UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

barterAPP

Manage your business time.

Presentado por Adrian Aguado
en Universidad de Burgos — 23 de mayo de 2017
Tutor: Luis R.Izquierdo

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. Luis R. izquierdo, profesor del departamento de Ingeniería Civil área de organización de empresas

Expone:

Que el alumno D. Adrian Aguado, con DNI 78759314T, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado barterAPP

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 23 de mayo de 2017

Vº. Bº. del Tutor:

Luis R. Izquierdo

Resumen

Existen numerosas ocasiones en las que se requiere un reparto de turno. En muchas ocasiones, este el reparto de turnos planificado supone un problema que podría mejorarse fácilmente si se llevaran a cabo cambios bilaterales que a veces no se llegan a producir simplemente porque las partes implicadas desconocen que existe esa oportunidad de mejora.

La aplicación desarrollada presenta una solución frente a los problemas que existen hoy en día a la hora de realizar un cambio de turno en determinadas empresas o instituciones públicas. Se ha pretendido realizar una interfaz lo más sencilla e intuitiva para el usuario que le facilite la tarea descrita.

BarterApp se presenta como un aplicación web y una interfaz híbrida móvil. Esta desarrollada con las últimas tecnologías web presentes en el mercado.

Descriptores

WebApp, Angularr, Nodejs, Express, MongoDB, MEAN

Abstract

There are a lot of occasions in life which a shift deal is required. At many times, this planned shift allocation is a problem that could easily be improved if bilateral changes are made, sometimes do not come about simply because the parts involved in the shift are unaware that there is an opportunity for improvement.

The application developed presents a solution to the problems that exist today when making a change of shift in public institutions or some companies. It pretends to be a simple interface, intuitive and easy to use for the user.

BarterApp is develop as a web application and hybrid mobile interface. It is developed with the latest web technologies present nowadays.

Keywords

WebApp, Angular, Nodejs, Express, MongoDB, MEAN,

Índice general

Índice de figuras

Índice de tablas

Introducción

Hoy en día existen multitud de situaciones en las que se requiere un reparto de turnos o guardias entre un grupo de personas (por ejemplo personal sanitario en servicios hospitalarios, servicios de emergencias extrahospitalarias, bomberos, protección civil y muchos). En muchas de éstas ocasiones, el reparto de turnos planificado podría mejorarse fácilmente si se llevaran a cabo cambios bilaterales que a veces no se llegan a producir porque las partes implicadas desconocen que existe esa oportunidad de mejora, o porque intuyen que los costes de encontrar esa mejora y llegar a hacerla posible son excesivamente elevados. Otras veces los cambios sí que pueden llegar a producirse pero tras un largo período de negociación.

Actualmente, tras un proceso de investigación inicial, he determinado que no existe una herramienta sencilla que presente una solución a este problema. Es más en servicios como los sanitarios el problema no esta ni si quiera digitalizado, es más se suele realizar simple y llanamente en papel. Por una parte esta no es una mala técnica pero la implantación de una herramienta digital accesible por cualquiera puede facilitar las cosas enormemente a nivel de tiempo, espacio y entendimiento.

El mundo de las aplicaciones, tanto web como móvil, resulta extremadamente competitivo hoy en día y posicionarse en el mercado es realmente complejo. Por una parte esta la tarea de adaptarse a las nueva tecnologías tanto web, prácticamente cada muy pocos meses cambian las tecnologías o los frameworks con los que realizar las webs; como también móvil, versiones que se van actualizando de todos los sistemas operativos.

1.1. Motivacion

Explicar por que he elegido este trabajo y no otro es también describir cuáles son mis objetivos personales de cara a un futuro cada vez más cercano. Me gusta el desarrollo web y me apasiona la posibilidad de mejorar el mundo

a través de la tecnología, si bien no tengo experiencia dentro del desarrollo web si que me gustaría encaminarme hacia ello en mis próximos años.

1.2. Estructura de la memoria

- **Introducción:** Aquí se explica la motivación que me ha llevado a escoger este trabajo y no otro, la descripción del problema propuesto y una breve introducción general a la solución que se ha pretendido dar. Así mismo también una estructura de toda la memoria
- **Objetivos del proyecto:** tanto a nivel académico, personal como los objetivos alcanzados con el proyecto. También se comentará lo que le espera en un futuro al proyecto, ya que me gustaría seguir con él.
- **Conceptos teóricos:** breve explicación de los conceptos teóricos que he debido adquirir previamente para la realización del proyecto. Necesarios para comprender mejor cómo funciona el proyecto.
- **Técnicas y herramientas:** software principal, metodologías empleadas durante el proyecto. frameworks empleados y herramientas.
- **Aspectos relevantes del desarrollo:** detalles sobre el la realización del proyecto.
- **Trabajos relacionados:** otros aspectos del desarrollo del proyecto.
- **Conclusiones y líneas de trabajo futuras:** conclusiones obtenidas tras la realización de este trabajo fin de grado y hacia donde va en el futuro.

Junto a la memoria se proporcionan los siguientes anexos:

- **Plan del proyecto software:** estudio de viabilidad y planificación del proyecto paso a paso.
- **Especificación de requisitos del software:** se describe la fase de análisis; los objetivos generales, el catálogo de requisitos del sistema y la especificación de requisitos necesarios para su correcto funcionamiento.
- **Especificación de diseño:** se describe la fase de diseño de la aplicación tanto front-end como back-end y el paso de la interfaz web a la móvil.
- **Manual del programador:** recoge los aspectos más relevantes relacionados con el código fuente y su correcto manejo por parte de un programador. **Manual de usuario:** guía de usuario para el uso de la aplicación.

Objetivos del proyecto

En este apartado se van a detallar los diferentes objetivos que se buscaban, a diferentes niveles, con la realización del proyecto.

2.1. Objetivos generales

Comenzaremos con los objetivos generales del proyecto.

- Desarrollar una *aplicación web* que permita la gestión de turnos de una manera sencilla e intuitiva.
- Desarrollar una aplicación móvil híbrida que permita acceder desde cualquier plataforma a esa aplicación para dar un servicio multiplataforma.
- Permitir tener una herramienta que resuelva un problema real.

2.2. Objetivos técnicos

A continuación explicaré los principales objetivos técnicos que se pretendían.

- Toma de contacto con el mundo web por parte del alumno, a su juicio el dónde .
- Aprendizaje de la implementación de una aplicación web con *frameworks* web, como puede ser el caso de Angular o Ionic.
- Breve toma de contacto con el mundo móvil (*PhoneGapp, Cordova*).
- Aplicar la metología Scrum.
- Utilizar ZenHub como Herramienta de gestión de proyectos

2.3. Objetivos personales

Los objetivos personales que he perseguido durante todo el desarrollo han sido los siguientes:

- El diseño e implementación de una aplicación web, realizada con *Node.js*y *Express* para la parte del servidor, y *Angular* y *Bootstrap* para la parte del cliente.
- Conocer y manejar bases de datos NoSQL, como *MongoDB*.
- Mejorar mi inquietud y curiosidad sobre el mundo web.
- Averiguar y aprender el mayor número de cosas sobre las tecnologías web del mercado actual.
- Aprender a realizar un proyecto de gran envergadura, y sobre todo, a gestionarlo.

2.4. Objetivos alcanzados

Los objetivos que finalmente se han alcanzado han sido:

- Aplicación web funcionando.
- Desarrollar una *aplicación web* .
- Hola hola.

2.5. Objetivos futuros

Los objetivos que se pretenden seguir en el futuro:

- Lograr una mejora sustanciar de la usabilidad de la aplicación en su versión móvil.
- Realizar encuesta para mejorar la aplicación base en función de lo que lo usuarios demanden .
- Implementar un chat que permita poner en contacto directamente a los usuarios.
- Posicionar la herramienta dentro del mercado.

Conceptos teóricos

La parte del proyecto más ardua ha sido la de aprender a utilizar *frameworks* para crear web apps, para lo cual me he nutrido de diversos cursos online, tanto gratuitos como de pago. A continuación, en este apartado voy a relatar todas las tecnologías empleadas y a justificar el por que he empleado unas y no otras. Las técnicas y tecnologías como tal se verán en el siguiente apartado pero todas ellas presentan unos conceptos teóricos comunes que es necesario conocer.

3.1. Introducción

El desarrollo de aplicaciones informáticas evoluciona continuamente para adaptarse a las tecnologías de la información y las comunicaciones (TIC). El auge de Internet y de la web ha influido notablemente en el desarrollo de software durante los últimos años. Hoy en día la interfaz de los sistemas de información se implementa utilizando tecnologías web que ofrecen numerosas ventajas tales como el uso de una interfaz uniforme y la mejora del mantenimiento del sistema. Sin embargo, la existencia de numerosos estándares y los intereses de los fabricantes de tecnologías web dificultan el desarrollo de este tipo de aplicaciones.

En los principios de la informática las relaciones en entre los usuarios y los programas de los que hacían uso era muy diferente a lo que tenemos ahora. Ahora se potencia el diseño basado en usuario, antes es potencia más que el programa estuviera implementado de manera idónea antes que la experiencia que el usuario tenía al hacer uso del mismo. Obviamente ahora también se impulsa esa manera de desarrollar código, lo que se conoce como *clean code* [**?**], pero en un mundo en el que la competitividad es tan alta prima sobre todo la experiencia del usuario final, que a fin y al cabo va a ser el que interactúa con tu producto ya sea web o móvil.

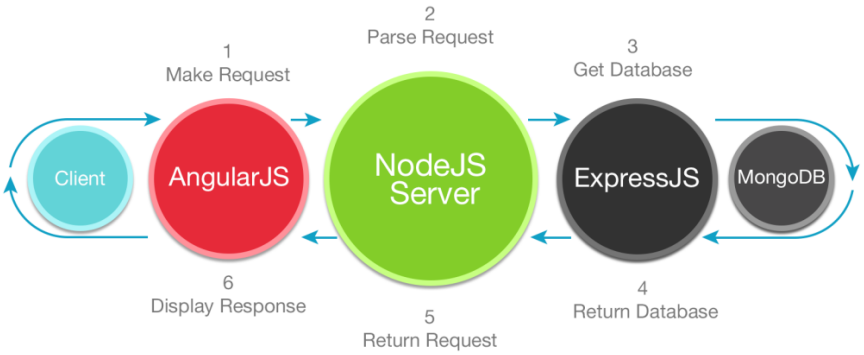

Figura 3.1: Esquema *UI, UX IxD*, fuente *dealfuel.com*.

El mundo de las aplicaciones web es un mundo de constante evolución y actualización, donde aparecen nuevas tecnologías que ofrecen tanto mejoras visuales que mejoran la experiencia del usuario como de rendimiento. Este tipo de tecnologías abren un abanico inmenso de posibilidades, y hacen pensar a los analistas y programadores de aplicaciones web que puede haber otras soluciones que mejoren su aplicación, pero que no estaban disponibles cuando definieron la arquitectura y el diseño de sus aplicaciones. Es un mundo que avanza tan rápido que en muchas ocasiones es imposible estar al día de todos los *frameworks* o tecnologías que van surgiendo, por eso es mejor focalizar los esfuerzos en alguna en concreto e intentar aprenderla de la mejor manera posible.

3.2. Concepto general del proyecto

Hace algún tiempo para realizar una web existía una gran barrera a la hora de entender el concepto de cliente y servidor. Por un lado estaba la parte del cliente, la cual se realizaba en lenguajes puros que todos conocemos como *HTML* y *CSS* para las hojas de estilo. Por otro lado estaba el servidor lo cuál significa cambiar totalmente de lenguaje, lo que suponía un salto para un programador web que debía conocer ambos lados para crear webs seguras y robustas, la parte del cliente es la que interactúa con el usuario, la que nosotros vemos, y la parte del servidor es la parte que conecta con la base de datos, en caso de que sea necesario.

Todo cambio cuando se popularizó Javascript para la realización webapps, la posibilidad de crear web con un mismo lenguaje en todas las partes resulta muy atractiva tanto para el programador como para la lógica del propio programa o web ya que se disminuye el número de errores o se consiguen localizar

de manera más sencilla al no tener que estar cambiando de lenguaje. En este punto es donde surge el stack MEAN:

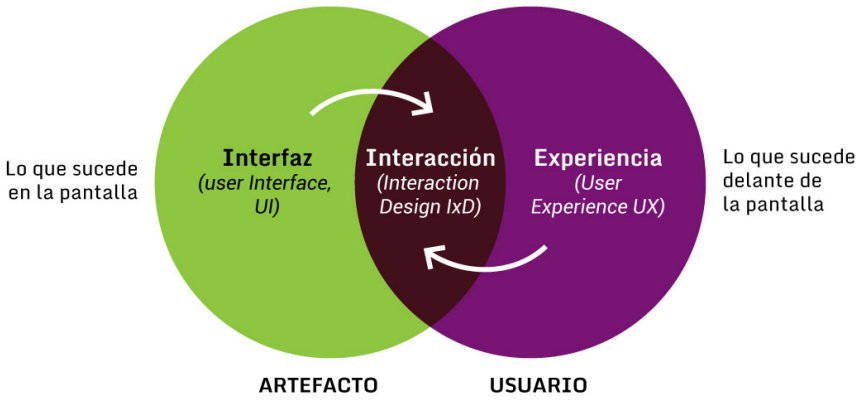

Figura 3.2: Esquema *UI, UX IxD*, fuente *kambrica.com*.

La presencia de *Javascript* para el desarrollo de software se está haciendo un hueco cada vez mayor en el mercado. Aquel humilde lenguaje que empezó en los años '90 como una vía sencilla de validar formularios, se ha convertido en parte fundamental del desarrollo de todo tipo de aplicaciones: web, móviles, bases de datos e, incluso, administración de sistemas. Esta proliferación ha llevado a *Javascript* a todas las capas de desarrollo, empezando por el lado cliente en sus inicios (el navegador), pero yendo también al servidor y a la capa de almacenamiento. En cualquiera de esos puntos podemos encontrar *Javascript* listo para ser utilizado.

¿Angular JS o Angular 2?

3.3. Análisis del sistema: MVC

3.4. Arquitectura

Como se ha comentado en secciones anteriores la aplicación tiene dos sistemas bien diferenciados: servidor y cliente. En esta sección se expondrán las arquitecturas de cada una de ellas así como de la aplicación competente.

Detalles

3.5. Secciones

Las secciones se incluyen con el comando section.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.6. Referencias

Las referencias se incluyen en el texto usando cite [**?**]. Para citar webs, artículos o libros [**?**].

3.7. Imágenes

Se pueden incluir imágenes con los comandos standard de LaTeX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

Figura 3.3: Autómata para una expresión vacía

3.8. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Tecnologías	Front-End	Back-End	BD	Memoria
HTML5	X			
CSS3	X			
BOOTSTRAP	X			
JavaScript	X			
AngularJS	X			
Bower	X			
PHP		X		
Karma + Jasmine	X			
Slim framework		X		
Idiorm		X		
Composer		X		
JSON	X	X		
PhpStorm	X	X		
MySQL			X	
PhpMyAdmin			X	
Git + BitBucket	X	X	X	X
MikTEX				X
TEXMaker				X
Astah				X
Balsamiq Mockups	X			
VersionOne	X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

-

3.9. Tablas

Igualmente se pueden usar los comandos específicos de LATEXo bien usar alguno de los comandos de la plantilla.

Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las tecnologías y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas y se pretende aqui realizar un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas.

No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas detalladamente, sino comentar los aspectos más destacados de cada opción.

He de decir que durante el desarrollo del proyecto tanto las herramientas cómo las técnicas a utilizar han ido mutando debido a que yo mismo iba descubriendo nuevas técnicas o nuevas herramientas que mejoraban mi manera de hacer el código o me facilitaban mi manera de trabajar. No ha sido así con la tecnología ya que esto hubiera supuesto más problemas que ventajas. El estar cambiando de herramienta puede que haya sido uno de los mayores errores del proyecto dado que siempre es mejor focalizarse en una sola antes de intentar abarcar demasiado. Expongo un ejemplo en las siguientes páginas.

4.1. Herramientas

En este capítulo se describen las herramientas utilizadas para el desarrollo del sistema. Las siguientes líneas servirán al lector para conocer algunas de las tecnologías más modernas y poco conocidas que se han usado en el proyecto. Se obvian el nombrar directamente lenguajes web como pueden ser *Javascript*, *HTML* o *CSS*. Existe mucha información accesible en la web sobre ellas y se hará más incapié en las tecnologías principales que usan propiamente éstos lenguajes que no son tan comunes.

Herramienta: Atom

Atom es un editor de texto moderno de código abierto desarrollado por GitHub que se ha escogido en este proyecto para desarrollar la aplicación debido a que esta desarrollado utilizando tecnologías web, luego ha sido creado por y para la web. Es posible ampliar sus funcionalidades a través de plugins desarrollados con *Node.js* que puede ser instalados de una forma sencilla a través del gestor de paquetes internos con el que cuenta, así como diferentes tipos de temas. Esta posibilidad hace que se convierta en un editor de texto muy personal, ya que es el mismo desarrollador el que elige las características que desea tener sin afectar mínimamente al rendimiento

Figura 4.4: fuente *atom.io*.

Detalles: IDE

Tal y como he nombrado en la introducción la necesidad o la curiosidad por mejorar mi trabajo hicieron que encontrara nuevas herramientas durante el desarrollo del proyecto, es el caso, por ejemplo, del entorno de desarrollo. Al comienzo empecé con brackets como IDE, ya que era el entorno que utilizaba el profesor de los cursos que me recomendó el tutor. Después y atraído por el uso a diario de Eclipse como herramienta principal de trabajo durante mi período de prácticas con la universidad decidí que podía ser una buena idea el emplearlo también para realizar la API, craso error dado que Eclipse no fue creado para el mundo del desarrollo web por lo que me vi con que cada vez me encontraba con menos soporte para lenguajes, sin ir más lejos eclipse no presenta, actualmente, soporte para *typescript* el lenguaje principal de desarrollo de *angular 2*. Por fin di con el IDE perfecto para desarrollar entornos web que es el descrito en el párrafo anterior.

Herramienta: Robomongo

Robomongo es una interfaz para *tMongoDB* que nos permite conectarnos al servidor de base de datos de forma sencilla ya que nada más arrancarlo

podemos crear una nueva conexión. Resulta también muy sencillo de utilizar.

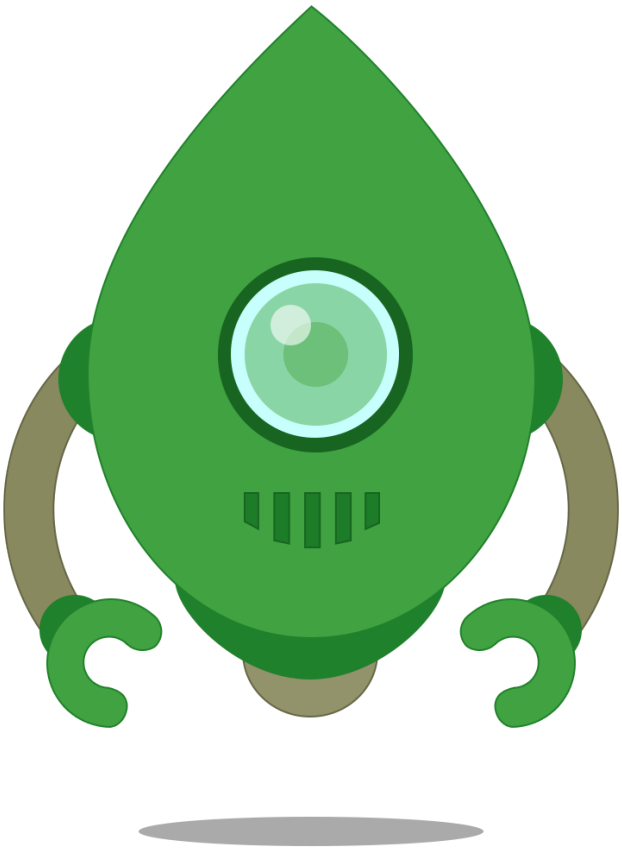

Figura 4.5: fuente *robomongo.org*.

Herramienta/Testing: POSTMAN

Las pruebas iniciales se realizaron con Postman. Esta herramienta nos permite construir y gestionar de una forma cómoda nuestras peticiones a servicios REST (Post,Get,etc). Su manejo es realmente intuitivo ya que simplemente tenemos que definir la petición que queremos realizar y pulsar el botón de enviar. Es realmente sencilla de usar y está disponible como extensión del navegador Chrome, pero también como aplicación de escritorio.

Figura 4.6: fuente *getpostman.com*.

Framework: Angular 2

Angular es un framework cliente MVC *Javascript* de código abierto creado en sus inicios por Google, permite crear *Single-Page Applications* (SPA) cuya principal característica es la de dar al usuario la impresión de que todo sucede en la misma página, sin hacer recargas de la misma. Es decir, trata de emular a las aplicaciones de escritorio, lo que se trata es de intercambiar las vista y no de recargar la página. En la actualidad cuenta con una amplia comunidad de desarrolladores que dan soporte al framework. Algunas características de Angular son:

- El sistema de databinding es muy completo y potente.
- Angular es una solución completa que incluye prácticamente todos los aspectos que puedes necesitar para crear una aplicación cliente en *Javascript* .

- La comunidad de desarrolladores crece cada día y la popularidad de Angular es un hecho.
- Está realizado para ser fácilmente testeable.

Figura 4.7: fuente *angular.io*.

Detalles: Angular Arquitectura

Un app de Angular esta basada en componentes. En su día resultó una revolución en el desarrollo web ya que lo que se persigue es que cada parte de la aplicación posea un componente de manera que se consigue hacer aplicaciones web reutilizables de algún modo.

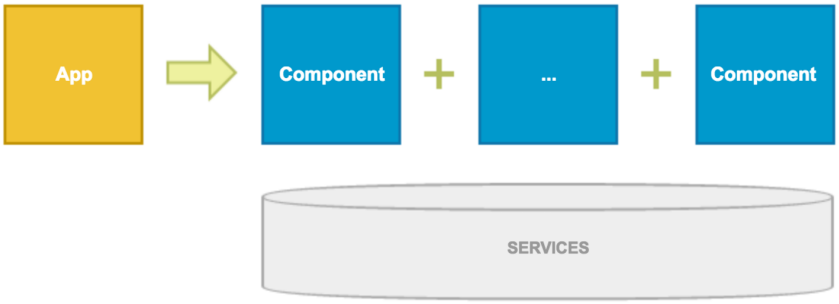

Figura 4.8: fuente *rldona.gitbooks.io/*.

- Vistas: es la parte visible por el usuario. Suelen estar parametrizadas, y todas las vistan tienen asociado un controlador.
- Controladores: Sirven los datos y funcionalidades a las vistas asociadas a él. Suelen ser estos los que utilizan los servicios para obtener los datos..
- Directivas: Ofrecen elementos nuevos o nuevos comportamientos en elementos ya existentes dentro de las propias vistas.
- Servicios: son los encargados de proveer los datos. Lo más normal es que estos datos provengan de una API externa

Detalles: Typescript

Angular 4 esta basado en Typescript , es un lenguaje de código abierto que resulta ser un superset de Javascript. Es fuertemente tipado y orientado a objetos basado en clases.

Detalles: Angular CLI

Es un intérprete de línea de comandos de Angular que te facilitará el inicio y desarrollo de proyectos, ocupándose de la creación del esqueleto de la mayoría de los componentes de una aplicación Angular. Es interesante utilizarlo dado que ahorra mucho trabajo al desarrollador además de evitar despistes innecesarios. Por poner un ejemplo cada vez que un componente es creado en Angular es necesario modificar el componente principal manualmente para poder usarlo, con Angular CLI tan solo es necesario introducir un comando para crear el nuevo componente y Angular añade todas las dependencias por ti. Se podría pensar que se trata de una heramienta de terceros pero no es asi, la interfaz de comandos es proporcionada directamente por el equipo de Angular. Se darán consejos de como usarlo en los anexos.

Por otro lado el uso de Angular CLI permite realizar en un solo comando el preparado de la aplicación para lanzarla al servidor, lo cual resulta también

muy interesante de cara a una parte más seria de la aplicación como puede ser por ejemplo lanzarla al mercado.

El aspecto de una aplicación hola mundo creada por Angular CLI contendría los siguientes elementos:

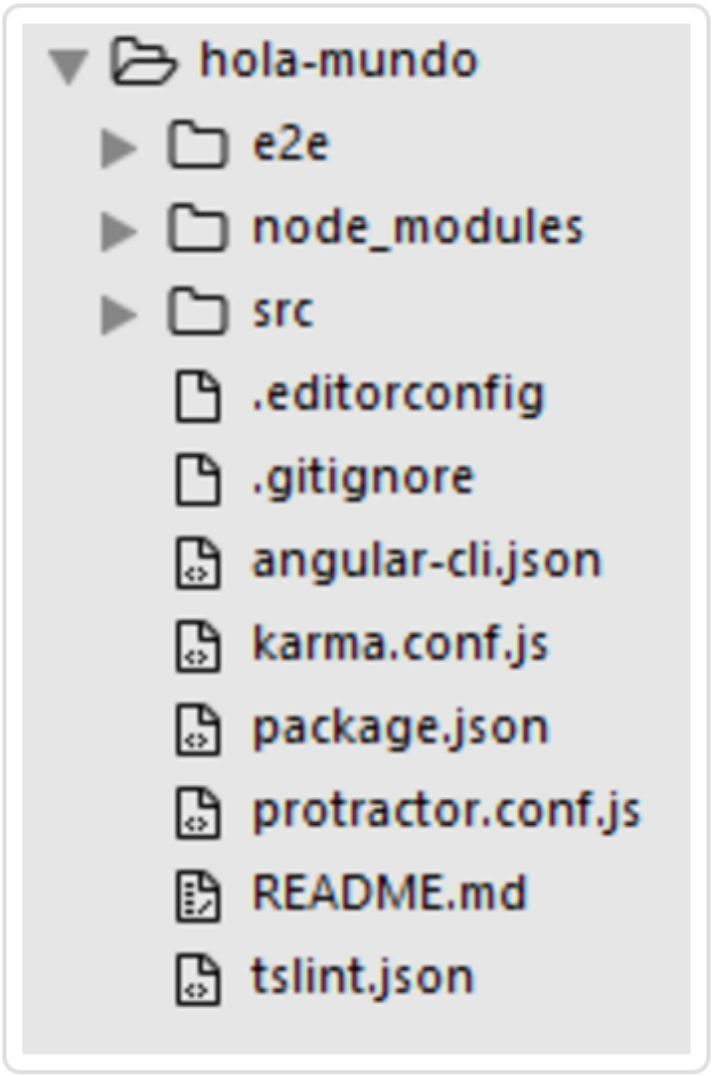

Figura 4.9: fuente *elaboración propia.*

Detalles: Módulos adicionales empleados en este proyecto

Algunos de los módulos adicionales empleados en el proyecto han sido:

- 1-
- 2-
- 3-
- 4-

Framework: Node js

Nodejs es un entorno de programación pensado para realizar funciones de servidor. Permite la construcción de servidores de forma muy sencilla y rápida, además se puede aplicar para otros usos. La ejecución es asíncrona. Esto significa que las funciones no son ejecutadas secuencialmente (si existen dos llamadas consecutivas, no es necesario que acabe la primera llamada para ejecutar la segunda). Está pensado para ser un gestor de entradas y salidas. No ha sido diseñado para ejecutar gran cantidad de código, sino para realizar comunicaciones muy rápidas y abundantes, tanto con eventos locales como por comunicación por red.

Figura 4.10: fuente *elaboración propia.*

Framework: Express

Express js es un framework web flexible para *Node js* que proporciona un conjunto robusto de características para aplicaciones web y móviles, proporciona una capa delgada de características fundamentales de aplicaciones web. Facilita la creación de API's gracias la gran variedad de métodos HTTP y middleware que proporciona.

Figura 4.11: fuente *expressjs.com*.

Base de datos: MongoDB

Para la parte de la base de datos he escogido MongoDB, estamos hablando de un sistema de gestión de bases de datos no relacionales, o noSQL . Es decir, un base de datos que no tiene tablas, se basa en colecciones de datos.

En las colecciones se almacenan contenidos que pueden tener diferentes campos. Este sistema almacena los datos en documentos de tipo JSON, cosa que puede favorecer la integración con las aplicaciones que trabajen con este tipo de formatos, como es el caso de una API REST.

Figura 4.12: fuente *mongodb.com*.

Framework: Boostrap

Boostrap es un framework CSS desarrollado inicialmente por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web.

Bootstrap esta orientado a facilitar el diseño y elaboración de aplicaciones web utilizando un sistema de cuadrículas que ayuda a conseguir interfaces claras y 'responsive', haciendo que las aplicaciones se vean con diferentes diseños en diferentes dispositivos, dependiendo del espacio del que disponga nuestra aplicación.

Figura 4.13: fuente *getbootstrap.com*.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.