

Desarrollo Web en Entorno Servidor

1. Protocolo de comunicaciones: IP, TCP, HTTP, HTTPS.....	1
2. Modelo de comunicaciones cliente-servidor y su relación con las aplicaciones web.....	1
3. Estudio sobre los métodos de petición al servidor HTTP/HTTPS.....	1
4. Estudio sobre el URI, URL y URN.....	2
5. Modelo de desarrollo de aplicaciones multicapa.....	2
6. Modelo de división front end-back end.....	2
7. Página web estática, dinámica, aplicación web y mashup.....	3
8. Componentes de una aplicación web.....	3
9. Programas ejecutados en el lado cliente y en el lado servidor.....	3
10. Lenguajes de programación utilizados en el lado servidor.....	3
11. Características de desarrollo de una plataforma XAMPP.....	4
12. Casos necesarios para la instalación de JVM y JDK en entorno de desarrollo y explotación. . .	4
13. IDE más utilizados.....	4
14. Servidores HTTP/HTTPS más utilizados.....	4
15. Apache HTTP y Tomcat.....	5
16. Navegadores HTTP/HTTPS más utilizados.....	5
17. Generadores de documentación HTML.....	5
18. Repositorios de software.....	5

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

Los protocolos de comunicación son conjuntos de normas que permiten que los dispositivos se comuniquen entre sí, estableciendo una forma de identificación, de transmisión de datos y de procesamiento de información.

IP: intenta transmitir los datos sin establecer una conexión previa.

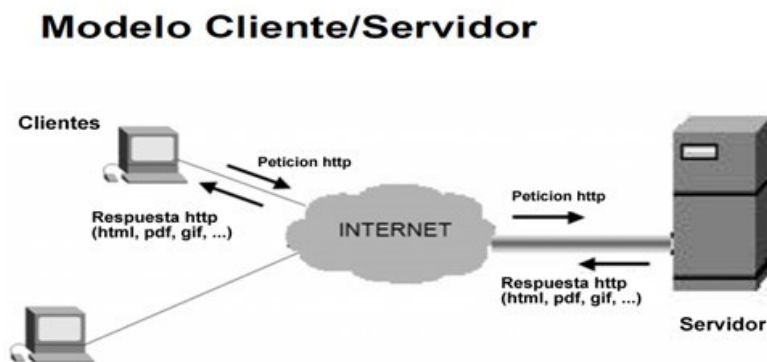
TCP: establece una conexión estable, descompone los archivos en paquetes y los envía.

HTTP: permite la transferencia de archivos web usando el puerto 80.

HTTPS: la versión segura de HTTP, usa el puerto 443.

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

Modelo de diseño en el cual las tareas se reparten entre el servidor y el cliente; el cliente realiza peticiones al servidor el cual le da respuesta.



3. Estudio sobre los métodos de petición al servidor HTTP/HTTPS

GET: solicita un objeto, recurso o documento.

POST: envía un objeto, recurso o documento.

HEAD: solicita el encabezado del recurso.

PUT: reemplaza o crea un nuevo elemento con los datos.

DELETE: borra un recurso.

CONNECT: intenta establecer conexión con el servidor.

OPTIONS: solicita los métodos de comunicación soportados por el servidor.

TRACE: efectúa una prueba de bucle de mensaje por el camino al recurso objetivo.

PATCH: aplica modificaciones parciales al recurso.

4. Estudio sobre el URI, URL y URN

El URI es una cadena de caracteres utilizados para identificar un recurso en la red y máquina local, el URL es un tipo de URI que indica la ubicación del recurso y el protocolo usado y el URN indica la ubicación y nombre, pero no el protocolo. Por ejemplo, del URI: <https://sub.dominio.com:443/ruta/adicional.php?parametro#ancla>, lo azul sería el URL, lo amarillo el URN y lo verde lo que tienen en común.

La estructura del URI se divide en varias partes:

Protocolo: proporciona información de cómo manejar la conexión.

Subdominio: extensión del dominio principal que actúa como sitio web independiente.

Dominio: dirección única que indica un sitio web en particular.

Puerto: el puerto usado.

Ruta: identifica la ubicación específica del recurso usado.

Parámetros: variable propia del URI para indicar características necesarias en el recurso.

Anclas: indica una parte específica del recurso.

<https://sub.dominio.com:443/ruta/adicional.php?parametro#ancla>

5. Modelo de desarrollo de aplicaciones multicapa

La arquitectura multicapa es aquella en la que las funciones de presentación, lógica y gestión de datos están separadas.

Capa de presentación la interfaz gráfica que ve el usuario; muestra la información y recoge los datos.

Capa de lógica procesa las peticiones de la capa de presentación y genera respuestas con la información que pide a la capa de datos.

Capa de datos la única encargada de acceder a los datos.

6. Modelo de división front end-back end

La funcionalidad front end es la cual corresponde a los usuarios corrientes, esta incluye la visualización del sitio, su interfaz y el diseño web; mientras que la back end corresponde a usuarios especiales como pueden ser administradores, incluye las bases de datos, su procesamiento y los servidores.

7. Página web estática, dinámica, aplicación web y mashup

Página web estática contenido fijo que se debe modificar manualmente, no se puede interactuar con la información.

Página web dinámica permita la interacción del usuario y es actualizada a tiempo real.

Aplicación web software con interacción cliente-servidor que envía, recibe y procesa información.

Mashup combinación de datos y funcionalidades de diferentes aplicaciones en un solo servicio.

8. Componentes de una aplicación web

Cliente web interfaz a través de la cual los usuarios interactúan con la aplicación.

Servidor web donde se almacena la aplicación.

Módulo para ejecutar el código programa que ejecuta los archivos.

Sistema gestor de base de datos software que maneja y administra los datos.

Ficheros escritos en lenguaje de programación archivos en los que se encuentra el código.

9. Programas ejecutados en el lado cliente y en el lado servidor

Los lenguajes de programación ejecutados en el lado del cliente son los que interpretan por el navegador web: **HTML, JavaScript, Java, VBScript, Flash** o **CSS**.

Los ejecutados en lado servidor se alojan en la aplicación: **PHP, ASP, JSP**, o **Perl**.

10. Lenguajes de programación utilizados en el lado servidor

PHP multiplataforma, ofrece facilidad de conexión con diversas bases de datos.

Ruby lenguaje interpretado, código abierto, orientado a objetos.

ASP.NET control de usuario personalizado, separación entre capas de diseño y código.

Java lenguaje compilado orientado a objetos, extensa biblioteca y comunidad de soporte.

JavaScript lenguaje interpretado, permite construir aplicaciones utilizando un único lenguaje.

Most popular server-side programming languages

	usage	change since 1 November 2024
© W3Techs.com		
1. PHP	75.4%	-0.3%
2. Ruby	6.1%	+0.1%
3. ASP.NET	5.5%	-0.1%
4. Java	5.0%	
5. JavaScript	3.9%	+0.2%

11. Características de desarrollo de una plataforma XAMPP

El XAMPP es un paquete de software que incluye Apache, MySQL, PHP y Perl, ofreciendo todos los componentes necesarios para desarrollar aplicaciones web.

Sus características principales son su portabilidad, compatibilidad multiplataforma y soporte de múltiples lenguajes de programación y base de datos.

12. Casos necesarios para la instalación de JVM y JDK en entorno de desarrollo y explotación

El Java Developers toolKit (JDK) permite desarrollar y compilar aplicaciones basadas en Java; por lo que sirve para el entorno de desarrollo.

La Máquina Virtual de Java (JVM) ejecuta el código compilado; haciéndolo necesario para el entorno de explotación.

13. IDE más utilizados

Visual Studio Code biblioteca de extensiones, depuración integrada, control de versiones.

Eclipse soporte de múltiples lenguajes, resalto de sintaxis, depuración de código.

NetBeans extensible mediante complementos y módulos.

IntelliJ autocompletado, refactorización.

14. Servidores HTTP/HTTPS más utilizados

Nginx eficiente manejo de conexiones simultáneas, bajo consumo de recursos.

Apache servidor de código abierto y gratuito, compatible con varios sistemas operativos, altos estándares de seguridad.

Cloudflare Server servicio gratuito de DNS, destaca su velocidad de búsqueda.

Most popular web servers

© W3Techs.com	usage	change since 1 November 2024
1. Nginx	33.8%	
2. Apache	28.1%	-0.6%
3. Cloudflare Server	22.6%	-0.1%
4. LiteSpeed	14.0%	
5. Microsoft-IIS	4.3%	-0.1%

15. Apache HTTP y Tomcat

Ambos son servidores web; mientras que HTTP fue desarrollado para PHP, Perl o Python, TomCat fue desarrollado para aplicaciones de Java.

16. Navegadores HTTP/HTTPS más utilizados

Google Chrome alto rendimiento, gran velocidad de carga, extensiones personalizables y sincronización entre dispositivos.

Mozilla Firefox enfoque en la privacidad del usuario, personalización mediante complementos, herramientas de desarrollo integradas.

Microsoft Edge integración de Windows, rendimiento mejorado, herramientas de seguridad avanzadas.

Safari optimizado para rendimiento y eficiencia energética, enfoque en la privacidad.

Opera VPN gratuita, bloqueador de anuncios, modo de ahorro de batería.

17. Generadores de documentación HTML

Los generadores de documentación se encargan de generar un api a partir de comentarios que se incluyan en el código.

PHPDocumentor soporta múltiples formatos de salida, diagrama de clases y personalización mediante plantillas.

ApiGen interfaz limpia y rápida, soporte de namespaces, opción de búsqueda, ideal para proyectos con muchas clases y funciones.

18. Repositorios de software

El control de versiones es un sistema que registra los cambios realizados en un proyecto a lo largo del tiempo, permitiendo gestionar, rastrear y revertir los cambios de manera eficiente.

Control de Versiones Centralizado (CVS) un único servidor almacena el proyecto y se desarrolla en copias locales.

Control de Versiones Distribuido (DVCS) cada desarrollador tiene una copia completa del repositorio, lo que permite trabajar sin conexión y realizar cambios localmente.