

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

DESARROLLO DE APLICACIONES WEB “JEOPARDY”

ARQUITECTURA: MEAN STACK.

ISEL TULA	A01410732
JENS WÖHRLE	A01670506
PATRICK WIJERAMA	A01670604

Tabla de Contenidos

1. Introducción	3
1.1 Propósito	3
1.2 Alcance	3
1.3 Referencias	3
1.4 Descripción	3
2. Requerimientos de aplicación web	4
2.1 Game-Master	4
2.2 Nuevo juego	4
2.3 Juego	5
3. Arquitectura	5
4. Diseño de la base de datos	Error! Bookmark not defined.

Documento de Arquitectura de Aplicación Web.

1. Introducción

1.1. Propósito

El siguiente documento presenta una visión general de la arquitectura de la aplicación Web de Jeopardy, siendo representado a través de una serie de diferentes vistas arquitectónicas que involucran diferentes aspectos de la Aplicación Web.

1.1. Alcance

En este documento se describe las vistas de Despliegue y de Implementación de la arquitectura de la aplicación que se desarrollará.

1.2. Referencias

Se tomaron por referencia los siguientes documentos de la aplicación Web:

- Angularjs Web Page. Recuperado de <https://angularjs.org/>
- Mean Stack Tutorial. Recuperado de <https://thinkster.io/angulartutorial/mean-stack-tutorial/>.
- Mean.js Web Page. Recuperado de <http://meanjs.herokuapp.com/#/>
- MongoDB Web Page. Recuperado de <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>
- Node.js Web Page. Recuperado de <http://nodejs.org/#download>.

1.3. Descripción

Este documento consta de ocho secciones las cuales detallan las vistas del sistema en base a las metas y restricciones de la arquitectura. Además se muestran las siguientes vistas: vista lógica, vista de física y vista de componentes y capas. Cada una de ellas incluye una breve descripción del contenido.

2. Requerimientos de la Aplicación Web.

La aplicación constará de tres partes fundamentales: el Game-Master que dirigirá el juego, la modificación y creación del juego y el Juego en sí.

2.1. Game-Master

- 2.1.1. Ingresar como usuario si ya tiene cuenta.
- 2.1.2. Registrarse como nuevo usuario.
- 2.1.3. La contraseña será enviada por medio de un correo electrónico.
- 2.1.4. Si escribes la contraseña incorrecta más de 3 veces el usuario es bloqueado.
- 2.1.5. Puede modificar su perfil.
- 2.1.6. Acceder a estadísticas de juegos anteriores.
- 2.1.7. Acceder a juegos no terminados.
- 2.1.8. Acceder a nuevo juego.

2.2. Nuevo juego

- 2.2.1. Seleccionar los temas que deseas en el juego.
 - 2.2.1.1. Puedes seleccionar Temas que ya están en la base de datos, modificarlos o crear nuevos.
- 2.2.2. Seleccionar las categorías.
 - 2.2.2.1. Puedes seleccionar categorías que ya están en la base de datos, modificarlas o crear nuevas.
- 2.2.3. Seleccionar las pistas que usarás en el juego.
 - 2.2.3.1. Puedes seleccionar pistas que ya están en la base de datos, modificarlas o crear nuevas.
- 2.2.4. Seleccionar las preguntas que usarás en el juego.
 - 2.2.4.1. Puedes seleccionar preguntas que ya están en la base de datos, modificarlas o crear nuevas.
 - 2.2.4.2. Puedes cambiar los valores de cada pregunta.
- 2.2.5. Puedes ver un previo de cómo quedaría el juego a medida que lo vas creando.

- 2.2.6. Iniciar el juego.
- 2.2.7. Seleccionar los jugadores.

2.3. Juego

- 2.3.1. Seleccionar la casilla que se desea responder.
- 2.3.2. Usar pistas si el jugador desea.
- 2.3.3. Adicionar y sustraer puntos.
- 2.3.4. Guardar el juego.
- 2.3.5. Terminar el juego.
- 2.3.6. Acceder a la página principal.

3. Arquitectura.


Para el desarrollo de este proyecto usamos lo que se conoce como MEAN STACK o MEAN.js. Este es el acrónimo que referencia Arquitecturas desarrolladas con MongoDB, Express.js, Angular.js y Node.js. Los cuatro son nuevos productos fuertemente ligados al mundo Javascript.

M

E

A

N



Cada una de estas tecnologías aporta una funcionalidad dentro del Stack:

- MongoDB: Es la base de datos NoSQL que nos permitirá almacenar información.
 - No SQL, Orientada a documentos.
 - Permite consultas avanzadas.
- Node.js: Permite ejecutar Javascript del lado del servidor. Es nuestro nuevo servidor de aplicaciones
 - Basado en el motor V8 de Google
 - Orientado a eventos, asíncrono, todo se ejecuta en un único thread.

Figure 1 MEAN Stack

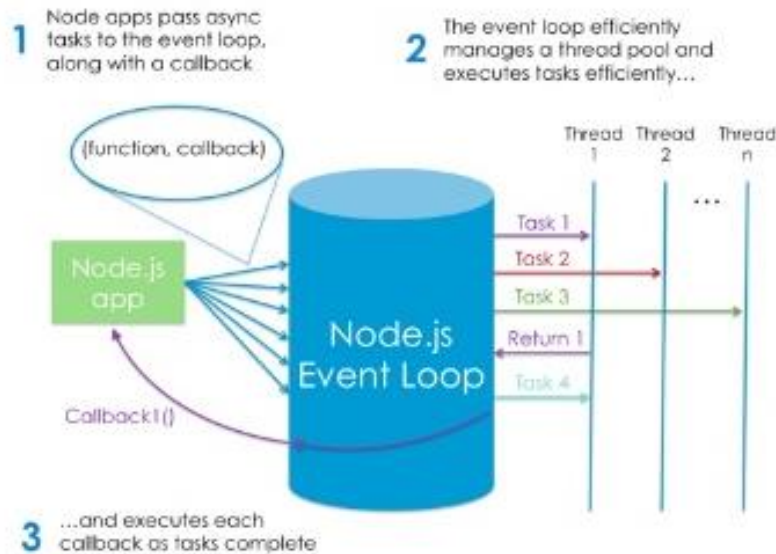


Figure 2 Funcionamiento Node.js

- Express.js : Es el framework JavaScript Web del Servidor
 - Aporta todo lo necesario para crear un servidor web: Middleware, routing, templating, archivos estáticos, cookies, manejo de errores, etc.
 - Inspirado en Sinatra (Ruby).
 - Asíncrono
- Angular.js: Es el framework JavaScript del Cliente.
 - Enfocado al modelo MVC
 - Two-way data binding.

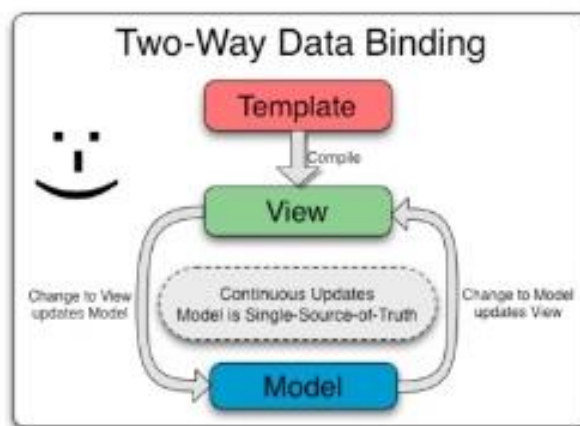


Figure 3 Doble vía Data Binding de Angularjs

Ventajas de usar MEAN:

- Arquitectura Nueva: Es una Arquitectura desarrollada desde cero y eso siempre es positivo ya que permite aprender de errores del pasado y no tener que arrastrar malas decisiones que se tomaron anteriormente.
- Arquitectura Homogénea: En todo el Stack MEAN destaca que prácticamente todas las tecnologías usadas se apoyan en el uso de JavaScript y eso genera una gran homogeneidad en la plataforma. Algo que muchas veces es crítico para tener éxito.

Las cuatro tecnologías unidas nos permiten desarrollar aplicaciones escalables, cercanas al mundo móvil y a la necesidad de tiempo real.

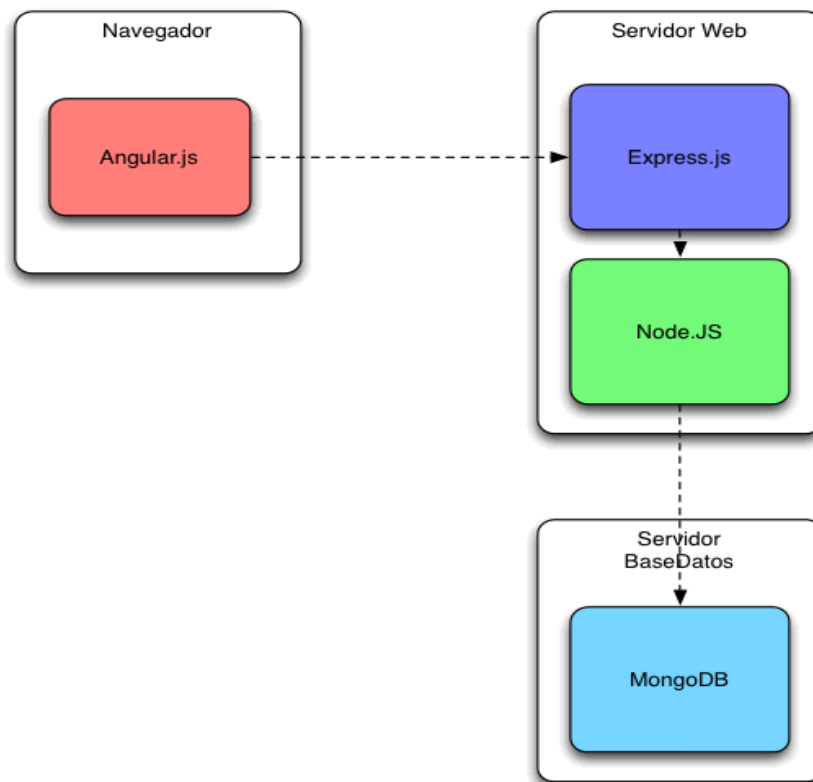


Figure 4 Funcionamiento del modelo MVC con MEAN

La arquitectura está diseñada bajo el patrón MVC (Modelo - Vista - Controlador) lo que permite la interacción entre las peticiones del usuario por medio de la presentación de la información (Vista), donde el Controlador define el comportamiento de la aplicación y el Modelo encargado de guardar la información y de las reglas del negocio que pueden alterar los datos. Logrando de esa manera la interacción del usuario y la aplicación. Los beneficios que nos brinda el patrón MVC es la fácil reutilización de código, descomposición funcional y modularidad permitiendo extensibilidad.

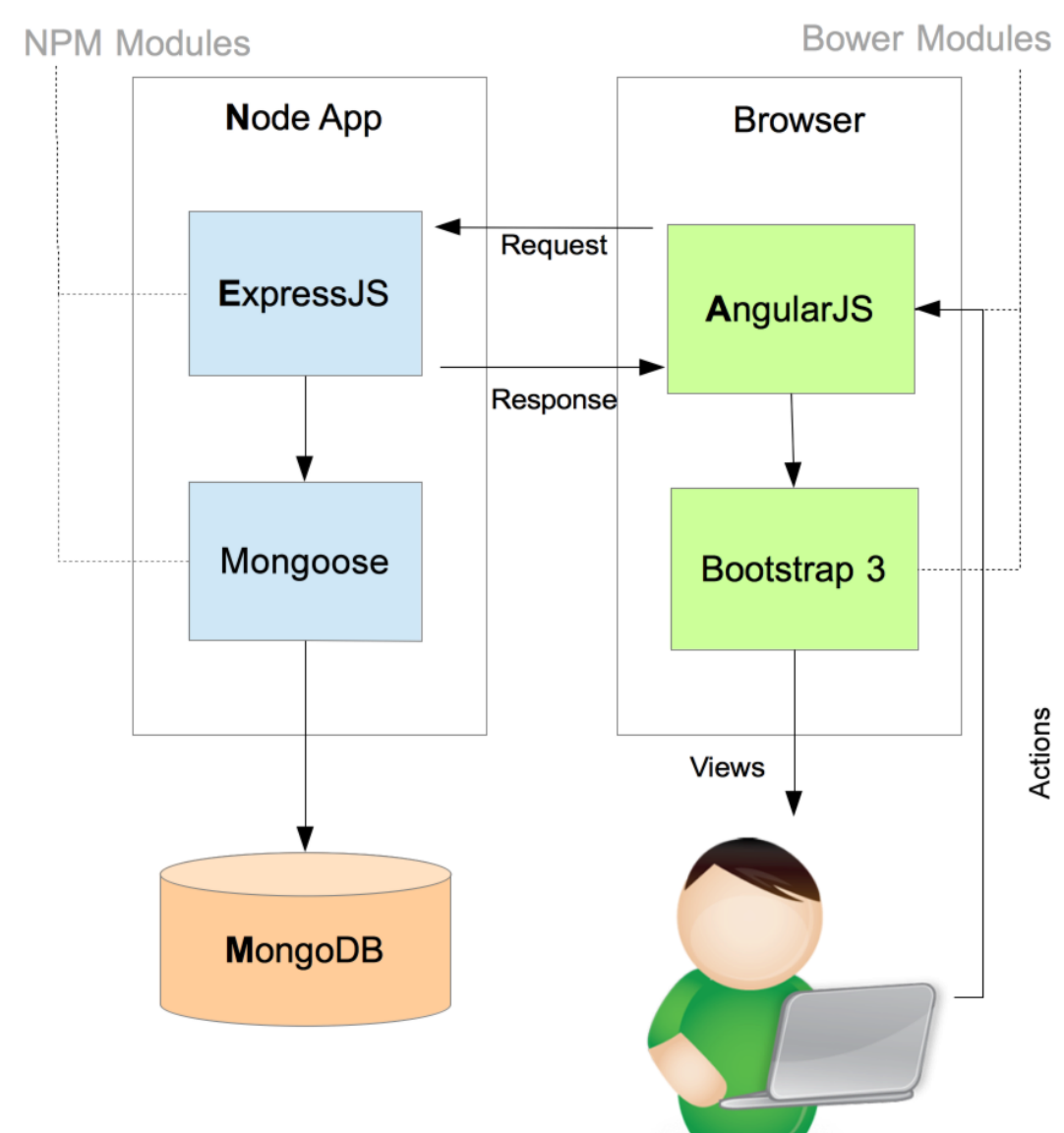


Figure 5 Modelo MVC

3.1. Aplicación.

Nuestra aplicación web está dividida en dos partes fundamentales: el administrador y la aplicación en general.

Administrador: El administrador puede acceder por medio de `http://localhost:3000/admin` para la cuenta puedes registrarte como nuevo usuario. En esta página podrás tener una contabilidad de los usuarios que se inscribieron en tu aplicación. Podrás crear nuevos, modificarlos, eliminarlos y ver la información de cada uno de ellos.

Aplicación: La aplicación es el juego en general. En este podrás crear un nuevo usuario o login con uno nuevo. Dentro podrás ver tu información de usuario, modificar o eliminar tu cuenta. Además puedes Crear un juego nuevo, en el que puedes acceder a diferentes categorías, clases y preguntas. Otra cosa que puedes hacer como usuario es Crear, modificar, eliminar clases, categorías, preguntas, respuestas y el valor de cada una de estas.

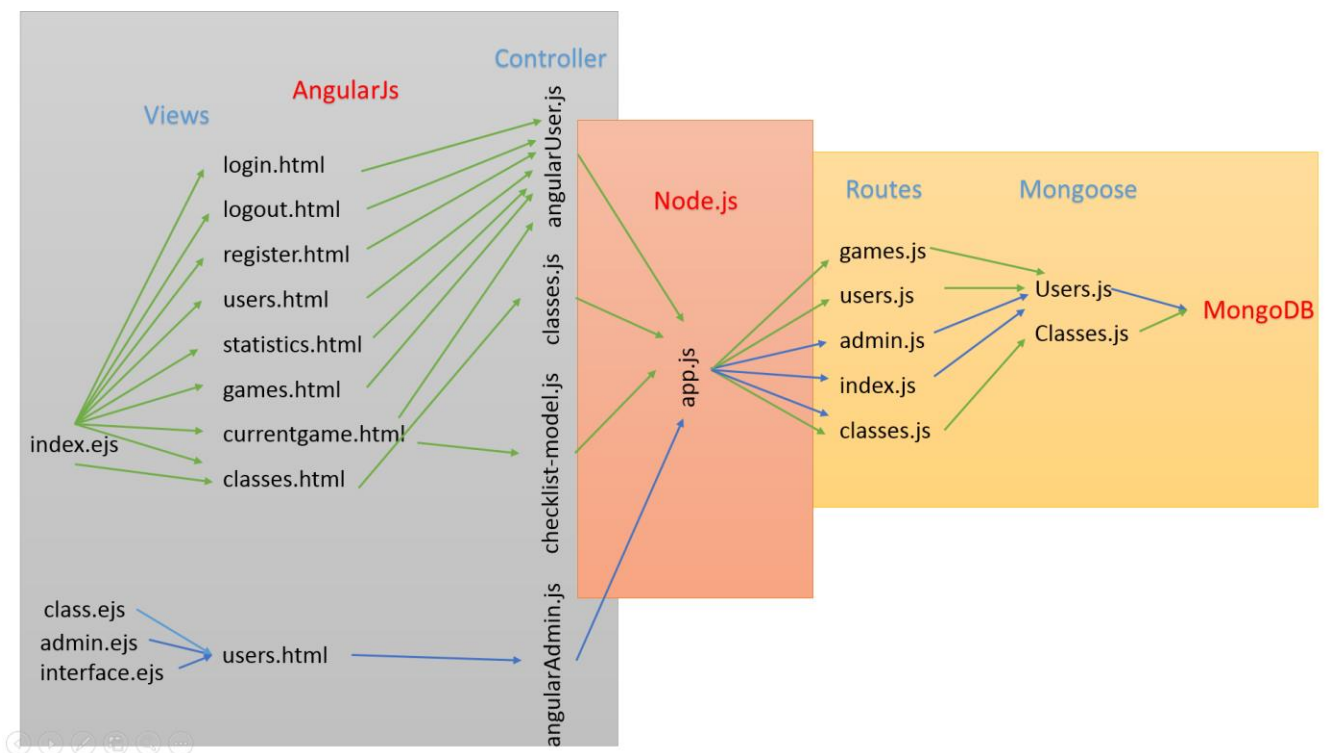


Figure 6 Relación dentro del modelo MVC

4. Diseño de la base de datos.

MongoDB es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

Cualquier campo en un documento de MongoDB puede ser indexado, al igual que es posible hacer índices secundarios. El concepto de índices en MongoDB es similar a los encontrados en base de datos relacionales.

Todos los objetos dentro de la base de datos tienen un número de indentificación conocido como `_id`, con el cual podemos identificar y enlazarlo con otros objetos.

En este proyecto contamos con cuatro documentos.

User: El cual guarda datos específicos del usuario.

Classes: Guarda la información de las clases, con sus categorías, preguntas y respuestas.

Game: Guarda información del juego como los jugadores, el estado, al igual que una conexión con el documento Classes para poder acceder a la información de esta.

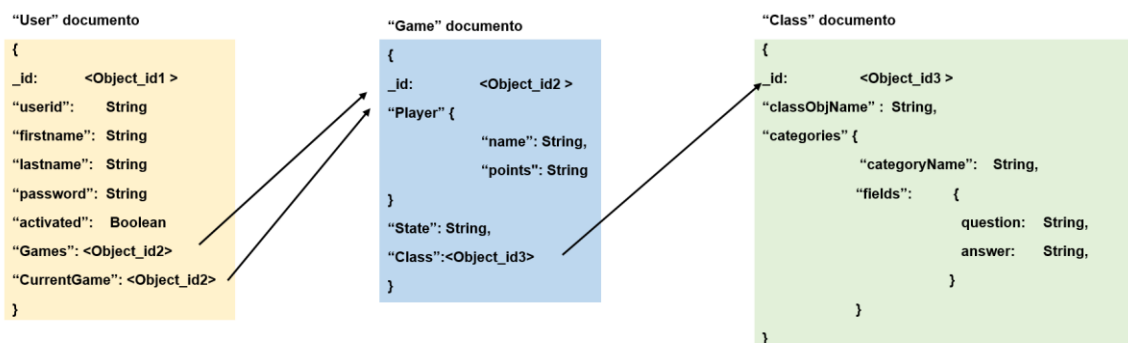


Figure 7 Diseño de la base de datos