
DESARROLLO DE APLICACIONES WEB 2

EJERCICIOS TEMA 1 - DWES

15 DE DICIEMBRE DE 2023

DESARROLLO WEB EN ESTORNO SERVIDOR

OSCAR PASCUAL FERRERO

AUTOR ORIGINAL

CARLOS GARCIA CACHON

1. Protocolos de Comunicaciones	2
2. Modelo de Comunicaciones Cliente-Servidor y su Relación con las Aplicaciones Web	3
3. Métodos de Petición HTTP/HTTPS más utilizados.....	4
4. Concepto de URI/URL/URN y su Relación con HTTP/HTTPS	5
5. Modelo de Desarrollo de Aplicaciones Multicapa.....	6
6. Modelo de División Funcional Front-end/Back-end para Aplicaciones Web.....	7
7. Página Web Estática, Página Web Dinámica, Aplicación Web, Mashup.....	8
8. Componentes de una Aplicación Web.....	10
9. Programas ejecutados en el lado del Cliente y Servidor.....	12
10. Lenguajes de Programación utilizados en el lado del servidor de una Aplicación Web	13
11. Características y posibilidades de desarrollo de una plataforma XAMPP.....	14
12. Casos en que se Necesita Instalar la Máquina Virtual Java (JVM) y el Software JDK.....	15
13. IDEs más utilizados	16
14. Servidores HTTP/HTTPS más utilizados.....	17
15. Apache HTTP vs Apache Tomcat	18
16. Navegadores HTTP/HTTPS más utilizados.....	19
17. Generadores de Documentación HTML (PHPDoc).	20
18. Repositorios de Software - Sistemas de Control de Versiones.....	21
19. Propuesta de Configuración del Entorno de Desarrollo	22
20. Propuesta de Configuración del Entorno de Explotación	22
21. Estudio sobre CMS y ERP	22

1. Protocolos de Comunicaciones

IP: (Protocolo de Internet) es un protocolo de comunicación que tiene como objetivo transmitir paquetes de datos de una forma bidireccional (que tiene un origen y un destino) a través de una red. Pertenece a la capa 3 del modelo OSI (capa de red) en la que se enruta el paquete dándole una dirección IP.

TCP: (Protocolo de control de transmisión) es un protocolo de comunicación que establece las comunicaciones entre los equipos para que los usuarios podamos navegar por internet y servicios en red. Pertenece a la capa de transporte del modelo OSI.

HTTP: (Hypertext Transfer Protocol) es un protocolo el cual nos permite realizar peticiones de datos que sigue la estructura de cliente-servidor (protocolo en el que se basan las aplicaciones web). Pertenece a la capa aplicación del modelo OSI.

HTTPS: consiste en lo mismo que el protocolo HTTP solamente que en este la comunicación es cifrada (agrega SSL).



Enlaces:

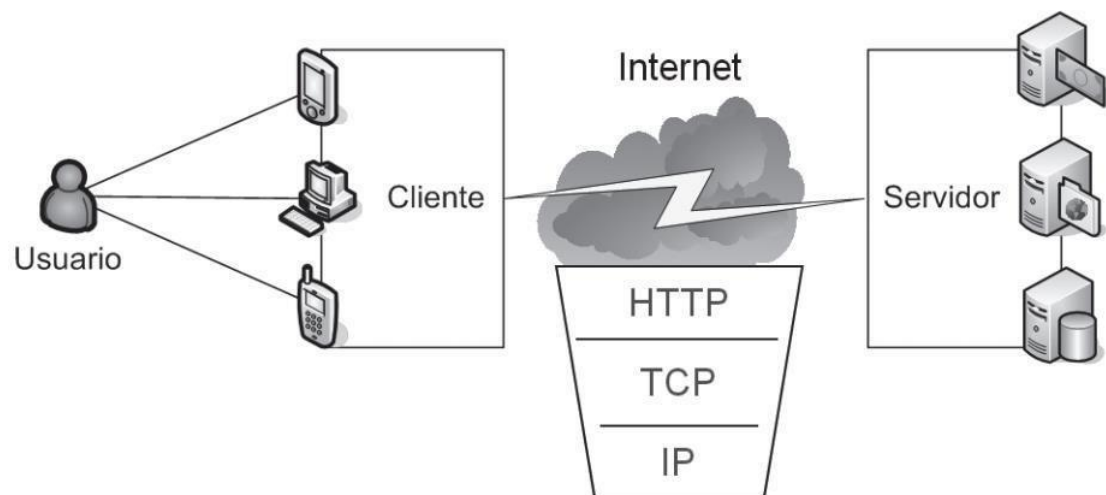
[https://www.cloudflare.com/es-es/learning/network-layer/internet-protocol/#:~:text=El%20Protocolo%20de%20Internet%20\(IP\)%20es%20un%20protocolo%20C%20o,trozos%20m%C3%A1s%20peque%C3%B1os%20llamados%20paquetes.](https://www.cloudflare.com/es-es/learning/network-layer/internet-protocol/#:~:text=El%20Protocolo%20de%20Internet%20(IP)%20es%20un%20protocolo%20C%20o,trozos%20m%C3%A1s%20peque%C3%B1os%20llamados%20paquetes.)

<https://developer.mozilla.org/es/docs/Web/HTTP/Overview>

Autor: Rebeca Sánchez Pérez

2. Modelo de Comunicaciones Cliente-Servidor y su Relación con las Aplicaciones Web.

Es un modelo de diseño de software que se compone de 2 partes (el cliente y el servidor) en el que cliente realiza una petición HTTP/S al servidor, que la procesa y envía una respuesta. Este tipo de arquitectura es la más utilizada en el desarrollo de aplicaciones web ya que nos permite conectar varios clientes a los servicios que provee un servidor.



Enlaces:

<https://nucba.medium.com/qu%C3%A9-es-la-arquitectura-cliente-servidor-eb9f402506cc>

Autor: Rebeca Sánchez Pérez

3. Métodos de Petición HTTP/HTTPS más utilizados.

Los métodos de petición HTTP/HTTPS comunes son GET, POST, PUT, DELETE, HEAD y OPTIONS.

GET se usa para recuperar datos.

POST para enviar datos al servidor.

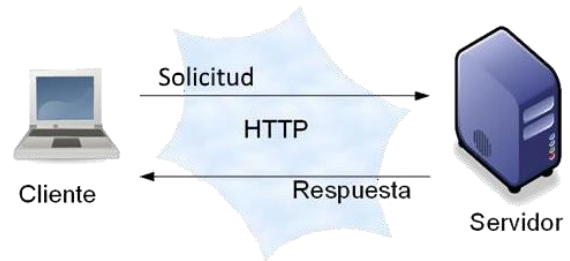
PUT para actualizar datos.

DELETE para eliminar recursos.

HEAD para obtener encabezados.

OPTIONS para obtener información sobre las opciones de comunicación permitidas.

Los más utilizados son POST y GET



4. Concepto de URI/URL/URN y su Relación con HTTP/HTTPS

URI (Identificador de Recursos Uniforme): Es una secuencia de caracteres que identifica de manera única un recurso en Internet. (Ej: <https://website.com/drive/info#part1>).

URL (Localizador Uniforme de Recursos): Es un tipo de URI que especifica la ubicación del recurso. (Ej: <https://elpais.com/>
<file:///localhost/8.8.8.8>).

URN (Nombre Uniforme de Recursos): Es otro tipo de URI que se usa para nombrar recursos de forma persistente incluso si su ubicación cambia.

(Ej: <urn:nbn:de:101:3-2019075675872913>).

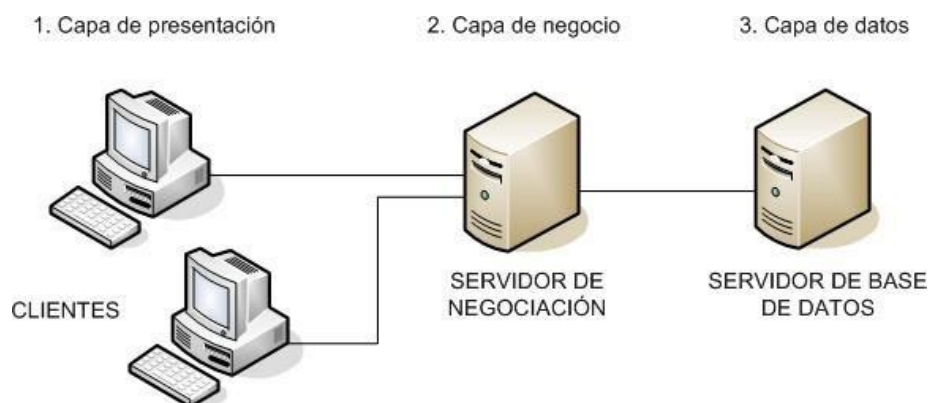
Se relacionan en los protocolos de HTTP y HTTPS para acceder a recursos en internet de manera segura o no segura.

Esquema URL:



5. Modelo de Desarrollo de Aplicaciones Multicapa.

Los modelos de aplicaciones multicapa o de 3 capas consisten en agrupar el código de la aplicación en capas dependiendo de la función que desempeñan y que permiten la comunicación entre ellas.



Capa de datos: en la que las líneas de código se agrupan en ficheros, dicho de otra forma, donde se encuentran los datos que utiliza la aplicación para el cliente.

Capa de negocio o aplicación: es el código que se encarga de realizar cálculos, es decir, el programa que estructura la aplicación.

Capa de presentación: las líneas de código que le dan apariencia a la web y muestra información al cliente desde su navegador (interfaz gráfica).

El número de capas de una aplicación web depende del tamaño de los componentes, de la naturaleza del servicio ofrecido y del reparto de las funciones entre el cliente y el servidor. Generalmente se componen por 3 capas, pero pueden aparecer más si trabajamos con un modelo de aplicación MVC o si la aplicación está destinada a la extracción de datos para un web service o para generar una API para el cliente.

Autor: Rebeca Sánchez Pérez

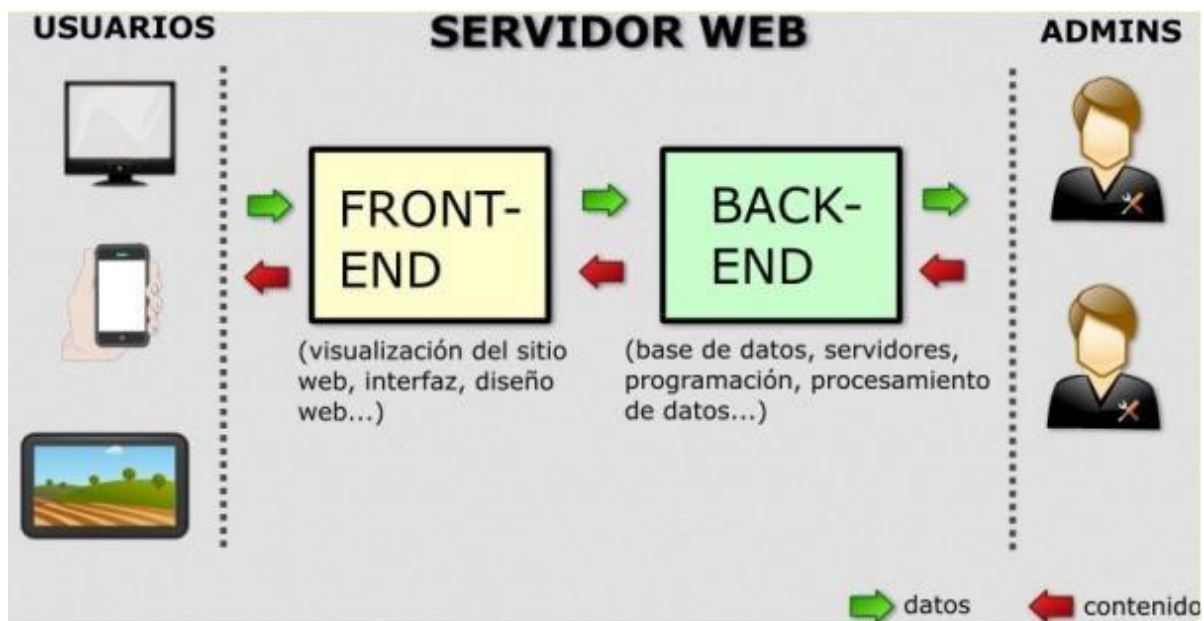
6. Modelo de División Funcional Front-end/Back-end para Aplicaciones Web.

Front-end se refiere a la parte de una aplicación web con la que los usuarios interactúan directamente, incluyendo botones, formularios, etc. Es responsable de la presentación de datos, la experiencia del usuario y la interacción en tiempo real.

El Back-end se refiere a la parte de una aplicación web que funciona detrás de escena y no es visible para el usuario final incluyendo el servidor web que recibe las solicitudes del navegador del usuario, almacena y gestiona los datos en una BD y contiene una lógica en la que procesa datos y autentica usuarios y a su vez proporciona una API (Interfaz de Programación de Aplicaciones) para permitir la comunicación entre el frontend y el backend.

CONCLUSIÓN:

El frontend se ocupa de la presentación y la experiencia del usuario, mientras que el backend se encarga de la lógica de la aplicación, la gestión de datos y la comunicación con el frontend. Ambos trabajan juntos para proporcionar una experiencia completa de la aplicación web.

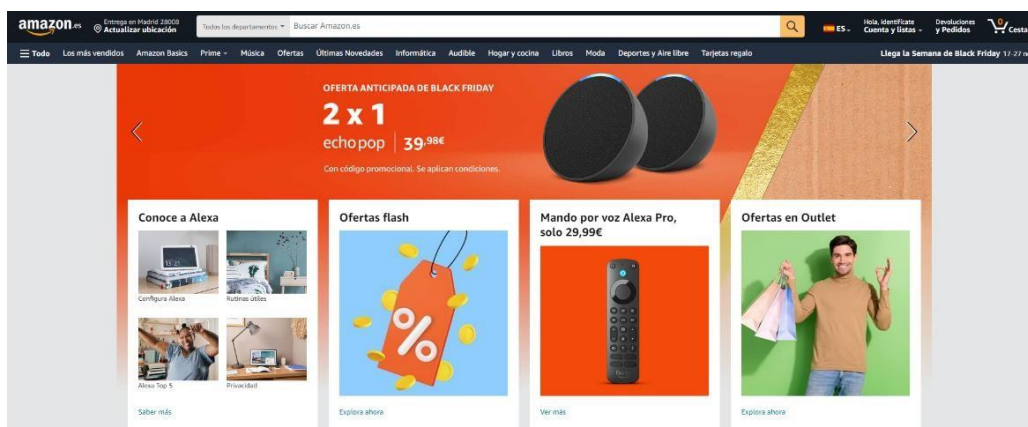


7. Página Web Estática, Página Web Dinámica, Aplicación Web, Mashup.

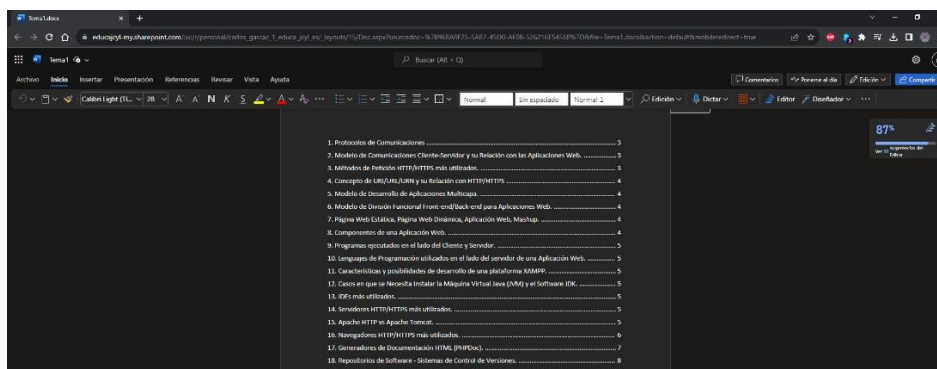
Página web estática: Contenido fijo que no cambia según la interacción del usuario, un ejemplo es el sitio "The World's Worst Website Ever," que se creó como un ejemplo de lo que no se debe hacer en diseño web.



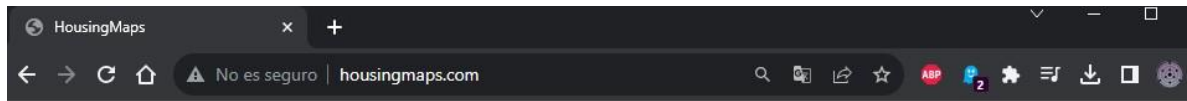
Página web dinámica: Contenido generado en tiempo real, adaptado según la interacción del usuario o datos externos, tiene control de acceso y gestión de datos. Un ejemplo sería "Amazon.es".



Aplicación web: Es un programa interactivo ejecutado en un navegador, un ejemplo de una aplicación web es "Google Docs".



Mashup: Integración de diferentes servicios web para crear una nueva aplicación compuesta, un ejemplo sería "HousingMaps," que combina datos de Craigslist y Google Maps para ayudar a los usuarios a encontrar alquileres y propiedades en un mapa interactivo.



8. Componentes de una Aplicación Web.

1. Frontend (Interfaz de Usuario):

- Interfaz de Usuario (UI): La parte de la aplicación con la que los usuarios interactúan. Incluye elementos como botones, formularios y páginas web.
- Diseño (Layout y Estilo): Define cómo se ve la interfaz de usuario, incluyendo la disposición de elementos y el estilo visual.
- JavaScript: Se utiliza para agregar interactividad a la interfaz de usuario y realizar acciones en el navegador del cliente.

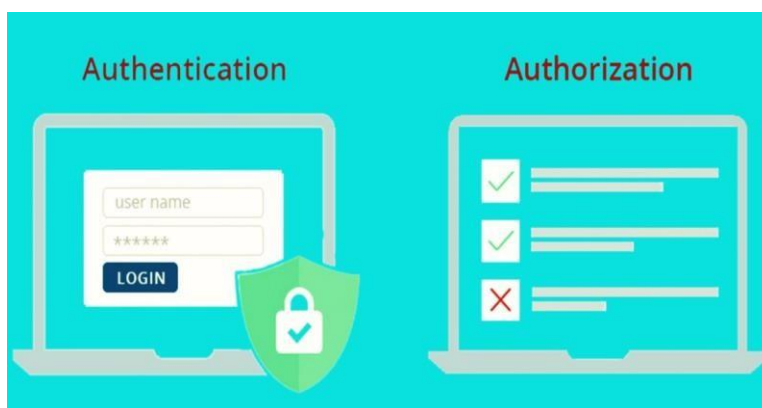
2. Backend (Lógica de Negocio):

- Servidor Web: El servidor que recibe las solicitudes de los clientes y responde con datos o páginas web.
- Base de Datos: Almacena datos utilizados por la aplicación, como información de usuarios, contenido, y configuraciones.
- Lógica de Aplicación: Las reglas y procesos que determinan cómo la aplicación funciona, incluyendo la autenticación, validación y procesamiento de datos.
- API (Interfaz de Programación de Aplicaciones): Permite la comunicación entre el frontend y el backend, permitiendo que las partes interactúen entre sí.



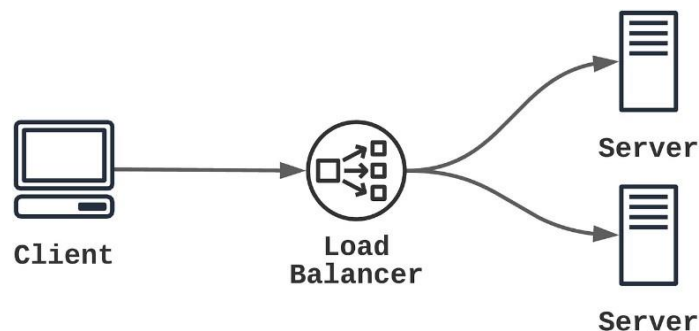
3. Seguridad:

- Autenticación y Autorización: Procesos que garantizan que los usuarios tengan acceso adecuado y seguro a la aplicación.
- Seguridad de Datos: Medidas para proteger los datos almacenados y transmitidos, como cifrado y políticas de acceso.



4. Infraestructura:

- **Servidores y Alojamiento:** Los servidores físicos o virtuales en los que se ejecuta la aplicación web.
- **Balanceador de Carga:** Distribuye el tráfico de la aplicación de manera equitativa entre varios servidores para mejorar el rendimiento y la disponibilidad.
- **CDN (Red de Distribución de Contenido):** Almacena en caché y entrega contenido estático, como imágenes y archivos CSS, desde ubicaciones geográficamente distribuidas para acelerar la carga de la aplicación.



5. Monitorización y Registro:

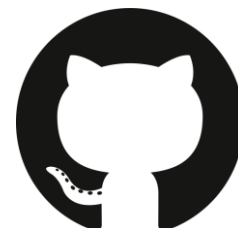
- **Herramientas de Monitorización:** Supervisan el rendimiento y la salud de la aplicación, identificando problemas y permitiendo la optimización.
- **Registro (Logging):** Registra eventos y actividades en la aplicación para su análisis y resolución de problemas.

	nivel	hilo	nombre	mensaje
▶	WARN	[main]	clases.Principal	Este es un mensaje de Warning
	ERROR	[main]	clases.Principal	Este es un mensaje de Error
	INFO	[main]	clases.Principal	Este es un mensaje de Info
	FATAL	[main]	clases.Principal	Este es un mensaje de Fatal
	TRACE	[main]	clases.Principal	Este es un mensaje de Trace

logs 2 Read Only

6. Gestión de Versiones y Despliegue:

- **Control de Versiones:** Utilizado para rastrear cambios en el código fuente y la colaboración entre desarrolladores.
- **Despliegue (Deployment):** El proceso de llevar la aplicación desde el entorno de desarrollo al entorno de producción.



7. Documentación:

- **Documentación de Usuario:** Instrucciones y guías para los usuarios de la aplicación.
- **Documentación Técnica:** Información para desarrolladores y administradores sobre cómo funciona la aplicación y cómo mantenerla.



9. Programas ejecutados en el lado del Cliente y Servidor.

Lado del Cliente:

Navegador web (por ejemplo, Google Chrome): Se ejecutan en el lado del cliente y su función principal es interpretar y representar contenido web, incluyendo páginas HTML, CSS y JavaScript.

Los navegadores permiten a los usuarios acceder a sitios web, interactuar con aplicaciones web y mostrar el contenido de manera amigable para el usuario. Además, pueden ejecutar scripts de JavaScript en el navegador del usuario para mejorar la interactividad y la experiencia del usuario en la web.

Lado del Servidor:

Servidor web (por ejemplo, Apache): Un servidor web es un programa que se ejecuta en el lado del servidor y responde a las solicitudes de los navegadores web de los clientes.

Su función principal es recibir las solicitudes HTTP/HTTPS de los clientes, procesarlas y entregar las respuestas adecuadas, que generalmente incluyen páginas web, datos o recursos.

Los servidores web también pueden ejecutar aplicaciones y lógica de servidor, como PHP o Python, para generar contenido dinámico antes de enviarlo al cliente. Estos servidores son esenciales para alojar sitios web y aplicaciones web y servir su contenido a los usuarios.



10. Lenguajes de Programación utilizados en el lado del servidor de una Aplicación Web.

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

PHP: Esta especialmente diseñado para el desarrollo web y 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 y actualmente ha perdido popularidad.

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.

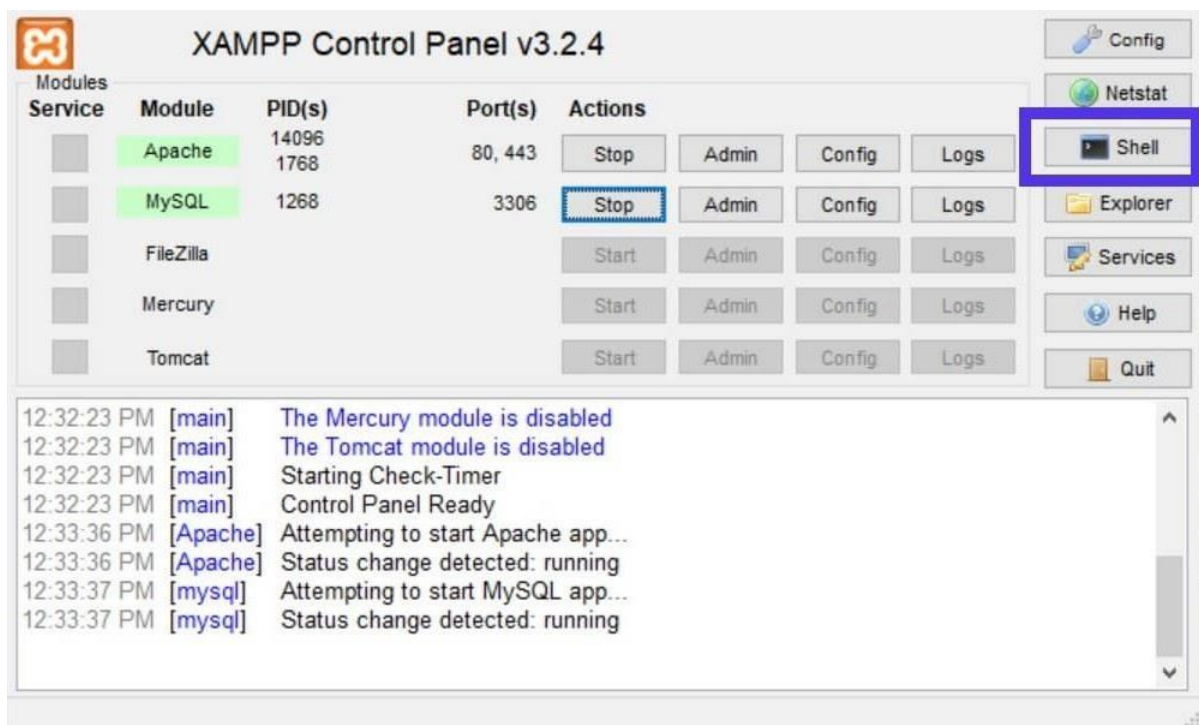
Características:

Multiplataforma compatible con Windows, Linux y macOS.

Fácil instalación y gran comunidad que ofrecen soporte y tutoriales.

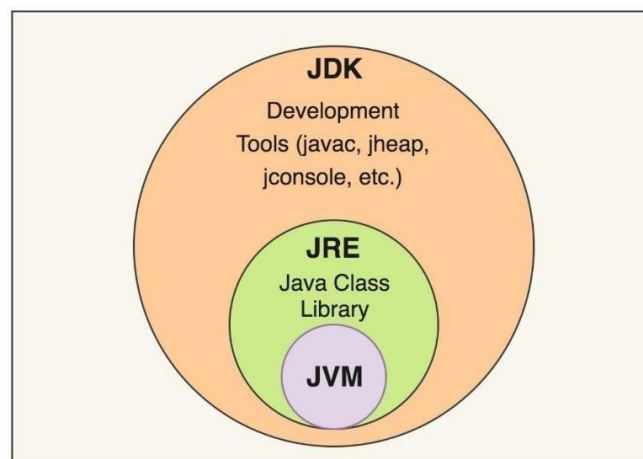
Compatibilidad con Apache uno de los servidores web más importantes, gestión de base de datos con MySQL o MariaDB conectable con phpMyAdmin para facilitar la gestión de bases de datos además de los lenguajes de programación más populares como PHP

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



12. Casos en que se Necesita Instalar la Máquina Virtual Java (JVM) y el Software JDK.

La instalación de JVM o JDK depende de si vamos a desarrollar una aplicación o solamente queremos usarla. Si queremos desarrollar una app en java necesitaremos el JDK (*Java Development Kit*) en el entorno de desarrollo (donde se llevan a cabo las pruebas) y de explotación (cuando ya puedes acceder a la app desde internet) y en el caso de querer usar una aplicación necesitaremos el JVM ya que este es el intérprete de Java.



13. IDEs más utilizados.

Un IDE, o Entorno de Desarrollo Integrado, es una herramienta de software que proporciona un conjunto completo de funciones para el desarrollo de aplicaciones informáticas. Están diseñados para facilitar y agilizar el proceso de desarrollo junto con herramientas y características integradas en una interfaz de usuario. Las más utilizadas son:

Visual Studio Code: Desarrollado por Microsoft.

Características:

- Amplia biblioteca de extensiones
- Ofrece sugerencias de código
- Integra depuración
- Control de versiones
- Multiplataforma
- Código abierto



Eclipse: Desarrollado originalmente por IBM y ahora mantenido por la Eclipse Foundation.

Características:

- Soporte para múltiples lenguajes de programación (Java, C++, Python, ...).
- Muy personalizable, por medio de extensiones.
- Incluye resaltado de sintaxis
- Depuración de código
- Control de versiones
- Plataforma RCP (Proporciona un conjunto de componentes y bibliotecas que permiten desarrollar aplicaciones de escritorio personalizables de manera eficiente.)



NetBeans: Desarrollado originalmente por Sun Microsystems y ahora mantenido por Oracle Corporation.

Características:

- Fácil uso y versatilidad particularmente en el desarrollo de aplicaciones Java
- Capacidad de arrastrar y soltar, que facilita la creación de interfaces de usuario gráficas y la construcción rápida de aplicaciones.
- Altamente extensible a través de complementos y módulos
- Admite numerosos lenguajes de programación como Java, PHP, HTML5, JavaScript.
- Proporciona herramientas de desarrollo web y móvil



14. Servidores HTTP/HTTPS más utilizados.

1. Apache HTTP Server:

- Apache es uno de los servidores web más populares y de código abierto del mundo. Es conocido por su estabilidad y flexibilidad.
- Ofrece una amplia gama de módulos y configuraciones personalizables que permiten a los administradores de servidores adaptarlo a sus necesidades.
- Es compatible con una variedad de sistemas operativos, incluyendo Linux y Windows.



2. Nginx:

- Nginx es otro servidor web de código abierto ampliamente utilizado, conocido por su alto rendimiento y capacidad de servir grandes cantidades de tráfico.
- Está diseñado para ser eficiente y escalable, y es una elección común para servir sitios web y aplicaciones web de alto rendimiento.
- También se usa comúnmente como servidor proxy inverso y equilibrador de carga.



3. (IIS):

- IIS es el servidor web de Microsoft para sistemas operativos Windows.
- Es conocido por su integración con otros productos de Microsoft y su capacidad para ejecutar aplicaciones web ASP.NET y servicios de Microsoft.
- Es una elección común para empresas que utilizan tecnologías de Microsoft.



Ambos servidores (Apache, Nginx e IIS) son capaces de admitir tanto HTTP como HTTPS.

Para habilitar HTTPS en estos servidores, generalmente se necesita obtener un certificado SSL/TLS válido y configurar el servidor para usarlo.

Esto garantiza la seguridad de las comunicaciones y permite la transmisión segura de datos entre el cliente y el servidor.

15. Apache HTTP vs Apache Tomcat.

Apache HTTP Server:

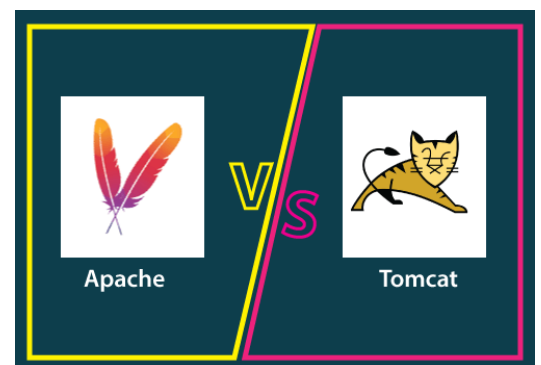
- Apache HTTP Server, conocido "Apache", es un servidor web de código abierto.
- Su función principal es servir páginas web estáticas y dinámicas, así como otros contenidos web, a través del protocolo HTTP.
- Es conocido por su estabilidad, flexibilidad y capacidad de configuración a través de archivos de configuración.
- Está diseñado para gestionar solicitudes HTTP y responder con contenido web, pero no ejecuta aplicaciones Java.

Apache Tomcat:

- Apache Tomcat, conocido como "Tomcat", es un servidor de aplicaciones web de código abierto desarrollado por la Apache Software Foundation.
- Está diseñado específicamente para alojar y ejecutar aplicaciones web Java, especialmente aquellas basadas en tecnologías como Java Servlets y JavaServer Pages (JSP).
- Tomcat actúa como un contenedor de servlets y JSP, lo que significa que puede recibir solicitudes HTTP, ejecutar aplicaciones web basadas en Java y devolver las respuestas adecuadas.
- Tomcat es ampliamente utilizado para ejecutar aplicaciones Java en servidores web.

Diferencias Clave:

- Apache se enfoca en servir contenido web, incluyendo páginas estáticas y dinámicas, a través del protocolo HTTP, mientras que Tomcat se enfoca en alojar y ejecutar aplicaciones web Java basadas en servlets y JSP.
- Apache no ejecuta aplicaciones Java, mientras que Tomcat es específicamente un contenedor de servlets y JSP para aplicaciones Java.
- En algunos casos, ambos servidores pueden trabajar juntos, con Apache gestionando las solicitudes web generales y redirigiendo solicitudes Java a Tomcat para su procesamiento.



En resumen, Apache se centra en servir contenido web a través de HTTP, mientras que Tomcat está diseñado para ejecutar aplicaciones web Java. La elección entre ellos depende de los requisitos específicos de tu proyecto web.

16. Navegadores HTTP/HTTPS más utilizados.

Google Chrome: es el navegador web y móvil desarrollado por Google. La mayor parte de su código fuente proviene de Chromium, el proyecto de software de código abierto y gratuito de Google.

Algunas de sus características son:

- Gestión de contraseñas.
- Generación de contraseñas seguras.
- Autocompletar formularios.
- Modo oscuro.
- Múltiples perfiles.

Safari: esta específicamente diseñado y optimizado para funcionar en dispositivos Apple y algunas de sus características frente a otros navegadores son:

- Notas rápidas. (Cuentan con ella de manera nativa, aunque el resto tiene plug-in para usar también la misma tecnología.)
- Streaming de vídeo en 4K.

Todo ello de manera nativa y sin descargar o configurar nada.

Microsoft Edge: Esta desarrollado por Microsoft para reemplazar a Internet Explorer y algunas de sus características a destacar son:

- Modo de navegación infantil. (Aunque las otras tienen Control Parental, pero hay que configurarlo.)
- Software Propietario

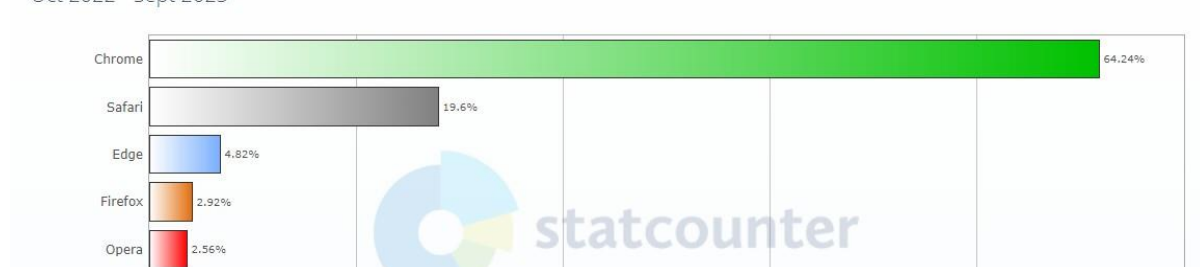
Mozilla Firefox: es un navegador de código abierto.

Comparte prácticamente todas las características anteriores que en Google Chrome.

Opera: es el más antiguo de los cinco y el código está basado en Chromium, el proyecto de software abierto de Google. La característica más destacable es:

- VPN integrada. (De manera nativa.)

Oct 2022 - Sept 2023



17. Generadores de Documentación HTML (PHPDoc).

phpDocumentor: Es uno de los generadores más antiguos y utilizados, proporciona una sintaxis simple y fácil de entender, es personalizable y extensible.

Características destacadas:

- Genera documentación en formatos como HTML, PDF, y otros.
- Permite enlazar fácilmente documentación por medio de clases, métodos y propiedades.
- Proporciona una amplia variedad de opciones de configuración para adaptarse a las necesidades específicas de un proyecto.



Link: <https://www.phpdoc.org/>

ApiGen: Se basa en phpDocumentor pero simplifica la sintaxis de comentarios, es rápido y fácil de usar, permitiendo a los desarrolladores generar documentación rápidamente.

Características destacadas:

- Proporciona una interfaz de línea de comandos simple para generar la documentación.
- Ofrece soporte para varias versiones de PHP y admite anotaciones de estilo PHPDoc.
- Permite personalización a través de temas y opciones de configuración.

Doxygen: Aunque no es específicamente un generador de documentación para PHP, es ampliamente utilizado y es compatible con varios lenguajes de programación.

Características destacadas:

- Admite varios formatos de salida, incluidos HTML, LaTeX, PDF, y más.
- Permite generar gráficos de dependencia, diagramas de clases, y otros elementos visuales para ayudar a comprender la estructura del código.
- Proporciona un gran conjunto de características avanzadas para personalizar la documentación y adaptarla a diferentes necesidades.



Link: <https://www.doxygen.nl/>

18. Repositorios de Software - Sistemas de Control de Versiones.

Git: Es el sistema de control de versiones más popular y utilizado actualmente debido a su eficiencia, velocidad y flexibilidad, es de código abierto, lo que fomenta una amplia aceptación en la industria de desarrollo de software y es utilizado por una gran cantidad de proyectos y organizaciones.



Características destacadas:

- Distribuido: Cada usuario tiene una copia local completa del repositorio y puede trabajar de forma independiente, lo que mejora la velocidad y la flexibilidad.
- Ramificación (branching) y fusión (merging): Permite trabajar en diferentes ramas de desarrollo y fusionar los cambios de manera efectiva.
- Integración con plataformas populares: Git se integra fácilmente con herramientas y servicios populares, como GitHub, GitLab y Bitbucket, facilitando la colaboración y la gestión del código fuente.

Apache Subversion (SVN): Ha sido muy utilizado durante muchos años y ofrece una transición más sencilla para aquellos que están familiarizados con sistemas de control de versiones centralizados.



Características destacadas:

- Centralizado: A diferencia de Git, SVN sigue un modelo centralizado, donde hay un único repositorio central que almacena toda la historia y las versiones del proyecto.
- Fácil administración de archivos binarios: SVN es eficaz para el manejo de archivos binarios, lo que lo hace adecuado para proyectos que contienen muchos recursos multimedia.
- Operaciones atómicas: Las operaciones de SVN son atómicas, lo que significa que se realizan por completo o no se realizan en absoluto, proporcionando una mayor integridad y coherencia en las versiones.

Mercurial: Es simple y eficaz, lo que lo convierte en una opción popular para muchos desarrolladores y proyectos, fácil de aprender y usar.

Características destacadas:

- Distribuido: Al igual que Git, Mercurial es un sistema de control de versiones distribuido, lo que permite un flujo de trabajo flexible y descentralizado.
- Rendimiento y eficiencia: Es conocido por su rapidez en operaciones cotidianas y su eficiencia en términos de uso de recursos.
- Simplicidad: La interfaz de usuario y la estructura de comandos de Mercurial son intuitivas y fáciles de entender, lo que facilita su adopción y uso.



19. Propuesta de Configuración del Entorno de Desarrollo.

Sistema operativo: Ubuntu Server 22.0.4(RAM, Disco, Particiones, Cuentas, Password, Red

Servidor administración remota :SSH

Servidor de transferencia de ficheros :SFTP (SSH)

Repositorio: GIT Hub

Servidor Web: Apache HTTP (mod_php, mod_ssl, ...),

SGBD: MySQL

Navegador: W3M

Autor: Álvaro Cordero Miñambres

20. Propuesta de Configuración del Entorno de Explotación.

Sistema operativo: Windows 10 Pro (RAM, Disco, Particiones, Cuentas, Password, Red

Servidor administración remota: SSH

Navegador: Mozilla y Google Chrome (Actualizados)

IDE: Visual Studio Code

Ofimática, multimedia: Word, Gimp.

Cliente SSH: Filezilla

Autor: Álvaro Cordero Miñambres

21. Estudio sobre CMS y ERP.

CMS (Content Management System/Sistema de Gestión de Contenidos): Es una aplicación de software que permite crear, editar, organizar y publicar contenido digital en línea de manera fácil y efectiva para usuarios sin conocimientos avanzados de programación o diseño web.

Algunos ejemplos populares de CMS son WordPress, Joomla, Drupal y Magento

ERP (Enterprise Resource Planning/Sistema de Planificación de Recursos Empresariales): Es un tipo de software que integra y centraliza las operaciones y procesos de una empresa en un sistema unificado. Su objetivo principal es optimizar la gestión de recursos y mejorar la eficiencia en todas las áreas de la organización. Algunos ejemplos de software pueden ser Odoo, Microsoft 365, NetSuite, etc.

Autor: Álvaro Cordero Miñambres