

Guía para el Desarrollo de Aplicaciones Web Responsivas Utilizando el Stack MEAN

Héctor Andrade ¹¹ and Victor Reboloso ²¹

¹Department of Computer Science, L^AT_EX University

²Department of Mechanical Engineering, L^AT_EX University

22 de septiembre de 2016

¹handrade@itver.edu.mx

²vrebo.deg@gmail.com

Resumen

El presente documento pretende ser una guía para el desarrollo de aplicaciones web utilizando el stack MEAN (MongoDB, Express, AngularJS, Node.js).

Índice general

1. Desarrollo de Aplicaciones Web Responsivas	2
1.1. Introducción	2
1.2. Patrón Arquitectónico MVC	2
1.3. HTML	3
1.4. CSS	3
1.5. JavaScript	4
1.6. Stack de Desarrollo MEAN	4
1.6.1. MongoDB	5
1.6.2. Express	5
1.6.3. AngularJS	5
1.6.4. Node.js	5
2. Instalación de las Herramientas	6
2.1. Node.js	6
2.1.1. Ejemplos	7
2.2. Express	7
2.3. Bower	7
2.4. Bootstrap	7
2.5. MongoDB	7
2.5.1. Mongoose	8
2.6. AngularJS	8
3. Desarrollo de una aplicación MEAN a través de servicios web REST	9
3.1. Descripción de la aplicación	9
3.2. Estructura de la aplicación	9
3.3. Front-end	9
3.4. Esquema de la Base de Datos	9
3.5. API	9
3.6. Conexión Front-end Back-end	9
4. Seguridad	10
4.1. Seguridad a través de cuentas de usuario	10
4.2. Seguridad a través de Facebook	10

Capítulo 1

Desarrollo de Aplicaciones Web Responsivas

1.1. Introducción

Es indudable que el uso de dispositivos móviles para la navegación en la web ha crecido de manera explosiva recientemente, y que esta tendencia continuará en los próximos años. El desarrollo de aplicaciones web que se adapten a este tipo de dispositivos es prácticamente una obligación para las empresas que desean de alguna forma interactuar con clientes o con el público en general a través de la web.

El diseño web responsivo o adaptivo como algunos autores lo llaman, consiste en diseñar y desarrollar aplicaciones para la web que puedan adaptarse a diferentes tipos de dispositivos desde pequeños dispositivos móviles hasta computadoras con monitores de gran tamaño.

El diseño responsivo no solo trata de ajustar la interface del usuario al tamaño de la pantalla. Algunos aspectos a considerar son también el comportamiento del usuario, la plataforma y tipo del dispositivo móvil e incluso la orientación.

La implementación de un diseño responsivo consiste en el uso adecuado de tecnologías tales como HTML, CSS y marcos de trabajo como Bootstrap y Foundation entre otros.

En esta guía mostraremos los conceptos, patrones arquitectónicos y algunas de las tecnologías y herramientas emergentes más utilizadas en el desarrollo de aplicaciones web responsivas. El primer concepto que veremos es el del patrón arquitectónico Modelo Vista Controlador.

1.2. Patrón Arquitectónico MVC

Un patrón arquitectónico es una descripción abstracta de una arquitectura de software que ha sido utilizada exitosamente y que es producto de buenas prácticas de desarrollo de software. En esencia, el patrón de diseño Modelo Vista Controlador (MVC) consiste en dividir conceptualmente la arquitectura de un sistema y, en la medida de lo posible, la implementación de la misma

en componentes que pertenezcan a una de tres categorías, Modelo, Vista y Controlador.

Dentro de la categoría de Modelo se ubican aquellos componentes encargados de: encapsular los datos utilizados en el sistema, representar el comportamiento de las entidades involucradas en los procesos de negocio del sistema y realizar tareas de persistencia de datos.

La categoría de Vista contiene los componentes con los que el usuario interactuará directamente, comúnmente también se les refiere como el Front-End del sistema. Es importante notar la independencia entre la vista y el modelo. Modificaciones en los componentes que pertenecen a la vista no necesariamente implican modificaciones en el modelo y viceversa.

Por último, los componentes pertenecientes a la categoría Controlador se encargan de atender las peticiones del usuario solicitadas a través de los componentes de la Vista haciendo uso de los componentes del Modelo. Los controladores son el puente entre el Modelo y la Vistas del sistema.

Algunos de los beneficios de desarrollar un sistema bajo el patrón MVC son:

- Simplifica el desarrollo brindando desglosando su arquitectura.
- Separa las distintas responsabilidades (lógica de negocios, interfaz de usuario) de sus componentes.
- Contribuye a su mantenibilidad.

1.3. HTML

El lenguaje de marcado de hipertexto, es un lenguaje basado en etiquetas utilizado para definir la estructura de páginas web y describir su contenido. Las estructuras de documentos HTML siguen una secuencia y una jerarquía, esto significa que un elemento puede seguir después de otro y que un elemento puede contener a otro.

Como todo lenguaje, HTML tiene reglas sintácticas y semánticas las cuales varían dependiendo de la versión en cuestión. La versión más reciente es HTML5.

Para profundizar en el tema se recomienda la siguiente lista de referencias:

- Especificación de HTML5 por la World Wide Web Consortium (W3C) ¹.
- Especificación de HTML5 por la Web Hypertext Application Technology Working Group (WHATWG) ².
- Excelente tutorial y sitio de referencia ³.

1.4. CSS

Las hojas de estilo en cascada (de su traducción del inglés Cascading Style Sheets) son un mecanismo que permite añadir estilo a los documentos web. Su

¹www.w3.org/TR/html5/

²html.spec.whatwg.org/multipage/

³www.w3schools.com/html/

uso delega la responsabilidad de definir la presentación de los documentos a los archivos CSS y se evita la práctica poco recomendable de añadir estilos mediante los elementos del documento.

CSS es un lenguaje basado reglas, las cuales describen al navegador como presentar los elementos de HTML. A diferencia de otros lenguajes, CSS no tiene versiones, en su lugar tiene niveles siendo el último el nivel 3.

Enlaces de referencia acerca de CSS:

- Documento de la W3C recopilatorio de las especificaciones que definen CSS. ⁴.
- Sitio web del grupo de desarrollo de CSS ⁵.
- Tutoriales de CSS ⁶ ⁷.

1.5. JavaScript

1.6. Stack de Desarrollo MEAN

El stack de desarrollo MEAN es un conjunto de tecnologías de código abierto que permite construir aplicaciones web basadas en el lenguaje JavaScript. Los componentes de este stack son los siguientes:

MongoDB: Base de datos NoSQL orientada a documentos.

Express.js: Permite el desarrollo de la programación de una aplicación web en el lado del servidor (Back-End) a través de Interfaces de aplicación (APIs).

Angular.js: Permite la implementación del patrón arquitectónico modelo vista controlador en la programación de los componentes de la Vista.

Node.js: Provee el ambiente de ejecución basado en JavaScript.

⁴www.w3.org/TR/css-2015/

⁵www.w3.org/Style/CSS/

⁶www.w3.org/Style/CSS/learning

⁷www.w3schools.com/css/



- 1.6.1. MongoDB
- 1.6.2. Express
- 1.6.3. AngularJS
- 1.6.4. Node.js

Capítulo 2

Instalación de las Herramientas

En este capítulo se explica el procedimiento de instalación de las herramientas utilizadas en esta guía y las configuraciones necesarias para su utilización en el sistema operativo Windows. Adicional a lo anterior, se proponen ejemplos que familiarizarán al lector en el uso de las herramientas.

2.1. Node.js

Para instalar Node basta con ejecutar el instalador adecuado para la arquitectura del equipo en que se instalará y seguir la configuración recomendada. Los paquetes de instalación se pueden descargar desde este sitio.

En la imagen 2.1 se puede notar que en la instalación de Node se incluye npm. npm es un administrador de paquetes al estilo apt y yum de Linux o homebrew de OS X para Node.

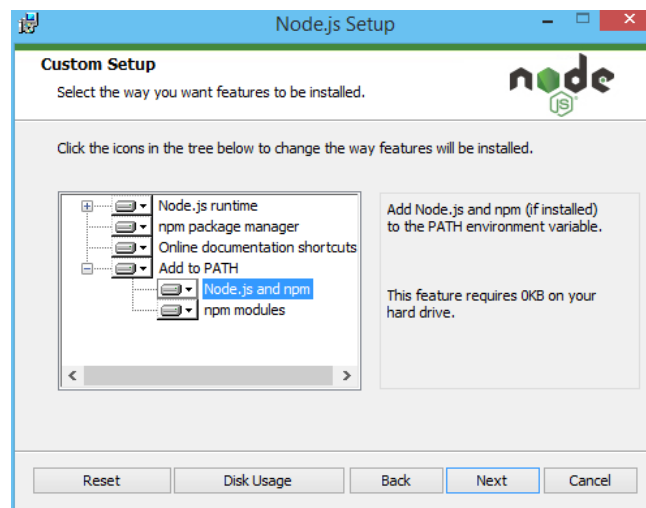


Figura 2.1: Configuración sugerida por el instalador de Node en Windows.

Una vez terminada la instalación es posible ejecutar programas escritos en JavaScript en una computadora fuera del navegador.

2.1.1. Ejemplos

Hello World

HTML Simple

2.2. Express

2.3. Bower

2.4. Bootstrap

2.5. MongoDB

MongoDB puede obtenerse desde el centro de descargas de su sitio web.

Su instalación en Windows es sencilla, sólo hay que ejecutar el paquete de instalación descargado y seguir la instalación recomendada. La instalación se realizará en el directorio *C:/Program Files/MongoDB/Server/3.2* o similar, dependiendo de la versión instalada.

En la imagen 2.2, se observa el resultado de la instalación. Dentro de la carpeta *bin* se encuentran los componentes principales y utilerías de MongoDB. Se recomienda la lectura del archivo *README* en el que se explica claramente la función de algunos de los componentes.

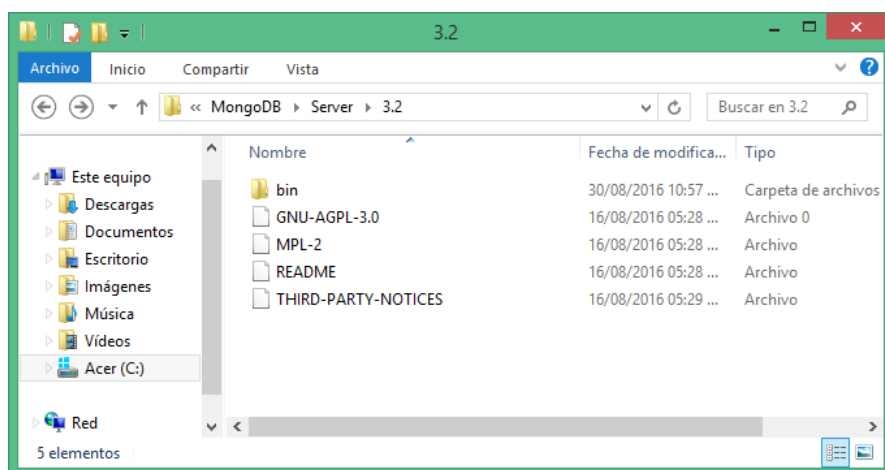


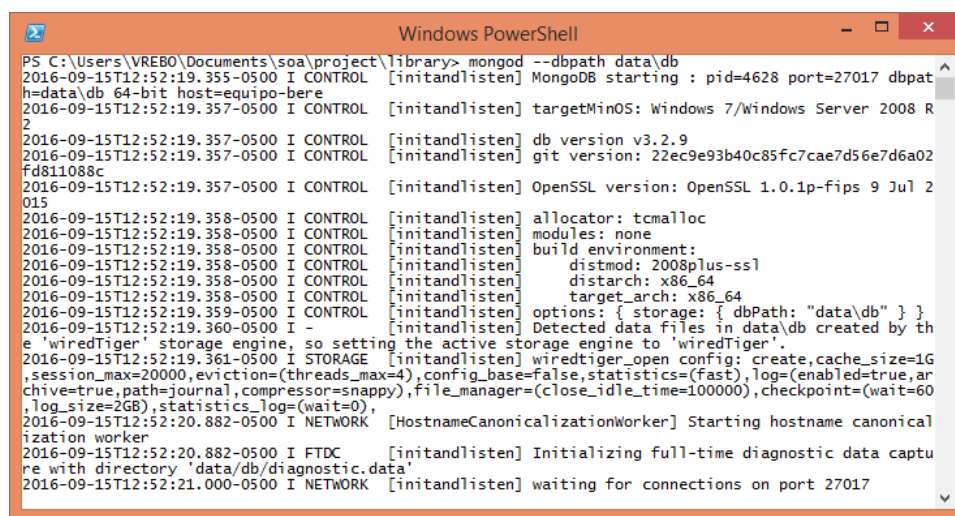
Figura 2.2: Directorio de instalación de MongoDB

Para poder hacer uso de Mongo desde el Command Prompt o Power Shell de Windows es necesario editar la variable de entorno *PATH* agregando la ruta de la carpeta *bin* del directorio de instalación de Mongo.

Para modificar la variable PATH se pueden seguir estos pasos. En el paso 4 se selecciona la variable PATH, se da click en Editar y se agrega *”;C:/Program Files/MongoDB/Server/3.2/bin”* al final del valor de la variable.

Para poner a andar MongoDB se utiliza el ejecutable **mongod** acompañado de la opción **–dbpath C:/../whatever/data/db**, esta opción indica al servicio de de Mongo cual es el directorio donde estarán o están contenidas las bases de datos. Es necesario que el directorio exista previo al inicio del servicio **mongod**.

En la figura 2.3 se observa el servicio de MongoDB en ejecución esperando por conexiones a través del puerto 27017, que es el puerto por defecto del servicio.



```
PS C:\Users\VREBO\Documents\soa\project\library> mongod --dbpath data\db
2016-09-15T12:52:19.355-0500 I CONTROL [initandlisten] MongoDB starting : pid=4628 port=27017 dbpat
h=data\db 64-bit host=equipo-bere
2016-09-15T12:52:19.357-0500 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R
2
2016-09-15T12:52:19.357-0500 I CONTROL [initandlisten] db version v3.2.9
2016-09-15T12:52:19.357-0500 I CONTROL [initandlisten] git version: 22ec9e93b40c85fc7cae7d56e7d6a02
fd811088c
2016-09-15T12:52:19.357-0500 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1p-fips 9 Jul 2
015
2016-09-15T12:52:19.358-0500 I CONTROL [initandlisten] allocator: tcmalloc
2016-09-15T12:52:19.358-0500 I CONTROL [initandlisten] modules: none
2016-09-15T12:52:19.358-0500 I CONTROL [initandlisten] build environment:
2016-09-15T12:52:19.358-0500 I CONTROL [initandlisten]   distmod: 2008plus-ssl
2016-09-15T12:52:19.358-0500 I CONTROL [initandlisten]   distarch: x86_64
2016-09-15T12:52:19.358-0500 I CONTROL [initandlisten]   target_arch: x86_64
2016-09-15T12:52:19.358-0500 I CONTROL [initandlisten] options: { storage: { dbPath: "data\db" } }
2016-09-15T12:52:19.359-0500 I CONTROL [initandlisten] Detected data files in data\db created by th
e 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
2016-09-15T12:52:19.361-0500 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=1G
,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,ar
chive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60
,log_size=2GB),statistics_log=(wait=0),
2016-09-15T12:52:20.882-0500 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonical
ization worker
2016-09-15T12:52:20.882-0500 I FTDC [initandlisten] Initializing full-time diagnostic data captu
re with directory 'data/db/diagnostic.data'
2016-09-15T12:52:21.000-0500 I NETWORK [initandlisten] waiting for connections on port 27017
```

Figura 2.3: Servicio mongod iniciado desde Windows Power Shell.

Ahora se puede ejecutar el cliente de MongoDB desde una línea de comandos para gestionar las colecciones de nuestras bases de datos.

2.5.1. Mongoose

2.6. AngularJS

Capítulo 3

Desarrollo de una aplicación MEAN a través de servicios web REST

3.1. Descripción de la aplicación

Para ejemplificar el uso del Stack MEAN y las herramientas antes descritas se desarrollará una aplicación web destinada a ser utilizada por bibliotecas. La aplicación permitirá por un lado, a la institución bibliotecaria hacer accesible el catálogo de su acervo a sus usuarios a través un sitio web, además los usuarios que hagan uso del servicio de préstamo de ejemplares a domicilio podrán consultar sus fechas de entrega por el mismo portal.

3.2. Estructura de la aplicación

3.3. Front-end

3.4. Esquema de la Base de Datos

3.5. API

3.6. Conexión Front-end Back-end

Capítulo 4

Seguridad

4.1. Seguridad a través de cuentas de usuario

4.2. Seguridad a través de Facebook