# Capítulo 6 – Stack MEAN

En éste capítulo, se analizará el stack MEAN y sus componentes. El mismo está compuesto por un conjunto de tecnologías que responden al siguiente acrónimo: MongoDB (acceso a datos), Express (framework web, *back end*), Angular (framework web, *front end*) y NodeJS (plataforma de aplicación web). Además, otros complementos, como Compodoc (documentador), bibliotecas y framework aplicados a la vista (o *front end*) como Bootstrap y JQuery.

## **6.1 ¿Qué es MEAN?**

“Se denomina MEAN, o MEAN *stack*, a un conjunto de capas de software para el desarrollo de aplicaciones, dónde la característica predominante es el uso del lenguaje de programación popularizado como JavaScript”. Más adelante, se visualiza el logotipo de este stack de tecnologías (Ilustración 31).



Ilustración 31 - Acrónimo MEAN

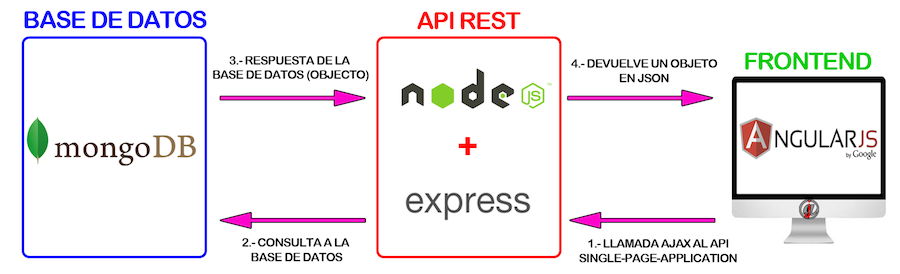


Ilustración 33 - Arquitectura de interacción MEAN

## **6.2 Componentes de MEAN**

Como podemos apreciar en la imagen (Ilustración 33 – Arquitectura de interacción MEAN) se puede observar que componentes interactúan entre si dentro del del *stack*. En las secciones 6.2 se ampliará cada tecnología.

En un primer momento la aplicación Angular se encuentra almacenada en un servidor WEB. Cuando es transferida al cliente (navegador web) comienza la comunicación entre el componente Angular y el servidor NodeJS, por medio de una API REST implementada sobre el framework Express. Los requerimientos producidos por los *endpoints* de la API provocan que NodeJS realice las consultas tanto de lectura como escritura sobre MongoDB. El formato de presentación de datos es JSON (AGREGAR REFERENCIA A 6.3.3 JSON) para todos los requerimientos.

### 6.2.1 MongoDB

Es un sistema de base de datos NoSQL, en el cual la información se almacena documentos en vez de filas en una tabla. Cada documento se trata de una estructura con formato JSON (Notación simple de objeto tipo JavaScript). Estos documentos son agrupados en colecciones en contraposición a las tablas de un RDBMS. Debido a la ausencia de comprobación de integridad referencial tiene un alto desempeño.

### 6.2.2 Express

Es un paquete de NodeJS que ofrece una interface mínima para manejo de solicitudes o peticiones HTTP. Uno de sus componentes principales se trata de un sistema de enrutamiento (Routing), que asocia URLs con funciones. Dentro del MEAN *stack* opera del lado del servidor, también conocido como *back-end*.

### 6.2.3 Angular

Es un framework orientado a crear aplicaciones web, basado en el sub-lenguaje TypeScript (JavaScript con verificación de tipos de dato *ahead of time*), mantenido por Google, enfocado en aplicaciones web de una sola página o SPA. Su objetivo es proponer un diseño de las aplicaciones basadas en navegador con basado en el patrón Modelo Vista Controlador (MVC), y facilitando el desarrollo y las pruebas.

### 6.2.4 NodeJS

En MEAN, NodeJS, es la plataforma encargada del funcionamiento del servidor. Se trata de un intérprete de JavaScript multiplataforma enfocado en la programación del lado del servior.

Utiliza el motor de ejecución de JavaScript de Google, denominado V8, y presenta una arquitectura orientada a eventos, en conjunto con una serie de APIs no-bloqueantes (asíncronas[[1]](#footnote-1)) que le proporcionan un rendimiento y una escalabilidad muy elevadas. Esta característica se debe a una librería en C llamada LibUV (Unicornio Velocirraptors), que proporciona soporte de E/S asíncronas basada en bucles de eventos.

Ilustración 2 - Logo del motor V8

Si bien NodeJS se puede utilizar para crear cualquier tipo de aplicación, dado a que incorpora un módulo de servidor web dentro de su biblioteca estándar, es especialmente popular para crear aplicaciones web, lo cual lo ha popularizado entre empresas que se dedican a servicios basados en Internet.

Su uso no se encuentra limitado a web, sino que también existen aplicaciones de línea de comandos, scripts para administración de sistemas, aplicaciones de red, etc. Su utilización es recomendada en aplicaciones cocurrentes por I/O como: chats, APIs REST, entrada de datos concurrentes, aplicacciones cuya interacción sea con servicios bloqueantes como escritura en RDBMS, procesamiento de archivos, transmisión de datos, proxies, aplicaciones como corredores de bolsa (tiempo real), visualización de interacciones, etc.

La principal razón de su utilización en la construcción y escalablamiento de aplicaciones de red, es su capacidad de afrontar la concurrencia mediante el procesamiento de eventos de manera no bloqueante (también conocido como, *event-driven I/O*). Podemos apreciar en la comparativa entre los servidores tradicionales y NodeJS en la Ilustración 40 Comparativa de servidores tradicionales y Nodejs).

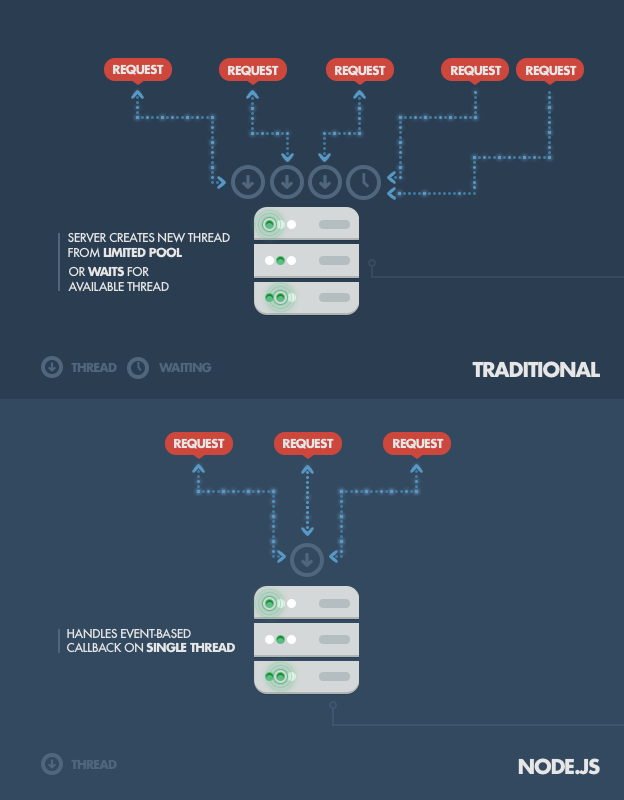


Ilustración Comparativa de servidores tradicionales y NodeJS

## **6.3 Otros complementos**

### 6.3.1 Twitter Bootstrap

Se trata de uno de los Frameworks más populares que integra HTML, CSS, y JS para el desarrollo de aplicaciones web del lado del cliente adaptables, es decir, que su presentación aproveche los diferentes medios de reproducción (Responsive).

Dentro de las ventajas que presenta este framework son:

* Facilita un sistema de maquetado por columnas.
* Cuenta con el soporte de una amplia comunidad.
* Admite la reconfiguración y recompilación mediante lenguajes como LESS (GLOSARIO)

### 6.3.2 Compodoc

Se trata de un generador de cocumentación, compatible con todas las definiciones de API de Angular. Genera contenido estático, “responsivas” y provee sistema de búsqueda basado en **lunr.js** para indexar los componentes, módulos, servicios y modelos.

### 6.3.3 JSON

Es el acrónimo de JavaScript Simple Object Notation. Se trata de un mecanismo de *marshaling*, que permite transmitir en formato de cadenas de texto objetos (o estructuras complejas) que pueden ser luego des-*marshalizadas* para recuperar los objetos originales.

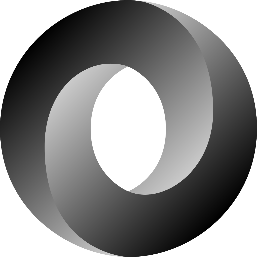
Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función JSON.parse(), lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

Ilustración - Logo de JSON

Podemos decir que en MEAN, JSON es el formato de transferencia de datos entre todas las capas: navegador, servidor web y servidor de datos.

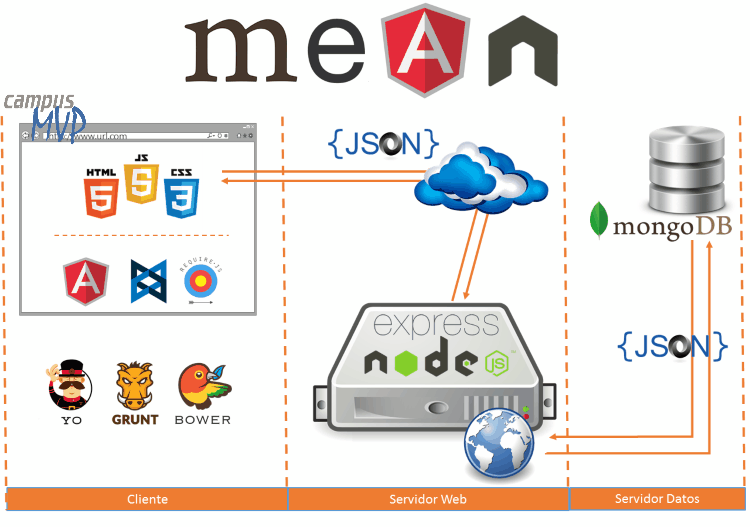


Ilustración 35 - Json pegamento de tecnologías

### 6.3.3 JQuery

Es una biblioteca multiplataforma de JavaScript del lado del cliente, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción mediante la simplificación de la utilización de AJAX.

### Resumen

En este capítulo, vimos el concepto de MEAN y sus componentes. El mismo está compuesto por un conjunto de tecnologías respetando el acrónimo como sigue: MongoDB, Express, Angular y NodeJS y que todas ellas se comunican mediante el formato JSON.

Finalmente, analizamos distintas herramientas complementarias como Compodoc (documentador), Frameworks y bibliotecas orientados a la vista como Bootstrap y JQuery.

1. https://en.wikipedia.org/wiki/Libuv [↑](#footnote-ref-1)