21 horas. Por eso, la cantidad que se le indicaría a las 9 de la mañana ya no es correcta a las 21. Los middlewares en tiempo real pueden ser de varios tipos:

- aplicación de base de datos en tiempo real;

- sensor de procesamiento de los datos;
- transmisión de la información solicitada y obtenida.