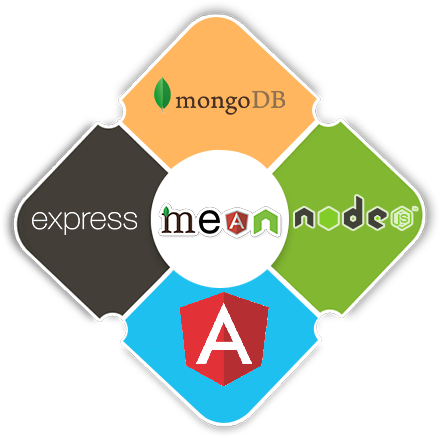
****

**[Por decidir]**

Miembros

**Pérez Llorente, Ángel** -[angperllo](mailto:serortagu@alum.us.es)@alum.us.es

**Valero García, Luis** -l[uivalgar@alum.us.es](mailto:luivalgar@alum.us.es)

Contenido

[Introducción y dominio 3](#_Toc6240851)

[Tecnologías y justificación 4](#_Toc6240852)

[Instalación 4](#_Toc6240853)

[NoSQL vs SQL 4](#_Toc6240854)

[Planificación 5](#_Toc6240855)

[Implementación 6](#_Toc6240856)

[Conclusiones 7](#_Toc6240857)

[Bibliografía y referencias 8](#_Toc6240858)

# Introducción y dominio

En esta memoria se analizarán el paquete de desarrollo MEAN, haciendo especial hincapié en la base de datos y la conexión de la aplicación con ella. Además, se dará una comparativa de los gestores de base de datos relacionales conocidos hasta la actualidad y el no relacional utilizado para este proyecto.

Para ello, se realizará un pequeño proyecto de microservicios característico del paquete de tecnologías seleccionado, más concretamente el proyecto se basará en una API cuyo modelo (NoSQL) consistirá en un usuario que podrá iniciar sesión en el sistema y poseer una librería que contenga su música, película o videojuegos favoritos.

Dichas librerías serán públicas y los demás usuarios podrán filtrarlas por sus distintos atributos como nombre, autor, tipo o calificación, posteriormente tendrán la posibilidad de añadir cualquiera de los elementos buscados a su propia librería. La API será consumida por Angular, quien dará soporte al frontend de nuestra aplicación.

# Tecnologías y justificación

En esta sección, se enumerarán las tecnologías seleccionadas junto a una breve descripción, posteriormente, se explicará detalladamente como se pueden instalar estas herramientas y comenzar un proyecto para ello.

1. **Ubuntu 18.04:** Sistema operativo sobre el que se ha desarrollado el proyecto, se ha elegido debido a su facilidad para instalar herramientas de desarrollo y para virtualizar el desarrollo, reduciendo los problemas de configuración por la instalación en diferentes clientes.
2. **Node.js:** Es el encargado de la ejecución de código javascript en la parte del servidor o API. Cabe destacar, que utiliza ejecución asíncrona y no bloqueante, pudiendo no bloquear el código hasta que se recibe respuestas como otras tecnologías típicas tales como PHP.
3. **Express:** Módulo de Node para facilitar el uso de la creación de una API en Node, utilizando el protocolo HTTP.
4. **Angular 7:** Framework de desarrollo frontend, es decir, de la parte de cliente de la aplicación web, a diferencia de los dos anteriores utiliza typescript en lugar de javascript, cuya diferencia principal es que se puede añadir tipado explícito a los objetos. Lo más característico de angular es que, una vez compilado, desarrolla la web en una sola página (SPA o single page application), modificándola a través de código javascript.
5. **MongoDB:** Gestor de base de datos no relacional y open source, basados en documentos en formato json, los cuales, pueden tener diferentes atributos.
6. **Mongoose:** Módulo de Node utilizado como conector de MongoDB, elegido por su similitud con las queries nativas y su facilidad de integración con la tecnología utilizada. Aporta, además, un gran middleware para la validación de objetos antes de la escritura en base de datos.
7. **MongoDB Compass:** Aplicación de escritorio de mongodb para poder ver los datos o indexar la base de datos más fácilmente.
8. **Git:** Utilizado como control de versiones junto a github, en concreto: <https://github.com/luisval11/cbd-project>
9. **Webstorm:** Entorno de desarrollo para javascript de JetBrains. Seleccionado por su cómoda instalación en Ubuntu, herramienta de debug e integración con repositorios de git.

En cuanto la justificación de la pila de trabajo MEAN (Mongo, Express, Angular y Node), se han elegido dichas tecnologías por su gran comunicación entre ellas, usando la mayoría javascript o habiendo sido desarrolladas con javascript. Todas usan el formato json para el traspaso de objetos.

Además, dada la asincronía de las herramientas y la programación orientada a eventos se pueden realizar muchas acciones en tiempo real sin necesidad de recargar la página o dar largas esperas al usuario y que pueda llegar a frustrarse.

## Instalación

## NoSQL vs SQL

# Planificación

# Implementación

[Hablar de como está montada la api y como funciona la app web]

# Conclusiones

# Bibliografía y referencias

1. <https://www.w3schools.com/nodejs/>
2. <https://nodejs.org/en/>
3. <https://expressjs.com/>
4. <https://angular.io/features>
5. <https://mongoosejs.com/>
6. <https://www.mongodb.com/products/compass>
7. <https://www.jetbrains.com/webstorm/>