

TEMA 1 – DESARROLLO WEB EN ENTORNO SERVIDOR

Ejercicios

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	2
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....	3
3. Estudio sobre los métodos de petición HTTP/HTTPS más utilizados.....	4
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme)/URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.....	5
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.....	6
6. Modelo de división funcional front-end / back-end para aplicaciones web.....	7
7. Página web estática – página web dinámica – aplicación web – mashup.....	7
8. Componentes de una aplicación web.....	8
9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor – lenguajes de programación utilizados en cada caso.....	9
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).....	11
11. Características y posibilidades de desarrollo de una plataforma XAMPP.....	12
12. En que 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.....	13
13. IDE más utilizados (características y grado de implantación actual).....	14
14. Servidores HTTP / HTTPS más utilizados (características y grado de implantación actual).....	15
15. Apache HTTP vs Apache Tomcat.....	16
16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	17
17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen,	18

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

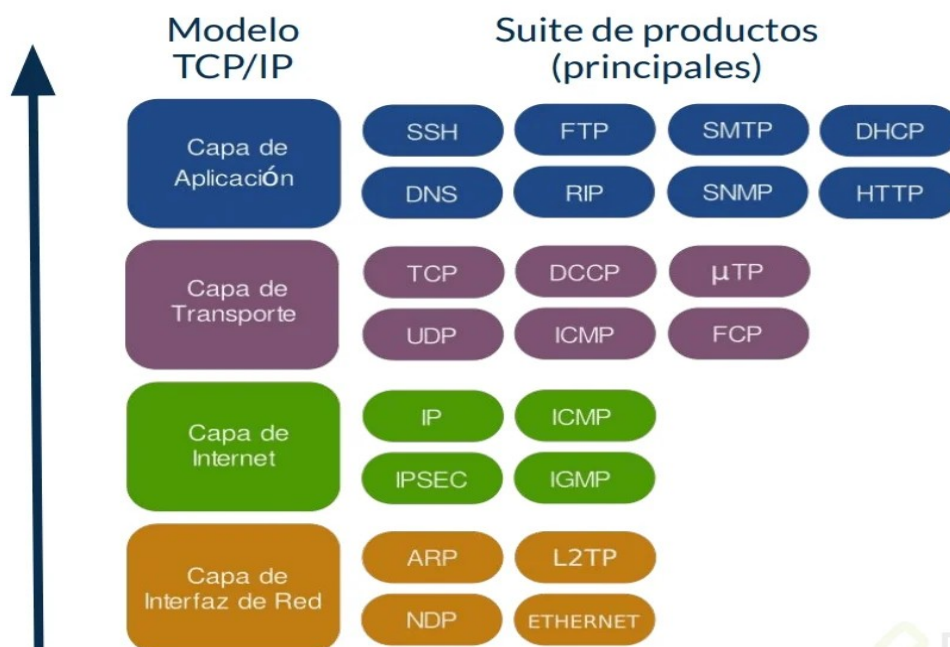
Consisten en un conjunto de normas que permiten que los ordenadores se comuniquen estableciendo la forma de identificación de estos en la red, la forma de transmisión de los datos y la forma en que la información debe procesarse.

IP: protocolo clasificado en la capa de red del modelo OSI cuya función es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un protocolo que no se orienta a la conexión que transfiere paquetes o datagramas.

TCP: protocolo clasificado en la capa de transporte del modelo OSI cuya finalidad es crear conexiones dentro de una red de datos compuesta por redes de computadoras para intercambiar datos. Distingue aplicaciones dentro de una misma máquina a través del concepto de puerto. TCP da soporte a muchas de las aplicaciones más populares de internet.

HTTP: protocolo de transferencia de hipertexto que permite transferencia de información a través de archivos (XML, HTML...). Desarrollado por la W3C y define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web para comunicarse. Usa el puerto 80.

HTTPS: protocolo de aplicación basado en HTTP destinado a la transferencia segura de datos de hipertexto. Utiliza un cifrado basado en SSL/TLS para crear un canal cifrado en el que la información sensible del usuario no pueda ser usada por un atacante. Usa el puerto 443.

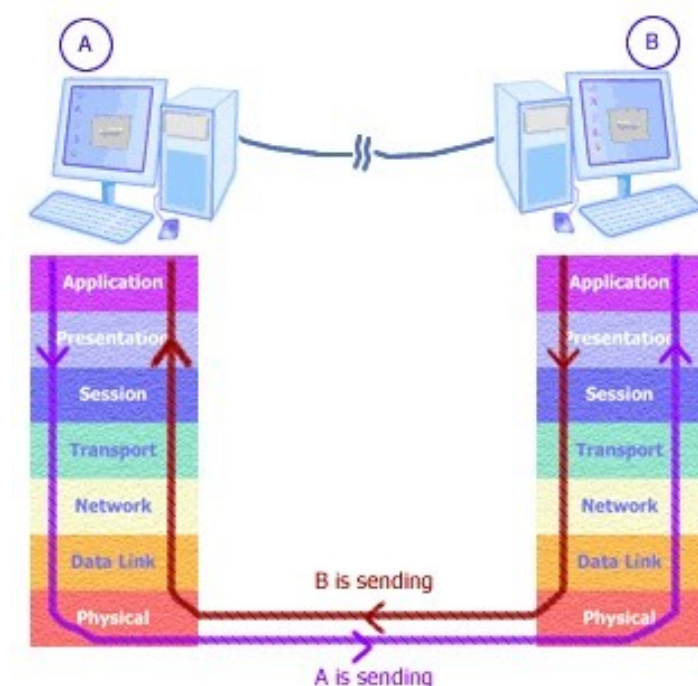


2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.

Modelo de diseño de software en el que las tareas se reparten entre los proveedores de servicios (servidores) y los demandantes (clientes). El cliente realiza peticiones al servidor que le da respuesta.

La relación con las aplicaciones web reside en que esta es una herramienta que los usuarios pueden utilizar accediendo a un servidor web (servidor) mediante un navegador (cliente).

[Página interesante sobre el tema](#)

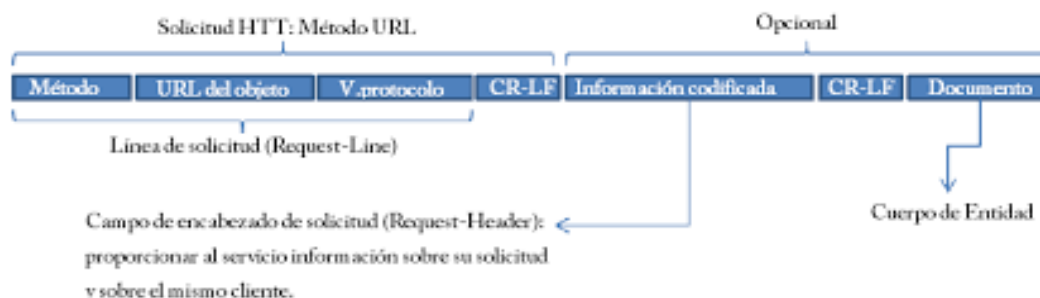


3. Estudio sobre los métodos de petición HTTP/HTTPS más utilizados.

Hay muchos pero los más utilizados son los métodos GET y POST.

Método GET: es un método de invocación en el que el cliente le solicita al servidor web que le devuelva la información identificada en la propia URL. Lo más común es que las peticiones se refieran a un documento HTML o a una imagen, aunque también se puede referir a un programa de base de datos. En tal caso, el servidor ejecuta ese programa y le devuelve al cliente el resultado generado tras esa petición.

Método POST: mientras que el método GET se utiliza para recuperar información, el método POST se usa habitualmente para enviar información a un servidor web. Estos casos suelen darse al enviar el contenido de un formulario de autenticación, así como entradas de datos o especificar parámetros para algún tipo de componente ejecutado en el servidor.

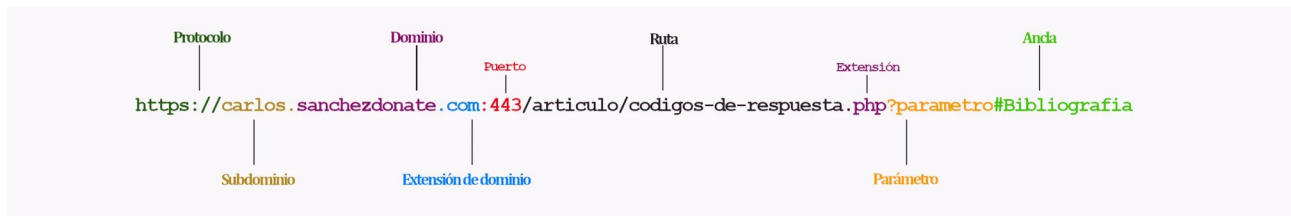


MÉTODOS HTTP	
Método HTTP	Significado en Restful Web Services
GET	Se utiliza para operaciones de sólo lectura. No generan ningún cambio en el servidor.
DELETE	Elimina un recurso en específico. Ejecutar esta operación múltiples veces no tiene ningún efecto.
POST	Cambia la información de un recurso en el servidor. Puede o no regresar información.
PUT	Almacena información de un recurso en particular. Ejecutar esta operación múltiples veces no tiene efecto, ya que se está almacenando la misma información sobre el recurso.
HEAD	Regresa solo el código de respuesta y cualquier cabecero HTTP asociado con la respuesta.
OPTIONS	Representa las opciones disponible para establecer la comunicación en el proceso de petición/respuesta de una URI.

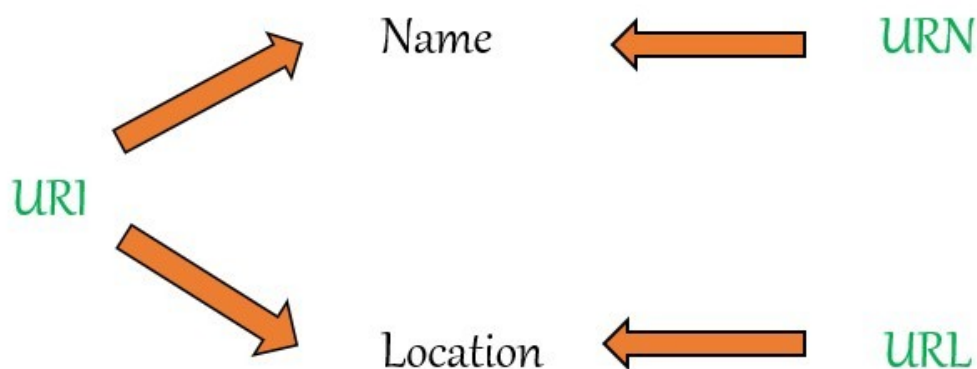
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme)/URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.

Un URI es una cadena de caracteres que se utilizan para identificar un recurso o un nombre en internet. La URL es un tipo de URI que indica la ubicación del recurso y permite localizarlo y la URN es una cadena especializada que indica el nombre de un recurso único en internet.

Se relaciona con el protocolo HTTP/HTTPS ya que estos son parte fundamental de la URI ya sea URN O URL como se ve en el esquema siguiente.



[Página que muestra algún tipo de URI más](#)



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

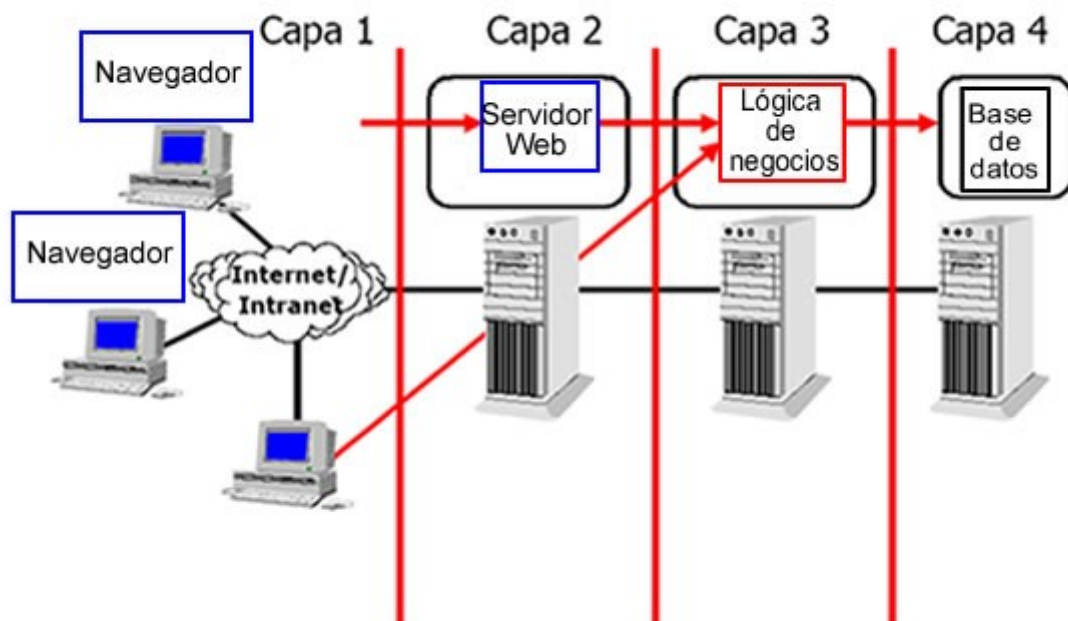
Una arquitectura multicapa es una arquitectura cliente-servidor en la que las funciones de presentación, lógica de negocio y gestión de datos están separadas físicamente.

Componentes: son las distintas capas de la aplicación (presentación, negocio, datos...).

- **Capa de presentación:** es la que ve el usuario conocida como interfaz gráfica le comunica a este la información y captura la información del usuario en un mínimo de proceso. Tiene que ser entendible y fácil de usar y se comunica solo con la capa de negocio.

- **Capa de negocio:** es donde están los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se comunica con la capa de presentación para recibir 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.

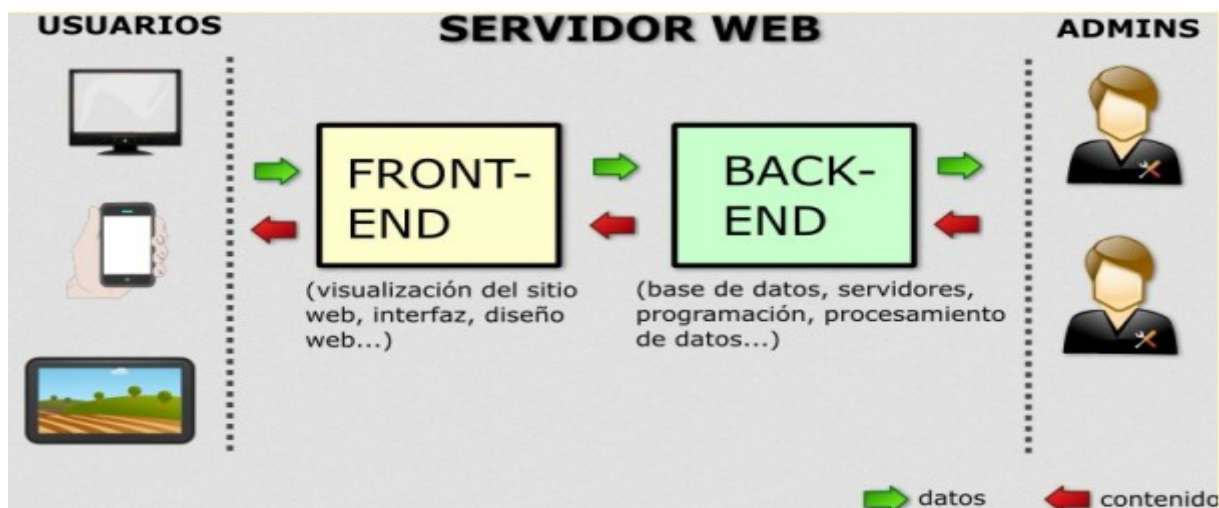
- **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



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

La funcionalidad front-end es la que tiene un usuario normal de una aplicación web esté o no autenticado y la funcionalidad back-end es la que tiene un usuario administrador, publicador, censor etc.

Al contrario de lo que muchos piensan no guarda relación ser programador del front o del back con este tipo de funcionalidades aunque tengan el mismo nombre.



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

Dependiendo del tipo de aplicación web creada salen unas u otras :

Aplicación web estáticas – Página web estática, **Aplicación web dinámica** – Página web dinámica, **Aplicación web interactiva** – Aplicación Web, **Aplicación web híbrida** – Mashup.

Página web Estática: es la que está compuesta por archivos HTML individuales por cada página que son pre-generados y presentados al usuario a través del navegador de la misma forma.

Página web Dinámica: está compuesta por elementos que cambian continuamente, son interactivos y funcionales, en lugar de ser simplemente informativos. Por supuesto, eso requiere utilizar más que solo código HTML y CSS.

Aplicación web: es el desarrollo de un programa de aplicación que se guarda y se almacena en un servidor remoto, al que el usuario accede a través del navegador.

Mashup: En desarrollo web, una mashup es una forma de integración y reutilización. Ocurre cuando una aplicación web es usada o llamada desde otra aplicación, con el fin de reutilizar su contenido o funcionalidad.

8. Componentes de una aplicación web.

Los componentes de una aplicación web del lado del servidor son:

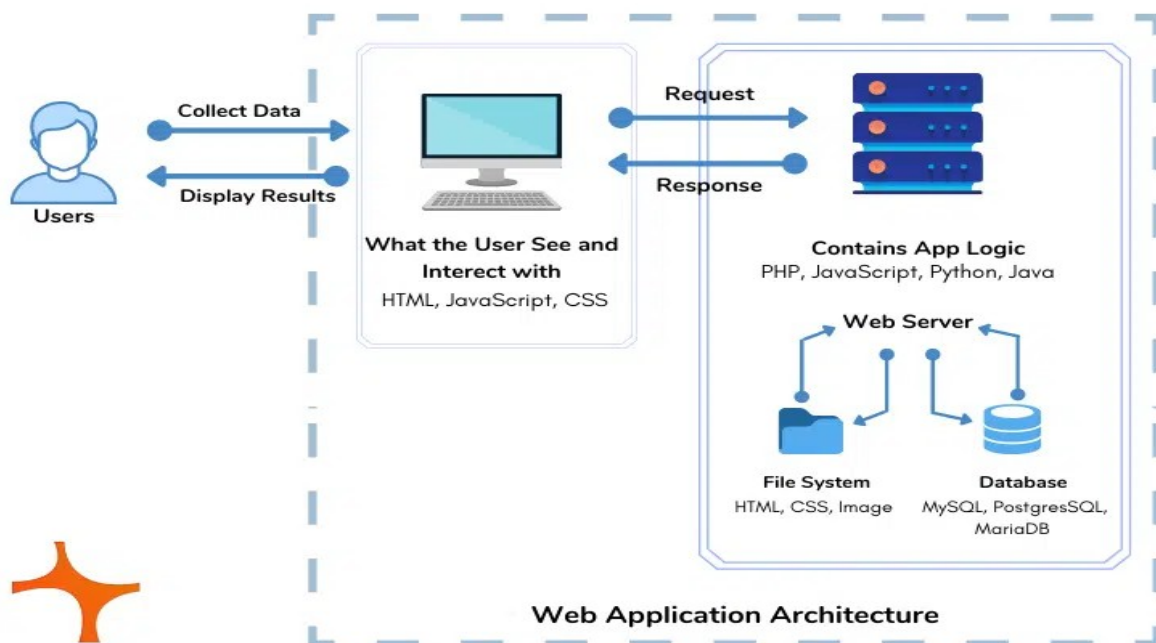
Servidor Web: es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

Módulo encargado de ejecutar el código: es un programa que ejecuta los ficheros escritos en el lenguaje de programación que se requiera hay módulos para PHP, Java etc.

Sistema gestor de base de datos (si se necesita): software que permite administrar una base de datos (MySQL, MariaDB...).

(Ficheros escritos en) Lenguajes de programación: ficheros utilizados para realizar la aplicación interpretados y ejecutados por el módulo que se mencionó anteriormente. Dependiendo del lado de la aplicación en el que nos encontremos los lenguajes de programación irán variando.

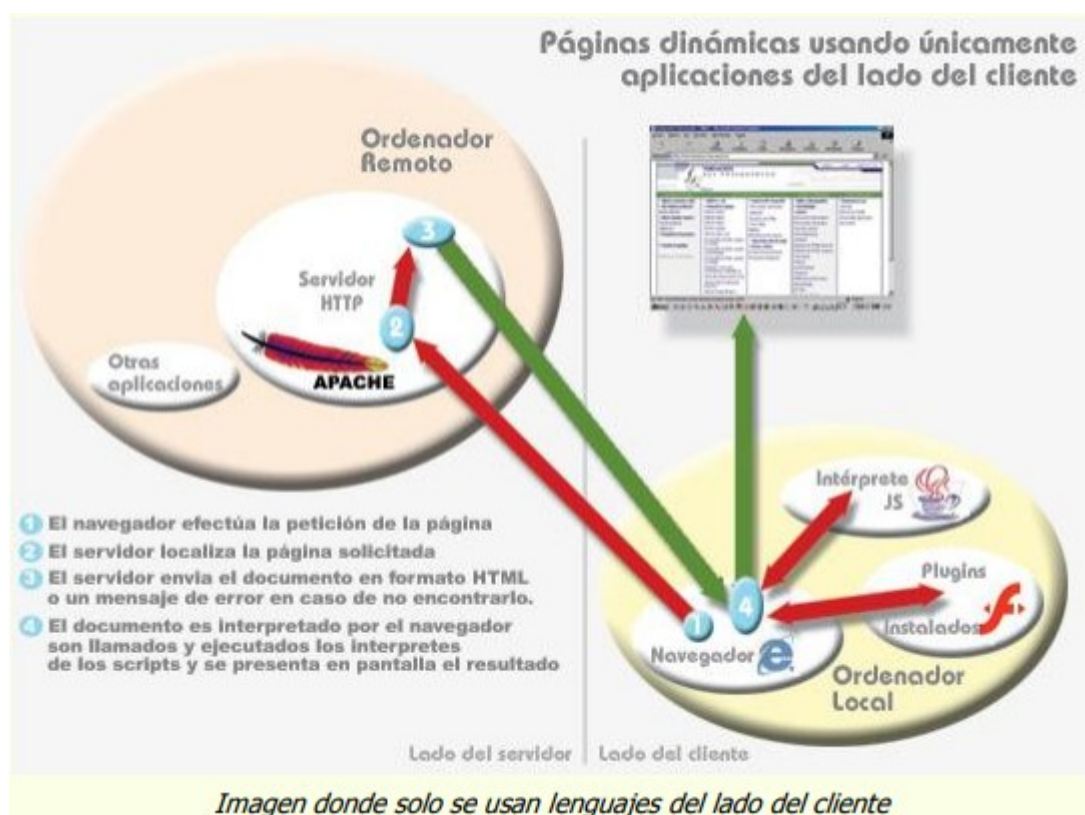
Web Application Architecture



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

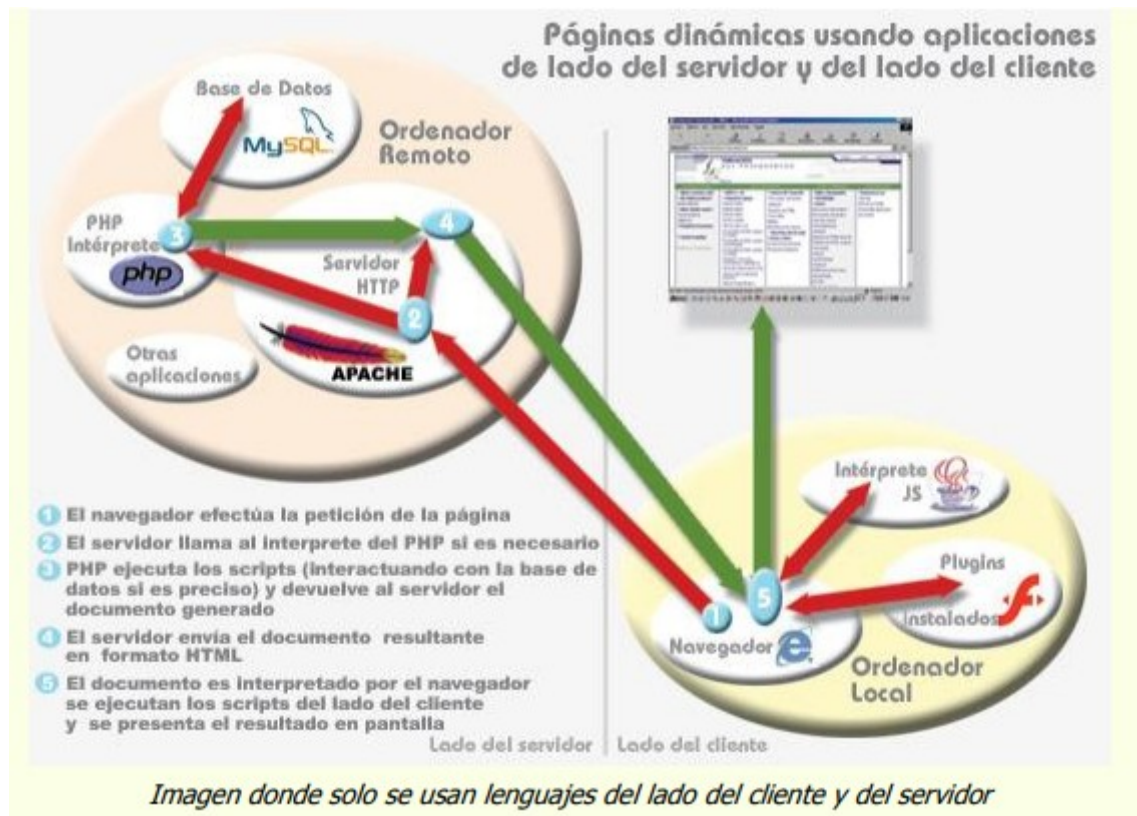
- Lado Cliente:

Los lenguajes que se utilizan en el lado del cliente son principalmente *HTML*, *JavaScript*, *Java*, *VBScript*, *Flash*, *CSS* etc.



- Lado Servidor:

Los lenguajes que se utilizan en el lado del servidor son **PHP**, **ASP** (Visual Basic .Net, C#), **JSP** y **Perl**.



<https://alfredcmmx.wordpress.com/wp-content/uploads/2013/02/tema-2-lenguajes-del-lado-del-cliente-y-servidor.pdf>

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

PHP: Una de las principales ventajas que ofrece es que es muy fácil de aprender; además, es multiplataforma. Incorpora una gran selección de funciones y ofrece una gran facilidad para establecer conexión con todo tipo de bases de datos, como Oracle o MySQL.

Ruby: Ruby es un lenguaje de programación dinámico y de código abierto, principalmente orientado a objetos, pero que también aborda la programación funcional. Entre sus características, destaca la simplicidad y la eficiencia: permite hacer mucho con pocas líneas de código.

ASP.NET: Comercializado por Microsoft. Una de las principales ventajas que ofrece ASP.NET es que permite el control de usuario personalizado y, además, ofrece una división entre la capa de diseño y el código. En cuanto a las desventajas, la más destacada es que consume una gran cantidad de recursos.

Perl: Su principal punto fuerte es que resulta muy útil en el procesamiento tanto de textos como de archivos. Además, está disponible en una gran selección de plataformas y sistemas operativos. Es uno de los lenguajes del lado del servidor preferidos por parte de los programadores ya que el desarrollo de aplicaciones con Perl es muy rápido. Además, ofrece una gran libertad para crear y desarrollar el programa “a medida”.

Server-side Programming Languages

Most popular server-side programming languages

© W3Techs.com	usage	change since 1 August 2024
1. PHP	75.8%	-0.3%
2. Ruby	6.0%	+0.1%
3. ASP.NET	5.8%	-0.1%
4. Java	5.0%	+0.1%
5. JavaScript	3.6%	+0.1%

percentages of sites

Fastest growing server-side programming languages since 1 August 2024

© W3Techs.com	sites
1. Scala	42.1
2. JavaScript	19.6
3. Java	15.0

daily increase of number of sites
per million

Find more details in the [server-side language surveys](#)

<https://w3techs.com/>

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

XAMPP es un paquete de software que incluye Apache, MySQL, PHP y Perl, lo que le permite ofrecer todos los componentes necesarios para desarrollar aplicaciones web con total garantía.

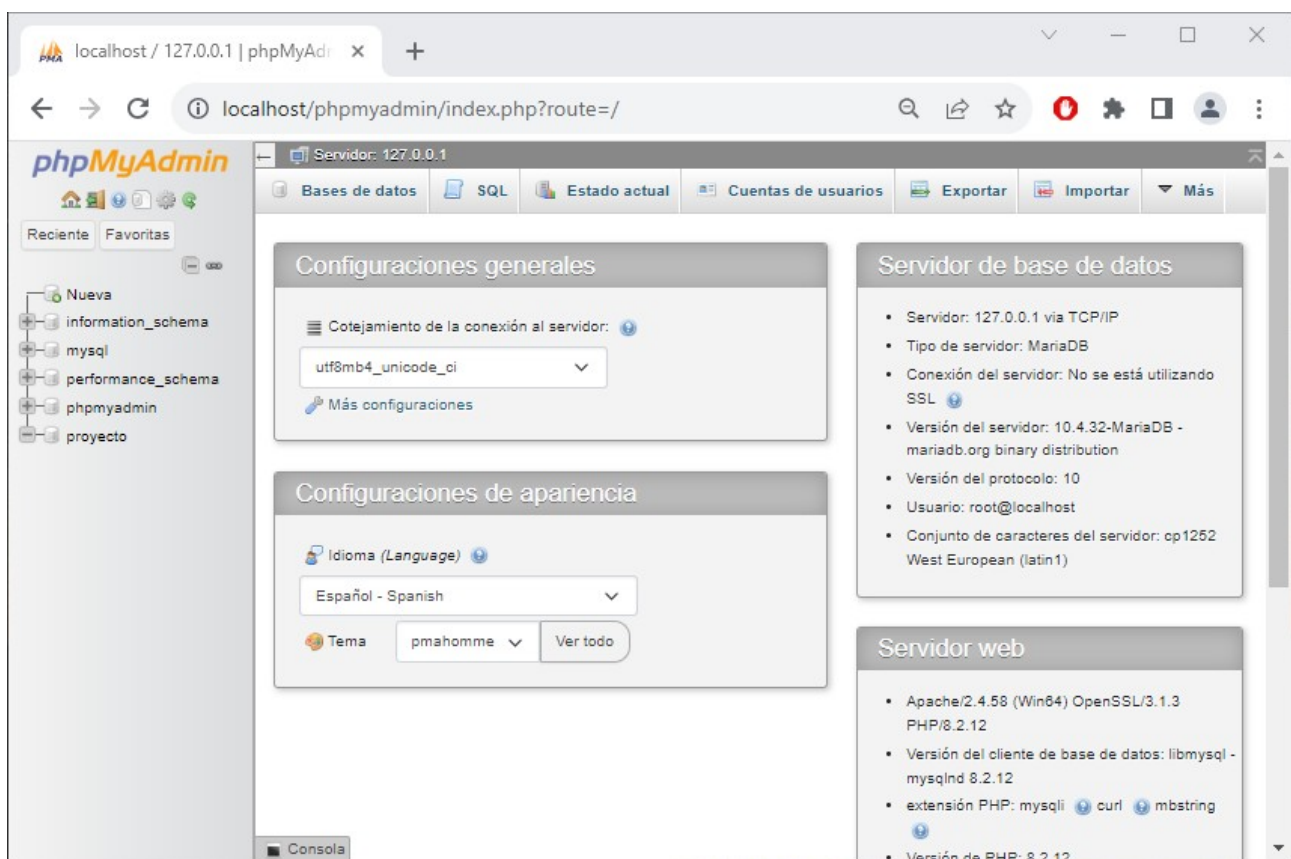
Sus características principales:

- Portabilidad y compatibilidad multiplataforma

XAMPP es compatible con diferentes sistemas operativos, incluyendo Windows, macOS y Linux. Además, la portabilidad de XAMPP facilita el traslado de proyectos entre diferentes ordenadores y sistemas operativos.

- Soporte para múltiples lenguajes de programación y bases de datos

Incluye soporte para PHP y Perl, además de gestionar bases de datos a través de MariaDB y MySQL.



<https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo>

12. En que 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.

Máquina virtual java para interpretar un programa precompilado y el JDK para precompilar el programa

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

Los IDE más utilizados son Visual Studio Code, Eclipse, NetBeans, IntelliJ Idea entre muchos otros. Eclipse y NetBeans son de código abierto.

Visual Studio Code: es un editor ágil que funciona muy bien en incluso en equipos con recursos limitados. Además, los desarrolladores lo aprecian porque su interfaz de usuario es muy intuitiva y permite comenzar a trabajar, incluso sin conocerlo de antemano, prácticamente sin necesidad de ninguna explicación. Otro de los factores de su rápida propagación es su capacidad para admitir una amplia variedad de lenguajes y frameworks, por lo que muchas comunidades lo usan en áreas tan dispares como el front-end o back-end.

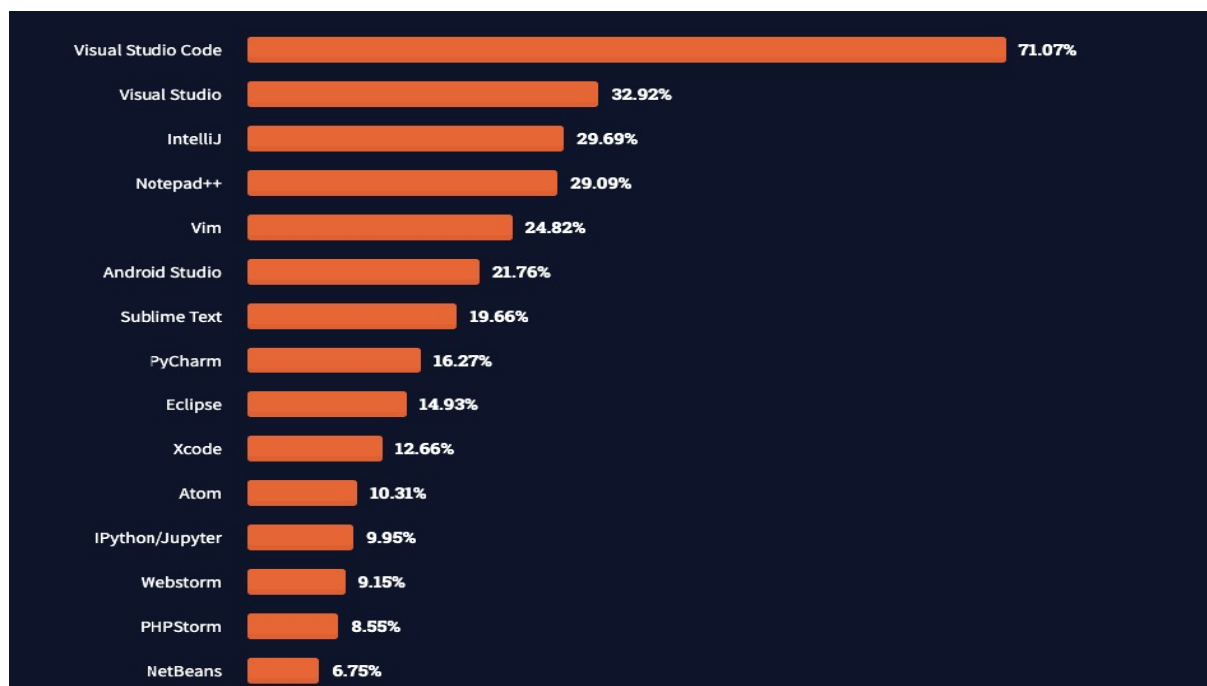
Eclipse: es posible desarrollar todo tipo de aplicaciones para web, escritorio o móviles usando Java, C, C++, JSP, Perl, Python, Ruby y PHP.

Su mayor punto fuerte reside en su asistente y capacidad para depurar errores. A medida que vas programando Eclipse lanza recomendaciones que nos ayudan a optimizar nuestro código hasta niveles que rozan el elegante guante blanco. Sin duda una excelente elección.

NetBeans: es el IDE oficial de Java pero también podemos desarrollar en otros lenguajes como PHP, C, C++ e incluso HTML 5.

Entre sus detractores, hay quien comenta que el desarrollo de una GUI sobre NetBeans es algo complicado y que no cuenta con la ingente cantidad de plugins y extensiones de las que sí cuenta Eclipse, además de carecer de soporte nativo para trabajar con webapps (.war, .jsp, y servlets).

IntelliJ Idea: se puede programar en diferentes lenguajes y proporciona soporte para estos pero lo más destacable es que ofrece integración con GIT.



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

Los más utilizados son Nginx, Apache y Cloudflare Server entre otros.

Nginx: es un servidor web de alto rendimiento que actúa como proxy inverso y balanceador de carga. Destaca por su eficiencia en el manejo de conexiones simultáneas, soporte para WebSockets y caché de contenido. Es ideal para sitios web de alto tráfico debido a su escalabilidad y bajo consumo de recursos.

Apache: es un servidor web de código abierto y gratuito, conocido por su flexibilidad y robustez. Es compatible con varios sistemas operativos, permite la adición de módulos para personalizar y ampliar sus funciones. Ofrece altos estándares de seguridad y actualizaciones regulares.

Cloudflare Server: ofrece un servicio gratuito de servidor de nombres de dominio (DNS) a todos los clientes que utilicen una red Anycast. Según W3Cook, el servicio de DNS de Cloudflare gestiona actualmente más del 35 % de los dominios DNS administrados. SolveDNS ha comprobado que Cloudflare cuenta de manera constante con una de las velocidades de búsqueda de DNS más rápidas del mundo, con una velocidad de búsqueda de 8,66 ms notificada en abril de 2016.

Web Servers

Most popular web servers

© W3Techs.com	usage	change since 1 August 2024
1. Nginx	33.8%	-0.2%
2. Apache	28.8%	-0.3%
3. Cloudflare Server	22.7%	+0.2%
4. LiteSpeed	13.8%	+0.3%
5. Microsoft-IIS	4.4%	-0.1%

percentages of sites

Fastest growing web servers since 1 August 2024

© W3Techs.com	sites
1. LiteSpeed	66.8
2. Cloudflare Server	34.2
3. Node.js	21.4

daily increase of number of sites
per million

Find more details in the [web server surveys](#)

<https://w3techs.com/>

15. Apache HTTP vs Apache Tomcat

Apache es un servidor web desarrollado por apache , para servir cosas estaticas y mediante extensiones para hostear aplicaciones dinámicas PHP, Python, .Net y otros.

Tomcat es un servidor web desarrollado por apache también, para servir aplicaciones que soporten el JVM aunque también soporta aplicaciones estáticas no es su fuerte. y no tiene algunas cosas para escalamiento que si tiene apache pero pueden trabajar juntos para tener esas.

Aspect	Apache HTTP Server	Apache Tomcat
Primary use case	Serves static content, handles HTTP requests.	Specialised for running Java Servlets and JSP applications.
Dynamic content	Limited support for dynamic content through modules.	Focuses on dynamic content using Java Servlets and JSP.
Architecture	Modular, process-based or threaded.	Java Servlet container, designed for Java-based applications.
Configuration	Configuration via text files.	Configuration through XML files and Java properties.
Language support	Supports various programming languages.	Primarily supports Java applications.
Performance	Efficient for serving static content.	Optimised for Java-based dynamic content.
Use cases	General web hosting, static content delivery.	Java web applications, servlets, and JSP deployment.

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

Un navegador web es un programa que permite ver la información que contiene una página web. El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar. Los más utilizados se detallan a continuación.

Chrome: es un navegador web de código propietario desarrollado por Google y compilado con base en componentes de código abierto (proyecto Chromium). Está disponible gratuitamente para diversas plataformas (Windows, MacOS, Android, iOS, Linux...).

Google Chrome es totalmente compatible con los estándares HTML5 y CSS3.

Safari: es un navegador web de código cerrado desarrollado por Apple Inc. Está disponible para Mac OS X, iOS (el sistema usado por el iPhone, el iPod Touch y el iPad) y Microsoft Windows.

Safari es totalmente compatible con los estándares HTML5 y CSS3.

Microsoft Internet Explorer: conocido comúnmente como IE, es un navegador web desarrollado por Microsoft para el sistema operativo Microsoft Windows desde 1995. Ha sido el navegador web más utilizado durante años, con un pico máximo de cuota de utilización del 95% entre el 2002 y 2003. Sin embargo, dicha cuota de mercado ha disminuido paulatinamente con los años. La versión 9 no es totalmente compatible con los estándares HTML5 y CSS3.

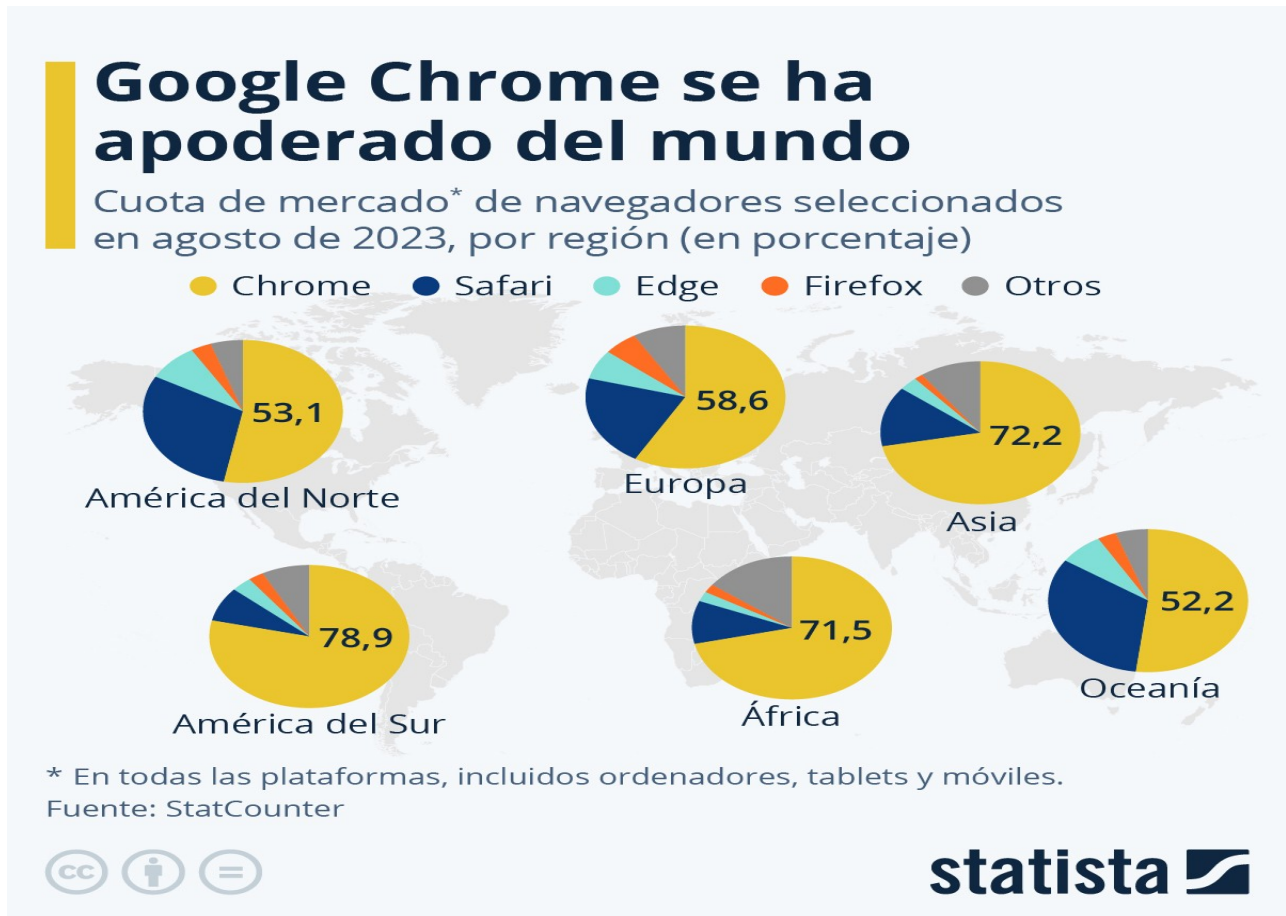
Mozilla Firefox: es un navegador web libre y de código abierto descendiente de Mozilla Application Suite y desarrollado por la Fundación Mozilla. Firefox puede ser utilizado en varios sistemas operativos (Windows, MacOS, Android, iOS, Linux...).

Firefox soporta completamente los estándares HTML5 y CSS3.

Microsoft Edge: es el navegador integrado por defecto por Microsoft en su sistema operativo Windows 10, y esta llamado a sustituir a Internet Explorer. Edge es compatible con los estándares HTML5 y CSS3.

Opera: es un navegador web y suite de Internet creado por la empresa noruega Opera Software, capaz de realizar múltiples tareas como navegar por sitios web, gestionar correo electrónico, contactos, fuentes web, charlar vía IRC y funcionar como cliente BitTorrent. Funciona en una gran variedad de sistemas operativos, incluyendo Microsoft Windows, MacOS, Android, iOS, GNU/Linux y FreeBSD.

Opera soporta totalmente HTML5 y CSS3.



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

PHPDocumentor es un generador de documentación de código abierto escrito en PHP. Automáticamente analiza el código fuente PHP y produce la API de lectura y documentación del código fuente en una variedad de formatos. PHPDocumentor genera la documentación en base al estándar formal PHPDoc. Es compatible con la documentación del código orientado a objetos y programación procedimental, además es capaz de crear documentos HTML, PDF, CHM y formatos Docbook. Se puede utilizar desde la línea de comandos o mediante una interfaz web.

[Manual de etiquetas PHPDocumentor](#)