

PREGUNTAS - TEMA 1

DESARROLLO WEB EN ENTORNO SERVIDOR

25/09/2023

Ismael Ferreras García

Índice

Índice.....	2
1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	3
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....	4
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.	5
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.	6
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.	7
6. Modelo de división funcional front-end / back-end para aplicaciones web.....	8
7. Página web estática – página web dinámica – aplicación web – mashup.....	9
8. Componentes de una aplicación web.	10
9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.....	11
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).	11
11. Características y posibilidades de desarrollo de una plataforma XAMPP.	12
12. En qué casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación.	13
13. IDE más utilizados (características y grado de implantación actual).	14
14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	16
15. Apache HTTP vs Apache Tomcat.....	17
17. Generadores de documentación HTML (PHPDoc):.....	19

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.

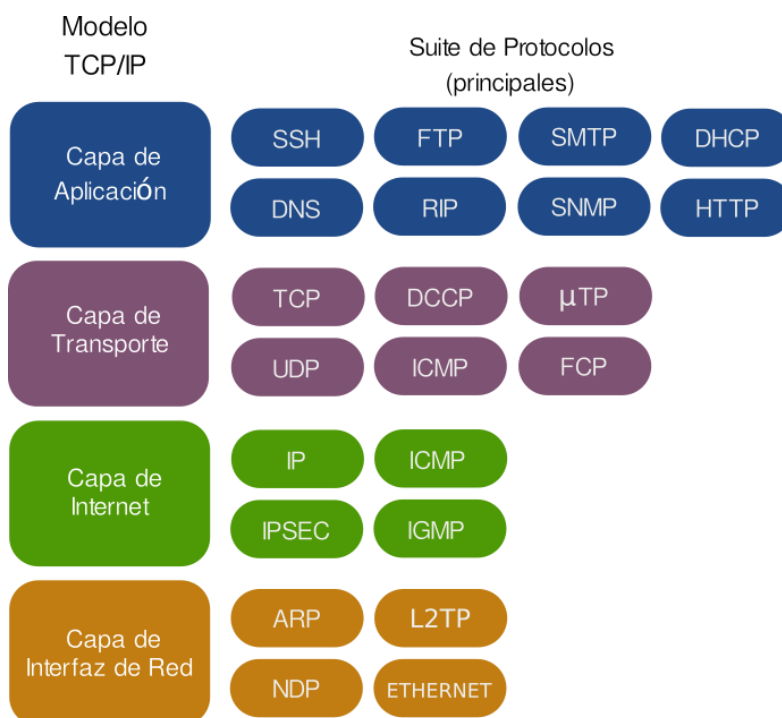
Son los protocolos más usados en internet y necesarios para su funcionamiento ya que es como se comunican los clientes web con los servidores web:

IP (Protocolo de Internet): **IP es el protocolo fundamental que se utiliza en Internet y redes para enrutar los paquetes de datos desde una fuente hasta un destino.** Hay dos versiones principales de IP, IPv4 (la versión que se utiliza ahora mismo) e IPv6 (creado para remplazar a IPv4 ya que no tiene suficiente capacidad).

TCP (Protocolo de Control de Transmisión): **TCP es un protocolo de nivel de transporte que se encarga de establecer y mantener conexiones entre dos dispositivos en una red.** Garantizando que los datos se transmitan sin errores ni pérdidas.

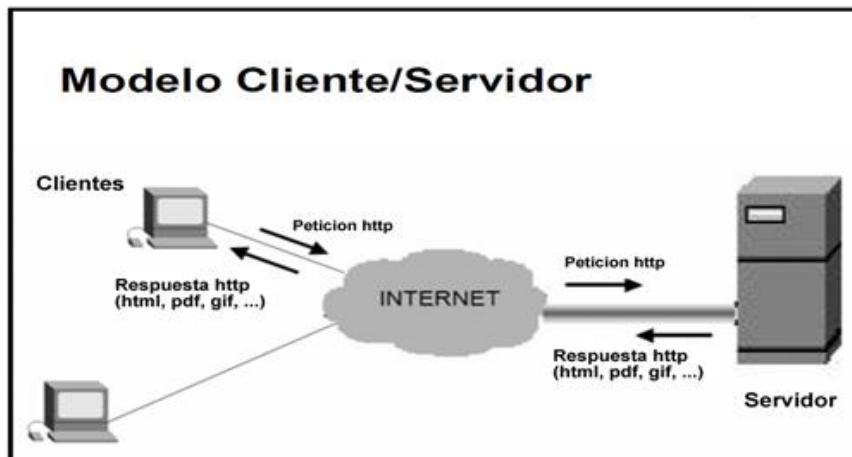
HTTP (Protocolo de Transferencia de Hipertexto): **HTTP es un protocolo de nivel de aplicación que se utiliza para la transferencia de datos en internet.** Usa la estructura **cliente-servidor** entre un cliente (un navegador web) y un servidor web. HTTP es un protocolo sin estado, lo que significa que procesa cada solicitud independientemente de las anteriores. HTTP tiene 3 versiones principalmente HTTP/1, HTTP/2, HTTP/3 El servidor web utiliza el puerto 80 para la comunicación cliente-servidor

HTTPS (Protocolo de Transferencia de Hipertexto Seguro): **HTTPS es una versión segura de HTTP que utiliza cifrado SSL/TLS para proteger la privacidad y la integridad de los datos transmitidos entre el cliente y el servidor.** Actualmente es utilizado en todos los sitios web que utilizan información privada y en la mayoría de los que no. El servidor web utiliza el puerto 443 para la comunicación cliente-servidor



2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web

Es una estructura en la que el cliente (mediante un navegador o aplicación web) hace una petición http/https y el servidor web recibe y procesa la solicitud enviando la respuesta mediante http/https junto al contenido solicitado normalmente en formato HTML. El servidor actúa de forma pasiva nunca comienza la comunicación.

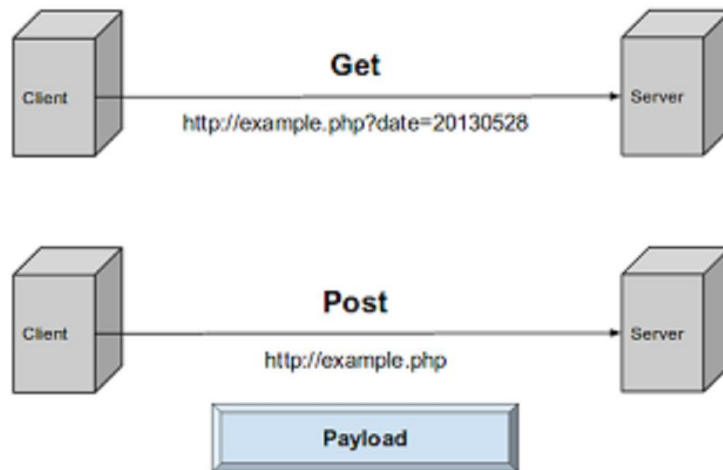


3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.

GET Y POST son los mas utilizados los dos son para enviar la información al servidor su principal diferencia es:

GET: Envía mediante la URL mostrando en ella lo que envía

POST: Envía los datos en segundo plano ocultándoselos a el usuario



HEAD: Pide la respuesta

PUT: Envía o actualiza datos al servidor para que sean procesados

DELETE: Borra el recurso

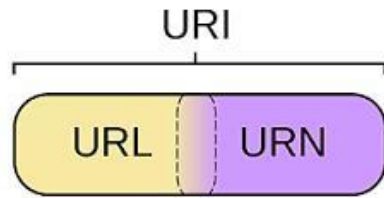
TRACE: Solicita al servidor que envíe de vuelta el mensaje (suele estar deshabilitado por seguridad)

OPTIONS: Devuelve las opciones de conexión HTTP que soporta la URL

CONNECT: Establece una conexión de red con el recurso de destino

4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.

URI es una cadena de caracteres utilizada para identificar un recurso de manera única ya sea en la web como la URL de una página web o fuera de ella como el ISBN de un libro.



Una URL es un tipo de URI para identificar un recurso en la red proporcionando además la ubicación precisa de este y si está disponible



Un URN es otro tipo de URI que se utiliza para identificar un recurso de manera única por su nombre, independientemente de su ubicación. A diferencia de las URL, los URN no indican cómo encontrar el recurso o si está disponible, no es tan usado como la URL.

5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.

PENDIENTE

6. Modelo de división funcional front-end / back-end para aplicaciones web.

Front-End (Interfaz de Usuario para Usuarios Finales):

La parte delantera de la aplicación web a la que acceden los usuarios finales, es decir, aquellos que utilizan la aplicación para su propósito principal.

Los usuarios normales o clientes acceden al front-end para interactuar con la aplicación, ver y manipular datos, realizar transacciones y utilizar las funciones ofrecidas.

Se enfoca en proporcionar una experiencia de usuario intuitiva, atractiva y eficiente.

Generalmente, el front-end se ejecuta en el navegador web del usuario y se encarga de la presentación y la interacción.

El 99% de los que usan la aplicación usan el Front-End

Back-End (Funcionalidades para Usuarios Especiales):

La parte de la aplicación a la que acceden usuarios especiales, como administradores, personal de servicio técnico...

Se utiliza para realizar tareas de gestión, configuración, mantenimiento y supervisión de la aplicación, así como para acceder a funciones y datos que no están disponibles para los usuarios normales.

Puede incluir paneles de administración, herramientas de diagnóstico, funciones de gestión de usuarios...

Se requiere autenticación adicional para acceder al back-end y garantizar que solo los usuarios autorizados puedan utilizar estas funciones especiales.

7. Página web estática – página web dinámica – aplicación web – mashup

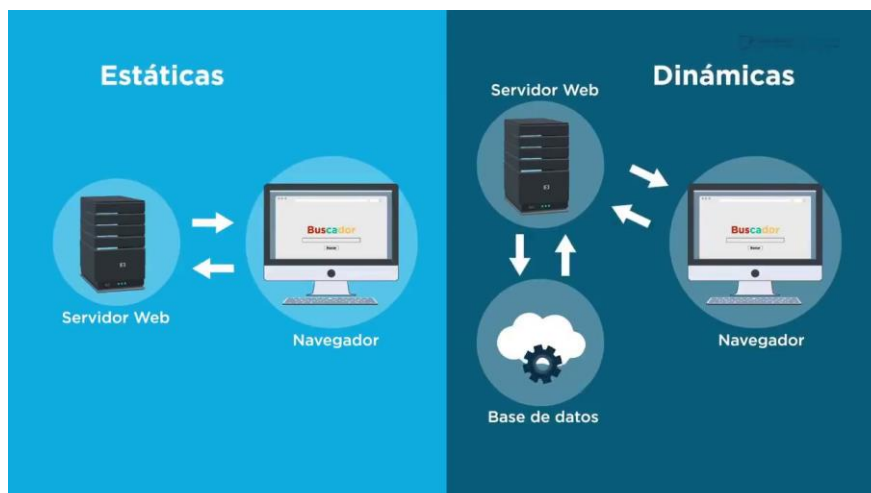
Estos términos se refieren a tipos de sitios web desde una página web estática siendo el más simple y que casi no permite al usuario interactuar hasta una aplicación web o un mashup siendo muy complejos, dinámicos e interactivos con el usuario.

Página web estática: Una página web estática es un sitio web que muestra contenido fijo y no cambia automáticamente según las interacciones del usuario. Por lo general, estas páginas se crean con HTML y CSS, y su contenido se mantiene sin cambios hasta que el propietario del sitio web lo actualiza manualmente. Suelen ser utilizadas para sitios web simples, como páginas de información o portafolios en línea.

Página web dinámica: Una página web dinámica es un sitio web que muestra contenido que puede cambiar en función de las acciones del usuario o la información de una base de datos. Se utilizan tecnologías como PHP, ASP.NET, Ruby on Rails o JavaScript para crear páginas web dinámicas. Estas páginas son capaces de mostrar contenido diferente a diferentes usuarios o en diferentes momentos, como una plataforma de redes sociales o una tienda en línea.

Aplicación web: Una aplicación web es una aplicación interactiva que se ejecuta en un navegador web. A menudo, ofrece funcionalidades similares a las aplicaciones de software tradicionales, pero no requiere una instalación en el dispositivo del usuario. Las aplicaciones web pueden ser tanto estáticas como dinámicas y pueden proporcionar una amplia gama de funciones, desde procesamiento de datos hasta interacción en tiempo real.

Mashup: Un mashup es una aplicación o sitio web que combina datos o funcionalidades de varias fuentes diferentes para crear una experiencia integrada. Los mashups suelen tomar información de diversas fuentes web, como APIs (interfaces de programación de aplicaciones), y la presentan de una manera combinada o enriquecida. Por ejemplo, un mashup de mapas podría combinar datos de mapas de Google con datos de ubicación de restaurantes de Yelp para mostrar restaurantes en un mapa.



8. Componentes de una aplicación web.

Cliente web que es normalmente un navegador su función es mandar peticiones al servidor y interpretar ciertos lenguajes como HTML, CSS y JS

Servidor web: Para recibir las peticiones de los clientes web (navegadores normalmente) y procesarlas

Base de datos: Almacena los datos de la aplicación.

Ficheros en un lenguaje de programación en el que está escrita la aplicación

Modulo encargado de ejecutar el código esto se encuentra integrado en el servidor.



9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso

Lado del cliente

El lenguaje más utilizado es JavaScript ocupando todo el mercado con más de un 99% de presencia

10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).

Lado del servidor

Los lenguajes más usados son PHP, Java, ASP.NET, Ruby y JavaScript

PHP esta especialmente diseñado para el desarrollo web es el mas usado para esto y es fácil de usar en la actualidad se suelen usar frameworks como Laravel o Symfony

ASP.NET es un framework propiedad de Microsoft en el que se usan lenguajes muy populares como C# y .net

Ruby es un lenguaje orientado a objetos fácil de leer fue muy popular en la década pasada ahora ya ha perdido popularidad se usa el framework Ruby on rails

Java es un lenguaje de programación orientado a objetos que ofrece alta escalabilidad y rendimiento. Es muy utilizado en aplicaciones empresariales y grandes sistemas se suele usar con el framework Spring.

JavaScript es versátil y se utiliza tanto en el lado del cliente como en el lado del servidor. Node.js es un entorno de ejecución de JavaScript en el lado del servidor que permite desarrollar aplicaciones web rápidas no tan usado para crear grandes aplicaciones.

© W3Techs.com	usage	change since 1 August 2023
1. PHP	76.9%	-0.5%
2. ASP.NET	6.8%	-0.1%
3. Ruby	5.5%	+0.1%
4. Java	4.7%	
5. JavaScript	3.0%	+0.3%
percentages of sites		

11. Características y posibilidades de desarrollo de una plataforma XAMPP.

XAMPP es una plataforma versátil y fácil de usar que proporciona un entorno de desarrollo web completo para la creación y prueba de aplicaciones web. Es especialmente útil para desarrolladores que desean trabajar en sus proyectos de manera local antes de llevarlos a un servidor en producción.

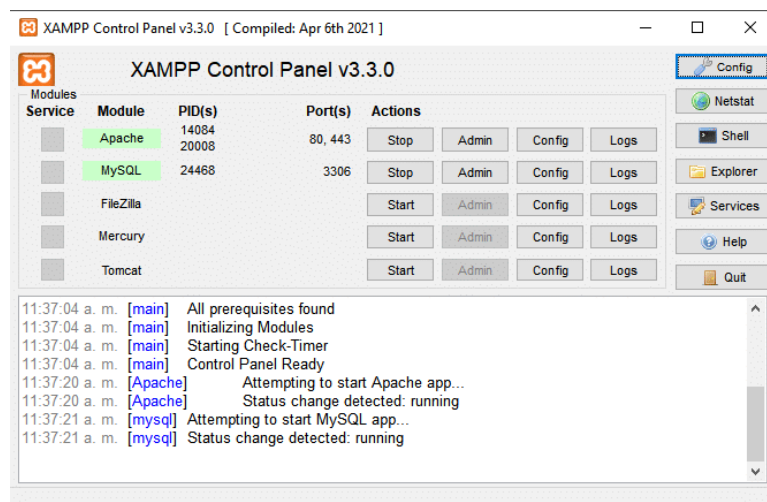
Características:

Multiplataforma compatible con Windows, Linux y macOS.

Fácil instalación y gran comunidad que ofrecen soporte y tutoriales.

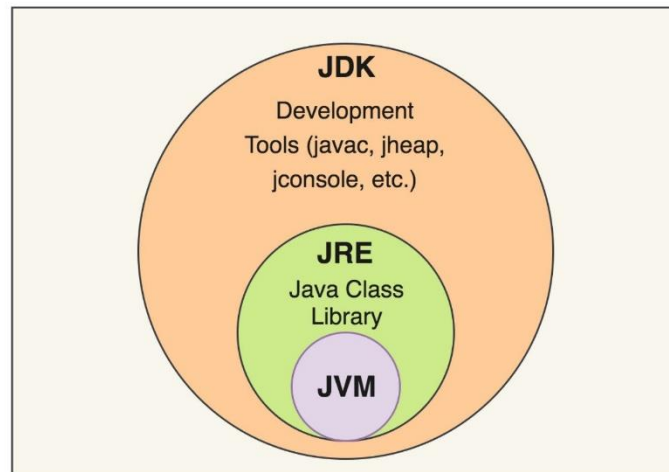
Compatibilidad con Apache uno de los servidores web más importantes, gestión de base de datos con MySQL o MariaDB conectable con phpMyAdmin para facilitar la gestión de bases de datos además de los lenguajes de programación mas populares como PHP

Seguridad XAMPP a sido creado para un entorno local puede tener graves problemas de seguridad en un entorno de producción



12. En qué casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación.

En el entorno de desarrollo si deseas programar Java necesitas el JDK que es una versión de JVM pero con añadidos para el desarrollo y en el entorno de explotación con el JVM que es el encargado de compilar es suficiente ya que solo necesitas ejecutar programas Java



13. IDE más utilizados (características y grado de implantación actual).

Los Entornos de Desarrollo Integrados son herramientas que ofrecen un conjunto de características y funcionalidades que simplifican y mejoran el proceso de desarrollo de software.

La elección de un IDE depende en gran medida del lenguaje de programación, las necesidades específicas del proyecto y las preferencias personales. Cada IDE tiene sus propias ventajas y desventajas.

Visual Studio Code (VS Code):

Características: VS Code es altamente personalizable, ligero y admite una amplia variedad de lenguajes de programación. Ofrece una gran cantidad de extensiones para añadir funcionalidades adicionales, depuración integrada y control de versiones, entre otras características.

NetBeans:

Características: NetBeans es un IDE de código abierto que admite varios lenguajes, incluyendo Java, PHP y C/C++. Ofrece una interfaz amigable y una amplia gama de funciones.

IntelliJ IDEA:

Características: Es una de las IDE más populares para Java y Kotlin, pero también ofrece soporte para otros lenguajes. Incluye características avanzadas de refactorización, depuración, integración con sistemas de control de versiones y análisis de código estático.

Eclipse:

Características: Aunque es ampliamente conocido por su uso en desarrollo de Java, Eclipse admite una amplia variedad de lenguajes a través de complementos. Ofrece un entorno de desarrollo extensible y sólido.

PyCharm:

Características: PyCharm está diseñado específicamente para el desarrollo en Python. Ofrece un editor avanzado, completado de código inteligente, depuración, administración de entornos virtuales y análisis de código.

Visual Studio:

Características: Visual Studio es un IDE de Microsoft que se utiliza principalmente para el desarrollo de aplicaciones Windows y .NET, pero también admite una variedad de lenguajes y plataformas. Incluye potentes herramientas de depuración y diseño de interfaces gráficas.

Android Studio:

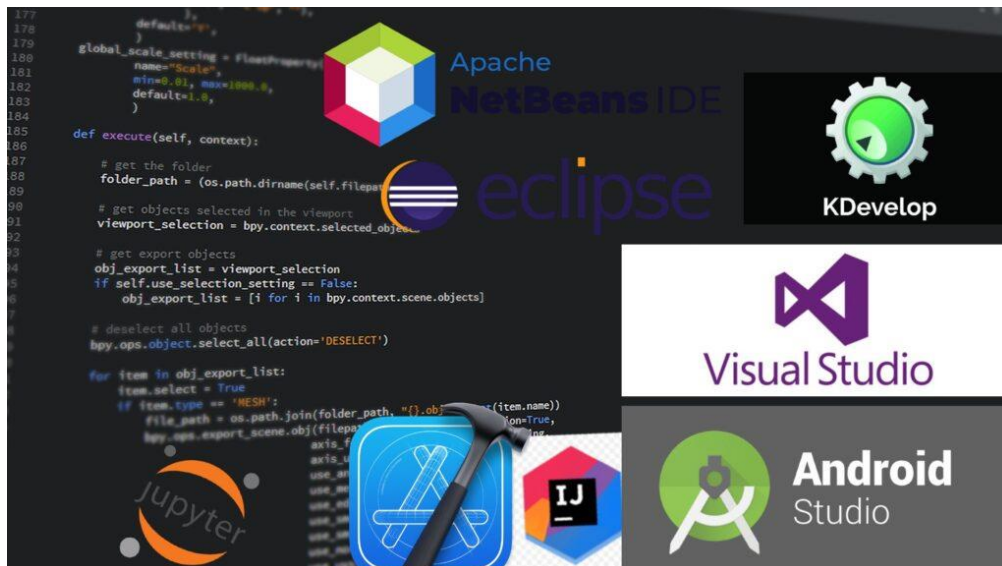
Características: Es el IDE oficial de Google para el desarrollo de aplicaciones Android. Ofrece una amplia gama de herramientas para el desarrollo de aplicaciones móviles, incluyendo un emulador de Android integrado.

Xcode:

Características: Xcode es el IDE de Apple para el desarrollo de aplicaciones iOS, macOS, watchOS y tvOS. Incluye un editor de código, herramientas de interfaz gráfica y emuladores para pruebas.

WebStorm:

Características: WebStorm es una IDE de JetBrains que se centra en el desarrollo web y ofrece soporte para JavaScript, TypeScript, HTML, CSS y otros lenguajes web. Incluye herramientas avanzadas de refactorización y depuración.



14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).

Nginx:

Características: Nginx es conocido por su alto rendimiento y bajo uso de recursos. Es altamente eficiente en el manejo de solicitudes HTTP y HTTPS. También es conocido por ser un servidor inverso y un proxy de alto rendimiento.

Grado de implantación: Nginx ha ganado una cuota de mercado significativa y se utiliza comúnmente como servidor proxy y balanceador de carga.

Apache:

Características: Apache es uno de los servidores web más antiguos y ampliamente utilizados. Es de código abierto y altamente configurable. Ofrece soporte para módulos adicionales, lo que lo hace versátil y adecuado para una amplia variedad de casos de uso.

Grado de implantación: A pesar de una disminución en su cuota de mercado en los últimos años, Apache sigue siendo un servidor web popular y se usa en muchos sitios web importantes.

CloudFlare:

No es un servidor web en si es una herramienta para mejorar la seguridad y rendimiento de servidores web siendo la mas popular de estas ofrecen firewall de aplicación web (WAF), servicios de DNS, gestión de tráfico, protección contra amenazas, CDN global...

LiteSpeed:

Características: LiteSpeed es conocido por su velocidad y eficiencia, y a menudo se utiliza en sitios web de alto tráfico. Ofrece funciones de servidor web y proxy inverso.

Grado de implantación: Aunque no es tan conocido como Apache o Nginx ha ganado popularidad y es utilizado en una variedad de sitios web.

Microsoft Internet Information Services (IIS):

Características: IIS es el servidor web de Microsoft para sistemas Windows. Ofrece una integración estrecha con otros productos de Microsoft y es una elección común para aplicaciones web en entornos Windows.

Grado de implantación: IIS es ampliamente utilizado en sitios web que se ejecutan en servidores Windows.

© W3Techs.com	usage	change since 1 September 2023
1. Nginx	34.2%	-0.1%
2. Apache	30.9%	-0.3%
3. Cloudflare Server	21.1%	+0.2%
4. LiteSpeed	12.6%	+0.2%
5. Microsoft-IIS	5.3%	-0.1%

percentages of sites

15. Apache HTTP vs Apache Tomcat

El Servidor HTTP de Apache (httpd) y Apache Tomcat son dos programas de código abierto ampliamente utilizados, pero tienen diferentes propósitos y, a menudo, se utilizan juntos en la implementación de aplicaciones web. Aquí tienes una comparación de ambos:

1. Propósito:

Apache : Es un servidor web diseñado principalmente para servir contenido estático, como archivos HTML, imágenes y otros recursos. Es conocido por su alto rendimiento en la entrega de páginas web y el manejo de solicitudes HTTP.

Apache Tomcat: Es un contenedor de servlets y JSP (JavaServer Pages) utilizado para alojar aplicaciones web Java. Tomcat es un servidor de aplicaciones que ejecuta servlets Java y páginas JSP, lo que lo hace adecuado para aplicaciones web dinámicas.

2. Lenguaje:

Servidor HTTP de Apache: Está escrito en C.

Apache Tomcat: Está escrito en Java.

3. Funcionalidad:

Servidor HTTP de Apache: Maneja solicitudes HTTP, sirve contenido estático y se puede ampliar con módulos para admitir varias funciones del servidor web, como autenticación, reescritura de URL y proxy.

Apache Tomcat: Es un contenedor de servlets Java, lo que significa que se utiliza para ejecutar aplicaciones web basadas en Java. Tomcat puede ejecutar servlets Java y páginas JSP y proporciona un entorno de tiempo de ejecución para aplicaciones web Java.

4. Configuración:

Servidor HTTP de Apache: La configuración generalmente se realiza a través de archivos .conf de Apache, que pueden ser complejos, especialmente para principiantes.

Apache Tomcat: La configuración se realiza generalmente a través de archivos XML, como server.xml y web.xml, y se centra más en la configuración de aplicaciones web Java.

7. Casos de uso:

Servidor HTTP de Apache: Es más adecuado para alojar sitios web estáticos y actuar como un servidor proxy inverso para varios servidores de aplicaciones.

Apache Tomcat: Es ideal para alojar aplicaciones web basadas en Java y proporcionar una plataforma para ejecutar servlets y JSP con contenido dinámico.

8. Integración:

Es común utilizar Apache y Apache Tomcat al mismo tiempo. El Servidor HTTP de Apache maneja contenido estático, equilibrio de carga y terminación SSL, al tiempo que reenvía las solicitudes dinámicas a Tomcat para su procesamiento.



17. Generadores de documentación HTML (PHPDoc):

Es un programa que mediante los comentarios impuestos por el programa genera código en un lenguaje de programación.

Estos son los mas usados para cada lenguaje:

PHP:

PHPDoc: Es una herramienta comúnmente utilizada para generar documentación en PHP en formato HTML. También admite otros formatos.

JavaScript:

JSDoc: Es una herramienta muy utilizada para documentar código JavaScript. Permite generar documentación en formato HTML a partir de comentarios en el código fuente.

Python:

Sphinx: Es una herramienta de documentación muy popular en la comunidad de Python. Aunque puede generar varios formatos de documentación, incluyendo HTML, es ampliamente utilizado para crear documentación HTML para proyectos de Python.

Java:

Javadoc: Es una herramienta estándar para la generación de documentación en Java. Genera documentación en formato HTML a partir de comentarios en el código fuente.

Ruby:

YARD: Es un generador de documentación para Ruby que se utiliza para crear documentación en HTML y otros formatos a partir de comentarios y anotaciones en el código.