

## EJERCICIO 1

Investiga y haz una breve redacción sobre algún framework que te llame la atención relacionado con la programación web. Indica qué características incluye, adjunta alguna imagen si tiene interfaz gráfica (GUI) o indica si únicamente tiene interfaz de línea de comandos (CLI), si está muy extendido y en qué tipo de proyectos se implementa, etc.

### CakePHP



CakePHP es un framework para PHP de código abierto creada para servir de ayuda a los programadores para crear, mantener y desplegar aplicaciones web. Utiliza patrones comunes como el modelo vista-controlador o el mapeo objeto-relacional. En la web del framework, dicen ayudar al cliente a centrarse en el código que realmente necesitas (“...consigue una copia de CakePHP, empieza con lo verdaderamente importante y no reinventes la rueda cada vez que te incorpores a un nuevo proyecto.”).

En cuanto a historia, CakePHP empezó en 2005, cuando “Ruby On Rails” estaba ganando popularidad y utiliza muchos de sus conceptos (Ruby on Rails, también conocido como RoR o Rails, es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma del patrón Modelo Vista Controlador).



(Icono de “Ruby On Rails”)

CakePHP dice tener desarrolladores y una comunidad apoyando el proyecto constantemente, lo cual nos ofrece frecuentes actualizaciones e información para ofrecernos en cualquier momento y sobre cualquier tema acerca del framework.

En cuanto a características importante que necesitamos saber, están estas:

- Licencia MIT (según ellos, perfecta para el uso de aplicaciones comerciales).

-CRUD integrado para la interacción con la base de datos (CRUD es el acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar). Este concepto se utiliza para describir las cuatro operaciones básicas que pueden realizarse en la mayoría de las bases de datos y sistemas de gestión de información).

-Soporte de aplicación [scaffolding] (El scaffolding consiste en la generación de código a partir de plantillas predefinidas y de una especificación proporcionada por el desarrollador. Se suele utilizar para generar código repetitivo que se puede especificar fácilmente y generar a partir de una plantilla).

-Validación integrada.

-Plantillas rápidas y flexibles (sintaxis de PHP, con ayudantes[helpers]).

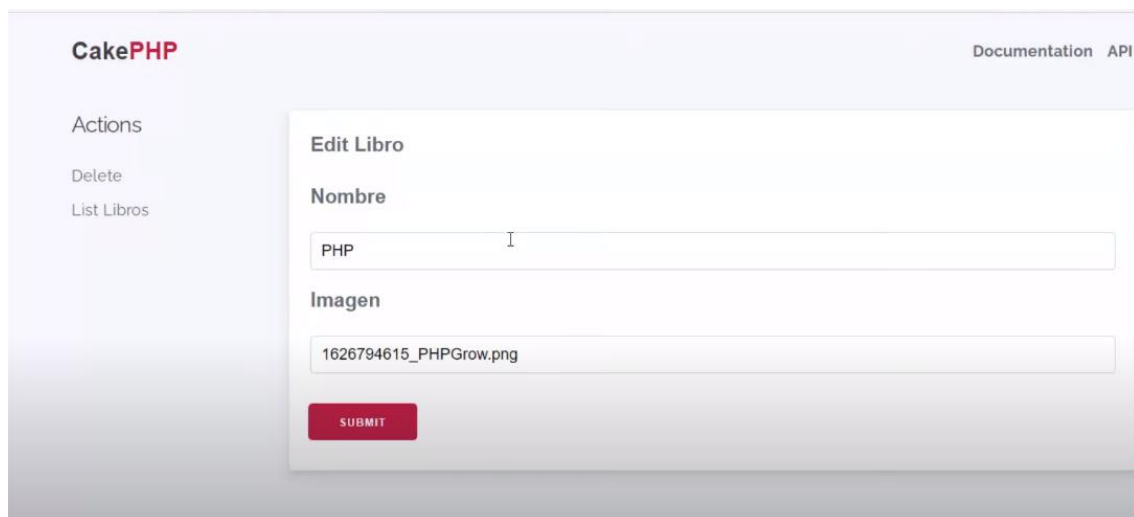
-Ayudantes para AJAX, JavaScript, formularios HTML y más.

-Componentes de Email, Cookie, Seguridad, Sesión y Manejo de solicitudes.

-Caché flexible.

-Funciona en cualquier subdirectorio del sitio web, con poca o ninguna configuración de Apache.

Ahora que ya sabemos todo del framework, vamos a ver como es visualmente y como se utiliza. CakePHP tiene tanto interfaz gráfica como interfaz de comandos. Aunque la mayoría de cosas las realizaremos en forma de comando, la interfaz gráfica nos permite realizar varias acciones también, facilitándonos el trabajo.



The screenshot displays the CakePHP web interface. At the top left, the 'CakePHP' logo is visible. To the right, there are links for 'Documentation' and 'API'. On the left side, under the heading 'Actions', there are links for 'Delete' and 'List Libros'. The main content area shows a form titled 'Edit Libro'. This form has two input fields: 'Nombre' with the value 'PHP' and 'Imagen' with the value '1626794615\_PHPGrow.png'. Below these fields is a red 'SUBMIT' button.

La interfaz visual del framework es algo así. Esta creado de una forma simple y sencilla de forma que sea fácil de entender y de usar para el usuario. De todas formas, el código es donde realizas todo.

Si quieres saber algo más de CakePHP, tiene un blog llamado “Bakery” (bakery en español significa panadería, juego de palabras ya que el framework se llama CakePHP, cake en español es tarta) en el cual suben todas las actualizaciones, artículos, plugins, tutoriales y muchas cosas más que nos pueden resultar de ayuda cuando lo utilicemos.

#### Fuentes:

- <https://blog.hubspot.es/website/que-es-crud#:~:text=CRUD%20es%20el%20acr%C3%B3nimo%20de,sistemas%20de%20gesti%C3%B3n%20de%20informaci%C3%B3n.>
- <https://book.cakephp.org/1.3/es/The-Manual/Beginning-With-CakePHP/What-is-CakePHP-Why-Use-it.html>
- <https://www.bestwebframeworks.com/web-framework-review/php/3/cakephp/>
- [https://lbd.udc.es/Repository/Publications/Drafts/1472199879813\\_CEDI\\_2016\\_paper\\_80.pdf](https://lbd.udc.es/Repository/Publications/Drafts/1472199879813_CEDI_2016_paper_80.pdf)
- <https://es.wikipedia.org/wiki/CakePHP>
- [https://es.wikipedia.org/wiki/Ruby\\_on\\_Rails](https://es.wikipedia.org/wiki/Ruby_on_Rails)

## EJERCICIO 2

Realiza un documento usando diagramas y detallando el proceso de carga de una página web. Contempla la posibilidad de tener que ejecutar script tanto en cliente como en servidor. Identifica en un bloque cada uno de los elementos (CLIENTE, SERVIDOR o MEDIOS) o programas necesarios (programa navegador, protocolo http en cliente, servidor dns, intérprete de JavaScript, etc).

El proceso de carga de una página web se divide en varios pasos:

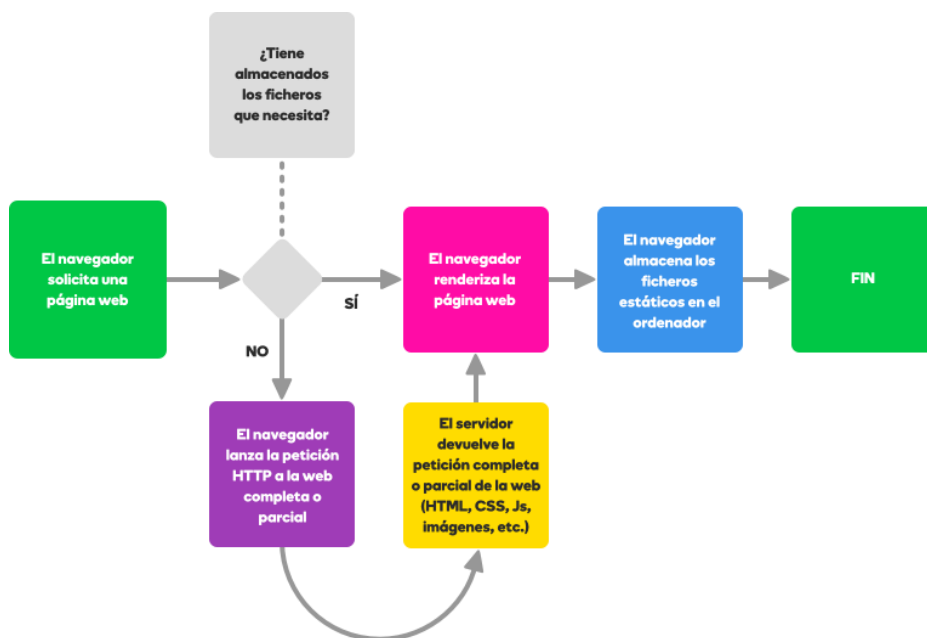
1º: Introduces la url en el buscador de tu navegador. Este lo interpreta y busca el servidor de ese sitio web.

2º: El navegador encuentra ese servidor y solicita el acceso y obtención de los archivos necesarios para cargar la página correctamente (normalmente archivos HTML, JavaScript, CSS e imágenes).

3º: Una vez el servidor nos da acceso a estos archivos, el navegador empieza a descargarlos y a la vez empieza a cargarlos para poder verlos en pantalla. Esto se realiza imprimiendo el documento HTML con su configuración del CSS y con el código de JavaScript. Al ser JavaScript, el script se debería ejecutar en el lado cliente, sin embargo, este también puede ser ejecutado del lado del servidor.

4º: Al final, tenemos toda la página cargada correctamente y ya el usuario puede interactuar sin ningún problema con la página, realizando acciones como entrar en otros links, ver imágenes y videos, o enviar información.

Aquí un diagrama representando esto que he dicho:



Fuentes:

- <https://www.mediummultimedia.com/web/como-se-carga-una-pagina-web/>
- <https://es.godaddy.com/blog/mejora-la-velocidad-de-carga-web-con-la-cache-del-navegador/>
- <https://www.cloudflare.com/es-es/learning/serverless/glossary/client-side-vs-server-side/>

### **EJERCICIO 3**

**Investiga las características y diferencias de cada uno de estos servidores web: plataforma en la que se ejecutan, lenguajes de programación que interpretan, propietario, etc.**

- **Apache HTTP Server**
- **Microsoft IIS (Internet Information Services)**
- **NGINX**
- **Lighttpd**
- **Sun Java System Web Server**

#### Apache HTTP Server:

- Se ejecuta en plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y más.
- Interpreta diferentes lenguajes como PHP, Python o Perl.
- Desarrollador: Robert McCool.
- Programado en C.
- Código abierto.

#### Microsoft IIS:

- Se ejecuta en sistemas operativos Windows.
- Interpreta lenguajes ASP/ASP.net, PHP, Perl, etc.
- Desarrollador: Microsoft.
- Programado en C++.
- No es código abierto.

### NGINX:

- Se ejecuta en sistemas operativos Unix y Windows.
- Interpreta PHP.
- Desarrollador: Igor Sysoev.
- Programado en C.
- Código abierto.

### Lighttpd:

- Se ejecuta en sistemas operativos Unix y existe una versión para Windows conocida como "Lighttpd For Windows" mantenida por Kevin Worthington.
- Interpreta PHP, Ruby, Python y otros.
- Desarrollador: Jan Kneschke.
- Programado en C.
- Código cerrado.

### Sun Java System Web Server:

- Se ejecuta en sistemas operativos Unix y Windows.
- Interpreta PHP, Perl, Python y más.
- Desarrollador: Oracle.
- Programado en Java.
- Código cerrado.

### Fuentes:

- [https://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache#:~:text=El%20servidor%20HTTP%20Apache%20es,seg%C3%BAn%20la%20normativa%20RFC%202616](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache#:~:text=El%20servidor%20HTTP%20Apache%20es,seg%C3%BAn%20la%20normativa%20RFC%202616).
- <https://www.hostinger.es/tutoriales/que-es-apache/#:~:text=Puedes%20utilizar%20Apache%20junto%20con,para%20ese%20prop%C3%B3sito%20que%20Apache>.
- [https://es.wikipedia.org/wiki/Internet\\_Information\\_Services](https://es.wikipedia.org/wiki/Internet_Information_Services)
- <https://es.wikipedia.org/wiki/Nginx>
- <https://es.wikipedia.org/wiki/Lighttpd>

- <https://www.martechforum.com/herramienta/sun-java-system-web-server/>
- <https://docs.oracle.com/cd/E19263-01/817-7518/index.html>

## **EJERCICIO 4**

**Investiga en la red buscando información de diferentes arquitecturas en un despliegue web:**

**-Arquitectura LAMP/WAMP**

**-Arquitectura MEAN**

**-Arquitectura Java EE**

**Interesa que hables de qué tipos de tecnologías se usan, así como del servidor web que atiende las solicitudes.**

### Arquitectura LAMP/WAMP:

Las letras L y W significan Linux y Windows respectivamente, y las otras letras significan Apache, MySQL y PHP. El servidor web Apache y la base de datos MySQL se ejecutan en el sistema operativo Linux o Windows y se comunican mediante PHP.

Apache es un servidor web HTTP de código abierto. Está desarrollado y mantenido por una comunidad de usuarios en torno a la Apache Software Foundation.

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto.

PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

El proceso del servidor web para atender las solicitudes es algo así:

### **Recepción de solicitudes**

El servidor web Apache recibe la solicitud entrante del navegador. Si la solicitud es para cargar un archivo estático, el servidor Apache responderá directamente con el contenido adecuado. Si la solicitud es para contenido dinámico, el servidor Apache pasará la solicitud al componente PHP. El componente PHP buscará y cargará el archivo PHP apropiado.

### **Procesamiento de solicitudes**

El archivo PHP contiene funciones PHP que son códigos para generar contenido dinámico. El componente PHP procesa las funciones de PHP, como la conversión de unidades de medida o la creación de un gráfico de ventas.

### **Respuesta de devoluciones**

El PHP pasa los resultados calculados al servidor web en formato HTML. Al mismo tiempo, también almacena datos nuevos en la base de datos MySQL. El servidor HTTP Apache envía los resultados HTML dinámicos al navegador del usuario.

### Arquitectura MEAN

Esta arquitectura está compuesta por cuatro tecnologías, las cuales son MongoDB, Express, Angular y Node.js.

MongoDB es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad, y un modelo de consultas e indexación avanzado.

Express es un web application framework para Node.js, es decir es un framework para hacer aplicaciones web. Es minimalista, rápido y fácil de usar y es la mejor opción cuando estás trabajando con Node para el lado del servidor, porque se adapta muy bien a su filosofía.

Angular es un framework de JavaScript con el que puedes desarrollar el frontend de tu aplicación, es decir la parte de tu aplicación con la que tus usuarios interactúan directamente.

Node.js es un runtime, es decir es un programa que corre JS, construido sobre el motor de JavaScript que usa Chrome. Node es el entorno de desarrollo de la capa del servidor, te permite desarrollar rápidamente aplicaciones escalables a nivel del servidor.

Node.js funciona mejor con muchas solicitudes de recursos bajos en comparación con las solicitudes de uso intensivo de recursos. Mientras una sola hebra protege contra los estancamientos de proceso, no es inmune a que un proceso grande congele el sistema para todos los clientes.

### Arquitectura Java EE

Es una plataforma de programación creada para desarrollar y ejecutar software de aplicaciones en el lenguaje Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

La plataforma divide el trabajo a realizar en la implementación en dos partes:



- La lógica de presentación y de negocio a implementar por el desarrollador.
- Los servicios estándar que ofrece la plataforma Java EE.

La plataforma Java EE utiliza un modelo de programación distribuido en distintas capas. La lógica de la aplicación se divide en distintos componentes dependiendo de su funcionalidad, y estos son desplegados en las distintas capas dependiendo de a cuál pertenecen. Las distintas capas son:

- Capa cliente: responsable de la interacción con el usuario.
- Capa web: responsable del control de la aplicación y en ocasiones también de la interacción con el usuario.
- Capa de negocio: responsable de la lógica de la aplicación propiamente dicha.
- Capa de datos: responsable de la persistencia de datos y/o lógica especializada.

En cuanto a las solicitudes, funciona así:

1. El cliente, por ejemplo, un navegador, solicitará una funcionalidad desde la interfaz visual.
2. Dicha petición entrará a través de un Java Servlet.
3. Dicho Java Servlet, analizará qué se está pidiendo, qué información adicional aporta y decidirá que Enterprise JavaBean cubre dicha petición.
4. Lo invocará, y tras recibir un resultado, decidirá qué JavaServer Page (JSP) muestra dicho resultado al cliente.
5. El resultado será devuelto y mostrado.

Fuentes:

- <https://aws.amazon.com/es/what-is/lamp-stack/#:~:text=base%20de%20datos.-,%C2%BFQu%C3%A9%20es%20la%20arquitectura%20LAMP%3F,administrar%20y%20ejecutar%20una%20aplicaci%C3%B3n.>
- <https://dinahosting.com/ayuda/que-es-apache-y-para-que-sirve/>
- <https://www.php.net/manual/es/intro-what-is.php>

- <https://platzi.com/blog/que-es-mean-full-stack-javascript/>
- <https://www.mongodb.com/es/what-is-mongodb>
- <https://kinsta.com/es/base-de-conocimiento/que-es-express/>
- [https://es.wikipedia.org/wiki/Java\\_EE](https://es.wikipedia.org/wiki/Java_EE)
- <https://sede.educacion.gob.es/publiventa/PdfServlet?pdf=VP15838.pdf&area=E>

Repositorio GitHub:

<https://github.com/javiidoce/EntornoServidor>