

Tema 1 - Introducción al funcionamiento de las aplicaciones web

Índice

Aplicaciones Web. Conceptos Básicos	3
Ventajas y desventajas de las aplicaciones web.	3
Elementos necesarios para ejecutar una aplicación web	4
Servidor Web	4
El navegador	5
El front-end y el back-end	5
Funcionamiento de una aplicación web	6
Funcionamiento del lado del servidor	7
Funcionamiento del lado del cliente	7
Tecnologías para la creación de aplicaciones web	8
La pila de aplicaciones LAMP (Linux+Apache+MySQL+PHP)	8
Desarrollo Full-Stack	8
El paradigma Modelo-Vista-Controlador (MVC)	9
Lenguajes de programación en el lado del cliente	10
Lenguajes de programación en el lado del servidor	11
Aplicaciones web para la creación de aplicaciones móviles	12
Despliegue de aplicaciones web	12
Sistemas Gestores de Contenidos (CMS)	13
Conclusiones	15

1.Aplicaciones Web. Conceptos Básicos

De forma tradicional, las aplicaciones que ejecutan las computadoras son creadas utilizando un lenguaje de programación y generando, bien en un proceso de compilación (comprobación de la corrección del código y después la ejecución del mismo), o bien un proceso de interpretación (se comprueba el código mientras se ejecuta el mismo) un fichero que puede ser ejecutado en una computadora.

Con la introducción de la tecnología web, los navegadores se convierten en programas capaces de interpretar código fuente de diversos lenguajes de programación asociados a la web como por ejemplo Javascript que permiten implementar funcionalidades similares a las obtenidas con lenguajes de programación tradicionales.

Toda esta evolución que ha permitido llegar a la ejecución de complejas aplicaciones web en los navegadores, evoluciona desde la aparición de las primeras páginas web de simples hiperenlaces, pasando por la aparición de los primeros navegadores, la popularización de Internet a mediados de la década de los 2000 y su masiva entrada en todo tipo de dispositivos en la década de 2010 hasta 2020. En el siguiente enlace se puede comprobar la evolución y los hitos más importantes acontecidos en el desarrollo de la World Wide Web desde su nacimiento hasta nuestros días.

2.Ventajas y desventajas de las aplicaciones web.

La principal ventaja de las aplicaciones web es la gran implantación de los navegadores en todo tipo de dispositivos con acceso a la red. Esto permite que este tipo de aplicaciones bien programadas sean capaces de adaptarse y utilizarse en casi cualquier dispositivo conectado independientemente de su sistema operativo. Además, también podemos destacar otras ventajas como pueden ser:

- La ejecución de una aplicación web solo necesita que el dispositivo pueda ejecutar un navegador actual.
- Se pueden centralizar los datos de todos los usuarios que utilizan esta aplicación.
- La capacidad de proceso de una aplicación web se puede delegar en un servidor.
- Al no tener que adaptarse a cada cliente, se puede realizar un soporte centralizado de la aplicación.
- Con los sistemas en la nube la potencia de los servidores necesaria para ejecutar una aplicación web puede escalarse en determinados momentos.

- Los usuarios suelen estar familiarizados con la forma de trabajar que proporcionan los navegadores web.

Pese a la rápida evolución de este tipo de aplicaciones, también muestran una serie de características desfavorables que hacen que no puedan ser utilizadas en cualquier contexto.

- Se necesita un navegador para ejecutarlas y esa capa de software extra, hace su ejecución más ineficiente, algo que puede no ser viable para aplicaciones como juegos o gráficos en 3D.
- Generalmente, se necesita que la computadora que esté ejecutando la aplicación web tenga conexión a Internet aunque en ocasiones pueda no ser necesario si la aplicación web se ejecuta completamente en el cliente.
- Aunque han aparecido nuevas tecnologías que tratan de mitigarlo, crear una aplicación requerirá conocer diversos lenguajes de programación y marcado (HTML+CSS+Javascript+PHP).
- Al implementar un modelo de datos centralizado en un servidor, estos datos pueden ser susceptibles de alguna violación que permitan ser consultados fuera de la organización que la utiliza.

3.Elementos necesarios para ejecutar una aplicación web

3.1. Servidor Web

Son los ordenadores encargados de componer la página web y ejecutar la mayor parte del código de la aplicación web. Estos servidores están a la espera de peticiones de clientes (los navegadores) a los cuales responderán con un fichero generalmente en HTML y posiblemente un fichero CSS que contenga los estilos más ficheros javascript con la funcionalidad del cliente.

Utilizan protocolos HTTP (Hypertext Transfer Protocol) o su versión segura HTTPS para recibir y enviar estas peticiones. En estos servidores suelen ejecutar versiones de sistemas operativos que contienen solamente las aplicaciones necesarias para su funcionamiento más eficiente.

Estos servidores suelen contener un software llamado servidor web que se encarga de recibir y gestionar peticiones (Apache o NGINX) y servidores de aplicaciones que son los encargados de formar los ficheros HTML+CSS+Javascript que se envían como respuesta a una petición.

Estos servidores suelen implementar una arquitectura en 3 niveles. Estos son:

- **La capa de negocio.** Suele contener los datos que almacena la organización para la que se ha desarrollado la aplicación web. Esto puede suponer una simple base de datos MySQL dentro del mismo servidor web si hablamos de una aplicación sencilla o un conjunto de servidores de bases de datos si hablamos de una implementación compleja como pueda ser un sistema sanitario o educativo.
- **La capa lógica.** Contiene toda la funcionalidad que implementa la aplicación web, que serían aquellos ficheros donde están programadas las funciones utilizadas. En una aplicación web con PHP, serían un conjunto de ficheros que pueden seguir un paradigma de programación (sistema para organizar el código fuente de un software) como el modelo MVC (Modelo-Vista-Controlador).
- **La capa de presentación.** Contiene el resultado de aplicar los ficheros con el código fuente de la capa lógica e incrustar en ellos los datos de las bases de datos. Este resultado serán ficheros HTML, CSS y Javascript que el servidor web transferirá al cliente del que ha recibido la petición.

3.2. El navegador

Es la herramienta necesaria para generar una petición web e interpretar los resultados de la misma. Estos sistemas han evolucionado desde sencillos presentadores de texto e intérpretes de enlaces a potentes sistemas software que permiten ejecutar múltiples funcionalidades como gráficos en 3D, sonido o vídeo consumiendo importantes recursos hardware.

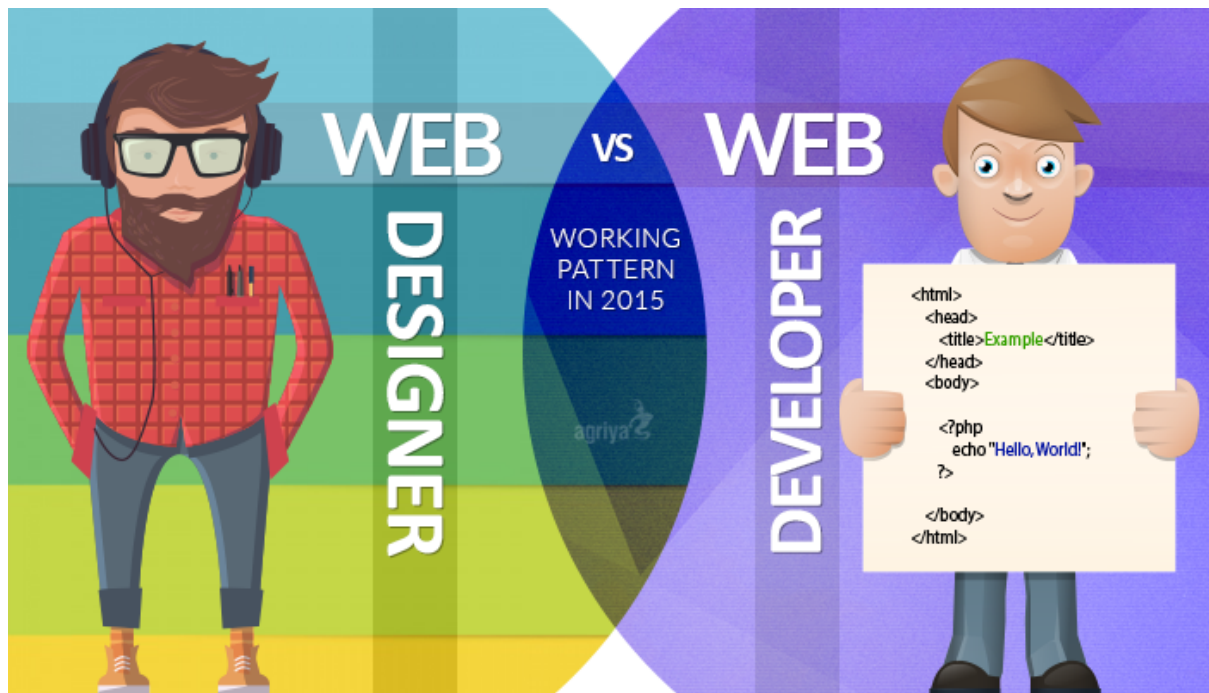
3.3. El front-end y el back-end

Se llama front-end a todo lo relacionado con la apariencia de la página web que será devuelta por el servidor al navegador. Los encargados del desarrollo de estos apartados necesitan conocimientos de usabilidad, diseño gráfico, lenguajes HTML o marcado CSS. En otras palabras, es la parte visible para el usuario de la aplicación web y por tanto parte importante del uso de la misma.

Por otra parte, el back-end desarrolla la funcionalidad que implementa la aplicación web. Es la que contempla el funcionamiento de la capa lógica y la capa de negocio comentadas anteriormente y utiliza tecnologías de bases de datos y programación web como PHP, ASP, etc.

Esta forma de dividir la funcionalidad de una aplicación web, permite que estos apartados puedan desarrollarse por separado por profesionales de diferentes

talentos (aunque no ocurre en proyectos pequeños) e incluso evoluciones del proyecto sin tener que modificar el otro apartado. Podemos mejorar la interfaz sin tener que modificar la presentación de la web y viceversa.



4. Funcionamiento de una aplicación web

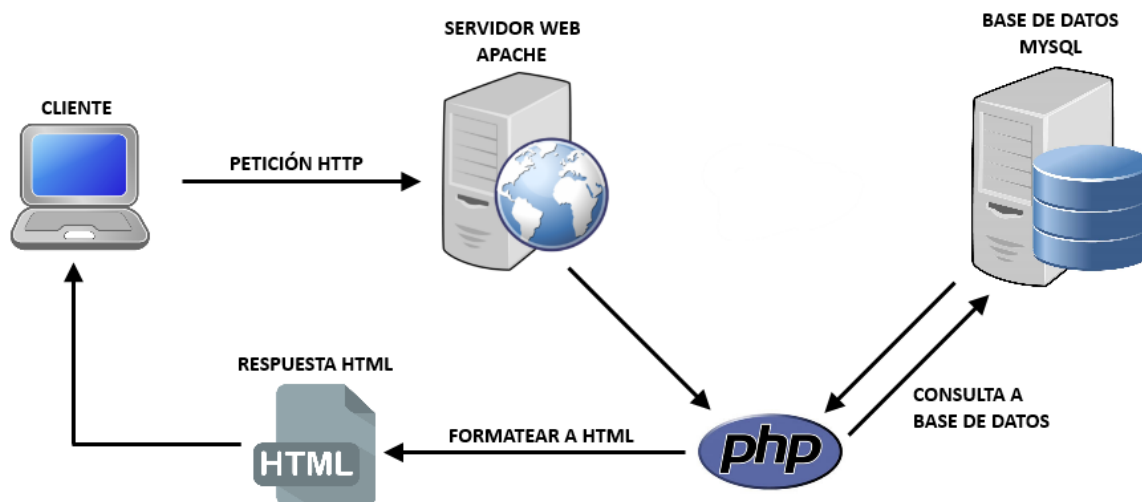
Las aplicaciones web suelen desarrollarse en dos apartados diferenciados que no pueden ser programados por separado o de forma conjunta. Existen la funcionalidad del lado del cliente y la funcionalidad del lado del servidor.

- **El lado del servidor.** Son aplicaciones web que utilizando una serie de lenguajes de programación y ejecutadas por el servidor implementan la funcionalidad deseada. En el lado del servidor se suelen utilizar entre otros PHP o NodeJS como lenguajes para desarrollar la funcionalidad.
- **El lado del cliente.** La funcionalidad en el lado del cliente se implementa junto con el código HTML que devuelve el servidor web. Esta funcionalidad será ejecutada por el navegador y por tanto utilizará la CPU del cliente. El lenguaje más utilizado para implementar esta funcionalidad es Javascript y sus frameworks derivados como Angular o React.

4.1. Funcionamiento del lado del servidor

Cuando tenemos implementada la funcionalidad de la web en el lado del servidor, todo empieza con una petición realizada por el usuario a través del navegador. El programa que actúa como servidor web, generalmente Apache o NGINX, tiene una

carpeta donde almacena los ficheros que componen la web y si detecta que estos contienen algún tipo de lógica programada, ejecuta estos ficheros mediante una llamada al servidor de aplicaciones. El resultado obtenido de esta ejecución que puede contener acceso a bases de datos generará un fichero HTML que se devolverá al usuario mediante el protocolo HTTP.



4.2. Funcionamiento del lado del cliente

Cuando la funcionalidad se implementa en el lado del cliente, es el navegador el que realiza toda la ejecución de la lógica de la aplicación web. En la actualidad la mayor parte de esta funcionalidad se implementa con Javascript y los navegadores se han convertido en programas pesados que actúan como una especie de máquina virtual. Ejemplos de la ejecución de la lógica en la parte del cliente puede ser la tecnología Flash necesaria para ejecutar muchos juegos que poblaban la web antes de la normalización del estándar HTML5, otro ejemplo podría ser la tecnología Silverlight de Microsoft y que se ha utilizado por algunas plataformas de streaming para reproducir sus contenidos en el navegador, ambas necesitan extender la funcionalidad de los navegadores con plugins externos.

En el funcionamiento de la lógica implementada en el cliente, el usuario realiza una petición mediante el navegador, que contacta con el servidor web que devuelve un solo documento o una serie de documentos que contienen tanto el código HTML como la funcionalidad que debe ser ejecutada en el navegador en una serie de ficheros que por norma general utilizarán el lenguaje de programación Javascript. Tras recibir todos los ficheros que contienen la web, será el navegador el encargado de ejecutar el código en Javascript y obtener así la funcionalidad que se ha querido implementar.



5. Tecnologías para la creación de aplicaciones web

La web es una plataforma altamente cambiante e independiente de una sola organización. Muchos son los competidores que se afanan por crear herramientas que faciliten la creación de aplicaciones web tanto del lado del cliente como del lado del servidor. Este panorama hace que existan múltiples opciones si se quiere desarrollar una aplicación web desde el formato más tradicional con PHP+MySQL o pasando por un modelo Full Stack con NodeJS como máximo exponente en estos momentos.

5.1. La pila de aplicaciones LAMP (Linux+Apache+MySQL+PHP)

Cuando se utiliza esta pila de aplicaciones el principal lenguaje utilizado para la creación de una aplicación web es PHP. Estas aplicaciones suelen realizar conexiones con bases de datos MySQL o su fork open source MariaDB y delegan en Apache como servidor web en el lado del servidor. La utilización de diversas tecnologías que se interconectan entre sí, hace que un desarrollador tenga que tener conocimientos del lenguaje de programación PHP, sentencias SQL con MySQL, creación de documentos HTML, diseño con CSS y programación del lado del cliente con Javascript.

5.2. Desarrollo Full-Stack

Se trata de una nueva visión para la creación de aplicaciones web donde tanto la programación que se realiza en el lado del cliente como la programación que se realiza en el lado del servidor utilizan la misma tecnología, por tanto se simplifica la tarea de los desarrolladores. El máximo exponente de esta nueva forma de crear aplicaciones web es NodeJS donde Javascript es el lenguaje que se utiliza tanto en el lado del cliente como en el lado del servidor, además también se fomenta el uso de bases de datos No-SQL que pueden proporcionar mejor rendimiento y por tanto mejores experiencias de los usuarios en la web.

5.3. El paradigma Modelo-Vista-Controlador (MVC)

Una de las técnicas más utilizadas en la programación para desarrollar aplicaciones de un considerable tamaño es dividir la funcionalidad de la misma en diferentes apartados que puedan tratarse por separado. El paradigma MVC trata de dividir la capa lógica de la aplicación web en tres subdivisiones, el Modelo, la Vista y el Controlador.

- **El modelo.** Contiene las funciones que trabajan con los datos de la aplicación. Aquí se situarán las funciones que permitan acceder a los datos que utilice la aplicación generalmente almacenados en la base de datos. Esta capa recibirá instrucciones del Controlador y le devolverá los resultados no interactuando de manera directa con la Vista.
- **La vista.** Contiene las funciones que definen el aspecto visual de los elementos de la web. Esta capa recibe instrucciones con datos del controlador y los representará de forma visual para que a través del servidor web se envíen al navegador desde donde se realizó la petición. Aunque puede comunicarse con el Controlador para comunicar resultados pero el envío de datos son principalmente la respuesta al navegador desde donde se realizó la petición.
- **El controlador.** Es el elemento básico que permite la comunicación entre Modelo y Vista. En una aplicación web el Controlador recibirá la petición del navegador y dependiendo del tipo de petición, será el controlador, según su lógica programada el que realizará una petición de datos al modelo o modificará la vista que se envía como respuesta al navegador.

Esta diferenciación de funciones que se realiza cuando se desarrolla una aplicación web permite que aunque incrementando el tiempo de desarrollo de la misma, se mejore el mantenimiento y posterior actualización de la misma. Si exponemos un ejemplo en una web, un desarrollador podría implementar un cambio en la funcionalidad de la web sin modificar de forma alguna la vista que se presenta al usuario. Imaginemos que Facebook quiere implementar un nuevo algoritmo de recomendaciones para actualizar las noticias que ven sus usuarios, para no detener completamente la web se empieza a trabajar en la nueva funcionalidad en el controlador, cuando esta esté suficientemente probada se modificará el controlador para que los datos que envíe a la vista sean aquellos proporcionados por el nuevo algoritmo de recomendaciones en lugar del antiguo. Si observamos un poco este cambio podemos comprobar que nada ha cambiado en la vista, los post siguen teniendo un número de likes, y un texto o enlace, el modelo seguirá consultando la base de datos y obteniendo la información de los posts seleccionados y solamente habrá cambiado el proceso de decisión de que posts encajan mejor con el perfil del usuario, tarea que debe realizar el controlador. Todo este proceso con el

paradigma MVC se realiza de forma transparente al usuario que apenas ha notado cambio alguno en el funcionamiento normal de la web.

5.4. Lenguajes de programación en el lado del cliente

En el lado del cliente la posición que ha obtenido Javascript como lenguaje de referencia es ampliamente dominante. Existen otros lenguajes que utilizan los complementos Flash o la tecnología Silverlight pero cuyo uso es residual. En el lado del cliente no existen múltiples lenguajes entre los que elegir, sin embargo existen múltiples frameworks derivados de Javascript que permiten aumentar la rapidez de desarrollo. Un framework es un conjunto de funcionalidades programadas en Javascript que se recopilan y empaquetan para que los desarrolladores puedan acceder a ellas de forma sencilla aumentando reutilizando parte del código ya escrito y mejorando la rapidez con las que se crean las web apps.

En la actualidad la evolución, uso y desarrollo de nuevos frameworks es trepidante, las nuevas versiones de los frameworks más populares evolucionan rápidamente y nuevos competidores aparecen constantemente. Esto, hace que pensar en utilizar unos u otros tenga que ser una decisión muy meditada pues la elección de un framework con una rápida evolución puede dificultar el mantenimiento y actualización de una aplicación web.



5.5. Lenguajes de programación en el lado del servidor

En el lado del servidor existen implementaciones de aplicaciones web que pueden ir desde sencillas implementaciones hasta integraciones completas con el sistema informático de una gran empresa o entidad. Por ello, existen diversos lenguajes que

son utilizados para diferentes tareas, desde las plataformas de desarrollo empresarial que ofrecen la posibilidad de desarrollar todo el software que sea necesario dentro de una empresa, lenguajes con los que poder crear aplicaciones que funcionen en el lado del servidor o lenguajes de script que permitan fusionar lenguaje HTML con elementos de programación que creen webs dinámicas.

- **Plataformas de desarrollo empresarial.** Son completos sistemas de desarrollo software que permiten crear aplicaciones de escritorio, servicios, interfaces web y un largo etcétera todo ello integrado dentro de un mismo paquete. Estas plataformas suelen ser utilizadas por las grandes organizaciones para integrar todo su desarrollo y ofrecer un entorno estricto y bien definido para desarrollar software de grandes dimensiones. En la actualidad las principales plataformas de desarrollo empresarial son J2EE de Oracle que se basa en el lenguaje Java y sus derivados además de la plataforma .Net de Microsoft que ofrece múltiples lenguajes de ellos el más destacado puede ser C#.
- **Lenguajes para crear aplicaciones en el lado del servidor.** Son lenguajes de programación que permiten además de crear aplicaciones de escritorio utilizando frameworks o librerías especiales el desarrollo de aplicaciones web. Aquí aparecen Java y C# que permiten crear estas aplicaciones pues están integradas en las plataformas de desarrollo empresarial comentadas anteriormente y Python, además de Javascript como las principales nuevas tendencias en el sector que intentan cambiar la forma de trabajo que se ha venido utilizando tradicionalmente.
- **Lenguajes de script.** Son lenguajes que se caracterizan por integrarse con el código HTML de una web. Utilizan etiquetas para indicar al procesador del documento cuando se inicia el código en el lenguaje cuya ejecución dará como resultado un código en HTML que será devuelto al navegador. Los principales ejemplos de estos lenguajes de script son PHP, JSP y ASP que generarán páginas con estas mismas extensiones que indicarán al servidor web que antes deben ser interpretadas por el servidor de aplicaciones antes de ser devueltas al navegador. En los dos últimos casos JSP es el lenguaje de script de Java mientras que ASP es el lenguaje de script de .Net

5.6. Aplicaciones web para la creación de aplicaciones móviles

En los últimos tiempos una tendencia aparece en el mundo de las aplicaciones web con proyectos que intentan utilizar toda la potencia de los lenguajes que se utilizan en la web para el desarrollo de aplicaciones móviles universales. Proyectos como Ionic, basado en Apache Cordova o Flutter de Google son propuestas de plataformas que permiten estos desarrollos que se integran en las plataformas IOS y Android sin tener que realizar dos desarrollos independientes para cada una. Estas

plataformas ofrecen una especie de capa intermedia que permita a las aplicaciones web el acceso a componentes de la máquina como la cámara de fotos, el acelerómetro, el GPS, etc.

6.Despliegue de aplicaciones web

Cuando la aplicación web ha sido desarrollada, esta debe ponerse disponible a los usuarios realizando un despliegue que utilice un servidor propio de forma local o autohospedada o de forma más profesional un servicio de hosting o un sistema de computación en la nube.

Dependiendo de las características de la aplicación web, el número potencial de usuarios que vayan a utilizarla o el estado de desarrollo de la misma nos decantamos por una u otra solución.

- **Servidor local.** Esto conlleva instalar en los equipos donde se está desarrollando la aplicación de un servidor web que permita utilizando la dirección de loopback de la tarjeta de red realizar pruebas como host local (127.0.0.1 o localhost). Es la forma básica de comprobar si el desarrollo funciona por parte de los programadores.
- **Servidor autohospedado.** En este escenario que puede ser válido para una pequeña organización se dedica un equipo con un servidor web instalado que junto con una conexión a Internet será capaz de servir páginas web de forma autónoma sin recurrir a servicios externos. El ahorro en costes conlleva sin embargo, la instalación y mantenimiento del servidor. La utilización de esta práctica puede ser útil en webs que no vayan a ser utilizadas por múltiples usuarios a la vez y que no necesiten estar completamente operativas el 100% del tiempo.
- **Servicio de hosting.** Son empresas que ponen a disposición de sus clientes una serie de planes para alojar una web en uno de sus servidores. Los planes van desde aquellos más modestos donde nuestra página web comparte espacio dentro de un servidor con otras y tiene restricciones de tráfico, espacio o número de bases de datos disponibles. En el otro extremo se encuentran planes donde podemos contratar servidores dedicados que permiten ser gestionados de forma remota por el cliente, configurando todas las características que se deseen. Estos últimos planes aunque mucho más caros también suelen presentar restricciones de tráfico y por supuesto las limitaciones que el propio hardware del servidor tenga.
- **Sistemas de computación en la nube.** Son grandes colecciones de servidores que se ofrecen a los clientes y donde estos utilizando máquinas

virtuales permiten al usuario crear toda la infraestructura necesaria para el despliegue de la aplicación web. Por estar basados en sistemas de virtualización estas plataformas ofrecen la posibilidad de aumentar la capacidad a demanda del usuario o la propia demanda de las necesidades de la aplicación web. En este caso, las principales plataformas son AWS de Amazon, Google Cloud de Google y Azure de Microsoft, aunque existen multitud de plataformas de menor tamaño como Heroku, Dokku o Firebase.

7. Sistemas Gestores de Contenidos (CMS)

Como puede observarse una aplicación web incrementa la potencia de las páginas web pudiendo adaptarse estas al usuario que la esté utilizando. Por este motivo, aparecieron en el mercado una serie de aplicaciones web llamadas genéricamente Sistemas Gestores de Contenidos (Content Management System) o CMS. Estas aplicaciones web aparecen generalmente en forma de colecciones de documentos HTML, CSS, Javascript y PHP u otros lenguajes de servidor agrupados en un fichero Zip que puede ser descomprimido en el servidor web.

Estos desarrollos pueden adoptar cualquier licencia software e ir desde aplicaciones web privativas desarrolladas por pequeñas empresas hasta grandes desarrollos open source que están implementados en multitud de páginas web de Internet. También existen variaciones de estos sistemas como LMS (*Learning Management System*) o DMS (*Document Management System*) que se centran en funcionalidades diferentes como los portales educativos o la gestión documental. Otro de las variantes más populares en la red son aquellos CMS que permiten crear una tienda on-line y que en algunos casos son aplicaciones web completamente centradas en este sector o CMS para crear portales de gestión genéricos con extensiones específicas de comercio electrónico.

En la actualidad algunos de los CMS más populares son los siguientes:

- **Wordpress.** En la actualidad es el CMS más popular y utilizado. Algunos estudios lo sitúan con una cuota de mercado de hasta un 30% de las páginas web de la red. Desarrollado en PHP, esta aplicación open source se actualiza constantemente y dispone de múltiples módulos que pueden extender la funcionalidad de la web.
- **Drupal.** Es otro de los grandes CMS del mercado, en él se puede configurar portales web con múltiples opciones de configuración. También está desarrollado en PHP con licencia libre y permite extender la funcionalidad de la web mediante componentes externos.
- **Ghost.** Es un CMS desarrollado para la plataforma NodeJS donde Javascript

es el lenguaje en el que se desarrolla la aplicación en el lado del servidor. Estos sistemas suelen hacer uso de nuevas bases de datos NoSQL que proporcionan mejoras en el rendimiento a las tradicionales bases de datos relacionales.

- **Prestashop.** Es una aplicación web que permite el despliegue de una tienda de comercio electrónico. Basada en PHP y MySQL, la aplicación web es de código abierto y permite la implantación de una tienda web de forma sencilla con un limitado número de posibilidades.
- **Magento.** También desarrollada en PHP y sobre bases de datos MySQL y MariaDB, esta aplicación para crear una tienda on-line tiene multitud de opciones que hacen que el proceso de configuración sea bastante complicado. Bien configurada esta opción ofrece mucha potencia para adaptarse a las características de cualquier tipo de negocio. Aunque la aplicación web es de carácter libre, la empresa desarrolladora fue adquirida por Adobe en 2018.
- **Moodle.** Este sistema está diseñado para crear una plataforma de aprendizaje on-line. Permite la creación de cursos, la comunicación entre profesorado y alumnado, la entrega de trabajos, gestión de calificaciones o desarrollo de exámenes. Es la plataforma más utilizada para la implantación de un sistema de e-learning autogestionado.
- **OwnCloud.** Es una plataforma que permite la creación de un sistema de gestión documental propio o de nube privada. En concreto la solución que propone Owncloud está implementada en PHP y puede utilizar MySQL o PostgreSQL si se desea un extra en fiabilidad. Es una solución acompañada de aplicaciones para dispositivos móviles que permite la creación de una nube sin tener que almacenar datos privados de una organización en servidores ajenos.

Estos sistemas son una pequeña muestra de los CMS que se pueden encontrar en la web. Dependiendo de la funcionalidad que se desee implementar se elegirá uno u otro teniendo en cuenta consideraciones como la cantidad de módulos, las opciones para futuros desarrollos, la plataforma base donde se debe implementar, capacidad para usuarios, etc.

8.Conclusiones

Las aplicaciones web son uno de los principales campos de desarrollo en la informática. En la actualidad, la web ofrece a través de la potencia que han

alcanzado los navegadores una plataforma unificada para la utilización de las aplicaciones pertenecientes a una entidad.

Para la utilización de una aplicación web son necesarias dos partes, por una parte un navegador en la parte del cliente y por otra un servidor que contenga la aplicación web y maneje la mayor parte de la parte del procesamiento de la misma. Esta última parte del servidor suele ser implementada con las tecnologías LAMP o los nuevos paquetes full-stack como NodeJS basado en Javascript.

Normalmente, las aplicaciones web suelen definir una parte *front-end*, basada en aquello que ve el usuario cuando entra a la web y una parte *back-end* que conlleva la configuración del funcionamiento de la misma. Estos dos términos pueden ser utilizados en el desarrollo de las aplicaciones web para referirse el primer término a la parte visual de la página web, mientras que el segundo término se referiría a la parte lógica o funcionalidad que tiene que desarrollar la *webapp*

Para que la aplicación web, pueda ser accesible se pueden definir varias formas de desplegarla realizando desde un despliegue local y autogestionado hasta la utilización de potentes sistemas de cloud computing que ofrecen escalabilidad y alta disponibilidad.

En la actualidad, se ofrecen en Internet aplicaciones web preprogramadas conocidas como CMS que pueden ser desplegadas de forma rápida y configuradas sin tener que programarla. Estos CMS cubren muchas de las funcionalidades desde portales web hasta tiendas on-line pasando por foros, comunidades o sistemas de aprendizaje.