

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

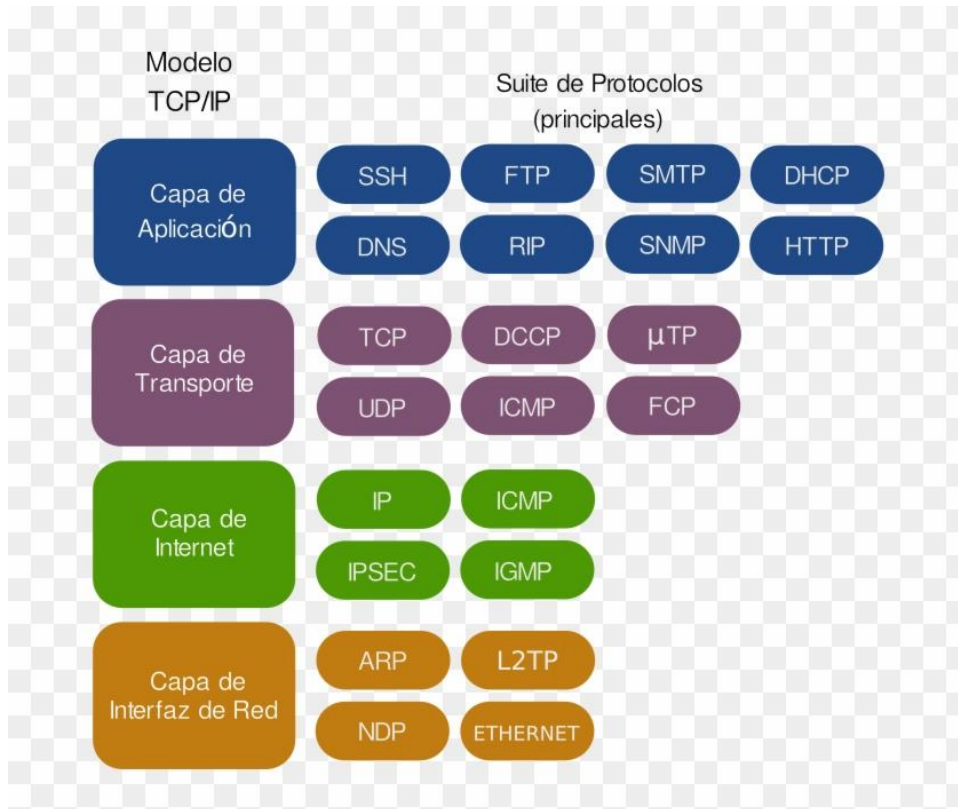
Ricardo Santiago Tomé

Alejandro Otálvaro Marulanda

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]	5
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]	6
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].	11
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.	14
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).	14
11. Características y posibilidades de desarrollo de una plataforma XAMPP.	17
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.	18
13. IDE más utilizados (características y grado de implantación actual). [David Aparicio Sir]	19
14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual). [David Aparicio Sir].....	20
15. Apache HTTP vs Apache Tomcat. [David Aparicio Sir]	22
16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual). [David Aparicio Sir].....	22
17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ... [Josué Martínez Fernández]	24
18. Repositorios de software – sistemas de control de versiones: GIT, CVS, Subversión, ... [Josué Martínez Fernández]	24
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. [Josué Martínez Fernández]	25
21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web [Alejandro Otálvaro Marulanda]:	26
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]:	26

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



- **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.

- **UDP (*User Datagram Protocol*)**: Este protocolo no garantiza la llegada de los paquetes al 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.

Realizado por Luis Pérez Astorga.

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

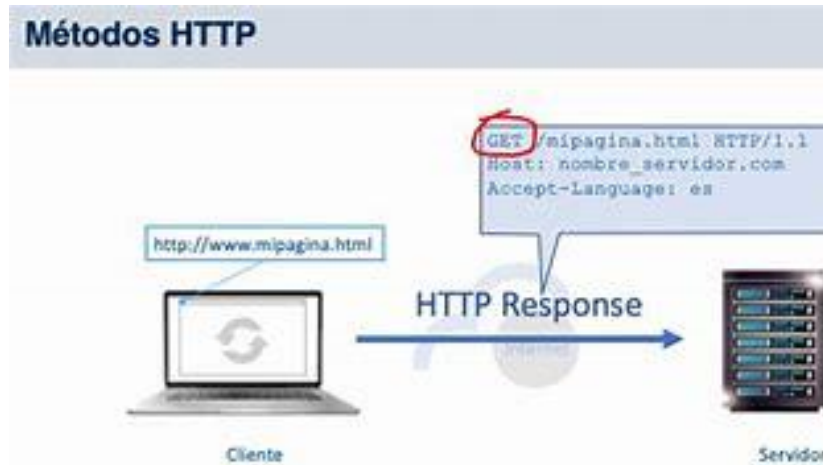
Este modelo se basa en un cliente que lanza peticiones a un servidor que está a la escucha, este a su vez cuando recibe una petición envía el recurso que el cliente a pedido mediante UDP o TCP (Dependiendo el tipo de servicio).

Dentro de este modelo se encuentran tres partes:

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

Realizado por Luis Pérez Astorga.

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



GET: Solicita una representación de un recurso en específico. No puede enviar datos, solo recuperarlos, la excepción se da cuando 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.

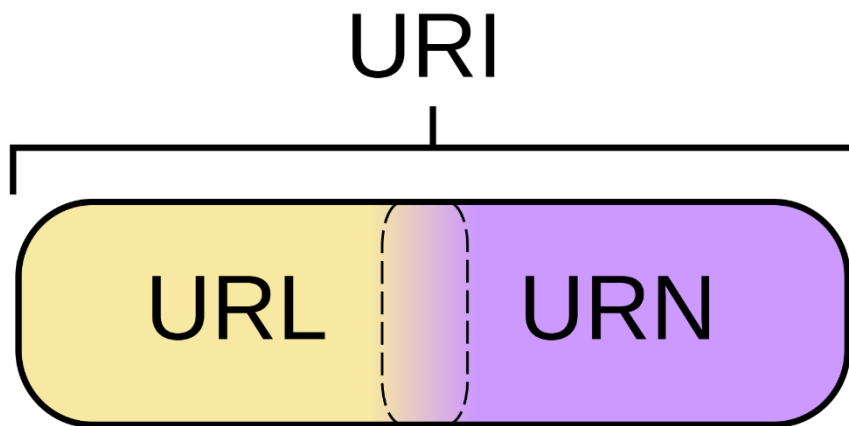
PATCH: Nos permite hacer modificaciones parciales de un recurso en concreto.

Los métodos más usados actualmente son el GET y el POST

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

Realizado por Luis Pérez Astorga.

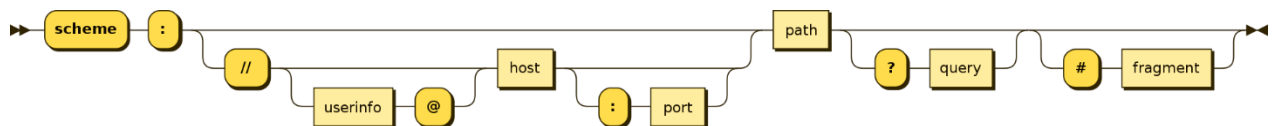
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.



URI: Es una cadena de caracteres que identifica un recurso físico o lógico. Es el conjunto de URL y URN

URL: Es el identificador del servidor.

URI: Identifica el recurso que le vamos a pedir al servidor.



[Video Explicativo](#)

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

Explicar

Realizado por Luis Pérez Astorga.

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

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.
- **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.
- **Capa de controlador:** es la capa que gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista.

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

<https://book.cakephp.org/2/es/cakephp-overview/understanding-model-view-controller.html#:~:text=La%20capa%20del%20controlador%20gestiona,modelo%20com%20de%20la%20vista.>

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.
- **Capa de controlador:** es la capa que gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista.

Realizado por Manuel Martín Alonso

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

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

- **Campos de membresía personalizables para el registro de usuarios front-end.**
- **Formulario de inicio de sesión front-end de estilo CSS puro, formulario de registro de usuario y formulario de perfil de edición.**
- **Los usuarios front-end admiten todos los tipos de entrada para los campos, lo que le permite crear un perfil de usuario personalizado.**
- **Incluya diferentes niveles de usuario de front-end / roles de usuario y restrinja el acceso al contenido en consecuencia.**
- **Opción para enviar correos electrónicos de registro y para requerir la aprobación del administrador de los usuarios de la interfaz.**
- **Redirecciones basadas en entrada de usuario de front-end.**

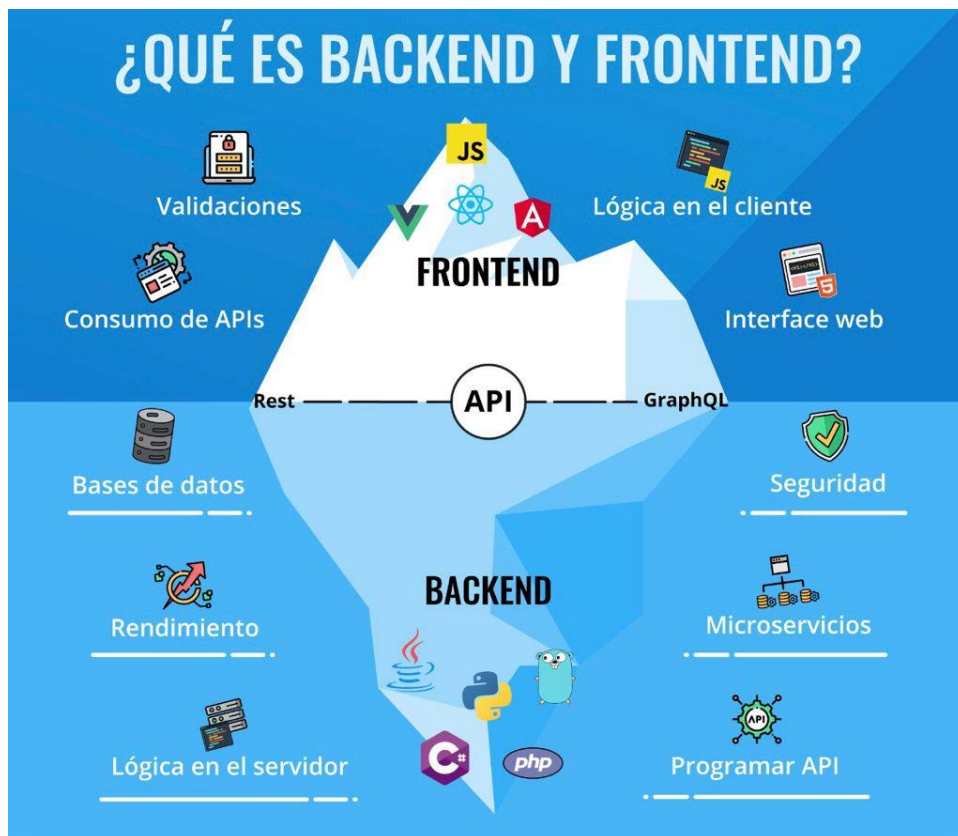
Back-end: es un conjunto de páginas dinámicas que utilizan las personas que producen el contenido y las que administran la aplicación web (usuarios internos).

- **Acceder a la información que se pide, a través de la app.**
- **Combinar la información encontrada y transformarla.**
- **Devolver la información al usuario.**
- **La subcapa lógica, relacionada con el lenguaje específico.**
- **La subcapa visual, relacionada con el diseño estético de la app.**

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

<https://kolmite.com/solo-usuarios-front-end/>

<https://www.mmaglobal.com/news/todo-lo-que-necesitas-saber-sobre-backend-all-you-need-know-regarding-backend#:~:text=El%20Backend%2C%20tambi%C3%A9n%20conocido%20como,y%20d%20evolverla%20al%20usuario%20final>



Realizado por Manuel Martín Alonso

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

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.

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**.

<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.

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.

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.

<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



Realizado por Manuel Martín Alonso

8. Componentes de una aplicación web.

- **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.
- **Aplicación:** aquella herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

Realizado por Manuel Martín Alonso

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

El programa que se ejecuta del lado del cliente es el navegador web, en función del lenguaje en que esté programada la aplicación web harán un uso más o menos intensivo de la capacidad de procesamiento del navegador.

En el lado del cliente tenemos desde sistemas gestores de bases de datos, pasando por intérpretes de los diferentes lenguajes hasta el servidor en sí mismo, que no es otra cosa que un programa que recibe peticiones y sirve respuestas en forma de documentos HTML.

También podemos considerar programas a las librerías y los frameworks que se utilizan para desarrollar con los distintos lenguajes de una manera más eficiente y práctica.

Preguntar a Heraclio: Los IDE's son programas que se ejecutan en el entorno de desarrollo, entonces ¿ES pertinente decir que también están presentes en el lado del servidor?

Lenguajes lado del cliente:

- Javascript. A través de React, Ember, Vue, Angular, Node.
- Visual Basic Script. En desuso al ser solo compatible con Internet Explorer.
- TypeScript.
- Applets de Java.

Lenguajes lado del servidor:

- Javascript.
- PHP. Frameworks Laravel o Symfony.
- C#.
- C, C++.
- VB.Net.
- Python.
- Ruby.
- Perl.
- Java Server Pages (JSP). Servlets de Java. Framework Spring.
- PL/SQL. Bases de datos.

Realizado por Ricardo.

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

Para el grado de implantación voy a usar el índice de la w3techs.com un índice que sitúa en un ranking numerado por posiciones de más usado a menos y asigna un

porcentaje a cada lenguaje. Esto lo hace investigando la presencia de los distintos lenguajes en una muestra de 10.000.000 de sitios web.

- **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 1ª posición en el ranking **w3techs** con un 77,5% de implantación.
- **ASP.NET** no es un lenguaje de programación, es un entorno para aplicaciones web desarrollado y comercializado por **Microsoft**. Apareció en enero de 2002 con la versión 1.0 del **.NET Framework**, y es la tecnología sucesora de la tecnología **Active Server Pages (ASP)**. **ASP.NET** está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código **ASP.NET** usando cualquier lenguaje admitido por el **.NET Framework**. En este ranking aparece en 2ª posición con un porcentaje de 7,6%. El principal lenguaje en que se programa en este entorno es [Visual Basic](#) V, del que dejo descripción en la segunda parte de esta lista.
- **Ruby**. Software libre, interpretado y orientado a objetos. Tercer puesto en el ranking de los 11 principales lenguajes de programación **w3techs**, implantación del 5.7%.
- **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 4º puesto en el ranking con un 4,5% de cuota.
- **Scala** Es un lenguaje de programación multi-paradigma, compilado e interpretado, diseñado para expresar patrones comunes de programación en forma concisa y con tipos seguros. Integra características de lenguajes funcionales y orientados a objetos. La implementación actual se ejecuta en la máquina virtual de Java y es compatible con las aplicaciones Java existentes. Fuente [Wikipedia](#).
- **Javascript** es un lenguaje interpretado orientado a objetos. Cuando se utiliza del lado del servidor se le denomina **SSJS (Server side JavaScript)**. 6ª posición en el índice **w3techs** con un 2,2% de implantación.
- **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 nº8 en uso en el mundo con un 1,4% de cuota de implantación.
- **Coldfusion**. Coldfusion es una plataforma de desarrollo rápido de aplicaciones web que usa el lenguaje de programación **CFML**. En este aspecto, es un producto similar a ASP, JSP o PHP. ColdFusion es una herramienta que corre en forma concurrente con la mayoría de los servidores web de Windows, Mac OS X, Linux y Solaris. Fuente [Wikipeda](#).
- **Perl**. Multiparadigma incluyendo la orientación a objetos, pertenece al grupo de los lenguajes interpretados. 10ª puesto del ranking con un 0.1% de cuota de mercado

Ejercicio Tema 1 - DWES

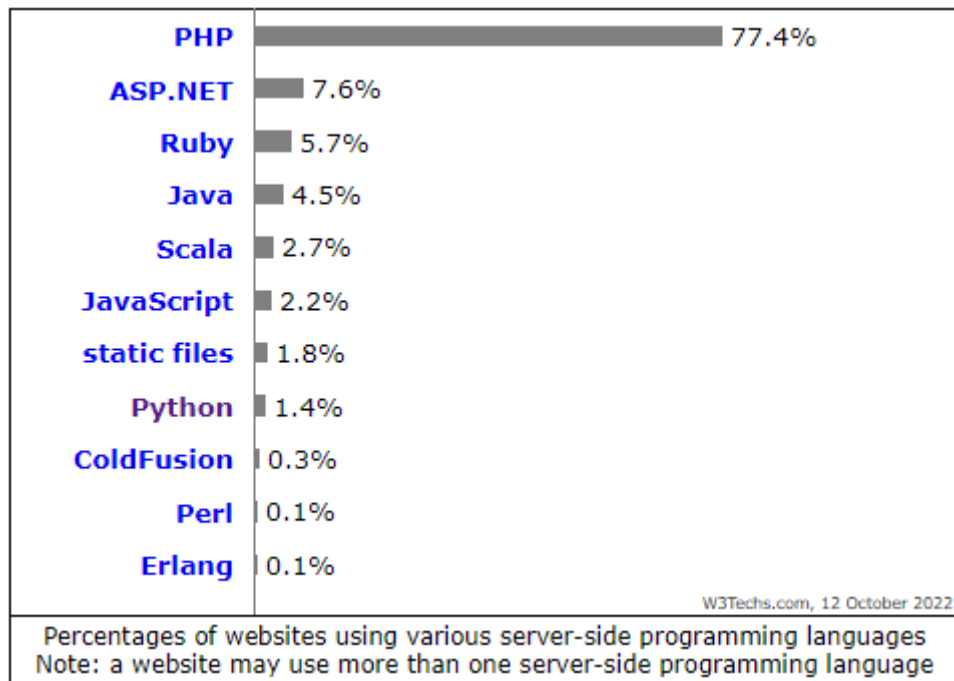
- **Erlang.** Es un lenguaje de programación orientado a la concurrencia y un sistema de ejecución que incluye una máquina virtual (BEAM) y bibliotecas (OTP). El subconjunto de programación secuencial de Erlang es un lenguaje funcional, con evaluación estricta, asignación única, y tipado dinámico. Fue diseñado en la compañía Ericsson para realizar aplicaciones distribuidas, tolerantes de fallos, soft-real-time y de funcionamiento ininterrumpido. Proporciona el cambio en caliente de código de forma que este se puede cambiar sin parar el sistema. Originalmente, Erlang era un lenguaje propietario de Ericsson, pero fue cedido como software de código abierto en 1998. La implementación de Ericsson es principalmente interpretada, pero también incluye un compilador HiPE (sólo soportado en algunas plataformas) .Fuente [Wikipedia](#)

A continuación, otros lenguajes presentes en el lado del servidor, pero con porcentaje de uso inferior al 0,1%.

- **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.
- **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.
- **VISUALBASIC.NET.** Es tanto interpretado como compilado y entre sus características figura el estar orientado a objetos. Es heredero de **VisualBasic** y como tal, desarrollado por **Microsoft**.

Cita aparte necesita PL/SQL. Es el único programa de consulta de datos que se cuela en las diferentes listas de uso de lenguajes de programación.

- **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. Puede abarcar sentencias completas de SQL en su código.



Fuente del diagrama: w3techs.com.

Realizado por Ricardo.

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



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**, **Python** y **Perl**.

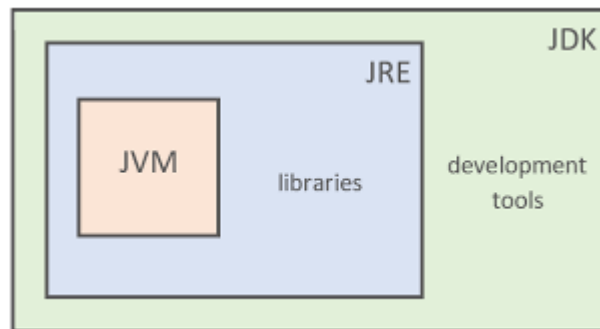
Te permite instalar de forma sencilla **Apache** en tu propio ordenador, sin importar tu sistema operativo (**Linux**, **Windows**, **MAC** o **Solaris**) también permite probar tu trabajo (páginas web o programación, por ejemplo) en tu propio ordenador sin necesidad de tener que acceder a internet. En concreto, se denomina **WAMP** cuando el **SO** es **Windows**, cuando está orientada a **MAC** se denomina **MAMP** y cuando lo es a **Linux**, se la llama **LAMP**.

Realizado por Ricardo.

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.

La JVM es necesaria en el entorno de explotación.

El JDK está presente únicamente en el entorno de desarrollo.



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 usar la aplicación. **Entorno de explotación**

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. **Entorno de Desarrollo**

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

Realizado por Ricardo.

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

Definición de un IDE : Es la herramienta de desarrollo de software utilizado para escribir, generar, probar y depurar un programa. También proporcionan a los desarrolladores una interfaz de usuario común (UI) para desarrollar y depurar en diferentes modos.

Los IDE's documentados son Eclipse, NetBeans, Visual Studio Code y IntelliJ idea.

En la lista que sigue, el nombre de cada versión es un enlace a la descarga de su última versión estable a fecha de octubre del 2022.

Eclipse:

Es un entorno de desarrollo de código abierto y multiplataforma, además es una comunidad de usuarios que hacen actualizan y proveen de nuevas utilidades constantemente al entorno.

[Última versión: 2022-06\(4.24\)](#)

- Editor de texto con analizador sintáctico.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit.
- Control de versiones con CVS.
- Integración con Ant.
- Refactorización.
- Asistentes (wizards) para creación de proyectos, clases, test, etc.

Netbeans: Código libre y orientado principalmente para java. Sun Microsystems lo fundó en junio del 2000.

[Última versión: 15.0.](#)

- 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

Ejercicio Tema 1 - DWES

- NetBeans IDE es libre, código abierto, multiplataforma con soporte integrado para el lenguaje de programación Java.

Visual Studio Code:

Desarrollado por Microsoft, gratuito y de código abierto.

[La última versión es la September 2022 \(version 1.72\).](#)

- Resaltado de Sintaxis
- Snippets
- Autocompletado de código
- Refactorización
- Depuración

IntelliJ IDEA:

Desarrollado por JetBrains con dos versiones, comercial y para la comunidad.

[Última versión 2022.2.3 Build : 222.4345.14](#)

- Búsqueda de coincidencia de patrones.
- Automatización del flujo de trabajo.
- Control de versiones.
- Análisis de código estático.
- Pruebas unitarias.
- Descompilador integrado.
- Múltiples atajos de teclado.

Realizado por David Aparicio Sir.

Revisado por Ricardo.

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

Apache HTTP: 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.

Apache Tomcat: Contenedor de aplicaciones contiene gratuito programado en Java. El servidor Apache tomcat se puede usar para manejar páginas estáticas y dinámicas.

Ejercicio Tema 1 - DWES

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.

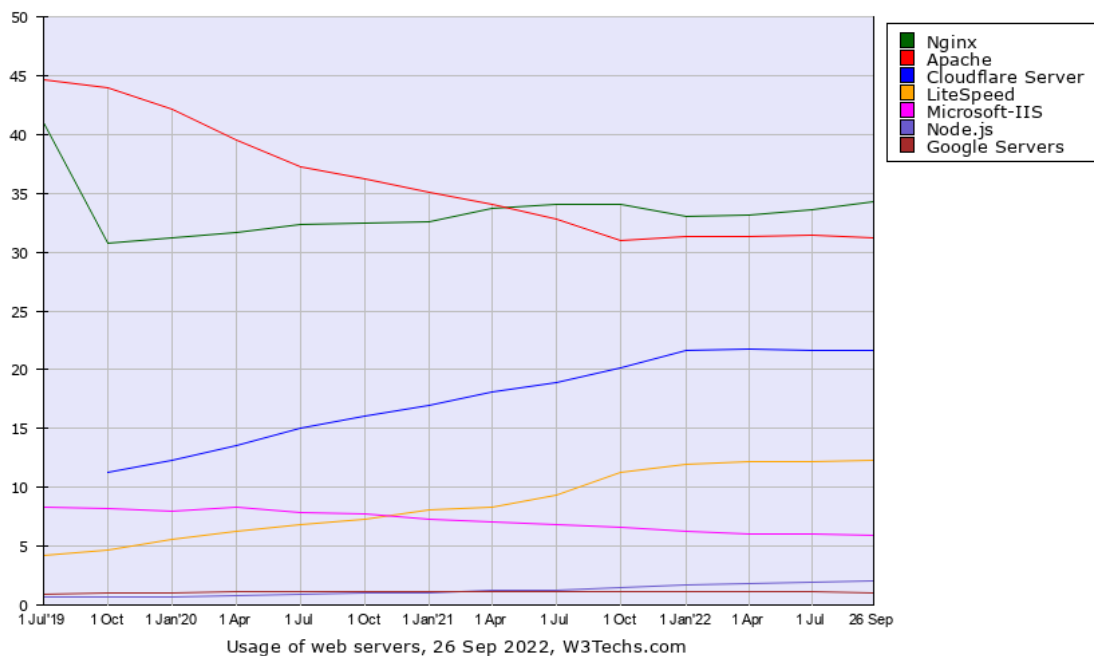
Compatible con IPv6

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...



Realizado por David Aparicio Sir.

15. Apache HTTP vs Apache Tomcat.

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.

Realizado por David Aparicio Sir.

16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).

Google Chrome programado en C++, Python y JavaScript

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

Ejercicio Tema 1 - DWES

- 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.

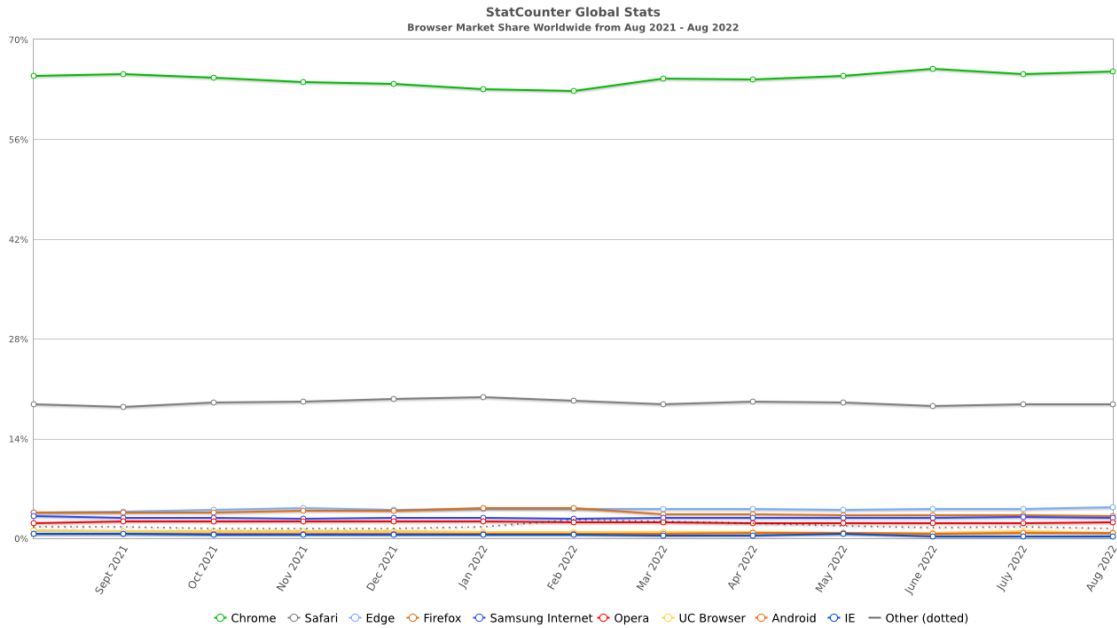
Cuota del 15.4%

Edge

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

Ejercicio Tema 1 - DWES



Realizado por David Aparicio Sir.

17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ...

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

[Enlace explicativo](#)

Realizado opor Josué Martínez Fernández.

18. Repositorios de software - sistemas de control de versiones: GIT, CVS, Subversión, ...

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

[Enlace explicativo](#)

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.

[Enlace explicativo](#)

[Enlace de interés](#)

Realizado opor Josué Martínez Fernández.

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.

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 (Apache2)

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

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

IDE: Microsoft Visual Estudio Code (Actualizado)

Realizado opor Josué Martínez Fernández.

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.

Sistema operativo: WINDOWS X pro.

Servidor administración remota: SSH

Servidor de transferencia de ficheros: Filezilla

Repositorio: GIT

Servidor Web: Apache HTTP

SGBD: MySQL

Cliente ssh: NetBeans / Putty

Navegador: Mozilla, Chrome

Realizado por Alejandro Otálvaro Marulanda.

21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web:

CMS - Sistema de gestión de contenidos: Es un software que ayuda a los usuarios a crear, administrar y modificar el contenido del sitio web sin la necesidad de conocimientos técnicos profesionales. Suelen tener interfaces sencillas fáciles de usar.

Un ejemplo de ello es el JOOMLA

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.

Realizado por Alejandro Otálvaro Marulanda.

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

- **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.

Ejercicio Tema 1 - DWES

- **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.
- **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.

Realizado por Alejandro Otálvaro Marulanda.