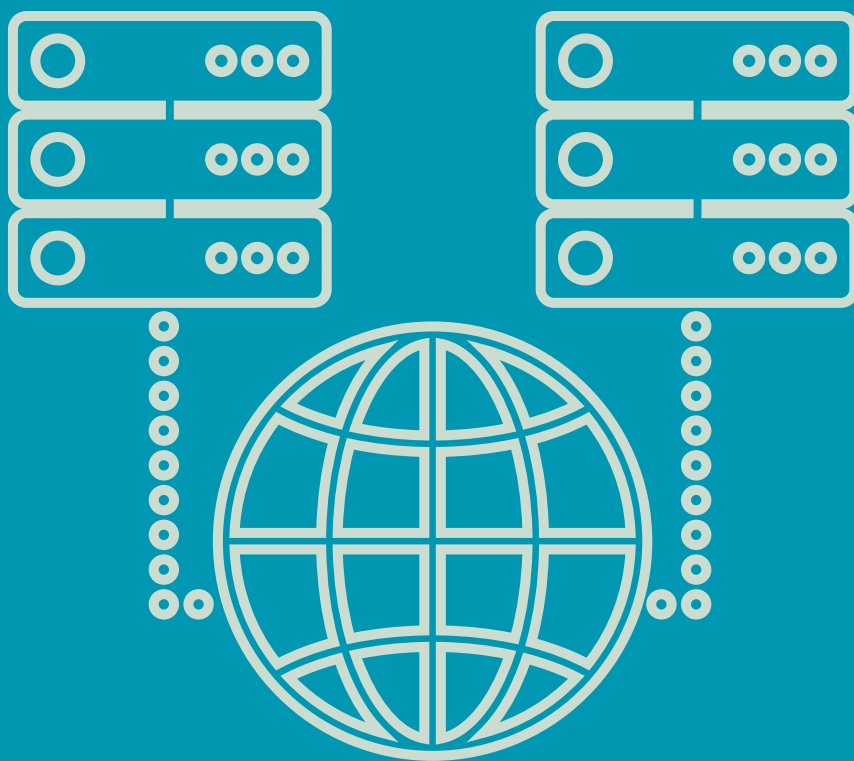


# SERVIDORES WEB



Alejandra Rubio  
Maria Domingo  
Candela Martinez  
Jingxiang Wang

## Introducción

Los servidores web son programas informáticos instalados en equipos conectados a Internet que, a través del protocolo HTTP, entregan contenido en este formato para que pueda ser visualizado desde cualquier navegador. Se basa en que cuando un usuario introduce una URL válida en el navegador, la solicitud es enviada al servidor web. Este, procesa la petición y devuelve el recurso solicitado, como una página web, un archivo o cualquier otro contenido almacenado en el servidor.

Por lo general, estos servidores están alojados en ordenadores de gran capacidad, diseñados para estar siempre en funcionamiento.

Estos servidores web permiten la existencia de hosting, que permite alquilar espacio en un servidor remoto, por ejemplo, para alojar una página web.

Actualmente, los servidores web más populares son Apache, Microsoft IIS, Nginx, Lighttpd y Apache Tomcat.

### 1) Historia y evolución de servidor web

El primer servidor web de la historia se remonta al año 1990, cuando Tim Berners-Lee creó el software CERN HTTPd como parte del proyecto WorldWideWeb en el CERN. Este software permitía a los usuarios acceder a páginas web a través de internet y marcaría el inicio de una nueva era en la comunicación y la información.

Con el paso del tiempo, surgieron nuevos servidores web que ofrecían mayor capacidad, velocidad y seguridad. Uno de los hitos más significativos fue la creación de Apache HTTP Server en 1995, que se convertiría en uno de los servidores web más populares y utilizados en todo el mundo.

Posteriormente, el lanzamiento de Microsoft Internet Information Services (IIS) en 1996 proporcionó a los usuarios de Windows una plataforma nativa para alojar sus sitios web. Esta diversificación en los servidores web contribuyó a la expansión de internet y al aumento de la presencia online de empresas y organizaciones.

En paralelo, el surgimiento de servidores web ligeros y eficientes como Nginx en 2004 revolucionó el rendimiento y la escalabilidad de las aplicaciones web. Su arquitectura modular y su capacidad para manejar grandes cargas de tráfico lo han convertido en una opción popular para sitios con alto volumen de visitas.

## **2) Plataformas en las que se puede utilizar**

Los servidores web se pueden ejecutar en sistemas operativos como Windows, Linux o macOS con diferentes tecnologías y lenguajes de programación. Los lenguajes más usados actualmente son PHP, Java, Ruby o Python que se usan para crear y entregar contenido web

Apache, Nginx, Microsoft IIS y LiteSpeed son algunos de los servidores web más populares del mundo. Un servidor web es uno de los pilares fundamentales de la arquitectura e infraestructura de la navegación en internet. Todos, en conjunto, alojan y entregan contenido web a los más de 5000 millones de usuarios de internet a nivel mundial.

## **3) Lenguajes de programación que soporta**

Los servidores web están diseñados para soportar una amplia variedad de lenguajes de programación. Esto permite que los desarrolladores elijan el lenguaje que mejor se adapte a sus necesidades y a las características del proyecto. A continuación, se detallan algunos de los lenguajes de programación más populares en el desarrollo web:

**PHP:** Uno de los lenguajes más utilizados para la creación de sitios web dinámicos. PHP es muy popular debido a su facilidad de uso, extensa documentación y compatibilidad con la mayoría de los servidores web. Además, es la base de muchos sistemas de gestión de contenidos (CMS) como WordPress, Joomla y Drupal.

**Java:** Es un lenguaje de programación robusto, seguro y de alto rendimiento, ampliamente utilizado en aplicaciones empresariales de gran escala. Java es compatible con diversos frameworks, como Spring o Hibernate, que facilitan la creación de aplicaciones web complejas y escalables.

C#: Este lenguaje fue desarrollado por Microsoft y es especialmente útil cuando se trabaja con el framework .NET. C# se utiliza comúnmente para crear aplicaciones web en entornos de empresas que usan tecnología Microsoft. Al combinarse con ASP.NET, permite desarrollar aplicaciones web rápidas y seguras.

Python: ha ganado popularidad en el desarrollo web gracias a su simplicidad y la potencia de frameworks como Django y Flask. Es ideal para construir aplicaciones web rápidas y eficientes.

Ruby: es conocido por su sintaxis sencilla y fácil de leer, y su popularidad ha crecido con el framework Ruby on Rails, que facilita el desarrollo rápido de aplicaciones web. Es muy utilizado para startups y proyectos que requieren una alta velocidad de desarrollo.

JavaScript (Node.js): Tradicionalmente, JavaScript era un lenguaje usado únicamente en el lado del cliente (navegador), pero con la llegada de Node.js, ha ganado terreno como lenguaje de backend.

Node.js permite crear aplicaciones web completas utilizando JavaScript tanto en el servidor como en el cliente, lo que lo convierte en una opción atractiva para desarrolladores que desean trabajar con un solo lenguaje en toda la pila tecnológica.

#### **4) Integración de control de versiones**

El control de versiones es una herramienta esencial para cualquier equipo de desarrollo que trabaje en proyectos web. Permite gestionar y rastrear los cambios en el código fuente a lo largo del tiempo, facilitando la colaboración entre múltiples desarrolladores, la prevención de errores y la recuperación de versiones anteriores del proyecto en caso de fallos.

Los sistemas de control de versiones más populares incluyen Git y Subversion (SVN), cada uno con sus propias características. Git es el más utilizado en la actualidad, y su flexibilidad y soporte para el trabajo distribuido lo han convertido en el estándar de facto para la gestión de código en proyectos de desarrollo de software.

## **Beneficios del control de versiones:**

- **Historial de cambios:** Los sistemas de control de versiones permiten almacenar todas las versiones anteriores del código, lo que facilita la revisión del historial de cambios, la restauración de versiones anteriores o la comparación entre diferentes versiones del proyecto.
- **Recuperación de errores:** Si se comete un error o se elimina accidentalmente parte del código, el control de versiones permite revertir esos cambios fácilmente, restaurando el estado anterior del proyecto.
- **Colaboración en equipo:** Facilita el trabajo colaborativo, ya que varios desarrolladores pueden trabajar en diferentes partes del mismo proyecto sin sobrescribir el trabajo de otros. Además, las herramientas de control de versiones gestionan los conflictos que pueden surgir cuando varios desarrolladores realizan cambios en las mismas líneas de código.
- **Ramas y fusiones:** Los sistemas de control de versiones permiten crear ramas para trabajar en nuevas características sin afectar la versión principal del proyecto. Posteriormente, se pueden fusionar los cambios de manera controlada y segura.

## **Herramientas para la integración del control de versiones:**

- **GitHub:** Es una de las plataformas más populares para alojar repositorios Git. Permite a los desarrolladores gestionar sus proyectos, colaborar en equipo y mantener un registro detallado del historial de cambios. GitHub también ofrece características como solicitudes de extracción (pull requests), donde los colaboradores pueden sugerir cambios y los mantenedores del proyecto pueden revisarlos antes de integrarlos.

**GitLab:** Similar a GitHub, GitLab es una plataforma para la gestión de repositorios Git, con la ventaja adicional de ser de código abierto. También incluye herramientas para la integración continua y el despliegue continuo (CI/CD), lo que facilita la automatización de pruebas y despliegues.

**Bitbucket:** Es otra plataforma de control de versiones que soporta Git y Mercurial. Bitbucket se integra bien con herramientas como Jira, lo que lo convierte en una opción preferida para equipos que ya utilizan el ecosistema de Atlassian para la gestión de proyectos.

## 5) Reconocimiento de sintaxis. Extensiones y componentes para la tecnología.

El reconocimiento de sintaxis es la capacidad de un servidor web pueda analizar o validar el código de programación como (como HTML, JavaScript, CSS, PHP, etc.)

### 1. Según el Lenguaje de Programación

- PHP: Utiliza el motor Zend, que tiene su propio analizador sintáctico incorporado.
- Python: Usa librerías como pygments o el módulo tokenize para el reconocimiento de sintaxis.
- Java: Con librerías como ANTLR o JavaCC para construir analizadores de sintaxis personalizados.

### 2. Motores y Frameworks de Servidor

- Apache
- NGINX: Utiliza extensiones para interpretar lenguajes como PHP a través de FastCGI y otras integraciones.
- Node.js: Utiliza su propio motor V8 para reconocer y ejecutar la sintaxis de JavaScript.

### 3. Plugins o Extensiones de IDE/Editor para Servidor

- JetBrains WebStorm, PHPStorm, IntelliJ: Proporcionan reconocimiento de sintaxis a través de sus motores internos.

### 4. Componentes para Preprocesadores y Compiladores

- Babel: Para transpilar JavaScript moderno (ES6+) a una versión más antigua reconocida por navegadores.
- Sass/LESS: Preprocesadores de CSS que reconocen su propia sintaxis y la compilan a CSS estándar.
- Webpack: Un empaquetador de módulos que usa loaders para reconocer la sintaxis de distintos lenguajes y transpilarlos a JavaScript o CSS.

## 5. Servicios de Análisis en la Nube

- SonarQube: Herramienta de análisis de código que reconoce la sintaxis de múltiples lenguajes para encontrar problemas y mejorar la calidad del código.
- CodeClimate: Proporciona análisis estático y reconocimiento de sintaxis.

## 6) Frameworks que tienen y cuáles son los más populares.

Son las Plataformas que simplifica el desarrollo de aplicaciones web, proporcionando herramientas y bibliotecas para la creación de un servidor. A continuación son algunos de los más populares, clasificados por lenguajes de programación:

### 1. Java

Spring Boot: Uno de los frameworks más utilizados en Java para crear aplicaciones web. Simplifica la configuración de Spring y proporciona un servidor embebido como Tomcat para iniciar rápidamente.

Jakarta EE (antiguamente Java EE): Un conjunto robusto de especificaciones para crear aplicaciones empresariales en Java. Incluye componentes como Servlets, JSP, EJB, JPA, entre otros.

### 2. JavaScript/Node.js

Express.js: Un framework minimalista y flexible para Node.js, conocido por ser ligero y modular. Es muy utilizado en la construcción de APIs y aplicaciones web.

NestJS: Un framework de Node.js progresivo que está inspirado en Angular, utilizado para crear aplicaciones del lado del servidor con TypeScript.

### 3. Python

Flask: Un microframework ligero que proporciona lo esencial para levantar un servidor web. Es muy flexible y permite agregar solo los componentes que necesitas.

#### 4. Ruby

Ruby on Rails: Un framework que sigue el principio de "convención sobre configuración", lo que hace que el desarrollo sea rápido y sencillo al proporcionar muchos componentes preconfigurados

.

#### 5. PHP

Laravel: Un framework con una sintaxis elegante y sencilla, que facilita la creación de aplicaciones robustas en PHP. Incluye características como migraciones de base de datos, un ORM llamado Eloquent, y herramientas para autenticación.

### 7) ¿Existen Manual de usuario y de ayuda? Indicar dónde y cuáles podemos utilizar.

Sí, cada servidor web cuenta con manuales de usuario y secciones de ayuda, disponibles en las páginas oficiales de cada uno, generalmente en la sección de documentación. En estos manuales oficiales se explica de manera detallada cómo realizar tareas como la instalación, configuración de opciones, preguntas frecuentes y otros temas relacionados con el uso del servidor.

Estos manuales suelen ser muy completos, aunque la mayoría de ellos están disponibles principalmente en inglés. Además de los manuales oficiales, hay una gran cantidad de recursos adicionales en internet. Si necesitas una explicación más específica o no encuentras la solución en la documentación oficial, puedes recurrir a:

- Otros manuales y guías publicadas en sitios web especializados.
- Foros y comunidades de usuarios que comparten soluciones y consejos prácticos.

Algunas de estas pueden ofrecer documentación o guías en español y con ejemplos para resolver problemas concretos



## 8) Ventajas y desventajas que presentan frente a otros competidores.

### Apache

- Es uno de los servidores web más populares junto con Nginx.
- Es gratuito.
- Soporta más de 1 millón de conexiones.
- Multiplataforma: soporta Linux, Unix, macOS y Windows.
- Ofrece buena flexibilidad y modularidad.

### Microsoft IIS

- Se destaca por su facilidad de administración y seguridad.
- Tiene una interfaz gráfica muy intuitiva, lo que facilita el aprendizaje y uso.
- Las últimas versiones pueden soportar un número ilimitado de conexiones.
- No es multiplataforma, está diseñado principalmente para sistemas Windows.

### Nginx

- Actualmente en 2024, ha superado a Apache en popularidad.
- Utilizado para sitios de alto tráfico, gracias a su eficiencia, estabilidad y bajo consumo de recursos.
- Gracias a su arquitectura de eventos asíncronos, Nginx es más eficiente que Apache al manejar conexiones concurrentes.
- La configuración puede ser más complicada que en Apache.
- Multiplataforma: soporta Linux, Unix, macOS y Windows.

### Lighttpd

- Capacidad para manejar muchas solicitudes simultáneas con bajo consumo de recursos.
- La comunidad de usuarios es pequeña, lo que puede dificultar la búsqueda de soluciones o documentación.
- Posibles problemas de compatibilidad con ciertas aplicaciones y sistemas operativos porque funciona mejor para servidores de alto rendimiento con pocos recursos.
- Multiplataforma.

## **Apache Tomcat**

- Principalmente, es un servidor de aplicaciones Java que puede actuar como un servidor web básico.
- Ideal para trabajar con aplicaciones Java, pero si solo necesitas servir contenido estático, lo mejor es optar por un servidor web como Apache HTTP Server o Nginx.
- Está específicamente diseñado para ejecutar aplicaciones Java.

También es importante fijarse en el rendimiento de cada uno dependiendo si el contenido es estático o dinámico:

Para contenido estático: Nginx y Lighttpd son los más eficientes, con su arquitectura basada en eventos. Apache puede configurarse para ser competitivo, pero requiere ajustes.

Para contenido dinámico: Apache es excelente con PHP y otros lenguajes, IIS sobresale con tecnologías de Microsoft, y Tomcat es líder para aplicaciones Java. Nginx, aunque no procesa contenido dinámico directamente, es excelente como frontend en combinación con otros servidores de aplicaciones.