

DESARROLLO DE APLICACIONES WEB EN ENTORNO SERVIDOR

JAVIER GUILLÉN BENAVENTE

CURSO 2025 - 2026

CONTENIDOS

- **Introducción y contextualización**
- **Modelos de programación en entornos cliente-servidor**
- **Generación dinámica de páginas web**
- **Componentes. Capas y funciones**
- **Lenguajes de programación en entorno servidor**
- **Integración con servidores web**
- **Tecnologías asociadas**

INTRODUCCIÓN

- El desarrollo web en entorno servidor es uno de los pilares fundamentales de la implementación web en la actualidad.
- La construcción de este entorno permitirá al cliente acceder a infinitos recursos servidos de forma estática o dinámica.
- Estos sistemas se organizan en módulos y capas que optimizan su construcción y funcionamiento.
- Analizaremos el funcionamiento de las peticiones y respuestas sobre un servidor, centrándonos en las páginas web dinámicas, gracias a las cuales la respuesta, en cada momento, se adaptará a la petición del usuario.

MODELOS DE PROGRAMACIÓN EN ENTORNOS CLIENTE/ SERVIDOR

- El funcionamiento de la Web es posible gracias a la coexistencia de una serie de componentes hardware y software:
 - Hubs, repetidores, puentes, pasarelas, enrutadores, etc.
 - Protocolos de comunicaciones (TCP, IP, HTTP, FTP, etc.)
 - Sistema de nombres de dominio (DNS) para la búsqueda y utilización de recursos.
 - Software específico para consumir los recursos.
- Componentes semánticos:
 - **URI:** Uniform Resource Identifier. Identifica los recursos web para su acceso y manipulación. Dos tipos. URN (Uniform Resource Name) y **URL** ([Uniform Resource Locator](#))
 - **HTML:** HyperText Markup Language. Lenguaje de marcas. Provee una representación de los documentos hipertexto en ASCII. Permite formatear texto, integrar imágenes, etc.
 - **HTTP:** Hypertext Transfer Protocol. Protocolo que permite a los componentes web comunicarse de una forma estándar y bien definida. Define el formato y el significado de los mensajes.

MODELOS DE PROGRAMACIÓN EN ENTORNOS CLIENTE/ SERVIDOR

- URL típica sería:
 - Protocolo (https, http, ftp, ...)
 - Anfitrión (servidor o host)
 - Puerto de red
 - Ruta al recurso

https://miapp.es:8080/admin/usuario.p
hp

esquema://[usuario];[password]@<maquina>:[puerto]/<camino>;[parametros]?[consulta]#[sección]

Los caracteres especiales se sustituyen por %<valor hex>

Localizador de recursos uniforme
(Uniform Resource Locator, URL)



- .com.es
- .nom.es
- .org.es
- .gob.es
- .edu.es



nombre del fichero

ciudad.html

provincia

subdirectorío

Como es
transmitida
la información:

http

Nombre del
ordenador anfitrión:

www.dominio.com.es

http://www.dominio.com.es/provincia/ciudad.html

MODELOS DE PROGRAMACIÓN EN ENTORNOS CLIENTE/ SERVIDOR

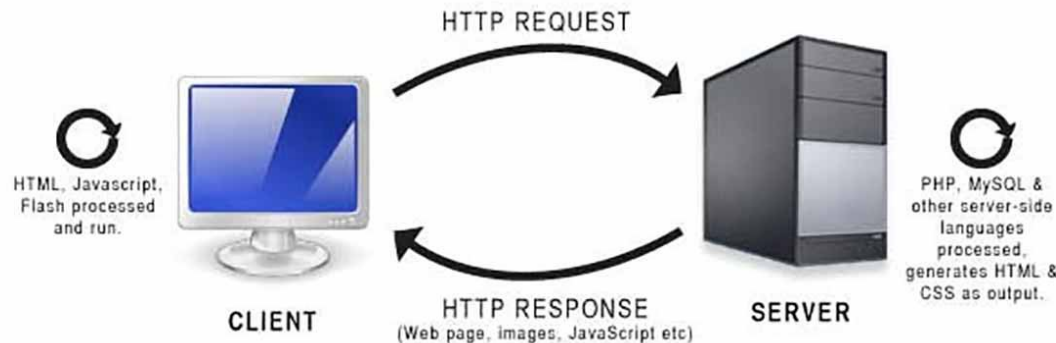


- El acceso a una página web es el resultado de un conjunto de interacciones entre varios entornos. Cuando accedemos a un sitio web a través de su URL, introducimos la dirección del sitio y la web se solicita a un servidor que, casi de forma automática, la devuelve y, de este modo, aparece en nuestro navegador.
- Estamos hablando de un **entorno de programación cliente-servidor**, en el que el cliente se comunica con un servidor para obtener cierto tipo de información. Por ejemplo, unos datos de consulta o, incluso, una interfaz de aplicación personalizada en función del usuario. La comunicación se establece mediante el intercambio de mensajes.
- Si el acceso se realiza a páginas cuyo funcionamiento no varía y se muestran tal y como se crearon, se trata de **páginas web estáticas**. En cambio, las **páginas web dinámicas** sí cambian su valor en función de diferentes variables o estados.

MODELOS DE PROGRAMACIÓN EN ENTORNOS CLIENTE/ SERVIDOR



- **Beneficios:**
 - Usabilidad
 - Escalabilidad
 - Flexibilidad
 - Interoperabilidad
- **Clientes** (originan la comunicación):
 - Navegadores
 - Robots (motores de búsqueda)
- **Servidores** (contestan generando respuesta HTTP)
 - Proveen acceso a los recursos
 - Controlan el acceso a los recursos restringidos
- **Servidores de altas prestaciones:**
 - ASP, JSP, PHP, ...
 - Caché
 - Más populares: Nginx, Apache, Microsoft IIS, etc.

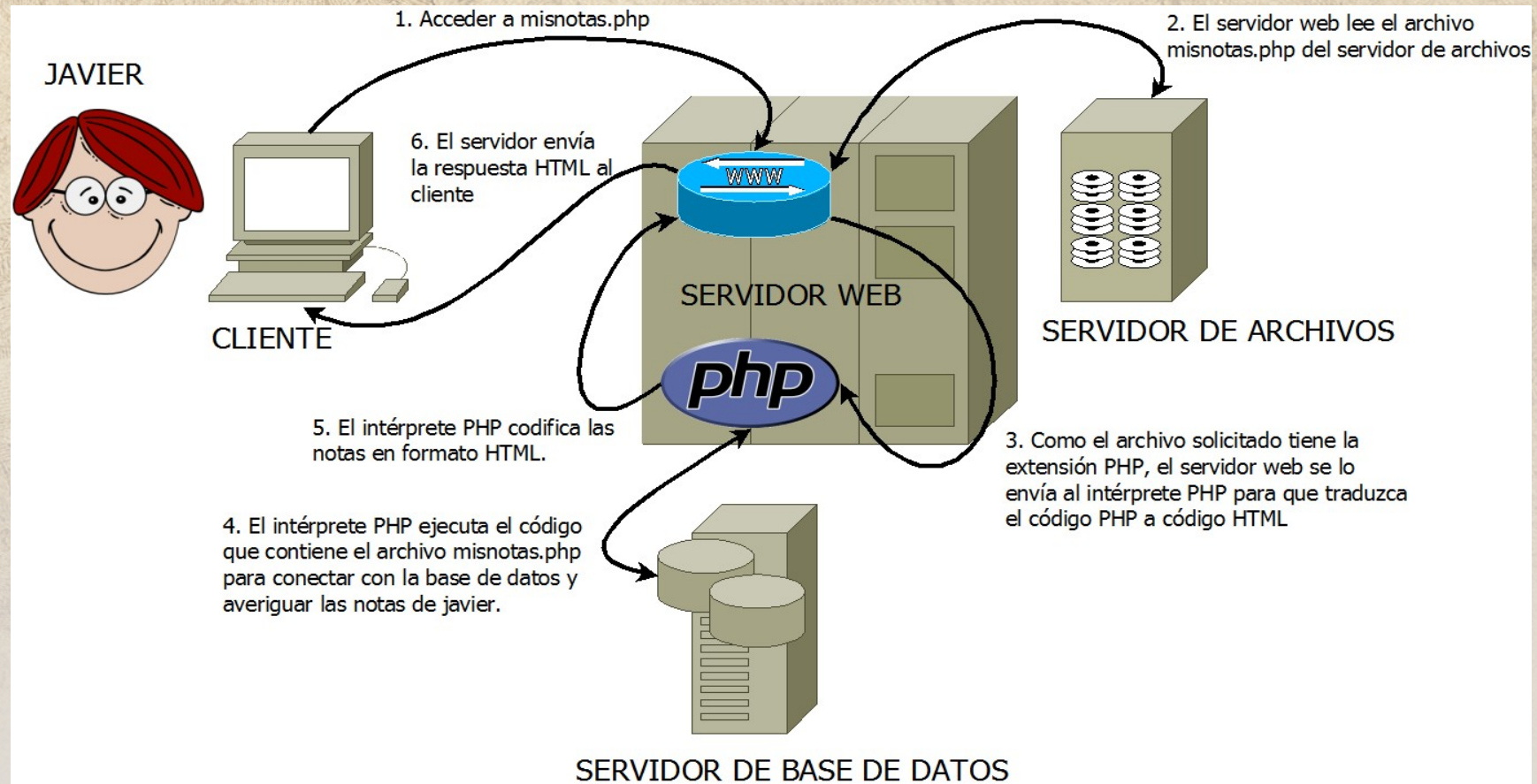


PÁGINAS WEB DINÁMICAS



- Las páginas web dinámicas muestran un contenido distinto en función de diferentes valores, como los datos de entrada o la autenticación del usuario.
- Funcionamiento:
 - El usuario introduce en el navegador una URL de la página o servicio que desea visitar.
 - Se envía una petición HTTP, en la que se suelen incluir diferentes datos de consulta al destino.
 - El servidor envía la petición al módulo donde se va a ejecutar el código para recuperar los datos de la consulta. Este se encargará de realizar las conexiones oportunas a los almacenes de datos (BBDD) y de recuperar el contenido.
 - Desde el servidor, se modela la respuesta en formato HTML. En función de los datos recuperados, se generará un contenido u otro.
 - Finalmente, el servidor envía la página resultado (solo HTML) al navegador (al cliente).

PÁGINAS WEB DINÁMICAS



PÁGINAS WEB DINÁMICAS



- Para construir este tipo de páginas, existen dos tipos de programación:
 - En un fichero HTML, en el que se incluyen funciones en lenguaje *JavaScript* que permiten evaluar el comportamiento del usuario y hacer que la página muestre un contenido u otro en función de esta entrada. En este tipo de casos los lenguajes principales serán HTML y JavaScript.
 - Existe otro tipo de páginas implementadas en lenguajes como *PHP*, *ASP.net* o *Java*. En este tipo de casos, las páginas mostradas al usuario son el resultado de la ejecución de un programa en el servidor cuyo resultado es un fichero en HTML que se devuelve al cliente. Hoy día también se puede hacer un servidor con Javascript (Node.js) y librerías Javascript.
- Algunos sitios web tienen un conjunto amplio de máquinas servidoras actuando como un único servidor (clústeres, máquinas virtuales, ...).
- Puede haber varios tipos diferentes: servidores frontales, de contenido, de base de datos.

COMPONENTES. CAPAS Y FUNCIONES



- Tras las diferentes estructuras de generación de páginas web, podemos observar que existe un conjunto de *componentes* clave que participan en el proceso:
 - **Lenguaje de programación:** utilizado para implementar las aplicaciones web.
 - **Servidor web:** se encarga de recibir la petición desde el cliente, de establecer la comunicación con el módulo encargado de generar la respuesta y, finalmente, de enviar la página resultante al navegador del cliente.
 - **Módulo de código:** se integra en el servidor web y, en función de la tecnología que implemente, llevará a cabo la generación de la página respuesta. El módulo de código suele establecer una comunicación con los componentes de almacenamiento de datos, habitualmente las bases de datos.
 - **Base de datos:** en el caso de ser necesario la recuperación, modificación o cualquier otra acción sobre datos, el módulo de código establece la conexión con una base datos.
 - **Entornos de desarrollo (frameworks):** se estructuran mediante un sistema de capas, puesto que, de esta forma, las funciones entre niveles quedan claramente diferenciadas. En concreto, hablamos de funciones a nivel lógico, de tal forma que será posible alojar cada una de estas en un servidor diferente.

COMPONENTES. CAPAS Y FUNCIONES



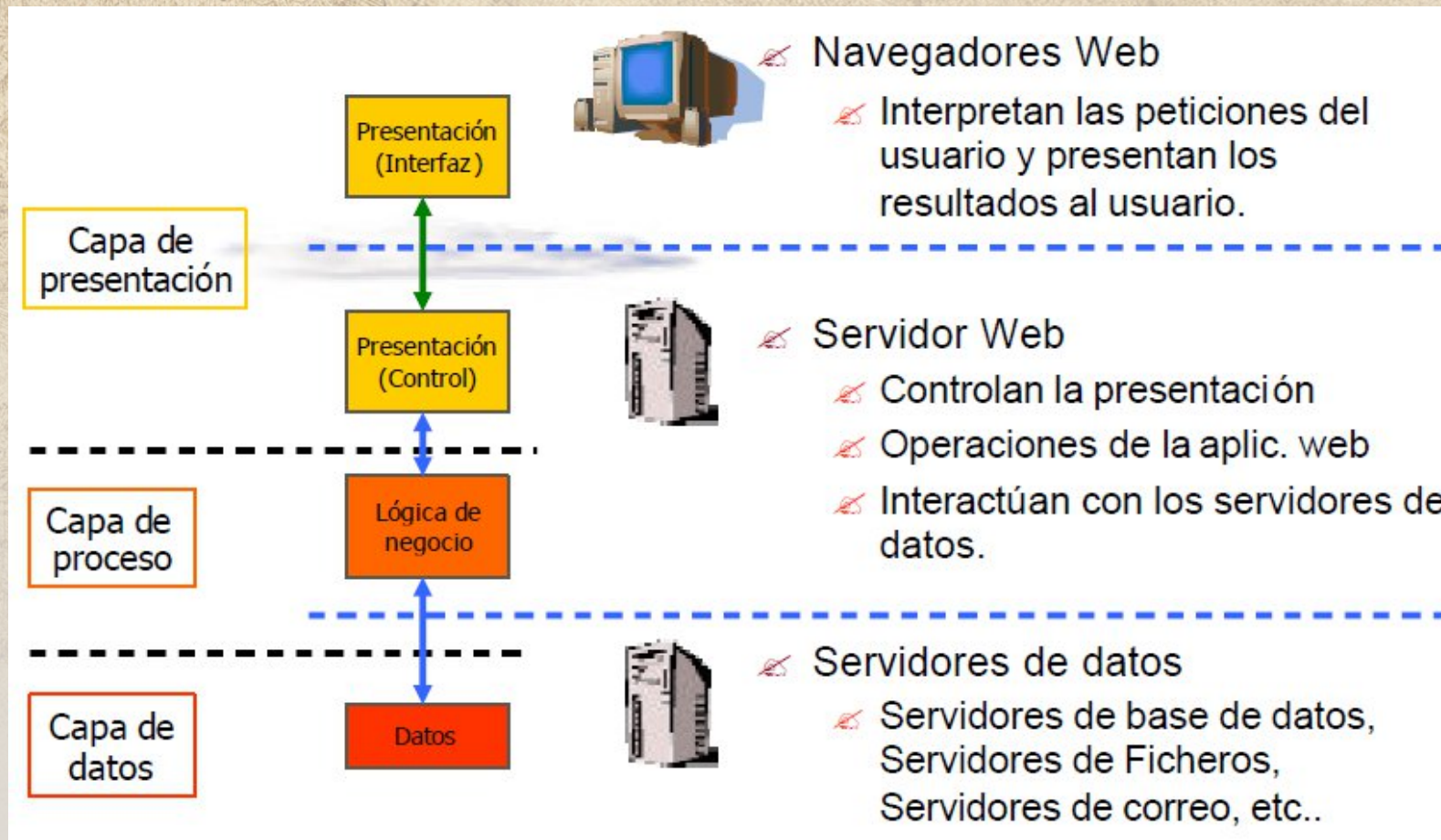
- **Aplicación web:** una aplicación basada en el modelo cliente/servidor, donde la aplicación es proporcionada por un servidor web, utilizando como interfaz páginas web (HTML). Hoy día tenemos también Servicios web y APIs Rest.
- **Componentes de las aplicaciones web:**
 - *Lógica de negocio:* la parte más importante, define los procesos que involucran la aplicación.
 - *Administración de los datos:* toda la parte de manipulación de BD y archivos.
 - *Interfaz:* los usuarios acceden a través de navegadores, móviles, tablets, otras aplicaciones, ...
- Las funcionalidades en la arquitectura cliente/servidor de la web se agrupan en diferentes **capas**. Cada capa se centra en la gestión de un aspecto determinado de la aplicación web.

COMPONENTES. CAPAS Y FUNCIONES



- **Modelo de tres capas.** Es el modelo más extendido, divide la estructura de la aplicación en tres capas:
 - **Capa de presentación:** toma la información del usuario y la envía al servidor. Recibe los resultados generando la presentación y la vista que se muestra al usuario.
 - **Capa de negocio:** recibe las peticiones del usuario, ejecuta los procesos necesarios, interactúa con la capa de datos y devuelve los resultados procesados a la capa de presentación.
 - **Capa de datos o de persistencia:** almacena, recupera y mantiene los datos, asegurando su integridad.
- Esta división origina un patrón de desarrollo web ampliamente utilizado en los frameworks de desarrollo web actuales, conocido como patrón **MVC** (Modelo Vista Controlador).

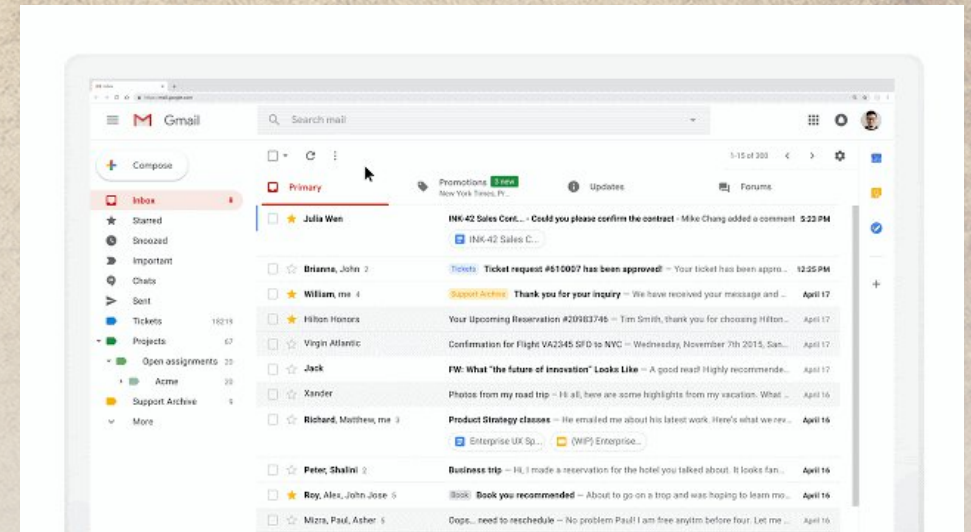
COMPONENTES. CAPAS Y FUNCIONES



COMPONENTES. CAPAS Y FUNCIONES



- Dependiendo de cómo se use el JavaScript en el cliente se diferencian tres tipos de aplicaciones web:
 - Aplicación web **interactiva** (cliente / servidor clásica, recargando en cada petición)
 - Aplicación web con **AJAX** (peticiones HTTP sin recargar la página del navegador)
 - Aplicación web **SPA**, Simple Page Application (Ajax al extremo). Se muestra la nueva página una vez cargada (sin pasar por la página en blanco). Los cambios en la página no suponen una recarga de la web, toda la interacción es por peticiones HTTP. A veces se utiliza el término RIA (Rich Internet Application).



LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR



- Una de las primeras tecnologías que se comenzó a utilizar en el desarrollo en entorno servidor fue CGI (Common Gateway Interfaces). Actualmente, está en desuso porque resultaba poco eficiente. Actualmente, los lenguajes de Scripting son una herramienta clave para el desarrollo en entorno servidor.
- Los **lenguajes de scripting** hacen referencia a aquellos lenguajes que pueden ser embebidos con el código HTML utilizado para la implementación de páginas web. A través de estos fragmentos, se posibilita la conexión con el servidor y se generan las páginas web dinámicas. El código se precompila o interpreta en el servidor y el resultado genera una página web HTML que se devuelve al cliente.

LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR



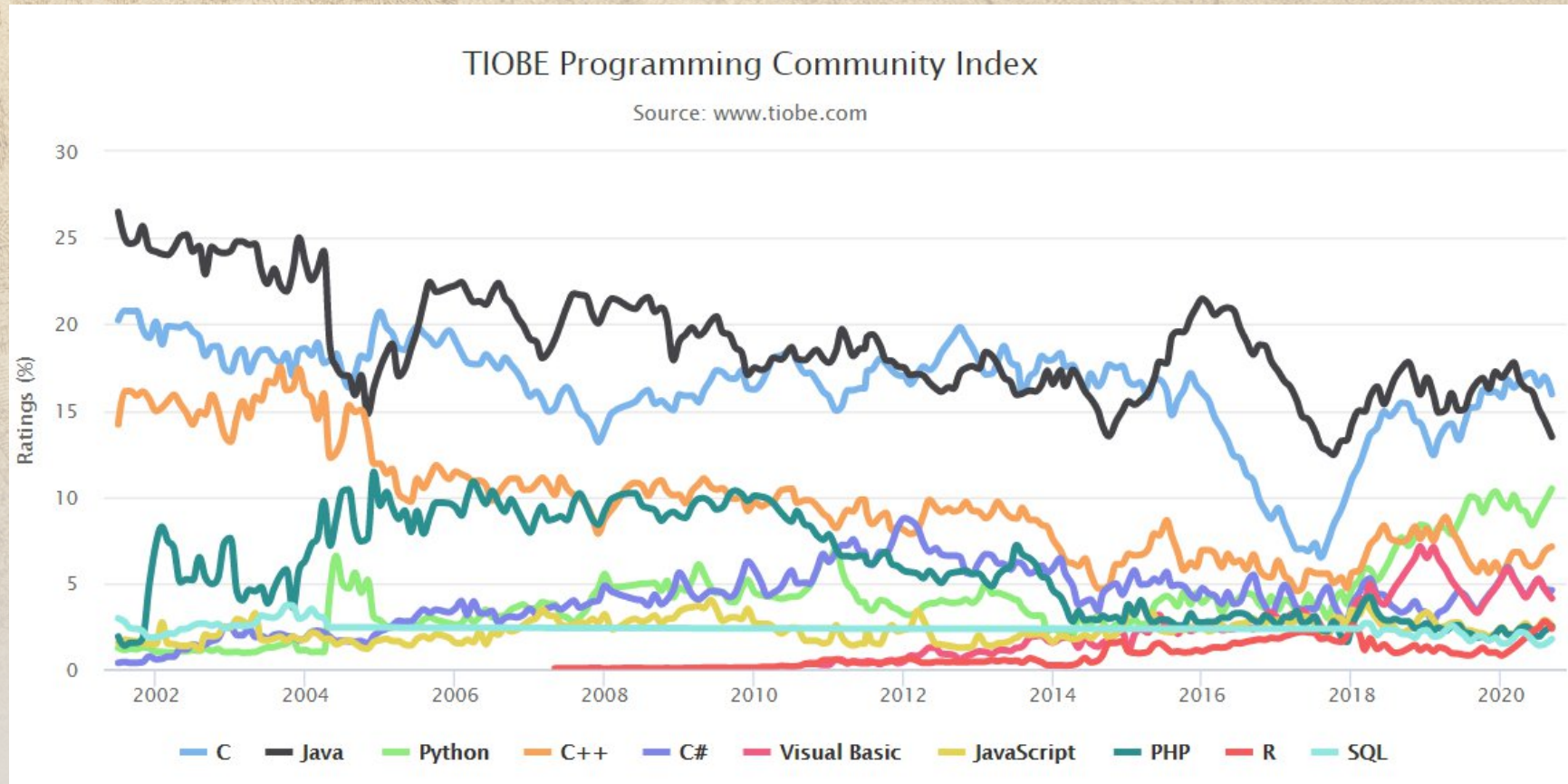
- **PHP** (Hypertext Processor): Lo soportan casi todos los servidores web y es de código abierto. En la actualidad, es uno de los lenguajes de programación en entorno servidor más utilizados, gracias a su potencia (interpretado) y simplicidad. Permite embeber fragmentos de código desarrollado en este lenguaje en una página web HTML. Se trata de una tecnología del lado del servidor, también conocido como **backend**. Permite POO. Framework *Laravel*.
- **JSP** (Java Server Pages): En el caso de servidores de aplicaciones empresariales, se utiliza Java como lenguaje de programación. Gracias a JSP, será posible desarrollar aplicaciones web que se ejecutan en varios servidores. Para el desarrollo de estas páginas, se utiliza HTML/XML, junto con etiquetas que permiten incluir sintaxis en Java (servlets). Framework *Spring*.
- **ASP** (Active Server Pages): Se trata de la tecnología desarrollada por Microsoft, encargada de crear páginas web dinámicas en el lado del servidor. Framework *.NET*
- **Python**: Un lenguaje de programación bastante utilizado en la actualidad, es orientado a objetos, multiplataforma, interpretado y con alto rendimiento. Además, *Django*, uno de los frameworks más utilizados en los servidores web más populares, está implementado en Python.

LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR



- **Frameworks con lenguajes de scripting.** Un framework es un conjunto de herramientas, librerías y buenas prácticas para resolver una serie de problemas. En desarrollo Backend, todos los frameworks tienen la principal tarea de crear un servidor web.
- Con estas herramientas puedes tener un desarrollo más rápido usando buenas prácticas y patrones de diseño, evitando escribir código repetitivo. Pero lo más importante, enfocar tu tiempo en solucionar problemas propios de tu desarrollo, en lugar de solucionar problemas que otros ya han solucionado.
- Python - Django
- Javascript - React, Angular y Vue
- Node.js - Express.js y Sail.js
- PHP - Laravel, Symfony
- Java - Spring
- Ruby - Ruby on Rails
- Microframework - Flask

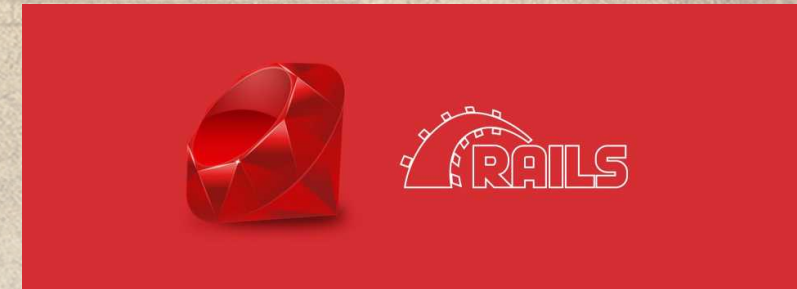
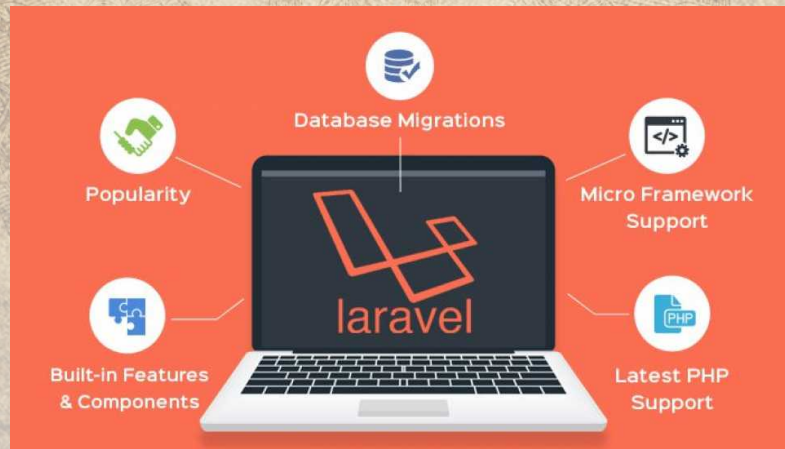
LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR



LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR



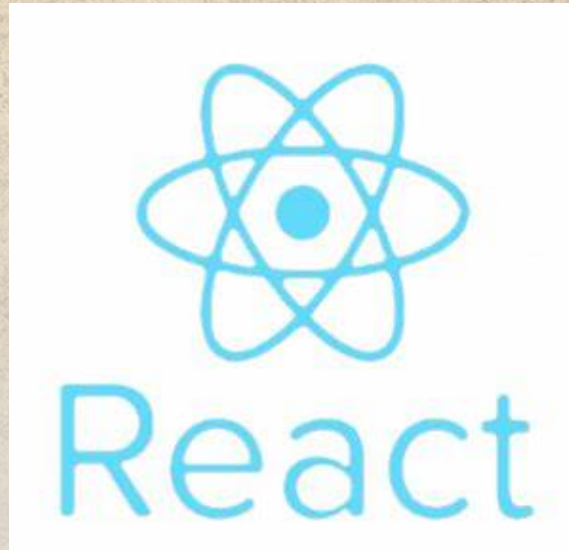
- **BACKEND FRAMEWORKS**



LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR



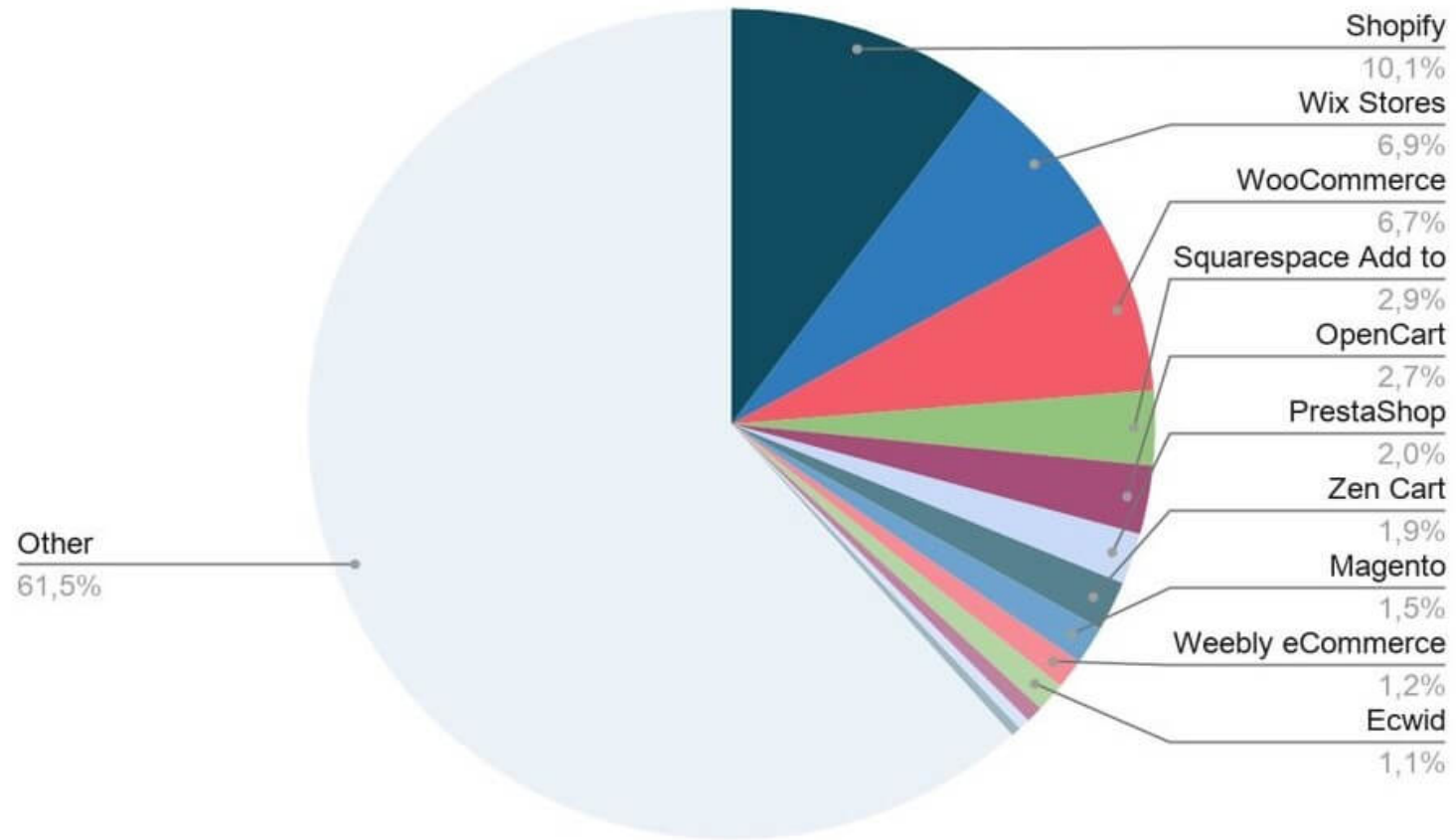
- **FRONTEND FRAMEWORKS**



BACKBONE.JS

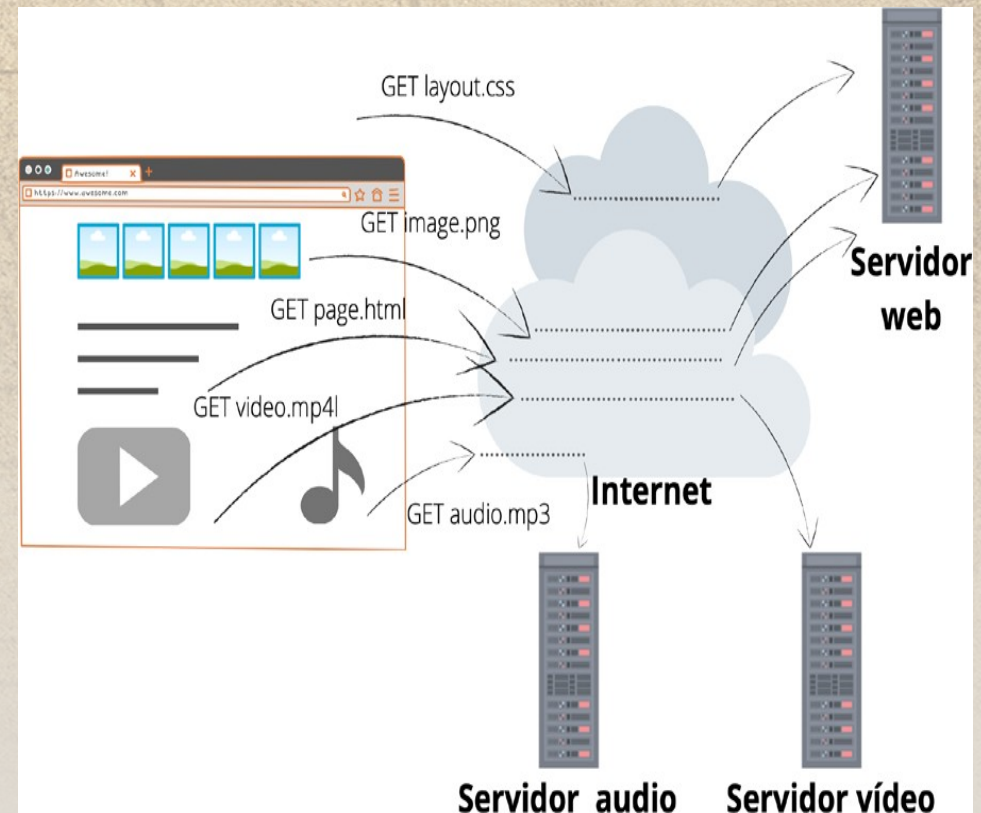


LENGUAJES DE PROGRAMACIÓN EN ENTORNO SERVIDOR



INTEGRACIÓN CON SERVIDORES WEB

- La conexión entre la petición en el entorno cliente y el servidor web se lleva a cabo mediante el modelado HTTP. Para que esta conexión se establezca, el cliente debe generar una petición con unos elementos concretos y una sintaxis exacta, comprensible por el servidor.



INTEGRACIÓN CON SERVIDORES WEB

- **PETICIÓN**

- **Método HTTP:** Se utilizan dos verbos, GET, mediante el cual el cliente realiza una petición al servidor para recuperar información. POST se utiliza para enviar información desde el cliente al servidor.
- **URL:** La dirección del recurso solicitado.
- **Versión del protocolo HTTP:** Por ejemplo, HTTP/1.1, HTTPS, ...
- **Cabeceras HTTP:** Elemento no obligatorio que se utiliza para añadir información adicional. Lo veremos.
- **Cuerpo del mensaje.**

- **RESPUESTA**

- Se incluyen los elementos que indican a través de un código numérico y un breve mensaje. Se indica si la petición ha tenido éxito, o ha habido un error, en cuyo caso se indica el error.

- | | |
|------------------------------------|----------------------------------|
| ▪ 1xx: Mensaje informativo. | ▪ 4xx: Error del cliente |
| ▪ 2xx: Exito | ▪ 400 Bad Request |
| ▪ 200 OK | ▪ 401 Unauthorized |
| ▪ 201 Created | ▪ 403 Forbidden |
| ▪ 202 Accepted | ▪ 404 Not Found |
| ▪ 204 No Content | ▪ 5xx: Error del servidor |
| ▪ 3xx: Redirección | ▪ 500 Internal Server Error |
| ▪ 300 Multiple Choice | ▪ 501 Not Implemented |
| ▪ 301 Moved Permanently | ▪ 502 Bad Gateway |
| ▪ 302 Found | ▪ 503 Service Unavailable |
| ▪ 304 Not Modified | |

TECNOLOGÍAS ASOCIADAS



- El desarrollo de aplicaciones web puede realizarse a través de varias tecnologías. En la actualidad, algunas de las más comunes son Java EE y AMP. En este módulo, nos vamos a centrar en la segunda. La tecnología **AMP** (*Apache + MySQL + PHP/Perl/Python*) es una arquitectura de código libre que permite desarrollar todo tipo de aplicaciones web.



Servidor de páginas web HTTP de código abierto, aloja los recursos web.



Sistema de gestión de bases de datos, que se utiliza para almacenar y administrar la información.

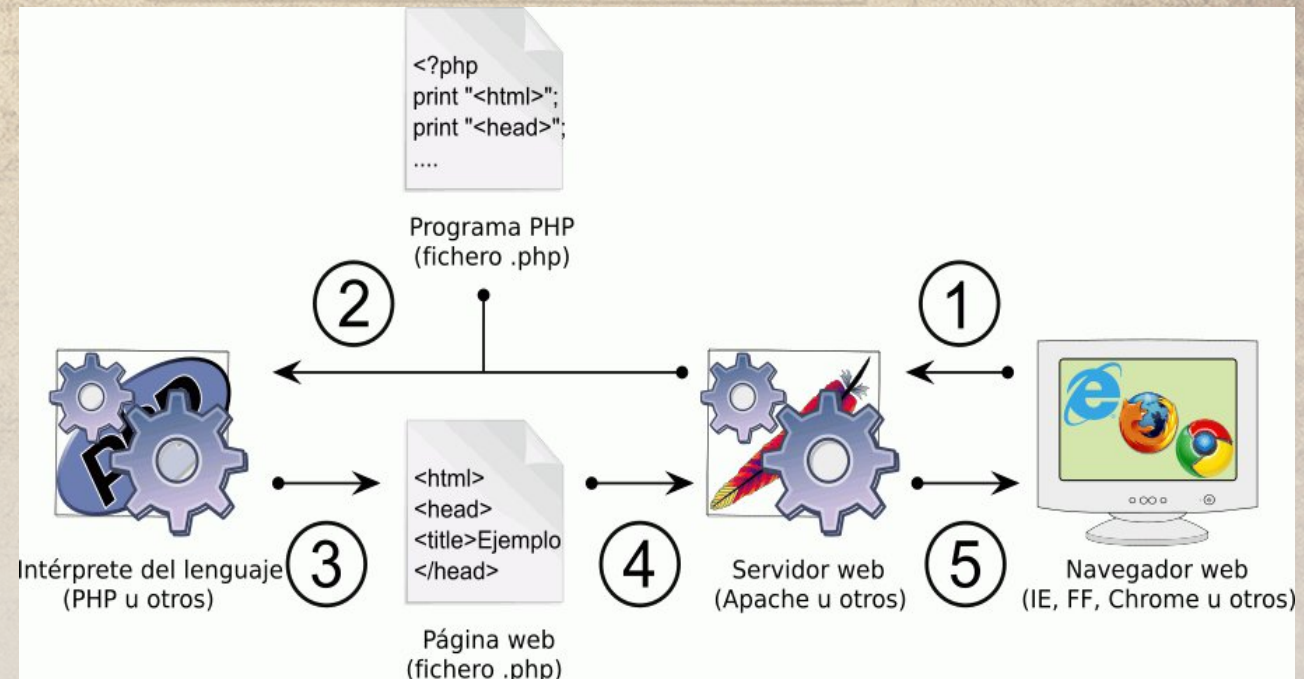


Lenguaje de código abierto que queda embebido en las páginas HTML y permite modelar la peticiones y posteriores respuestas de recuperación de un recurso. [Historia](#)

TECNOLOGÍAS ASOCIADAS - PHP



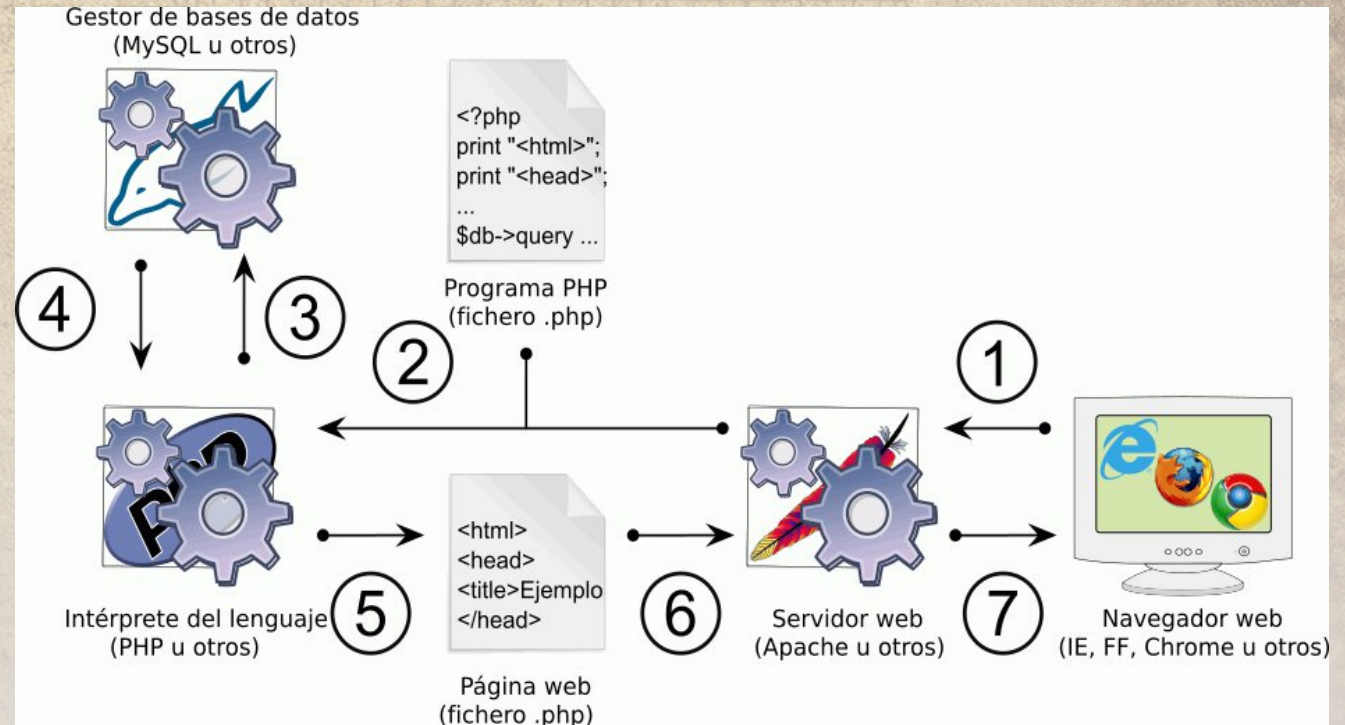
- El usuario escribe la dirección de la página web en su navegador
- (1) El navegador la solicita al servidor web correspondiente (este paso requiere la participación de máquinas intermedias que no se comentan aquí)
- (2) El servidor detecta que es un programa PHP y lo envía al intérprete del lenguaje
- (3) El intérprete del lenguaje completa la ejecución del programa.
- (4) El resultado final del programa (por ejemplo, el código fuente de una página web) se envía al servidor
- (5) El servidor envía el fichero al navegador
- El navegador muestra la página web al usuario



TECNOLOGÍAS ASOCIADAS – PHP + MYSQL



- El usuario escribe la dirección en su navegador
- (1) El navegador la solicita al servidor web correspondiente (este paso requiere la participación de máquinas intermedias que no se comentan aquí)
- (2) El servidor detecta que es un programa PHP y lo envía al intérprete del lenguaje
- (3) El intérprete ejecuta el programa. Si el programa incluye consultas a la BBDD, estas se realizan.
- (4) La base de datos entrega al intérprete el resultado de las consultas
- (5) El intérprete del lenguaje completa la ejecución del programa.
- (6) El resultado final del programa (por ejemplo, el código fuente de una página web) se envía al servidor
- (7) El servidor envía el fichero al navegador
- El navegador muestra la página web al usuario



TECNOLOGÍAS ASOCIADAS – XAMPP



- XAMPP es un entorno de desarrollo para Windows que incluye un servidor web Apache, una base de datos con MySQL y el software de desarrollo en entorno servidor PHP. Este proceso es clave para poder continuar con éxito el resto del módulo.
- Instalación y uso de Xampp. [Link](#).
- Configuración de Apache. [Link](#).

A screenshot of the XAMPP website. At the top, it says "XAMPP Apache + MariaDB + PHP + Perl". Below this, there's a section titled "What is XAMPP?" which describes it as a popular PHP development environment. To the right of this text is a video player showing an "Introduction to XAMPP" video with the XAMPP logo. At the bottom, there are three buttons for downloading XAMPP for Windows, Linux, and OS X, all version 7.4.7 (PHP 7.4.7).

XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

Download
Click here for other versions

 **XAMPP for Windows**
7.4.7 (PHP 7.4.7)

 **XAMPP for Linux**
7.4.7 (PHP 7.4.7)

 **XAMPP for OS X**
7.4.7 (PHP 7.4.7)