MÓDULO DESPLIEGUE DE APLICACIONES WEB

TEMA 3

Administración de servidores de aplicaciones

CICLO DE GRADO SUPERIOR INFÓRMATICA

DESARROLLO DE APLICACIONES WEB

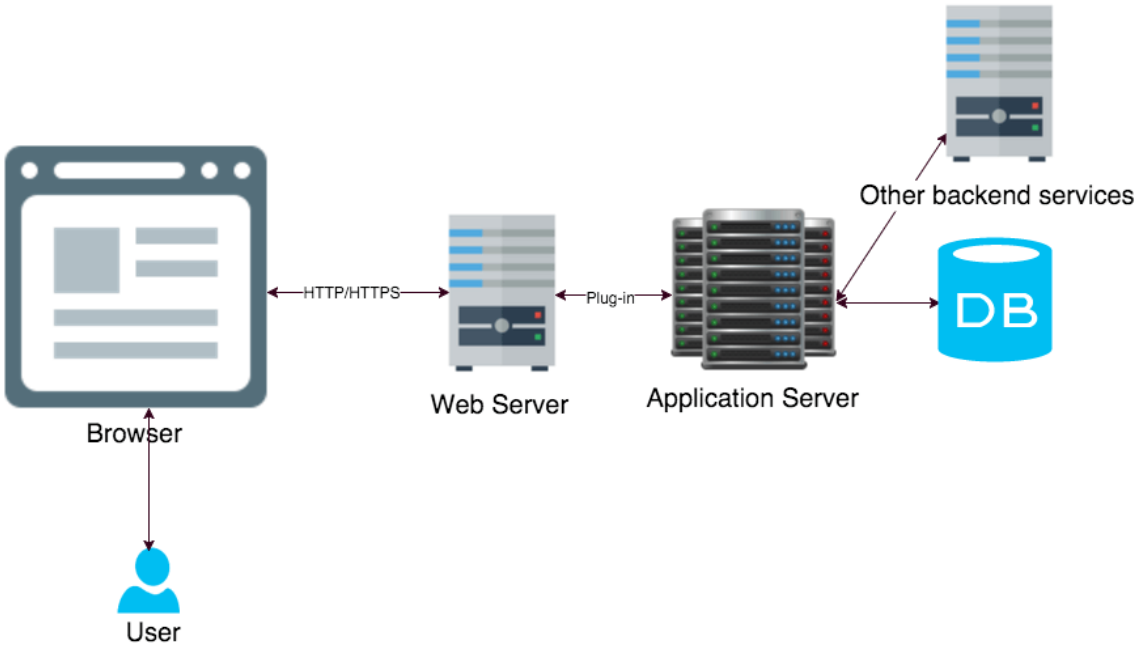
Índice

1. Introducción

Un servidor de aplicaciones permite la **creación de aplicaciones web** y en general proporciona un entorno de servidor para ejecutarlas.

A menudo puede ser una pila compleja de diferentes elementos computacionales que ejecutan tareas específicas que necesitan trabajar como uno solo para alimentar múltiples nubes y software y aplicaciones basadas en la web.

Situado **entre el servidor web** y el nivel de backend del **servidor de bases de datos**, el servidor de aplicaciones es esencialmente un intermediario para el servidor de bases de datos y los usuarios de las aplicaciones empresariales o de consumo que soporta mediante el uso de varios protocolos e interfaces de programación de aplicaciones (API).



Es habitual que se **utilice junto con un servidor web** o que contenga un servidor web, por lo que ambos pueden converger y denominarse **servidor de aplicaciones web**.

Los servidores de aplicaciones también pueden contener sus propias interfaces gráficas de usuario para su gestión a través de PC, pero también pueden ocuparse de sus propios recursos, así como del procesamiento de transacciones, la mensajería, la agrupación de recursos y conexiones, y la realización de tareas de seguridad.

2. Servidor de aplicaciones

Los servidores de aplicaciones son los ordenadores de gran potencia que proporcionan recursos de aplicaciones a los usuarios y clientes web.

Los servidores de aplicaciones, como ya hemos dicho, se sitúan física o virtualmente entre los servidores de bases de datos que almacenan los datos de las aplicaciones y los servidores web que se comunican con los clientes. Los servidores de aplicaciones y el middleware afín son los sistemas operativos que soportan el desarrollo y la entrega de una aplicación. Ya sea una aplicación de escritorio, móvil o web, los servidores de aplicaciones desempeñan un papel fundamental en la conexión de un mundo de dispositivos.

Cuando los usuarios de las aplicaciones, ya sea usuarios físicos o los clientes web, solicitan acceso a una aplicación, el servidor de aplicaciones suele hacer el trabajo pesado en el backend para almacenar y procesar las solicitudes dinámicas de las aplicaciones.

A la hora de trabajar con servidores de aplicaciones, es posible que tengamos que utilizar diferentes términos que se repiten con cierta frecuencia:

- **Servidor web:** visto en anteriores unidades, es responsable de almacenar, procesar y entregar los datos de E/S de las páginas web.
- **Cliente web:** punto final que intenta acceder a los recursos de la web o de la aplicación
- **Protocolo de comunicación entre el servidor web y los clientes web:** conjunto de reglas que define cómo los datos deben intercambiarse entre un cliente (como un navegador web o una aplicación móvil) y un servidor web. A destacar: HTTP/HTTPS, REST, SOAP o WebSockets.
- **Lenguaje para el intercambio entre los servidores web y de aplicaciones:** formatos que utilizan los protocolos anteriormente nombrados (JSON, XML, HTML, …)
- **Lógica de negocio:** reglas para el almacenamiento de datos y la transferencia de recursos de la aplicación.
- **Aplicación:** programa o un sitio web unido a una base de datos.

Para conseguir una agilidad óptima del servidor web, no sirve gestionar tanto las peticiones HTTP de los clientes web como pasar o almacenar recursos de múltiples sitios web. Los servidores de aplicaciones llenan este vacío con un diseño de alta potencia construido para **manejar las solicitudes de contenido web dinámico**.

Los servidores de aplicaciones también **proporcionan redundancia de programas y una capa adicional de seguridad**. Una vez desplegado entre una base de datos y un servidor web, el trabajo de preservar y duplicar la arquitectura de la aplicación a través de la red es más factible. Dado que los servidores de aplicaciones pueden procesar solicitudes de lógica empresarial, un intento de inyección SQL es también mucho más difícil.

Las organizaciones pueden proteger aún más sus datos con un servidor proxy inverso colocado delante de sus bases de datos. Los servidores proxy y las VPN

pueden hacer maravillas para anonimizar y encriptar la comunicación para proteger a los usuarios y los datos de la empresa.

2.1 Necesidad de un servidor de aplicaciones

Un servidor de aplicaciones es necesario porque **proporciona un entorno especializado y optimizado para ejecutar aplicaciones empresariales y web**, manejando tareas clave que de otro modo serían complejas o difíciles de gestionar directamente en un servidor web.

Alguna de sus principales funcionalidades son las siguientes:

Gestión de la Lógica de Negocio: un servidor de aplicaciones facilita la gestión de transacciones distribuidas y garantiza que todas las operaciones en una aplicación se realicen de manera coherente, segura y sin errores. Esto es esencial para aplicaciones empresariales donde varias operaciones deben completarse con éxito o, en caso de fallos, revertirse. Organiza la lógica de negocio en capas, separando la presentación (front-end), la lógica de negocio y la persistencia (base de datos), lo que mejora la escalabilidad, el mantenimiento y la reutilización del código.

- **Escalabilidad:** los servidores de aplicaciones gestionan de forma eficiente la carga, permitiendo que una aplicación crezca sin problemas al manejar más usuarios simultáneamente mediante características como el balanceo de carga.
- **Alta Disponibilidad:** ofrecen características como clustering, que permite que la aplicación se ejecute en varios servidores, asegurando que la aplicación esté disponible incluso si uno de los servidores falla.
- **Gestión de la Seguridad:** autenticación y autorización: Los servidores de aplicaciones proporcionan mecanismos de seguridad como autenticación (verificación de identidad) y autorización (control de acceso), para garantizar que solo los usuarios autorizados puedan acceder a determinadas funcionalidades de la aplicación. Cifrado y manejo de datos sensibles: Permiten el cifrado de comunicaciones (por ejemplo, usando HTTPS), protegiendo datos sensibles durante la transmisión entre el cliente y el servidor.
- **Integración con Otros Sistemas:** los servidores de aplicaciones son compatibles con varios protocolos (comentados anteriormente). También poseen capacidades para integrar fácilmente las aplicaciones con bases de datos.
- **Facilidad de Desarrollo y Mantenimiento:** los servidores de aplicaciones permiten a los desarrolladores crear aplicaciones sin preocuparse por detalles como la gestión de la memoria, las conexiones a bases de datos o el manejo de transacciones. Permiten la automatización de ciertas tareas.

- **Rendimiento y Optimización:** muchos servidores de aplicaciones proporcionan mecanismos de optimización como la caché de resultados y el pooling de conexiones, lo que mejora el rendimiento de las aplicaciones, especialmente en aplicaciones con alta carga.

2.2. Funcionamiento de servidor de aplicaciones

Como la mayoría de los servidores de hoy en día, los servidores de aplicaciones contienen características de seguridad, transacciones, servicios, clustering, diagnósticos y bases de datos. En lo que se diferencian los servidores de aplicaciones es en su capacidad para procesar peticiones de servlets (programas Java) desde un servidor web.

En la imagen anterior, se muestra el flujo general de los servidores de aplicaciones web:

1. El cliente abre un navegador y solicita acceso a un sitio web.
2. El servidor web recibe la petición HTTP y responde con la página web deseada
3. El servidor web gestiona las peticiones de datos estáticos, pero el cliente quiere utilizar una herramienta interactiva.
4. Al tratarse de una petición de datos dinámicos, el servidor web transfiere la petición a un servidor de aplicaciones.
5. El servidor de aplicaciones recibe la petición HTTP y la convierte en una petición de servlet.
6. El servlet llega al servidor de la base de datos, y el servidor de aplicaciones recibe una respuesta del servlet.
7. El servidor de aplicaciones traduce la respuesta del servlet al formato HTTP para el acceso del cliente.

Al recibir una solicitud de servlet de un servidor web, el servidor de aplicaciones procesa la solicitud y responde al servidor web mediante la respuesta de servlet. Dado que los servidores de aplicaciones trabajan principalmente con peticiones

de lógica de negocio, el servidor web traduce la respuesta del servlet y pasa una respuesta HTTP accesible para el usuario.

Con los contenidos vistos hasta el momento ya es posible realizar una diferenciación entre servidor de aplicaciones y servidor web.

	Servidor de aplicaciones	**Servidor web**
Utilidad	Sirve peticiones HTTP y de otra lógica de negocio	Sirve peticiones HTTP
Almacena y proporciona	Lógica de negocio	Contenido web estático
La utilización de los recursos es	Pesada	Ligera

2.3. Casos reales de servidores de aplicaciones en entornos profesionales

En entornos reales de desarrollo y despliegue de aplicaciones web, los servidores de aplicaciones no son un concepto abstracto, sino herramientas concretas ampliamente utilizadas en empresas, centros de datos y plataformas cloud. A continuación se describen algunos casos reales habituales.

Apache Tomcat en aplicaciones Java web

Uno de los servidores de aplicaciones más utilizados es Apache Tomcat, especialmente en aplicaciones desarrolladas en Java. Tomcat actúa como contenedor de servlets y permite ejecutar aplicaciones web basadas en tecnologías como JSP y Servlets.

En un escenario típico podemos encontrarnos una situación como la que se describe a continuación:

- Un servidor web (Apache o Nginx) gestiona las peticiones HTTP iniciales.
- Las peticiones dinámicas se redirigen a Tomcat, donde se ejecuta la lógica de negocio.
- Tomcat se comunica con una base de datos (MySQL, PostgreSQL, etc.).
- El resultado se devuelve al servidor web y finalmente al cliente.

Este tipo de arquitectura es muy habitual en aplicaciones corporativas, plataformas educativas, intranets empresariales y sistemas de gestión desarrollados en Java.

WildFly / JBoss en entornos empresariales

En aplicaciones de mayor complejidad se utilizan servidores de aplicaciones más completos como WildFly (antes JBoss). Estos servidores ofrecen gestión avanzada de transacciones, seguridad integrada, clustering y alta disponibilidad.

Son frecuentes en grandes empresas, aplicaciones críticas y sistemas con múltiples usuarios concurrentes.

Servidores de aplicaciones basados en Node.js

En arquitecturas modernas, especialmente con aplicaciones JavaScript, se utilizan entornos basados en Node.js. En este caso, el propio proceso Node.js actúa como servidor de aplicaciones, gestionando peticiones HTTP, ejecutando la lógica de negocio y conectándose con bases de datos y servicios externos.

Este tipo de servidores de aplicaciones es muy común en APIs REST, aplicaciones SPA y arquitecturas de microservicios.

Servidores de aplicaciones comerciales

En grandes organizaciones también es habitual el uso de servidores de aplicaciones comerciales como Oracle WebLogic o IBM WebSphere. Estos entornos están pensados para sistemas críticos, aplicaciones financieras y grandes volúmenes de usuarios y datos.

Relación con el despliegue en la nube

Actualmente, estos servidores de aplicaciones se despliegan con frecuencia en infraestructuras cloud. El servidor de aplicaciones se instala en una instancia, se configura el acceso por puertos, se gestiona su arranque y parada como servicio y se integra con medidas de seguridad y escalabilidad.