*fuente:* [*IBM*](https://www.ibm.com/es-es/topics/web-server-application-server)

**Servidor web frente a servidor de aplicaciones: ¿en qué se diferencian?**

Por definición, un servidor web es un subconjunto común de un servidor de aplicaciones.

Los servidores web ofrecen contenido web estático (por ejemplo, páginas HTML, archivos, imágenes, vídeo), principalmente en respuesta a solicitudes de protocolo de transferencia de hipertexto (HTTP) de un navegador web.

Los servidores de aplicaciones normalmente también pueden ofrecer contenido web, pero su tarea principal es habilitar la interacción entre los clientes de usuario final y el código de aplicación del lado del servidor (el código que representa lo que a menudo se denomina *lógica empresarial*) para generar y entregar contenido dinámico, como resultados de transacciones, soporte de decisiones o análisis en tiempo real. El cliente de un servidor de aplicaciones puede ser la propia interfaz de usuario final de la aplicación, un navegador web o una aplicación móvil, y la interacción cliente-servidor puede producirse a través de diversos protocolos de comunicación.

En la práctica, sin embargo, se ha desdibujado la línea entre los servidores web y los servidores de aplicaciones, sobre todo porque el navegador web ha surgido como el cliente de aplicaciones preferido, y porqué han aumentado las expectativas de los usuarios y el rendimiento de las aplicaciones web.

**Servidores de aplicaciones y servidores web de código abierto**

El mercado está inundado de servidores web y servidores de aplicaciones, demasiados para nombrarlos aquí. En lugar de ello, hemos pensado que puede ser más útil nombrar las opciones disponibles de código abierto más populares:

Nginx

[Nginx](https://nginx.org/en/) es un servidor web de código abierto que incluye funciones de proxy inverso, equilibrio de carga, proxy de correo y memoria caché HTTP. También hay disponibles versiones comerciales y compatibles de Nginx en [Nginx, Inc.](https://www.nginx.com/) (enlace externo a ibm.com). De acuerdo con la compañía de investigación de internet y prevención de ciberdelincuencia [Netcraft](https://www.netcraft.com/) (enlace externo a ibm.com), en diciembre de 2019 Nginx era el servidor o proxy de casi el 38 % de todos los sitios web del mundo y de más del 25 % del millón de sitios más ocupados. Algunas empresas usuarias de Nginx conocidas a nivel mundial: Dropbox, Netflix y Zynga.

Apache HTTP Server

Apache HTTP Server, que se publicó por primera vez en 1995, es otro servidor web gratuito de código abierto, muy popular, que hasta hace muy poco servía como base a más sitios web que cualquier otro servidor web —llegó a alcanzar el 71 % en su punto álgido—, antes de ser superado por Nginx en abril de 2019. En diciembre de 2019, Apache era el servidor de más del 24 % de todos los sitios del mundo y del 31 % del millón de sitios más ocupados.

Apache Tomcat

[Apache Tomcat](https://tomcat.apache.org/) es un servidor de aplicaciones de código abierto que ejecuta servlets Java, representa y entrega páginas web que incluyen código JavaServer Page y actúa como servidor para aplicaciones Java Enterprise Edition (Java EE). Tomcat se publicó en 1998 y es el servidor de aplicaciones [Java](https://www.ibm.com/es-es/topics/java) de código abierto más utilizado.

Glassfish

[Glassfish](https://javaee.github.io/glassfish/) es un servidor de aplicaciones Java EE de código abierto lanzado por Sun Microsystems en 2006 y alojado en la actualidad por [Eclipse Foundation](https://www.eclipse.org/org/foundation/) (enlace externo a ibm.com). Al igual que la mayoría de los servidores de aplicaciones Java, Glassfish admite servlets Java, Enterprise JavaBeans (EJB) y más, pero también puede funcionar como servidor web y dar servicio a contenido web en respuesta a solicitudes HTTP.

*fuente:* [*Ionos*](https://www.ionos.es/digitalguide/servidores/know-how/servidor-de-aplicaciones/)

**¿Cómo funciona un servidor de aplicaciones?**

Para comprender mejor el funcionamiento general de un servidor de aplicaciones, es importante entender primero su lugar dentro de una red. Se sitúa entre los clientes individuales y la base de datos. Recibe peticiones HTTP y responde a ellas, pero también puede emitir peticiones servlet y procesar las respuestas correspondientes. Esta es una de las principales diferencias entre un servidor de aplicaciones y un servidor web. En nuestra comparación entre servidores web y servidores de aplicaciones profundizamos en las diferencias entre ambos modelos. Un servidor de aplicaciones funciona junto con un servidor web de la siguiente manera:

1. Un usuario quiere visitar una página web. Abre un navegador y navega hasta la página web.
2. La petición HTTP se envía al servidor web, que procesa la petición y devuelve al cliente la página web solicitada. Esto funciona siempre que sean peticiones estáticas.
3. Si en su lugar se utiliza una herramienta interactiva, el procedimiento es el mismo, pero el servidor web no emite la respuesta directamente. En su lugar, se reenvía al servidor de aplicaciones.
4. El servidor de aplicaciones recibe la petición HTTP y la convierte en una petición servlet.
5. Se reenvía a la base de datos.
6. El servidor de la base de datos procesa la petición y envía una respuesta servlet de vuelta al servidor de aplicaciones.
7. En el último paso, el servidor de aplicaciones envía el formato del servlet al servidor web, que convierte la respuesta a HTTP y la pone a disposición del cliente.

**¿Qué ventajas tienen los servidores de aplicaciones?**

Un servidor de aplicaciones ofrece numerosas ventajas. Entre ellas están:

Compatibilidad

Al instalar un servidor de aplicaciones en tu red, te aseguras de que todos los clientes trabajan con la misma versión de software. Como resultado, todos los datos intercambiados y almacenados son compatibles. Si renuncias a un servidor de aplicaciones, pueden surgir problemas por falta de actualizaciones o por software obsoleto.

Seguridad

La implantación de un servidor de aplicaciones puede resultar ventajosa por cuestiones de seguridad. Por ejemplo, impide una conexión directa entre una página web y su propia base de datos. A menudo se pueden evitar ataques de malware como los programas de inyección SQL.

Rendimiento

Los servidores de aplicaciones pueden mejorar el rendimiento general. Las mejoras de rendimiento son especialmente notables en los programas de gran tamaño y/o de uso frecuente. Las opciones de escalado pueden ayudar a mejorar el rendimiento, así como a agrupar y controlar el acceso a la base de datos.

Configuración

La centralización a través de un servidor de aplicaciones implica que la configuración no tiene que realizarse en cada dispositivo individual. Esto simplifica la configuración y ahorra tiempo y recursos.

Escalabilidad

Gracias a las numerosas posibilidades de conexión con la base de datos, es posible conseguir una mayor escalabilidad. Esto también garantiza una mayor flexibilidad para los usuarios.

¿Qué inconvenientes tienen los servidores de aplicaciones?¶

Sin embargo, cuando se recurre a un servidor de aplicaciones también existen algunos inconvenientes. Merece la pena mencionar los siguientes:

Costes adicionales

La instalación y el mantenimiento de un servidor de aplicaciones cuesta dinero y tiempo. Debes sopesar detenidamente si la implantación es necesaria. Si es así, hay que tener en cuenta los gastos.

Posibles retrasos

Pueden producirse retrasos durante la implantación de un servidor de aplicaciones. Esto es especialmente habitual si la instalación se realiza detrás de un firewall.

Programación exigente

Los requisitos de programación son mayores para un servidor de aplicaciones. Por tanto, es conveniente tener un poco de experiencia y conocimientos previos.

Ancho de banda

Cuando se utilizan simultáneamente varias aplicaciones grandes y varios usuarios acceden a ellas, esto puede tener un impacto negativo en la velocidad y el rendimiento general.

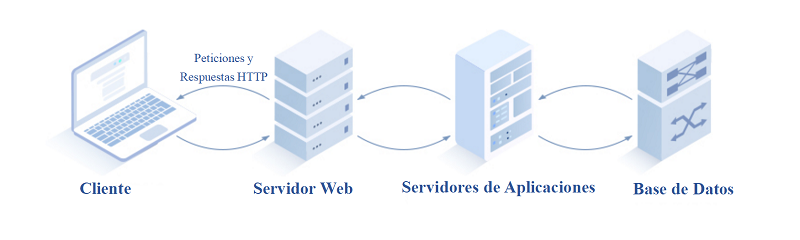
Impacto de los problemas y errores

Aunque la agrupación es una de las ventajas de utilizar un servidor de aplicaciones, los problemas y fallos son mucho más graves como consecuencia de ello. Dado que numerosos clientes dependen de una única fuente, los problemas con un software afectan a varios usuarios. Las posibles soluciones a estos problemas suelen ser un poco más complicadas.

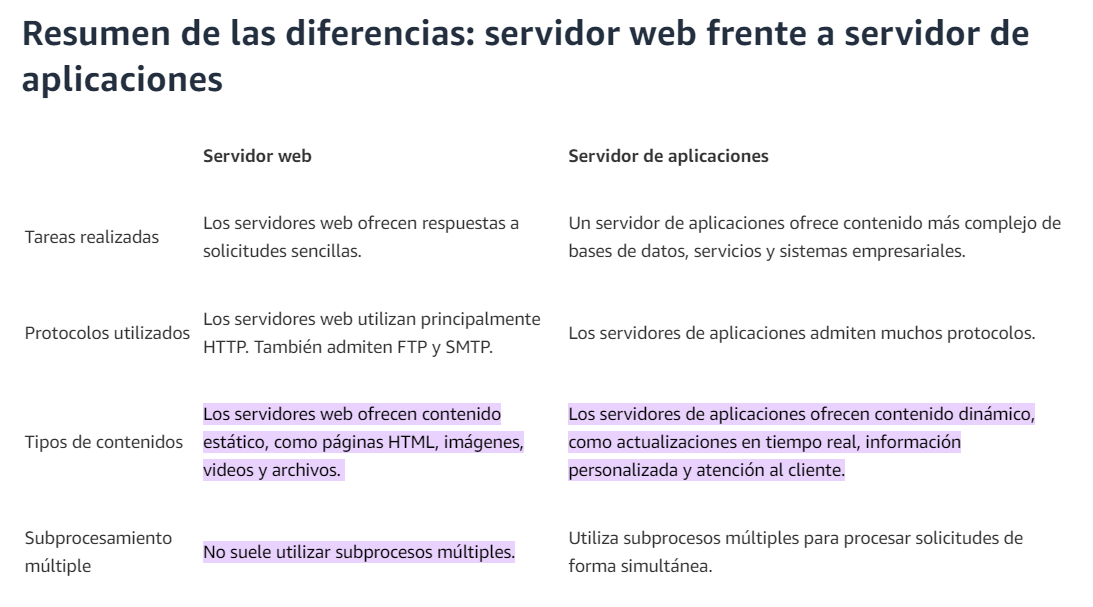
*fuente:* [*Amazon*](https://aws.amazon.com/es/compare/the-difference-between-web-server-and-application-server/#:~:text=Los%20servidores%20web%20ofrecen%20contenido,personalizada%20y%20atenci%C3%B3n%20al%20cliente.&text=No%20suele%20utilizar%20subprocesos%20m%C3%BAltiples.)

**¿Cómo funcionan juntos los servidores de aplicaciones y los servidores web?**

Los servidores de aplicaciones y los servidores web trabajan juntos para administrar las solicitudes de los clientes y ofrecer el contenido correcto al usuario. El servidor web siempre recibe primero una nueva solicitud. Si puede producir la información por sí mismo, lo hace y envía una respuesta HTTP. También comprueba que los datos que el usuario solicitó no estén ya en su caché.



Si el servidor web no puede acceder al contenido que el usuario necesita, reenvía la solicitud al servidor de aplicaciones. El servidor de aplicaciones procesa los datos y utiliza la lógica empresarial para proporcionar la información correcta. A continuación, devuelve la solicitud al servidor web, que la pasa al usuario. En ciertas arquitecturas, también puede configurar los servidores de aplicaciones para que gestionen las solicitudes HTTP por sí mismos.

****