

Ejercicios Tema 1

Desarrollo Web en Entorno Servidor

DAW2

IES Los Sauces

25/09/2022

Manuel Martin Alonso

David Aparicio Sir

Josué Martínez Fernández

Luis Pérez Astorga

Ricardo Santiago Tomé

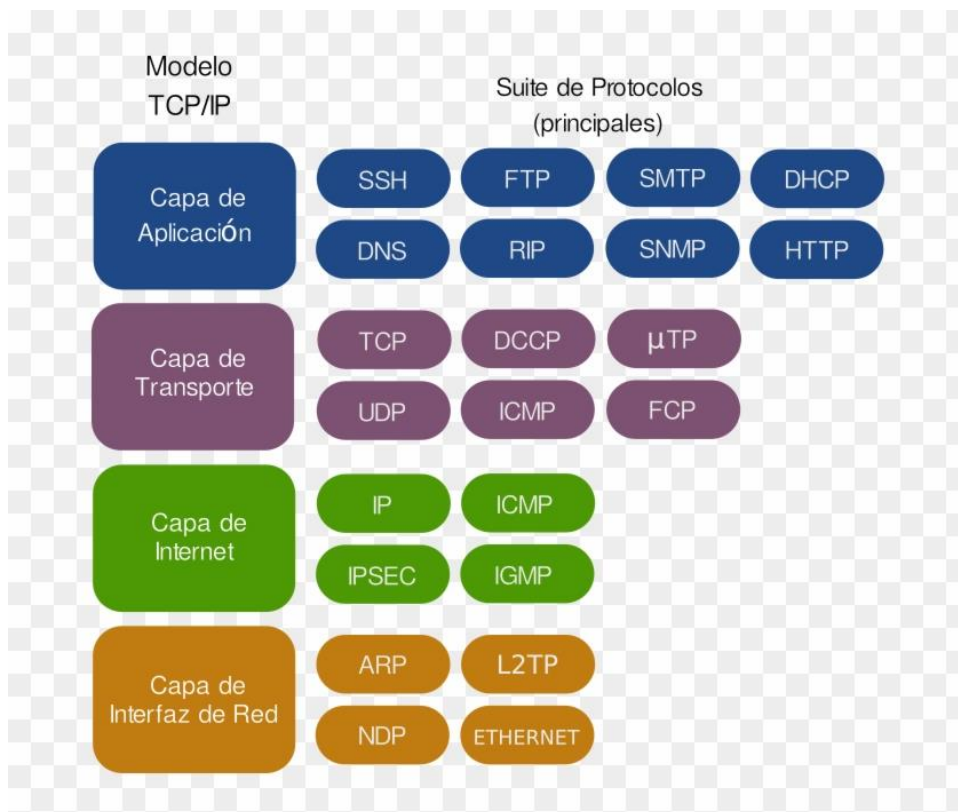
Alejandro Otálvaro Marulando

Ejercicio Tema 1 - DWES

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.[Luis Pérez Astorga]	3
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web. [Luis Pérez Astorga].....	4
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados. [Luis Pérez Astorga]	6
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme)/URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS. [Luis Pérez Astorga]	7
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa[Manuel Martín Alonso].	7
6. Modelo de división funcional front-end / back-end para aplicaciones web.[Manuel Martín Alonso]	9
7. Página web estática – página web dinámica – aplicación web – mashup[Manuel Martín Alonso].	10
8. Componentes de una aplicación web[Manuel Martín Alonso].....	13
9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.[Ricardo Santiago Tome]	13
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual). [Ricardo Santiago Tome]	14
11. Características y posibilidades de desarrollo de una plataforma XAMPP. [Ricardo Santiago Tome]	15
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? [Ricardo Santiago Tome]	16
13. IDE más utilizados (características y grado de implantación actual).[David Aparicio Sir]	17
14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual). [David Aparicio Sir]	17
15. Apache HTTP vs Apache Tomcat. [David Aparicio Sir]	18
16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual). [David Aparicio Sir]	18
17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ... [Josue Martinez Fernandez]	20
18. Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion, ... [Josue Martinez Fernandez].....	20
19. Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-WXED. [Alejandro Otálvaro Marulanda]	21
21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web[Alejandro Otálvaro Marulanda]:	22
22. Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web [Alejandro Otálvaro Marulanda]:	22

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

[Luis Pérez Astorga]



- **IP (*Internet Protocol*)**: Es un protocolo de comunicación de datos, que se encuentra dentro de la Capa de Red del modelo OSI.

El funcionamiento del protocolo IP está organizado en paquetes, dentro de estos paquetes hay una parte llamada encabezado que contiene la ip de origen, ip de destino y otra información acerca de los datos.

- **TCP (*Transmission Control Protocol*)**: Este protocolo garantiza que los datos lleguen a su destino.

- **HTTP (*Hypertext Transfer Protocol*)**: Es el protocolo utilizado para transferencia de archivos XML, HTML, CSS...

- **HTTPS (*Hypertext Transfer Protocol Secure*)**: Es como el protocolo HTTP, pero Cifrado.

2. Modelo de comunicaciones cliente - servidor y su relación con las aplicaciones web. [Luis Pérez Astorga]

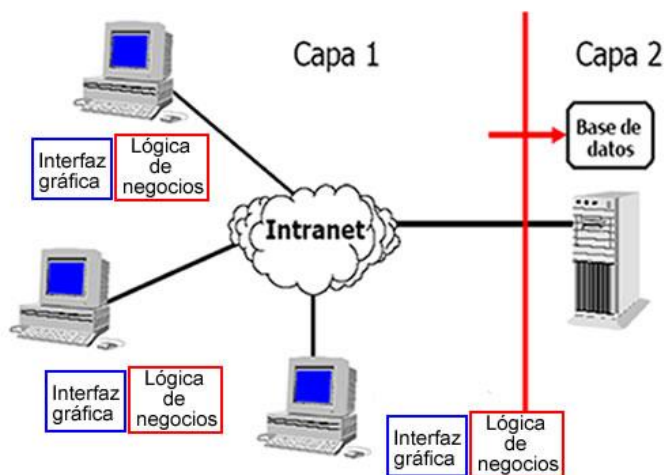
Este modelo se basa en un cliente que lanza peticiones a un servidor que está a la escucha, este a su vez cuando recibe una petición envía el recurso que el cliente a pedido mediante UDP o TCP (Dependiendo el tipo de servicio).

Dentro de este modelo se encuentran tres partes:

- Cliente: Puede ser cualquier dispositivo, lo único que necesita el software para hacer peticiones.
- Red: Es una conexión entre varios ordenadores.
- Servidor: Es un ordenador que se le instalado un software que escucha y responde peticiones del cliente.

Este modelo tiene diferentes arquitecturas:

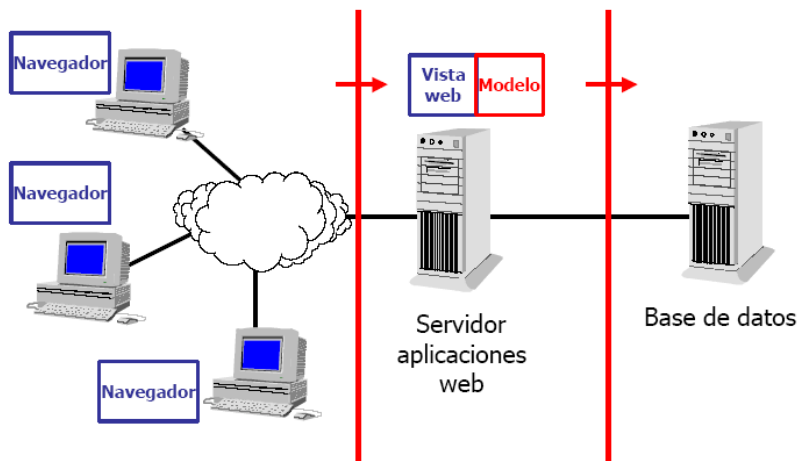
- Arquitectura de dos capas:



Esta arquitectura comunica directamente el cliente con el servidor y el responde con los recursos que tiene. Esta arquitectura puede estar separada en dos ordenadores o puede estar instalador el servidor y el cliente en el mismo ordenador

Ejercicio Tema 1 - DWES

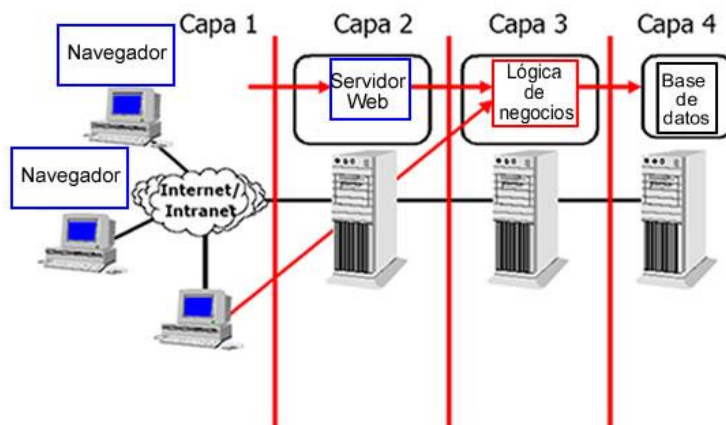
- Arquitectura de tres Capas



Esta arquitectura se basa en tres capas cliente, aplicación (servidor web) y datos.

Para que un cliente obtenga el recurso primero debe enviar una petición al servidor web este a su vez obtendrá el recurso de una base de datos, después el servidor web envía el recurso al cliente.

- Arquitectura de N Capas:



Esta arquitectura se parece mucho a la de tres capas, pero el servidor web puede utilizar recursos de otro servidor para dar su servicio al cliente.

3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados. [Luis Pérez Astorga]



GET: Solicita una representación de un recurso en específico. No puede enviar datos a excepción si dichos datos se envían como un parámetro en la URL que realiza la petición.

HEAD: Es similar al Get pero este no envía datos como un parámetro URL.

POST: Este método puede enviar datos al servidor por medio del cuerpo(body). EL tipo de cuerpo de solicitud se define en la cabecera Content-Type.

PUT: Crea un nuevo elemento o reemplaza una representación del elemento de destino con los datos de la petición. Este método es Idempotente (Significa que da igual cuantas veces lo llames siempre va a tener el mismo efecto).

DELETE: Nos permite eliminar un recurso específico. Este método es Idempotente (Significa que da igual cuantas veces lo llames siempre va a tener el mismo efecto).

CONNECT: Inicia la comunicación entre el cliente y el servidor, también se utiliza para crear una comunicación de túnel.

OPTIONS: Comprueba que opciones de comunicación posee el recurso de destino

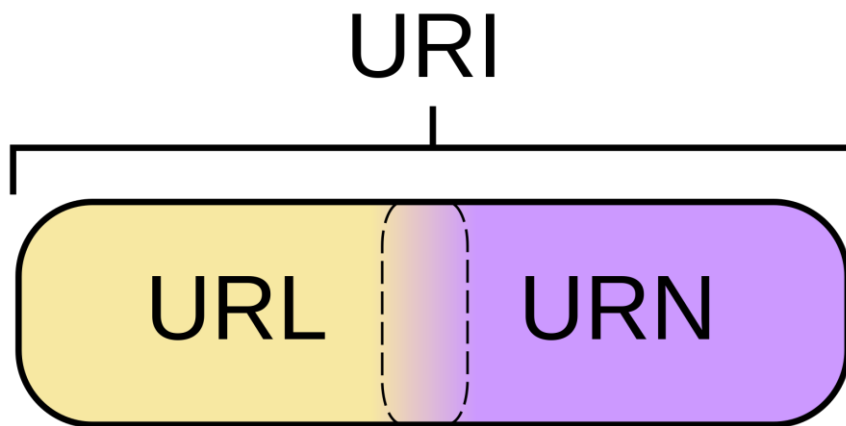
TRACE: Efectúa una prueba del camino que efectúa el recurso.

PATCH: Nos permite hacer modificaciones parciales de un recurso en concreto.

Los métodos más usados actualmente son el GET y el POST

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

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



Video Explicativo

La relación de estos conceptos con HTTP y HTTPS. Se utilizan para obtener recursos de un servidor web.

5. Modelo de desarrollo de aplicaciones multicapa - comunicación entre capas - componentes - funcionalidad de cada capa [Manuel Martín Alonso].

La **programación por capas** es un **modelo de desarrollo software** en el que el objetivo es la **separación (desacoplamiento)** de las partes que componen un sistema software o también una arquitectura cliente-servidor.

El modelo multicapas se divide en tres capas:

- **Capa de presentación:** es la **capa encargada de interactuar con el usuario de la aplicación mediante una interfaz de usuario**. Esta capa se comunica únicamente con la capa de negocio.

Ejercicio Tema 1 - DWES

- **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- **Capa de datos:** es la **capa encargada de gestionar el almacenamiento de los datos**, generalmente en un sistema gestor de bases de datos relacionales, y de la comunicación del sistema con cualquier otro sistema que realice tareas auxiliares.

https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas

<http://elvex.ugr.es/decsai/csharp/design/layers.xml>

<https://docplayer.es/2857560-Tema-6-arquitectura-web-multicapa.html>

https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas#:~:text=Capa%20de%20negocio%3A%20es%20donde,las%20reglas%20que%20deben%20cumplirse

Resumen:

El **desarrollo de aplicaciones multicapa** es un modelo de desarrollo software en el que el objetivo principal es la separación de las partes que componen un sistema software.

Este modelo se divide en tres capas:

- **Capa de presentación:** capa encargada de interactuar con el usuario de la aplicación mediante una interfaz de usuario.
- **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso.
- **Capa de datos:** es la capa encargada de gestionar el almacenamiento de datos.

6. Modelo de división funcional front-end / back-end para aplicaciones web. [Manuel Martín Alonso]

Front-end: es el conjunto de páginas que ven la gran mayoría de usuarios que las usan (usuarios externos).

Quienes desarrollan el Front-end son responsables del código orientado a la inmersión del sujeto en el sitio. Para lograrlo los expertos deben conocer tres idiomas esenciales:

- Javascript;
- CSS;
- HTML.
- Así como estar familiarizados con frameworks como:
- Foundation;
- Bootstrap;
- AngularJS;
- Ember JS;
- Backbone.

Back-end: es un conjunto de páginas dinámicas que utilizan las personas que producen el contenido y las que administran la aplicación web (usuarios internos) para crear contenido, organizarlo, decidir la apariencia externa, etc.

Por lo general, dicho código está compuesto por uno o más lenguajes de programación empleados para orientar el desarrollo de estructuras web.

<https://rockcontent.com/es/blog/front-end-y-back-end/>



7. Página web estática - página web dinámica - aplicación web - mashup[Manuel Martín Alonso].

Página web estática:

Se conoce como web estática a aquella que es meramente informativa, cuyo principal objetivo es el de mostrar una información de forma permanente. Únicamente puede darse el caso de envío de formularios de contacto, o suscripciones.

Ventajas:

- Proporciona una mayor seguridad en los sitios web dinámicos (los sitios web dinámicos corren el riesgo de sufrir ataques de web shell si existe una vulnerabilidad).
- Mejor rendimiento para los usuarios finales en comparación con los sitios web dinámicos.
- Menos o ninguna dependencia de sistemas como bases de datos u otros servidores de aplicación.
- Ahorro de costos por la utilización del almacenamiento en la nube, en lugar de un entorno alojado.

Desventajas:

- Una página web estática puede ser más difícil de actualizar.
- Agregar contenido a la página web o realizar actualizaciones puede incurrir en costos adicionales.
- Agregar nuevas páginas o funcionalidades a una web estática puede ser más difícil que hacerlo para una web dinámica.

<https://axarnet.es/blog/pagina-web-estatica>

https://es.wikipedia.org/wiki/P%C3%A1gina_web_est%C3%A1tica#Ventajas_de_un_sitio_web_est%C3%A1tico

<https://openwebinars.net/blog/paginas-web-estaticas-vs-paginas-web-dinamicas/>

Página web dinámica

Una página web dinámica es una página web generada bajo demanda, a diferencia de una página web estática. Su contenido varía en función de los datos como la hora, el nombre del usuario, las preferencias fijadas por el usuario, etc.

Los sitios web dinámicos basan su comportamiento y funcionalidad en dos tipos de programación, front-end (del lado del cliente) y back-end (del lado del servidor). Las instrucciones del lado del cliente es código JavaScript que se ejecuta en el navegador. Mientras que las instrucciones que se ejecutan del lado del servidor son instrucciones escritas en lenguajes de scripting o programación, como ASP.Net, PHP, Python, etc.

Ventajas:

- Puede gestionar información a través de bases de datos.
- El contenido se puede gestionar a través de un CMS.
- El coste de mantenimiento es menor.

Desventajas:

- Pueden existir limitaciones en el diseño: dependiendo del CMS, puede resultar difícil crear varios diseños o plantillas que permitan mostrar diferentes tipos de contenido de diferentes formas.
- Puede involucrar altos costos de construcción iniciales: Al coste del desarrollo de la página web se le suma el coste del desarrollo de las bases de datos donde se guardará el contenido a mostrar, etc.

El esquema de funcionamiento de una página web dinámica es el siguiente:

Pasos:

1. El cliente web (navegador) de tu ordenador solicita a un servidor web una página web.
2. El servidor busca esa página y la recupera.
3. En el caso de que se trate de una página web dinámica, es decir, que su contenido deba ejecutarse para obtener el HTML que se devolverá, el servidor web contacta con el módulo responsable de ejecutar el código y se lo envía.
4. Como parte del proceso de ejecución, puede ser necesario obtener información de algún repositorio (Cualquier almacén de información digital, normalmente una base de datos), como por ejemplo consultar registros almacenados en una base de datos.

Ejercicio Tema 1 - DWES

5. El resultado de la ejecución será una página en formato HTML, similar a cualquier otra página web no dinámica.
6. El servidor web envía el resultado obtenido al navegador, que la procesa y muestra en pantalla.

https://es.wikipedia.org/wiki/P%C3%A1gina_web_din%C3%A1mica

<https://openwebinars.net/blog/paginas-web-estaticas-vs-paginas-web-dinamicas/>

Una aplicación web es aquella herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

Una Mashup o aplicación web híbrida es una aplicación que combina servicios de varias páginas web en una sola para ofrecer un nuevo servicio. Es habitual que la integración de datos se obtenga desde API's abiertas.

Ventajas de las Mashups:

- Generación rápida de contenido. La integración con soluciones ya existentes permite acortar los tiempos de desarrollo.
- Facilidad de uso. Cualquiera con conocimientos de desarrollo puede crear servicios novedosos integrando las funcionalidades ya existentes con otras propias.
- Sinergias. Las marcas pueden crear experiencias innovadoras sin que los usuarios tengan que salir de sus páginas web.

<https://www.arimetrics.com/glosario-digital/mashup-aplicacion-web-hibrida>

https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web#:~:text=En%20la%20ingenier%C3%ADa%20de%20software,una%20intranet%20mediante%20un%20navegador.

Ejemplos de Mashups:

- Google Maps
- Flickr



8. Componentes de una aplicación web [Manuel Martín Alonso].

- Servidor web para recibir las peticiones de los clientes web (normalmente navegadores) y enviarles la página que solicitan (una vez generada puesto que hablamos de páginas web dinámicas). El servidor web debe conocer el procedimiento a seguir para generar la página web: qué módulo se encargará de la ejecución del código y cómo se debe comunicar con él.
- Módulo encargado de ejecutar el código o programa y generar la página web resultante. Este módulo debe integrarse de alguna forma con el servidor web, y dependerá del lenguaje.
- Sistema gestor de base de datos (si es necesario): que normalmente también será un servidor. Este módulo no es estrictamente necesario, pero en la práctica se utiliza en todas las aplicaciones web que utilizan grandes cantidades de datos para almacenarlos.
- (Ficheros escritos en) Lenguajes de programación que utilizarás para desarrollar las aplicaciones.

9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso. [Ricardo Santiago Tomé]

Lado del cliente:

- Javascript.

Lado del servidor:

- Javascript.
- PHP.
- C#.
- C, C++.
- VB.Net.
- Python.
- Ruby.
- Perl.
- Java.
- PL/SQL.

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

actual). [Ricardo Santiago Tomé]

Para el grado de implantación voy a usar el índice [TIOBE](#): un índice que sitúa en un ranking numerado por posiciones de más usado a menos y asigna un porcentaje a cada lenguaje según su grado de implantación entre los desarrolladores. Este ranking lo realiza en base a las consultas realizadas a través de los principales motores de búsqueda a nivel mundial.

- **Javascript** es un lenguaje interpretado orientado a objetos. Cuando se utiliza del lado del servidor se le denomina **SSJS (Server side JavaScript)**. 7ª posición en el índice **TIOBE** con un 2,82% de implantación.
- **PHP** es un lenguaje interpretado, libre y orientado a objetos. Diseñado sobre todo para hacer aplicaciones web y es especialmente útil en el acceso a datos. Ocupa la 10ª posición en el ranking **TIOBE** con un 1,68% de implantación.
- **C#** es una evolución de los lenguajes **C** y **C++**; a diferencia de los dos anteriores es compilado y no interpretado, también orientado a objetos. Puesto 5ª en el ranking y cuota de implantación del 4,88%.
- **C, C++**. Estos dos lenguajes son una evolución uno del otro y ambos están en la base de **C#**. Los dos son compilados y su principal diferencia es que **C** a pesar de ser un lenguaje de alto nivel, posee características de los de bajo nivel, pudiendo mezclar su código con ensamblador y acceder directamente a memoria o dispositivos periféricos. Por otra parte, **C++** añade a **C** la orientación a objetos. Ambos están ampliamente implantados y ocupan los puestos 2º de la lista **TIOBE** para **C** con un 13,96% de implantación y 4º para **C++** con un 9,76%.
- **VisualBasic.Net**. Es tanto interpretado como compilado y se entre sus características figura el estar orientado a objetos. Es heredero de **VisualBasic** y como tal, desarrollado por **Microsoft**. 6º puesto en el ranking con un 4,39% de implantación en el mercado.
- **Python**. Es un lenguaje interpretado y multiparadigma, aunque su fuerte es la orientación a objetos. No fue diseñado con la idea de hacer aplicaciones Web, pero gracias al framework **Django** está ganando popularidad para la creación de páginas **HTML**. Actualmente es el lenguaje nº1 en uso en el mundo con un 15,74% de cuota de implantación.

- **Perl**. Multiparadigma incluyendo la orientación a objetos, pertenece al grupo de los lenguajes interpretados. 19ª puesto del ranking con un 0.72% de cuota de mercado.
- **Ruby**. Software libre, interpretado y orientado a objetos. Último puesto en el ranking de los 20 principales lenguajes de programación **TIOBE**, implantación del 0.66%.
- **Java**. No es solo un lenguaje, es toda una plataforma de aplicaciones informáticas. Como lenguaje es compilado a un lenguaje intermedio (**bytecode**) y este es después interpretado, también está orientado a objetos. Es multiplataforma, pudiendo ejecutarse en cualquiera de ellas gracias a la **JVM (Java Virtual Machine)**. Esta máquina virtual de **Java** puede ejecutarse incluso en un navegador web, de ahí su 3º puesto en el ranking con un 11.72% de cuota.
- **PL/SQL o Procedural Language/Structured Query Language** es un lenguaje de consultas a bases de datos desarrollado por **Oracle** que complementa a **SQL** añadiendo procedimientos para poder ejecutar bloques de código completos, a diferencia de **SQL**, que solo puede realizar las consultas de una en una. Es el 9º e en el ranking de lenguajes más usados con un 2.01% de cuota de uso.

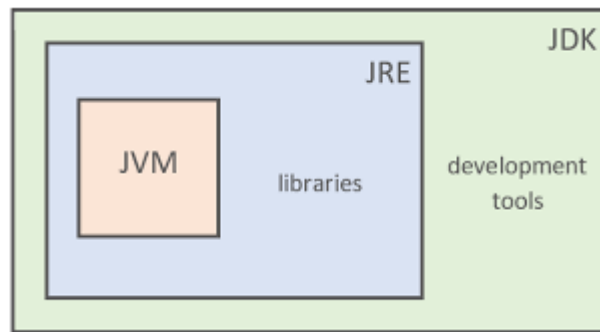
11. Características y posibilidades de desarrollo de una plataforma XAMPP. [Ricardo Santiago Tomé]



XAMPP es un servidor independiente de plataforma, software libre, que consiste en la base de datos **MySQL**, el servidor web **Apache** y los intérpretes para **PHP** y **Perl**. En concreto, la X de **XAMPP** hace referencia a la plataforma orientada al desarrollo en **Windows**, cuando está orientada a **MAC** se denomina **MAMPP** y cuando lo es a **Linux**, se la llama **LAMPP**.

Te permite instalar de forma sencilla **Apache** en tu propio ordenador, sin importar tu sistema operativo (**Linux**, **Windows**, **MAC** o **Solaris**) también permite probar tu trabajo (páginas web o programación, por ejemplo) en tu propio ordenador sin necesidad de tener que acceder a internet.

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 [Ricardo Santiago Tomé]



La **Java Virtual Machine (JVM)** forma parte del entorno **Java Runtime Environment (JRE)** y es necesaria cuando vamos a ejecutar una aplicación programada en **Java**, por lo tanto, ha de estar presente tanto en el entorno de desarrollo como en el de explotación. En el de desarrollo para probar el funcionamiento del código escrito en **Java** y en el de explotación para que el usuario final pueda usarla.

Por su parte, el **Java Development Kit (JDK)**, que también incluye el **JRE** y por lo tanto el **JVM** solo estará presente en el entorno de desarrollo al tratarse de la principal herramienta para la escritura y prueba de las aplicaciones, es imprescindible en este entorno, pero innecesaria en el de explotación, pues en él, el usuario final no tiene por qué tener acceso al código fuente de las aplicaciones, únicamente a la ejecución de las mismas.

<https://dharmawebstudio.com/blog/conoces-los-entornos-de-trabajo-utilizados-en-el-desarrollo-de-una-aplicacion-web>



13. IDE más utilizados (características y grado de implantación actual). [David Aparicio Sir]

Eclipse: Eclipse dispone de un Editor de texto con un analizador sintáctico. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, test, etc., y refactorización.

NetBeans:

- Gestión de la interfaz de usuario (menús y barras de herramientas)
- Gestión de configuración de usuario
- Gestión de almacenamiento (guardar o cargar algún tipo de dato)
- Gestión de ventana
- Marco Asistente (soporta diálogos paso a paso)
- Librería visual de NetBeans
- Herramientas de desarrollo integrado
- NetBeans IDE es libre, código abierto, multiplataforma con soporte integrado para el lenguaje de programación Java.
- Visual Studio Code
- Resaltado de Sintaxis
- Snippets
- Autocompletado de código
- Refactorización
- Depuración

IntelliJ IDEA: Sus funciones incluyen coincidencia de patrones, automatización del flujo de trabajo, control de versiones, análisis de código estático, pruebas de unidades, descompilador integrado y atajos de teclado.

14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual). [David Aparicio Sir]

Apache: Servidor HTTP gratuito programado en C. Es gratuito y de fuente abierta. Instalación y configuración sencilla, altamente extensible y adaptable mediante módulos, funciones incorporadas para autenticación y validación de usuarios. Soporte para lenguajes como Perl, PHP y Python.

Ejercicio Tema 1 - DWES

Tomcat: Servidor HTTP gratuito programado en Java

Nginx: Servidor HTTP gratuito programado en C Servidor de archivos estáticos, índices y auto indexado.

Proxy inverso con opciones de caché.

Balanceo de carga.

Tolerancia a fallos.

Soporte de HTTP y HTTP2 sobre SSL.

Soporte para FastCGI con opciones de caché.

Servidores virtuales basados en nombre y/o en dirección IP.

Streaming de archivos FLV y MP4.24

Soporte para autenticación.

Compatible con IPv6

Soporte para protocolo SPDY

Compresión gzip.

Habilitado para soportar más de 10 000 conexiones simultáneas

Fue inicialmente creado para Rambler ahora se utiliza en WordPress, Badoo...

15. Apache HTTP vs Apache Tomcat. [David Aparicio Sir]

La diferencia fundamental es que Tomcat proporciona contenido dinámico empleando lógica basada en Java, mientras que el propósito principal del servidor web Apache es simplemente brindar contenido estático como HTML, imágenes, audio y texto.

16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual). [David Aparicio Sir]

Google Chrome programado en C++, Python y JavaScript

- Add-ons y extensiones web.
- Alertas de filtraciones de datos.
- Autocompletar formularios.
- Modo oscuro.

Ejercicio Tema 1 - DWES

- Múltiples perfiles.
- Gestión de contraseñas.
- Picture-in-picture.
- Navegación privada.
- Modo de lectura.
- Corrección ortográfica.
- Generación de contraseñas seguras.
- Sincronización en múltiples dispositivos.
- Navegación con pestañas.
- Grupos de pestañas.
- Texto a voz

Cuota del 64.9%

Safari programado en C++,Objective-C y Swift

- Add-ons y extensiones web.
- Alertas de filtraciones de datos.
- Autocompletar formularios.
- Modo oscuro.
- Gestión de contraseñas.
- Picture-in-picture.
- Navegación privada.
- Notas rápidas.
- Modo de lectura.
- Corrección ortográfica.
- Generación de contraseñas seguras.
- Sincronización en múltiples dispositivos.
- Navegación con pestañas.
- Grupos de pestañas.
- Texto a voz.
- Cookies de rastreo de terceros y rastreados sociales bloqueados.
- Streaming de vídeo en 4K.

Cuota del 15.4%

Edge

- Add-ons y extensiones web.
- Alertas de filtraciones de datos.
- Autocompletar formularios.
- Modo de navegación infantil.
- Modo oscuro.
- Múltiples perfiles.
- Gestión de contraseñas.

Ejercicio Tema 1 - DWES

- Picture-in-picture.
- Navegación privada.
- Modo de lectura.
- Corrección ortográfica.
- Generación de contraseñas seguras.
- Sincronización en múltiples dispositivos.
- Navegación con pestañas.
- Grupos de pestañas.
- Texto a voz.
- Cookies de rastreo de terceros y rastreados sociales bloqueados.

Cuota de 6.3%

17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ... [Josué Martínez Fernández]

Son un conjunto de herramientas que permiten generar documentación de forma automática a partir del código fuente.

[Enlace explicativo](#)

18. Repositorios de software - sistemas de control de versiones: GIT, CVS, Subversión, ... [Josué Martínez Fernández]

Un repositorio de software es un lugar de almacenamiento del cual pueden ser recuperados e instalados los paquetes de software en un ordenador.

Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo.

19. Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-WXED. [Alejandro Otálvaro Marulanda]

Herramientas de desarrollo de aplicaciones web:

Sistema operativo: Ubuntu Server (Versión 22.04 LTS)

Servidor administración remota: SSH

Servidor de transferencia de ficheros: SFTP (SSH)

Repositorio: GIT Hub

Servidor Web: Apache (Versión a elegir por la asignatura de despliegue para evitar conflictos)

SGBD: MySQL (Ultima versión con soporte de largo plazo)

Navegador: Mozilla, Chrome, ... (Actualizados)

IDE: Microsoft Visual Estudio Code (Actualizado)

20. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE. [Alejandro Otálvaro Marulanda]:

Sistema operativo: Ubuntu Server (Versión 22.04 LTS)

Servidor administración remota: SSH

Servidor de transferencia de ficheros: SFTP (SSH)

Repositorio: GIT

Servidor Web: Apache HTTP, XAMPP

SGBD: Mariadb / MySQL

Cliente ssh: NetBeans / Putty

Navegador: Mozilla, Chrome

21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web [Alejandro Otálvaro Marulanda]:

CMS – Sistema de gestión de contenidos: Son las siglas de Content Management System. Es un sistema online que nos permite poner en marcha un sitio web.

ERP – Sistema de planificación de los recursos empresariales: Es un conjunto de aplicaciones de software integradas que nos permiten automatizar la mayoría de las prácticas de un negocio. Una de sus principales características es su modularidad y escalabilidad, es decir, la posibilidad de implementar las funciones que se necesiten en cada momento.

Un ejemplo de ello es el sistema SAP.

22. Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web [Alejandro Otálvaro Marulanda]:

- **MEAN:** (con MongoDB y con MySQL) MEAN Stack (acrónimo para MongoDB, Express.js, AngularJS, Node.js) es un framework o conjunto de subsistemas de software para el desarrollo de aplicaciones y páginas web dinámicas que están basadas, cada una de estas, en el lenguaje de programación JavaScript. Gracias a esta característica el conjunto se integra exitosamente en una plataforma autosuficiente.
- **Java EE vs Spring**
 - Java EE: Java Platform Enterprise Edition es una plataforma de programación. Sirve para desarrollar y ejecutar software de aplicaciones en Java.
 - Spring: Es un framework para el desarrollo de aplicaciones Java. Es de código abierto.
Las características pueden ser usadas en cualquier aplicación Java, pero hay extensiones para la construcción de aplicaciones web sobre Java EE.
- **Microsoft .NET:** Es una plataforma de aplicaciones que permite la creación y ejecución de servicios web y aplicaciones de internet. Es una colección de diferentes plataformas de software de Microsoft. Se desarrollo como competencia directa de Java.

Ejercicio Tema 1 - DWES

- **Angular 7:** Es un framework de JS de código abierto escrito en TypeScript, cuyo objetivo principal es desarrollar aplicaciones de una sola página. Google se encarga del mantenimiento y constantes actualizaciones de mejoras.
- **Symfony:** Symfony es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Esta desarrollado en php.
- **Laravel:** Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5, PHP 7 y PHP 8. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti".
- **CakePHP:** Es un framework que facilita la actividad de programación de PHP.
 - **Modelo:** Contiene los datos y el mecanismo para acceder a la información, utiliza una base de datos para actualizar el estado de la aplicación.
 - **Vista:** Contiene el código HTML que produce la visualización a la interfaz del usuario.
 - **Controlador:** Conecta el modelo y la vista a través de un código para responder a las acciones dentro de la aplicación.