

ARQUITECTURAS WEB

Trabajo sobre descripción y análisis de
arquitecturas web

Joaquín Vera Vallés
2º DAW Distancia

Vera Valles, Joaquin
[Email address]

Contents

1. Arquitecturas de Aplicaciones Web	2
2. Modelos de arquitecturas Web	3
2.1 Modelo de dos capas.....	3
2.2 Modelo de tres capas	4
2.3 Modelo de 4 capas.	5
3. Arquitectura Web LAMP y WAMP	6
3.1 Apache.....	6
3.2 MySQL.....	7
3.3 MariaDB.....	7
3.4 PHP	7
4. Arquitectura Web MEAN.....	8
4.1 MongoDB.....	8
4.2 Express.....	8
4.3 Angular	9
4.4 Node.JS	9
5. Lamp VS Mean	10
5.1 LAMP	10
5.2 MEAN.....	10
5.3 Conclusión.....	11
6. FRAMEWORK	12
6.1 Introducción.....	12
6.1.1Pasos del desarrollo WEB	12
6.2 ¿Qué es Spring Boot?.....	13
6.3 ¿Qué ofrece Spring Boot?.....	14
6.4 Motivo de la elección (personal).....	14

1. Arquitecturas de Aplicaciones Web

Una aplicación Web es proporcionada por un servidor Web y la utilizan los Usuarios (Clientes) que se conectan desde cualquier punto vía clientes Web, como, por ejemplo, los navegadores.

La arquitectura de un Sitio Web tiene tres componentes principales:

- Un Servidor Web
- Una conexión de red
- Uno o más clientes.

El servidor Web procesa la solicitud HTTP / HTTPS y distribuye la información de la página a los clientes que la solicitan. Los requerimientos son hechos a través de una conexión de red.

Al ser un servicio Distribuido, el cliente hace una petición con el protocolo HTTP, la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y envía la página Web al cliente que ha hecho la solicitud.

El cliente es el que se encarga de interpretar la respuesta y un intérprete (navegador, por ejemplo) es el que se encarga de que el código sea visual y funcional.

Una aplicación web distribuida que está basada en el modelo Cliente / Servidor que gestionan servidores web.

Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe, es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente). Cuando el browser ejecuta un script en el cliente, éste no tiene acceso directo a los recursos del servidor.

Las colecciones de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario. El acceso a las páginas está agrupado también en el tiempo (sesión). Los componentes de una aplicación Web son:

1. Lógica de negocio.
 - Parte más importante de la aplicación.
 - Define los procesos que involucran a la aplicación.
 - Conjunto de operaciones requeridas para proveer el servicio.
2. Administración de los datos.
 - Manipulación de BD y archivos.
3. Interfaz
 - Los usuarios acceden a través de navegadores, móviles, PDAs, etc.
 - Funcionalidad accesible a través del navegador.
 - Limitada y dirigida por la aplicación.

2. Modelos de arquitecturas Web

Las aplicaciones web se modelan mediante lo que se conoce como modelo de capas, Una capa representa un elemento que procesa o trata información. Los tipos son:

2.1 Modelo de dos capas

Es uno de los modelos más fáciles de implementar.

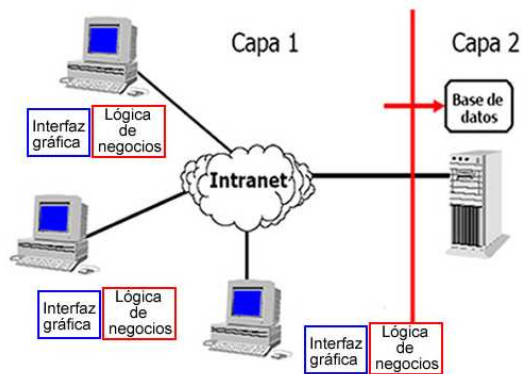
La mayor parte de la lógica de este modelo de arquitectura corre en la parte del cliente (fat client).

Las 2 capas son:

- Cliente: La lógica de negocio está inmersa dentro de la aplicación. Esta aplicación se realiza en la interfaz de cliente, es decir que la lógica de negocio corre parte del cliente.
- Servidor: Es la parte que gestiona los datos y que los contiene. Suele ser una base de datos normal y corriente.

Este modelo de aplicaciones Web tiene una serie de desventajas:

- Es difícilmente escalable
- Número de conexiones reducida
- Alta carga de la red.
- La flexibilidad es restringida
- La funcionalidad es limitada.



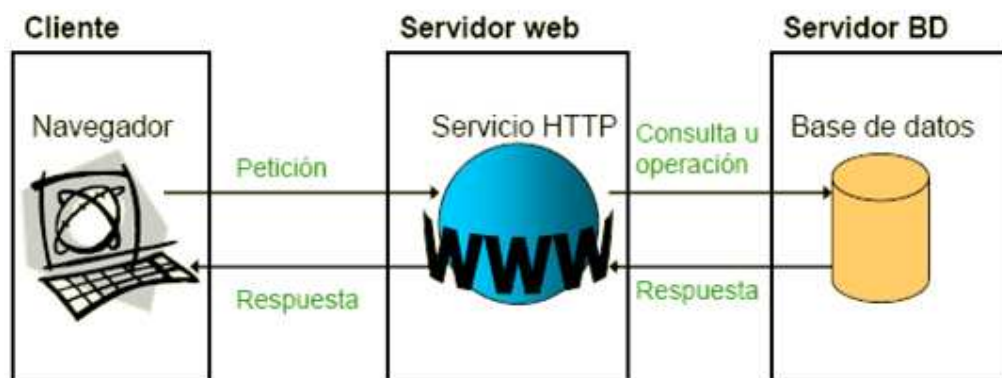
2.2 Modelo de tres capas

Es una arquitectura diseñada para superar las dificultades de las arquitecturas de 2 capas, es decir, que es mucho más potente que las arquitecturas de 2 capas.

La diferencia en estos modelos es que no cae tanta funcionalidad en el cliente y se crea una capa intermedia (Capa de servidor) entre la capa de cliente y la capa de datos.

Ahora, el modelo 3 capas, las capas quedarían con las siguientes funcionalidades:

1. Capa de presentación (parte en el cliente y parte en el servidor)
 - Recoge la información del usuario y la envía al servidor (cliente)
 - Manda información a la capa de proceso para su procesamiento
 - Recibe los resultados de la capa de proceso
 - Generan la presentación
 - Visualizan la presentación al usuario (cliente)
2. Capa de proceso (servidor web)
 - Recibe la entrada de datos de la capa de presentación
 - Interactúa con la capa de datos para realizar operaciones
 - Manda los resultados procesados a la capa de presentación
3. Capa de datos (servidor de datos)
 - Almacena los datos
 - Recupera datos
 - Mantiene los datos
 - asegura la integridad de los datos



2.3 Modelo de 4 capas.

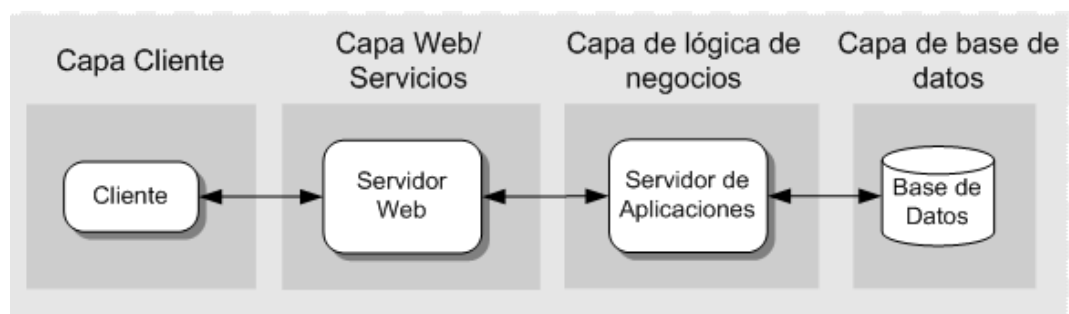
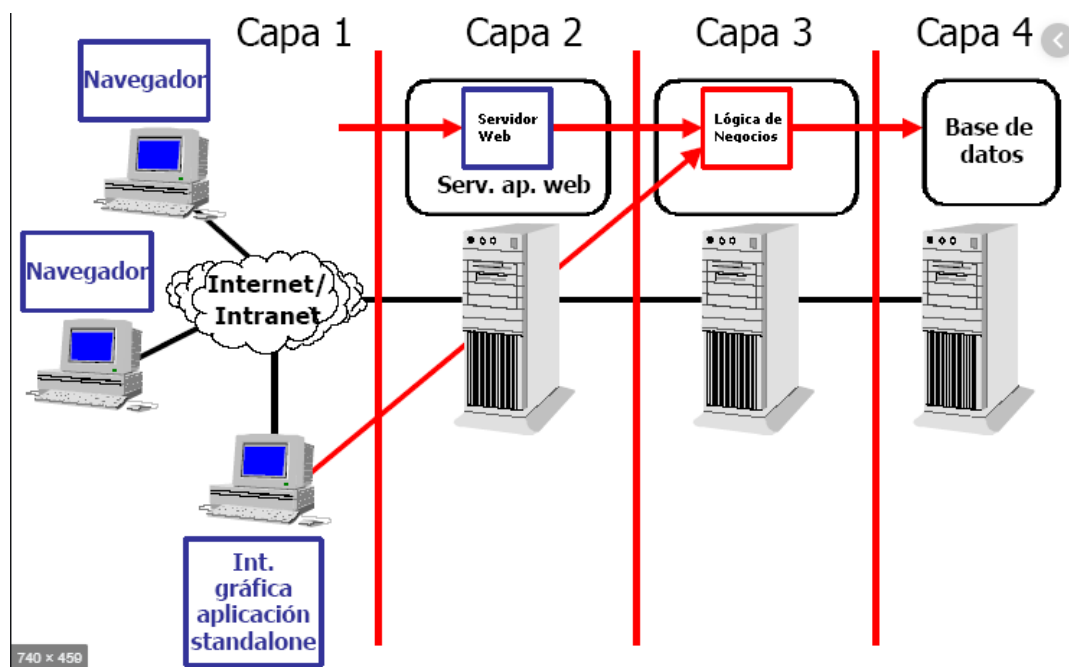
Es una arquitectura que solo es utilizada en las empresas más grandes. Es el modelo más difícil y costoso de implementar pero es el mejor preparado y es el más escalable y funcional.

Ahora, en la capa de cliente solo se queda la interfaz gráfica y la ejecución de códigos en cliente (Visualizaciones, etc...)

La capa de Servidor ahora se divide en 2 partes y se crea la capa de lógica de negocios, anteriormente gestionada por el servidor.

Los clientes pueden llamar a la parte del servidor Web cuando necesite una interfaz o a la capa de lógica de negocios cuando intente algo de la lógica de negocios.

El servidor también puede llamar a la capa de negocios.



3. Arquitectura Web LAMP y WAMP

El sistema LAMP es un conjunto de sistemas, lenguajes o tecnologías que se utilizan habitualmente, de manera conjunta, en el desarrollo de aplicaciones web.

LAMP hace referencia a Linux, Apache, MySQL y PHP/Perl/Python.

LAMP es un grupo de software de código libre que se instala normalmente en conjunto para habilitar un servidor para alojar sitios y aplicaciones web dinámicas.

Es un acrónimo que representa un sistema operativo Linux con un servidor web Apache. Los datos son almacenados en una base de datos MySQL o MariaDB (NoSQL) y el contenido dinámico es procesado con PHP.

Hay variaciones sobre este conjunto de aplicaciones para este tipo de sistema, que son WAMP y MAMP.

WAMP corre sobre un servidor Windows y Mamp sobre un servidor Mac Os.

3.1 Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh...

Es desarrollado y mantenido por una comunidad de usuarios dentro del proyecto HTTP Server (httpd).



Developer	March 2017	Percent	April 2017	Percent	Change
Apache	79,942,445	45.82%	78,489,472	46.28%	0.46
nginx	34,317,972	19.67%	33,176,490	19.56%	-0.11
Microsoft	15,611,256	8.95%	14,033,779	8.28%	-0.67
Google	11,684,677	6.70%	12,048,089	7.10%	0.41

3.2 MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

3.3 MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License).

Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, APIs y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

3.4 PHP

PHP es un acrónimo recursivo que significa PHP Hypertext Preprocessor (inicialmente PHP Tools, o, Personal Home Page Tools).

PHP puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores.

Permite la conexión a diferentes tipos de servidores de bases de datos tanto SQL como NoSQL tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite o MongoDB.

4. Arquitectura Web MEAN

El desarrollo web utilizando stack MEAN es cada vez más utilizado. Es un sistema de desarrollo web basado en JavaScript.

El lenguaje JavaScript es un lenguaje cada vez más utilizado. Esto quiere decir que ya no solo funciona en la capa de cliente sino que se ha llevado a todas las capas de desarrollo, incluidos el servidor.

A este concepto y las tecnologías que lo posibilitan se les ha bautizado con el nombre de MEAN, acrónimo formado por las iniciales de las cuatro tecnologías principales que entran en juego: MongoDB, Express, Angular y Node.js.

4.1 MongoDB

Actualmente, los tipos de información que suelen requerir las aplicaciones web demandan mayor flexibilidad, menos coherencia y sobre todo mayor capacidad de escalar. Para dar respuesta a todo esto surge la tendencia tecnológica en almacenes de datos que se denomina NoSQL.

Estos almacenes de datos NoSQL pueden ser de diversos tipos, pero en muchos casos utilizan JavaScript para representar la información, recibiendo, enviando y almacenando datos usando la notación JSON (ver más abajo).



4.2 Express

Node.js por sí mismo está muy bien para crear la lógica de las aplicaciones, y como dispone de un módulo para el protocolo HTTP es posible crear aplicaciones web con lo que trae por defecto. Sin embargo, es complejo, costoso y tenemos que hacer todo el trabajo a bajo nivel nosotros mismos.

Para ayudar a crear aplicaciones web más fácilmente nació Express. Este Framework está escrito en JavaScript para Node.js. Su objetivo es que no tengamos que reinventar la rueda cada vez que queramos crear una aplicación web, ofreciéndonos soporte para las principales necesidades en este tipo de aplicaciones: gestión de peticiones y respuestas, cabeceras, rutas, vistas...



4.3 Angular

En el navegador desde siempre han existido bibliotecas de funciones que nos facilitan mucho la vida. La más conocida y utilizada es sin duda jQuery, que también fue una de las primeras en ser adoptada universalmente. En los últimos años se ha trasladado al navegador el patrón de diseño denominado MVC (Modelo-Vista-Controlador) y han surgido cientos de bibliotecas especializadas en facilitarnos su uso. De entre todas ellas destaca Angular que ha tomado un especial protagonismo en los últimos tiempos, entre otras muchas cosas porque está creada y soportada por Google, es gratuita y de código abierto.



4.4 Node.JS

Node.js utiliza por debajo el motor de JavaScript de Google, denominado V8, y provee de una arquitectura orientada a eventos (como la de los navegadores) así como una serie de APIs no-bloqueantes (asíncronas) que le proporcionan un rendimiento y una escalabilidad muy elevadas. Se puede utilizar para crear cualquier tipo de lógica de aplicación, pero dado que incorpora un módulo para poder actuar como un servidor web, es especialmente popular para crear aplicaciones web.



5. Lamp VS Mean

5.1 LAMP

PROS

- Es una tecnología madura
- Está bien documentada
- Funciona bien en apps web de tamaño medio y grande donde la gestión de los datos en tiempo real no es vital
- Tiene una comunidad muy grande
- PHP es el lenguaje de servidor más utilizado del mundo, lo usan WordPress, Drupal, Prestashop, Magento, Moodle y un largo, largo etcétera

CONTRAS

- No funciona bien en aplicaciones de tiempo real que manejan grandes cantidades de datos de forma concurrente
- No está de moda
- Es más lento que MEAN
- Emplea varios lenguajes de programación

5.2 MEAN

PROS

- Lo utilizan las startups más punteras
- Es la mejor solución para programar aplicaciones de tiempo real que manejan grandes cantidades de datos de forma concurrente
- Es más rápido que LAMP
- Emplea un solo lenguaje para todo: JavaScript

Está de moda

CONTRAS

- La orientación a eventos no es siempre el mejor marco para desarrollar aplicaciones
- MEAN está todavía en fase de investigación y desarrollo, lo que lo convierte en un entorno extremadamente volátil
- Cada dos por tres salen nuevas librerías que no siempre funcionan bien
- La documentación no es tan buena como en LAMP
- Muchas empresas pequeñas y medianas no pueden permitirse el riesgo de utilizar tecnologías tan cambiantes

5.3 Conclusión

Si es para un aprendizaje, ambas son recomendadas, pero si se quiere crear una página web sin demasiada inversión, lo recomendable es la estructura LAMP.

No hay una opción correcta, además se puede usar partes de ambas tecnologías.

Dependiendo del tipo de entorno y de las necesidades, se puede utilizar una u otra.

6. FRAMEWORK

6.1 Introducción

Para la elaboración de un análisis de un Framework de desarrollo web se ha elegido Spring Framework, especialmente, el bloque Spring Boot.

Spring es una aplicación normalmente utilizada en proyectos Java, ya que incorpora librerías muy bien desarrolladas y fácilmente utilizables.

El desarrollo Web con Spring Boot Framework se divide en distintos pasos.

6.1.1 Pasos del desarrollo WEB

Lo primero de todo hay que crear el proyecto Maven. Maven es una herramienta de software para la gestión de proyectos Java, cuya funcionalidad principal es la de importar librerías basado en un formato XML. Su desarrollador es Apache.

Después, hay que importar y seleccionar las librerías en el proyecto Maven,
<https://mvnrepository.com/artifact/org.springframework/spring-core/5.2.0.RELEASE>

Después hay que crear y desarrollar la aplicación web

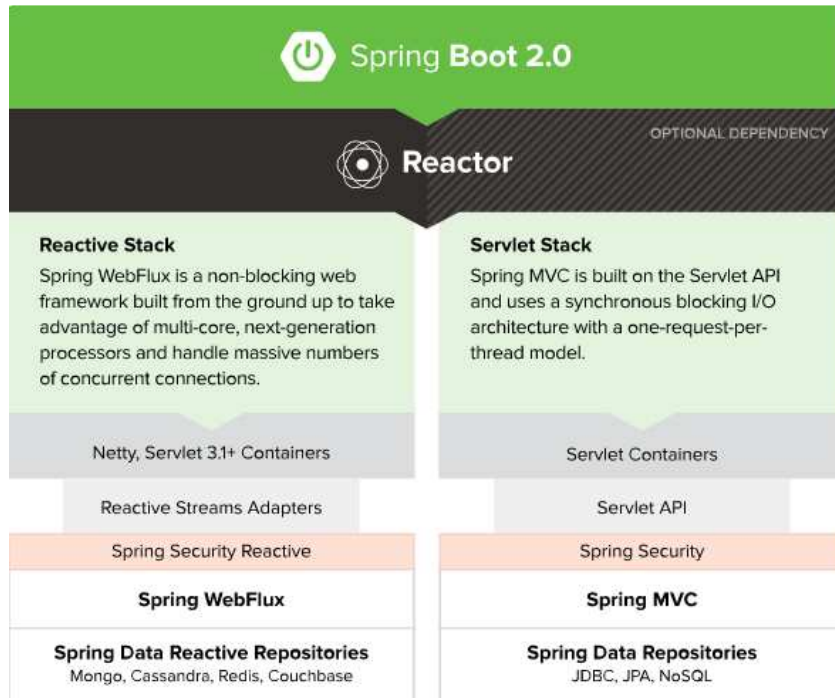
Y por último hay que desplegar la aplicación web en el servidor.

Un ejemplo de funcionamiento puede ser encontrado en el siguiente enlace:

<https://www.arquitecturajava.com/que-es-spring-boot/>

6.2 ¿Qué es Spring Boot?

Según la página web oficial de Spring, Spring Boot es un proyecto que está diseñado para ponerlo en funcionamiento lo más rápido posible, con una configuración inicial mínima de Spring. Spring Boot tiene una visión obstinada de crear aplicaciones listas para la producción.



Spring Boot es el punto de partida para construir todas las aplicaciones basadas en Spring.

Spring Boot está diseñado para ponerlo en funcionamiento lo más rápido posible, con una configuración inicial mínima de Spring.

6.3 ¿Qué ofrece Spring Boot?

Las funcionalidades más importantes que ofrece Spring Boot son las siguientes:

- Se puede crear cualquier entorno web: REST API, WebSocket, web, streaming, tasks y más.
- Soporte para SQL y NoSQL.
- Soporte de tiempo de ejecución incorporado: Tomcat, Jetty y Undertow.
- Herramientas de productividad para desarrolladores como LiveReload y Auto Restart.
- Funciones listas para entornos de producción, como tracking, gráficas de rendimiento y seguimiento del estado del servidor.
- Funciona en su Muchos IDEs: Spring Tool Suite, IntelliJ IDEA, eclipse y NetBeans.

6.4 Motivo de la elección (personal)

En el proyecto de fin de grado del DAM, desarrollé una aplicación con un modelo 3 capas, Cliente – Servidor – Datos.

Para la parte del servidor, quería desarrollar una REST API y centralizar todo el código en Java. Empecé desarrollando la REST API con Spring Boot, aunque acabé haciéndolo con ASP.NET y Entity Framework Core.

Por eso esta ha sido la elección del Framework a describir.