

DESARROLLO WEB EN ENTORNO SERVIDOR

TEMA 1

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

1. Protocolos de Comunicaciones

1.1. IP (Internet Protocol)

IP es el protocolo que establece las reglas para enviar y recibir datos en una red. Es la base de Internet y permite la comunicación entre diferentes dispositivos.

1.2. TCP (Transmission Control Protocol)

TCP es un protocolo de transporte que garantiza la entrega ordenada y sin errores de datos entre dispositivos. Es esencial para la comunicación fiable y establece conexiones punto a punto.

1.3. HTTP (Hypertext Transfer Protocol)

HTTP es un protocolo utilizado para transferir datos, generalmente páginas web, entre un servidor web y un navegador. Es sin estado y basado en solicitudes y respuestas.

1.4. HTTPS (HTTP Secure)

HTTPS es una extensión segura de HTTP que utiliza cifrado SSL/TLS para proteger la transferencia de datos entre el servidor y el navegador, garantizando confidencialidad y autenticidad.

1.5 HTTP (Hypertext Transfer Protocol 2)

HTTP2 está diseñado para ser más rápido, eficiente y flexible en comparación con HTTP/1.1

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

El modelo cliente-servidor implica que un cliente envía solicitudes y un servidor responde. En las aplicaciones web, el navegador actúa como cliente que solicita información al servidor web, que responde con datos.

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 y OPTIONS para obtener información sobre las opciones de comunicación permitidas.

Más utilizados GET y POST.

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.

URL (Localizador Uniforme de Recursos): Es un tipo de URI que especifica la ubicación del recurso.

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

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

5. Modelo de Desarrollo de Aplicaciones Multicapa.

Este modelo implica dividir una aplicación en capas: presentación (front-end), lógica de negocio y acceso a datos (back-end). La comunicación entre capas se realiza a través de interfaces definidas y se busca separar las responsabilidades para facilitar el desarrollo y el mantenimiento.

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

El front-end incluye la interfaz de usuario y la interacción del usuario, mientras que el back-end gestiona la lógica, procesamiento y almacenamiento de datos. Se busca dividir la aplicación para facilitar la escalabilidad y la colaboración en el desarrollo.

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

Página web estática: Contenido fijo que no cambia según la interacción del usuario.

Página web dinámica: Contenido generado en tiempo real, adaptado según la interacción del usuario o datos externos. (Control de acceso, gestión de datos)

Aplicación web: Programa interactivo ejecutado en un navegador.

Mashup: Integración de diferentes servicios web para crear una nueva aplicación compuesta.

8. Componentes de una Aplicación Web.

Incluyen la interfaz de usuario (HTML, CSS, JavaScript), lógica de aplicación (back-end), sistema gestor de base de datos y comunicación con servidores.

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

Lado del cliente: Se ejecutan en el navegador del usuario y utilizan principalmente HTML, CSS y JavaScript.

Lado del servidor: Se ejecutan en el servidor y pueden usar varios lenguajes como PHP, Python, Ruby, Java, etc.

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

Los lenguajes comunes son PHP, Python (con frameworks como Django), Ruby (con Rails), Java (con Spring), Node.js (con JavaScript), y .NET (con C#).

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

XAMPP es un paquete de software que facilita la creación de un entorno de desarrollo local para aplicaciones web. Incluye Apache, MySQL, PHP y Perl, simplificando la configuración de servidores.

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

La JVM se instala para ejecutar aplicaciones Java, mientras que el JDK se necesita para desarrollar aplicaciones Java.

13. IDEs más utilizados.

Algunos IDEs populares son Visual Studio Code, Sublime Text, IntelliJ IDEA, Eclipse y NetBeans. Ofrecen funciones como resaltado de sintaxis, depuración y control de versiones.

14. Servidores HTTP/HTTPS más utilizados.

Apache HTTP Server, Nginx y Microsoft IIS son servidores HTTP populares.

Para HTTPS, los mismos servidores se utilizan con configuraciones seguras.

15. Apache HTTP vs Apache Tomcat.

Apache HTTP Server es un servidor web para servir páginas web, mientras que Apache Tomcat es un contenedor de servlets y JSP para ejecutar aplicaciones web basadas en Java.

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

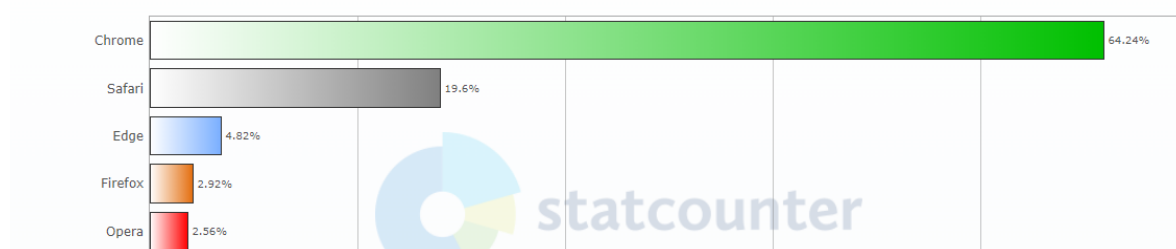
Mozilla Firefox: es un navegador de código abierto.

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.



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.



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.

Una máquina virtual con Ubuntu Server 22.04 LTS 64 Bit, con Apache, SSH y MySQL. El nombre del equipo sería DAW-USED y el desarrollo se alojaría en '/www/ProyectoDWES'

20. Propuesta de Configuración del Entorno de Explotación.

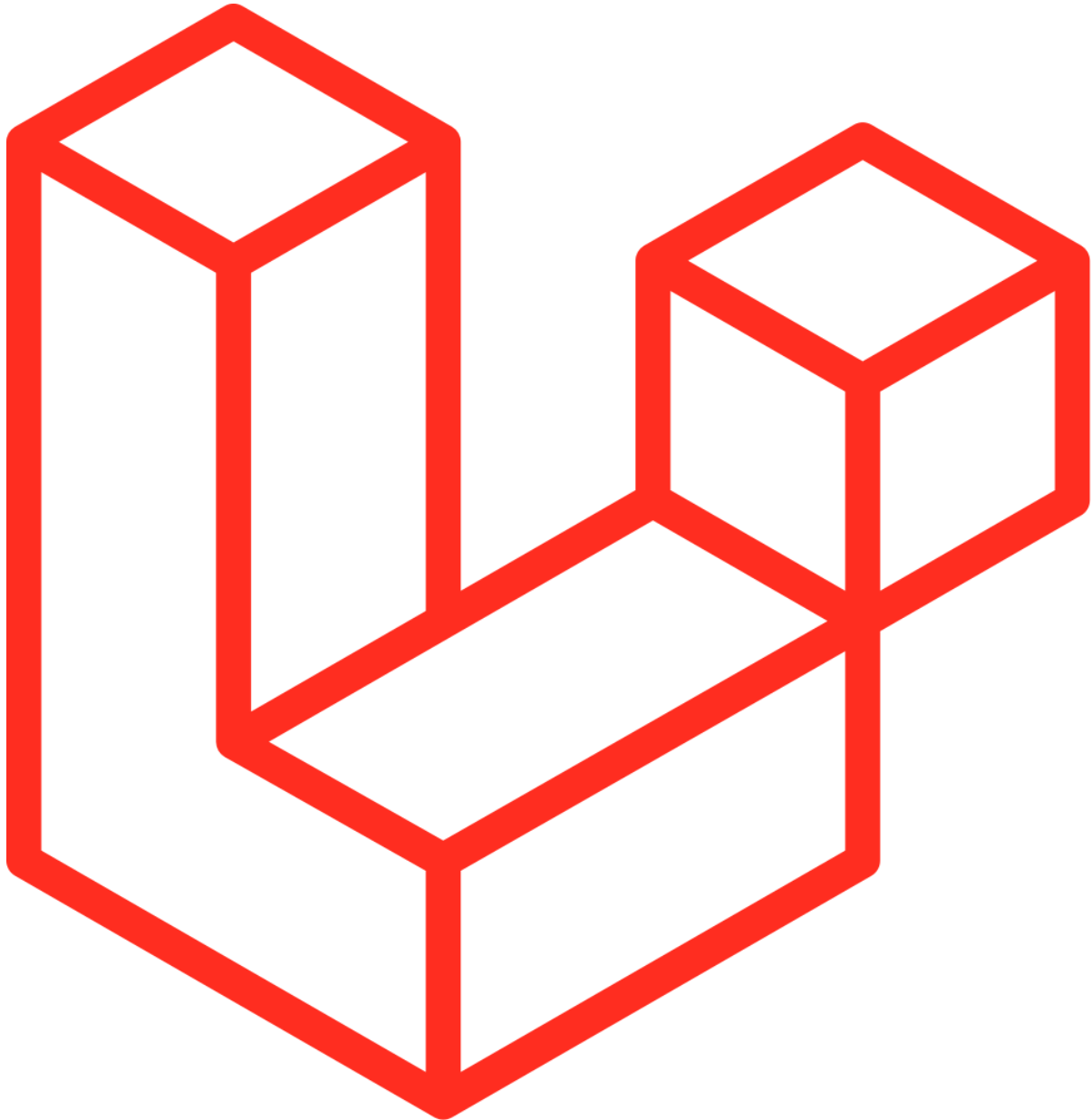
Una máquina virtual con Ubuntu Server 22.04 LTS 64 Bit, con Apache, SSH y MySQL. El nombre del equipo sería DAW-USEE y el desarrollo se alojaría en '/www/ProyectoDWES'

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.

22. Laravel.



Un popular framework de desarrollo de aplicaciones web en PHP, conocido por su sintaxis y sus potentes características que facilitan la construcción de aplicaciones web robustas y escalables.

1. Introducción a Laravel

Laravel es un framework de código abierto para el desarrollo de aplicaciones web y aplicaciones móviles utilizando el lenguaje de programación PHP. Fue creado por Taylor Otwell y lanzado en 2011. Laravel sigue el principio de desarrollo basado en el patrón de diseño Modelo-Vista-Controlador (MVC).

2. Características clave de Laravel

2.1. Artisan

Artisan es la interfaz de línea de comandos que viene con Laravel. Permite la automatización de tareas comunes de desarrollo, como migraciones de base de datos, generación de controladores, modelos y más.

2.2. Sistema de rutas

Laravel proporciona un sistema de rutas flexible y fácil de usar que permite definir las rutas de la aplicación de manera clara y concisa.

2.3. Plantillas Blade

Blade es el motor de plantillas de Laravel que permite crear vistas de manera eficiente y elegante. Tiene características como herencia de plantillas, inclusión de subvistas y directivas condicionales.

2.4. Eloquent ORM

Eloquent es un Object-Relational Mapping/ Mapeado Relacional de Objetos (ORM) avanzado que simplifica la interacción con la base de datos y facilita la manipulación de los datos utilizando modelos y consultas fluidas.

2.5. Migraciones y esquema de base de datos

Laravel proporciona un sistema de migraciones que facilita la administración y versionado de la base de datos, permitiendo a los desarrolladores mantener un control estructurado de los cambios en la base de datos.

2.6. Control de versiones y gestión de dependencias

Laravel utiliza Composer para administrar las dependencias de la aplicación y facilitar la instalación de paquetes de terceros.

2.7. Seguridad

Laravel incluye herramientas y características para garantizar la seguridad de la aplicación, como el sistema de autenticación, autorización, protección contra CSRF (Cross-Site Request Forgery) y protección XSS (Cross-Site Scripting).

3. Beneficios de utilizar Laravel

3.1. Productividad

Laravel permite a los desarrolladores ser altamente productivos al proporcionar herramientas y características que reducen la cantidad de código que deben escribir, acelerando así el desarrollo de aplicaciones.

3.2. Comunidad activa y soporte

Laravel cuenta con una comunidad activa y comprometida que contribuye con documentación, paquetes, tutoriales y soporte en línea, lo que facilita resolver problemas y obtener ayuda cuando se necesita.

3.3. Escalabilidad

Laravel permite la construcción de aplicaciones escalables mediante la implementación de prácticas recomendadas y la estructura de MVC, lo que facilita la expansión de la aplicación a medida que crece la base de usuarios y los requisitos.

3.4. Flexibilidad

Laravel es altamente flexible y permite a los desarrolladores elegir y utilizar las herramientas y bibliotecas que mejor se adapten a sus necesidades, lo que facilita la adaptación del marco a diferentes tipos de proyectos.

4. Casos de uso de Laravel

Laravel es ampliamente utilizado en una variedad de proyectos, incluyendo:

- Aplicaciones web empresariales
- Plataformas de comercio electrónico
- Sistemas de gestión de contenido (CMS)
- Aplicaciones de redes sociales
- Aplicaciones móviles (a través de Laravel API y Laravel Sanctum)

5. Conclusión

Laravel es un marco de trabajo PHP popular que brinda a los desarrolladores las herramientas necesarias para construir aplicaciones web de manera eficiente y escalable. Sus características, beneficios y casos de uso demuestran por qué es una elección popular en la comunidad de desarrollo de aplicaciones web.