

2020

REE y Calidad del Aire



José Granados Rodríguez
IES PUNTA DEL VERDE

Índice

CAPÍTULO 1: INTRODUCCIÓN	2
1.1. Introducción del Proyecto.....	2
1.2. Propósito.....	2
1.3. Objetivos del proyecto.....	2
1.4. Costes del proyecto	3
1.4.1. Costes de desarrollo.....	3
1.4.2. Costes de implantación	3
CAPÍTULO 2: ANÁLISIS DEL SISTEMA.....	4
2.1. Introducción	4
2.2. Análisis de requisitos	4
2.2.1. Requerimientos Hardware	4
2.2.2. Requerimientos Software.....	5
2.2.3. Requerimientos de Red	6
CAPÍTULO 3: ESTUDIO DE LA INFORMACIÓN RELEVANTE.	7
3.1. Introducción.	7
3.2. Valoración y comparación con antecedentes	7
CAPÍTULO 4: TÉCNICAS UTILIZADAS.....	8
4.1. Introducción	8
CAPÍTULO 5: PRUEBAS DE EJECUCIÓN	9
5.1. Introducción	9
CAPÍTULO 6: CONCLUSIONES	13
6.1. Conclusiones	13
6.2. Propuestas futuras	13
CAPÍTULO 7: BIBLIOGRAFÍAS Y REFERENCIAS.....	14
7.1. Referencias bibliográficas.	14

CAPÍTULO 1: INTRODUCCIÓN

1.1. Introducción del Proyecto

Este proyecto va a tratar sobre la calidad del aire que respiramos por tanto es necesario que hacer una pequeña concienciación a la gente, por tanto, gracias al profesorado que nos han dado las herramientas necesarias hemos tomado los datos de los distintos contaminantes que hay en el aire y lo hemos mostrado en una página web mediante gráficas más visuales en vez de mostrar solo unos números.

1.2. Propósito

El propósito de esta página web es concienciar a la gente de que tenemos que cambiar nuestro estilo de vida, también poder ver día a día cual va siendo la evolución y posteriormente poder hacer un estudio de porqué en dichos días hubo más contaminación o menos contaminación

1.3. Objetivos del proyecto

Los objetivos de este proyecto son los siguientes:

- 1) Realizar un estudio de los requisitos necesarios, así como en el estudio de las fuentes de los datos que obtenemos
- 2) Desarrollo de la aplicación web que mostrará los diferentes datos que obtenemos, es decir, ya sea mediante una API externa oficial o nuestros propios datos.
- 3) Desarrollo de una interfaz con facilidad de uso e intuitiva para que el usuario pueda visitar nuestra aplicación web con una buena experiencia
- 4) Redacción de los manuales de usuario e instalación
- 5) Redacción de la documentación del proyecto

1.4. Costes del proyecto

El coste del proyecto estará dividido en 2 partes:

1.4.1. Costes de desarrollo

En el coste de desarrollo para la realización de la aplicación web constará el estudio previo de las tecnologías utilizadas (Angular, TypeScript, MongoDB, Node.js...), por tanto, dicho eso, se añadirán a las horas dedicadas al proyecto.

Los **recursos informáticos** por ahora, según el uso estimado que va haber podemos quedarnos en el plan básico en MongoDB para el almacenamiento de los datos oficiales y propios, debido a que no habrá tantas peticiones al servidor. La tecnología de Angular no requiere ninguna licencia, ni requiere que se haga un pago para su uso, por tanto, su costo es 0, al igual que Node.js. Heroku tampoco tiene un costo adicional por que nos permite ver nuestra app desde cualquier dispositivo, porque nos ofrece un dominio (bajo su dominio "herokuapp.com") y la aplicación se aloja en GitHub por lo que para que funcione no es necesario que se eleve el coste en caso contrario si se añadiría al costo final del proyecto.

Los **costes de personal** es lo que encarecería el coste por que le pagaríamos al programador de la aplicación web por horas. Cada hora del programador se paga a 10€ y las horas estimadas para el estudio previo de las tecnologías y el desarrollo del proyecto serían unas 250 horas, por tanto, el proyecto tiene un costo aproximado de 2500 euros.

1.4.2. Costes de implantación

El coste de implantación depende de la plataforma que queramos ponerla, es decir, se puede hacer poniendo un servidor propio, es decir, preparar un servidor y acondicionarlo para un correcto funcionamiento, y las peticiones se harían a nuestro servidor que tendrá implantado nuestra aplicación web, pero también se puede implantar en un hosting sin necesidad de pagar nada pero con menos prestaciones, es decir, atenderá a menos peticiones simultaneas pero por el momento esta forma será la implantación para la reducción de los costes del proyecto

CAPÍTULO 2: ANÁLISIS DEL SISTEMA

2.1. Introducción

En los siguientes puntos explicarán como se ha desarrollado el proyecto según el guion que se puede ver anteriormente.

2.2. Análisis de requisitos

Se define el análisis de requisitos como “el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento de dichos requisitos”.

El proyecto tiene como propósito ofrecer de una manera más intuitiva y una manera mas amistosa de ver la calidad del aire, gracias a que sus datos están representados con graficas y contrastados tanto con los datos que nos ofrece la Unión Europea con nuestros datos propios que tomamos constantemente cada día. Para que esto sea posible, necesitamos unos requisitos de:

- Hardware
- Software
- Red

2.2.1. Requerimientos Hardware

Los requisitos hardware para cumplir con dicho propósito son unos servidores que alberguen tanto las bases de datos que necesitamos para almacenar los datos que obtenemos de diferentes fuentes, como el node.js que es el encargado de ejecutar la api rest que nos ofrece los datos y la aplicación web montada en Angular, que se pueden hacer mediante aplicaciones que ya existen en internet o hacerlo de manera local, es decir, montar nosotros un servidor propio en el que tenga instalado el software necesario para funcionar y tendríamos que abrir los puertos que se necesiten para que se pueda acceder a la página web.

Otros requisitos hardware serían los sensores que tomarían los datos propios que ofrecemos.

2.2.2. Requerimientos Software

Los requisitos de software para cumplir con los propósitos tendremos que tener unas mínimas aplicaciones instaladas, para ello las dividiremos en diferentes fases para poder identificar las diferentes partes del proyecto:

La primera fase que veremos es la parte mínima que tenemos que tener para empezar a desarrollar la aplicación, es decir, instalar una serie de programas sobre las que va a correr nuestra aplicación web

Los programas son los siguientes:

- Node.js: Esta aplicación es necesaria, ya que dentro de ella existen muchos módulos o paquetes que nos permiten ejecutar nuestra página web.
- Angular 9: Este software es un módulo que se puede instalar dentro de node.js que permite ejecutar una página web como si fuera una aplicación móvil, es decir, ejecuta y carga toda la aplicación cuando entramos en la página, por lo que solo tendría que cargar las peticiones “get” cada vez que se ejecute una página
- Express: Este software es un módulo que se puede instalar dentro de node.js que permite la ejecución de un servidor web, es decir, pone el servidor a la escucha de peticiones y las redirige al proyecto de angular para mostrar la página web
- HighCharts: Es un módulo que se puede instalar en node.js y ejecutarlo como un framework que representa graficas una vez recibido los datos después de haber hecho la petición get o post.

La segunda fase veremos los módulos de node.js que necesitamos para la ejecución de la api rest que ofrece los datos que nosotros necesitamos.

- Express.
- MongoClient: Es un módulo de Node.js que permite hacer una petición a una base de datos MongoDB que, en este caso, ésta estará alojada en Mongo Atlas. (en la nube). Esto permite hacer una conexión a una base de datos y hacer consultas, envío de datos, eliminación de datos, etc.

La tercera fase constará de una base de datos MongoDB alojada en la nube gracias a los servicios que ofrece MongoDB con el software de Mongo Atlas. Mongo Atlas es una especie de hosting, pero dedicado solo a base de datos y en específico las de MongoDB.

La última fase constará de un hosting que ejecute todo esto mencionado anteriormente, por tanto, utilizaremos Heroku.

Heroku es un hosting en el que subiremos la api rest por si necesitamos hacer otras aplicaciones como una aplicación móvil y ofrecer los mismos datos tanto en la web como en la aplicación. Y también tendremos subida bajo otro dominio diferente la página web.

2.2.3. Requerimientos de Red

Los requisitos de red mínimos deben ser que siempre se mantengan bien interconectados los sensores para su correcto almacenamiento en la base de datos ya que, si no se tuviera una conexión a internet a la hora de la toma de los datos, no se almacenarían en la base de datos por tanto no se mostrarían en la web.

CAPÍTULO 3: ESTUDIO DE LA INFORMACIÓN RELEVANTE.

3.1. Introducción.

Aquí en este apartado veremos la diferencia entre una aplicación como puede ser la de Rossi ya que toda la idea general partió de este proyecto.

3.2. Valoración y comparación con antecedentes

Más que una comparación con la aplicación de Rossi esto es una actualización debido a que hemos incluido diferentes apartados, es decir, se han incluido el apartado de los datos propios que ofrecemos a partir de una placa de Arduino con los sensores indicados en la página web.

También se ha cambiado en la interfaz en la que corre, es decir, ha pasado de correr en HTML, CSS y JavaScript a una aplicación web en Angular que utiliza los diferentes recursos mencionados anteriormente más los módulos que ofrece Angular y Node.js.

CAPÍTULO 4: TÉCNICAS UTILIZADAS

4.1. Introducción

Las técnicas utilizadas para la realización de este proyecto fue primero el aprendizaje de las tecnologías.

Lo segundo fue poner en práctica lo aprendido, es decir, empezamos instalando los módulos necesarios para el correcto funcionamiento, es decir, instalando dentro de Node.js el módulo de angular, express y HighCharts

Una vez que tengamos todo eso instalado pasamos a crear componentes, es decir, los componentes son partes que pueden hacer algo en específico y añadirlo a una página en concreto y hacer uso de el en cuantas páginas se requiera. Después de generar el componente identificamos que partes tiene cada componente por lo que veremos 3 archivos importantes, que son el html css y ts (TypeScript) que son los necesario para realizar las operaciones necesarias para que el componente funcione como se desee. También tendremos los servicios que son como los componentes, pero solo ofrecen un servicio en concreto como puede ser una petición a una api rest.

Por tanto, una vez sabido esto, las técnicas utilizadas en cada página de la web que se carga una gráfica automáticamente realizaremos una petición get a una api para obtener los datos necesario para que la gráfica pueda representarlos de la manera que queremos. Por lo contrario, para las páginas como son la de los datos externos y propios la técnica utilizada es rellenar un formulario y en cuanto le damos al botón de enviar lo que hacemos es una petición post a nuestra api en la que nos ofrece los datos que necesitamos, es decir, según nuestros parámetros que pasamos mediante el formulario la consulta que hacemos a la base de datos está preparada para que cuando le pasemos el formulario nos ofrezca los datos que necesitamos y poder representar los datos en nuestra gráfica elegida.

Para subir esta aplicación tenemos que preparar el servidor express que es el encargado de estar a la escucha y redirigirlo a nuestro index, pero también tenemos que añadir en el archivo package.json los scripts necesarios para subir la aplicación a Heroku y desplegarla en su versión de producción.

Los scripts son los siguientes:

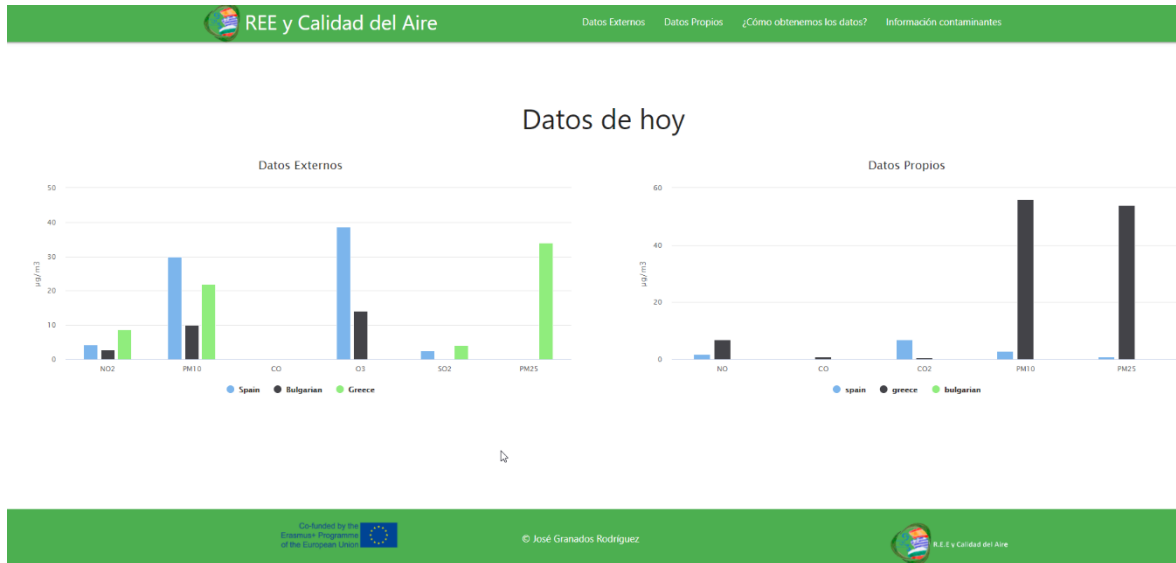
```
"scripts": {  
  "ng": "ng",  
  "start": "node server.js",  
  "build": "ng build",  
  "test": "ng test",  
  "lint": "ng lint",  
  "e2e": "ng e2e",  
  "lite": "lite-server",  
  "postinstall": "ng build --prod"  
},
```

CAPÍTULO 5: PRUEBAS DE EJECUCIÓN

5.1. Introducción

Para las pruebas de ejecución del proyecto lo probaremos ya subido en Heroku para comprobar que funciona correctamente:

Al cargar la página de inicio tendrá que cargar dos gráficas con los datos de hoy ya sean los datos externos o propios



Al hacer clic en Datos Externos en el menú nos aparecerá la siguiente imagen:

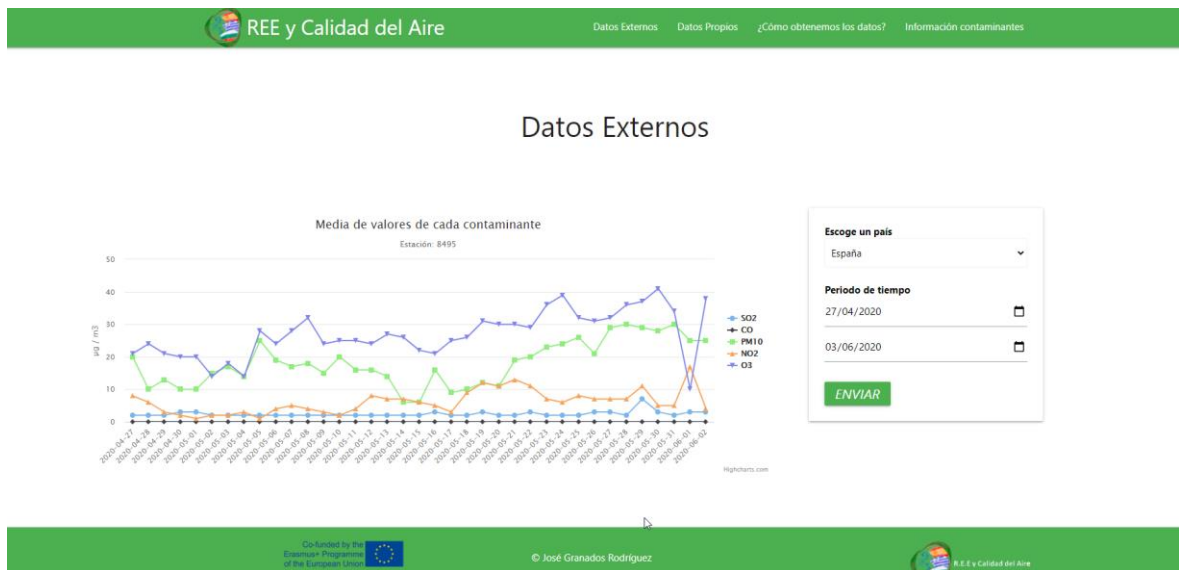
Datos Externos

Escoge un país
Elige una de las opciones

Periodo de tiempo
dd/mm/aaaa
dd/mm/aaaa

ENVIAR

Para que nos aparezca la gráfica tendremos que rellenar el formulario que tenemos a la derecha y hacemos clic en enviar



A continuación, nos pasaremos a la pagina de Datos Propios y nos aparecerá de esta manera:

REE y Calidad del Aire

Datos Externos Datos Propios ¿Cómo obtenemos los datos? Información contaminantes

Datos Propios

Escoge un país
Elige una de las opciones

Periodo de tiempo
dd/mm/aaaa
dd/mm/aaaa

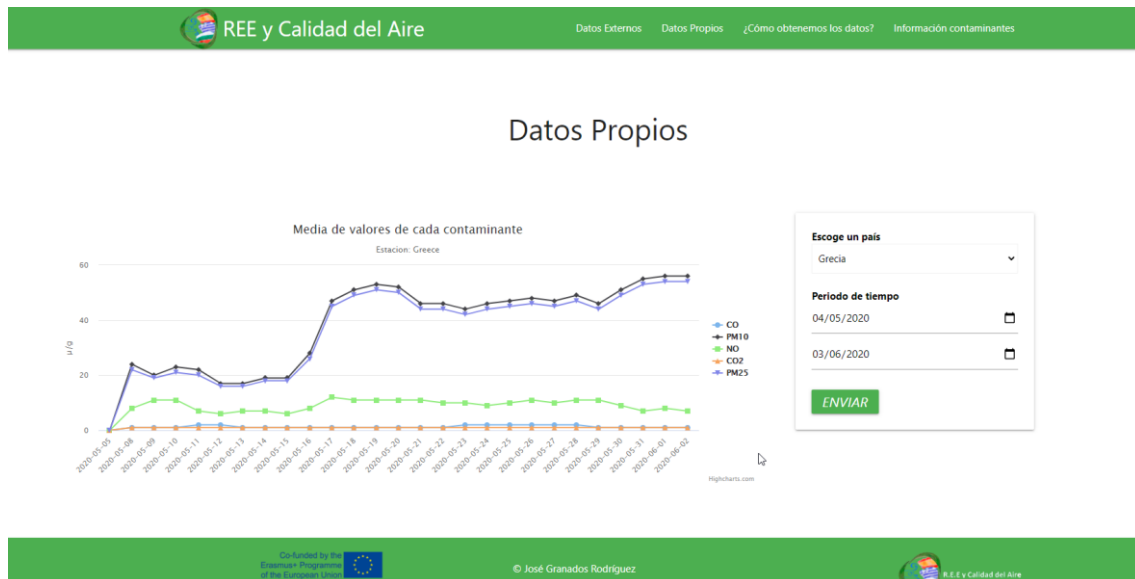
ENVIAR

Co-funded by the Erasmus+ Programme of the European Union

© José Granados Rodríguez

REE y Calidad del Aire

En la imagen anterior vemos lo mismo que en la de datos externos pero los datos que muestra son los nuestros por tanto ahora mostraremos la página con el formulario relleno y ver como muestra la gráfica correctamente con los datos que le hemos indicado.



Ahora las páginas siguientes solo mostrarán información sobre la realización del proyecto.

En la siguiente imagen vemos como nos muestra la página de ¿Cómo obtenemos los datos?

Gracias al estudio de varios alumnos y profesores del instituto [IES PUNTA DEL VERDE](#) que debido al descontrol de contaminación que existe (como pueden ser la expulsión masiva de gases a la atmósfera y muchos más factores), están dañando a nuestro planeta haciendo que la calidad del aire que se respira sea cada vez más y más perjudicial para los seres vivos.

Los datos que ofrecemos son de dos fuentes de datos, una fuente de datos ajena a nosotros proporcionada por la Unión Europea de sus diferentes estaciones que están tomando datos de los contaminantes diariamente. Y la otra fuente de datos es creada por nosotros, es decir, mediante una placa de Arduino y diferentes sensores de medida hemos sido capaces de proporcionar datos de los contaminantes que están en el aire. Los datos que tomamos con estos dispositivos, son almacenados en una base de datos para poder posteriormente hacer un estudio de la variación de los datos y saber el porqué de esa variación.

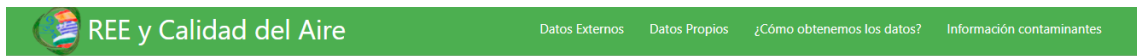
Hardware utilizado

El hardware que hemos utilizado ha sido una placa NodeMCU que la hemos elegido porque tiene un módulo de Wi-Fi ya integrado, para poder subir los datos a una base de datos alojada en la nube.

¿Con qué medimos los contaminantes?

Los contaminantes se miden mediante unos sensores añadidos a la placa de NodeMCU, estos sensores son:

Ahora veremos la página de la información de los contaminantes que nos mostrará los contaminantes y su información sobre ellos, si son muy perjudiciales y cual es su medida máxima por día y por mes



¿Qué contaminantes son nocivos?

Los contaminantes que mostramos en esta web son entre todos los contaminantes que hay los que pueden ser mas perjudiciales para el para el medioambiente, por tanto para nosotros y nuestro planeta.

A continuación os explicaremos que son, para que sirve cada contaminante y entre qué valores deben estar los diferentes contaminantes:

Ozono (O₃)

El ozono a nivel del suelo —que no debe confundirse con la capa de ozono en la atmósfera superior— es uno de los principales componentes de la niebla tóxica. Éste se forma por la reacción con la luz solar (fotoquímica) de contaminantes como los óxidos de nitrógeno (NO_x) procedentes de las emisiones de vehículos o la industria y los compuestos orgánicos volátiles (COV) emitidos por los vehículos, los disolventes y la industria. El límite recomendado en las Directrices de la OMS sobre la Calidad del Aire es 100 µg/m³ de media en 8h.

Dióxido de nitrógeno (NO₂)

Como contaminante atmosférico, el NO₂ puede correlacionarse con varias actividades:

- En concentraciones de corta duración superiores a 200 mg/m³, es un gas tóxico que causa una importante inflamación de las vías respiratorias
- Es la fuente principal de los aerosoles de nitrato, que constituyen una parte importante de las PM_{2.5} y, en presencia de luz ultravioleta, del ozono.

Las principales fuentes de emisiones antropogénicas de NO₂ son los procesos de combustión (calefacción, generación de electricidad y motores de vehículos y barcos)

El límite recomendado en las Directrices de la OMS sobre la Calidad del Aire es 40 µg/m³ de media anual y 200 µg/m³ de media en 1h

Partículas (PM)

Las PM son un indicador representativo común de la contaminación del aire. Afectan a más personas que cualquier otro contaminante. Los principales componentes de las PM son los sulfatos, los nitratos, el amoníaco, el cloruro de sodio, el hollín, los polvos minerales y el agua. Consisten en una compleja mezcla de partículas sólidas y líquidas

Las imágenes anteriores tienen mas información si hacemos scroll hacia abajo.

CAPITULO 6: CONCLUSIONES

6.1. Conclusiones

Las conclusiones que saco con el propósito de este proyecto es que debemos de concienciarnos y cuidar nuestro planeta ya sea mediante los gases o con los residuos electrónicos y eléctricos porque si todo lo que no queremos porque se haya roto o no lo uses, no tienes porqué tirarlo, es decir, puedes darle una segunda vida o como última opción reciclarlo.

Y respecto a la realización del proyecto, las conclusiones que saco es que dedicándole el tiempo que necesites todo más tarde o más temprano se acaba sacando con constancia y trabajo. He aprendido mucho con este proyecto tanto académicamente como en lo personal ya que te prepara para seguir avanzando y a ser ambicioso, aunque no dentro del mismo proyecto, no estén implementadas todas las ideas que a uno se le ocurren.

6.2. Propuestas futuras

Las propuestas que hago son que se podría mejorar la parte de front-end y quizás se podría depurar el código para que la carga del dispositivo sea menor. También se podrían añadir a un mapa las tomas de datos de los dispositivos portátiles.

CAPÍTULO 7: BIBLIOGRAFÍAS Y REFERENCIAS

7.1. Referencias bibliográficas.

La información sobre Angular las he sacado de un curso de open webinars:

<https://openwebinars.net/academia/aprende/angular/6546/>

MongoDB lo he aprendido de lo que hemos visto en clase de base de datos y también la página oficial de clase:

<https://docs.mongodb.com/manual/>

Un curso de MongoDB para aumentar los conocimientos adquiridos en clase

https://www.youtube.com/watch?v=3hdO_LXgZ_E&list=PLi26yBUldF2-ppuQIOywLdoszLy5rnX83

A administrar un servidor de node.js con express y angular lo he sacado de varios canales de YouTube aquí dejo una lista de reproducción:

<https://www.youtube.com/watch?v=khClunNAEHI&list=PLC2spcg-z-TwSVF0KNJplqyUljAfEIQZN>

Y las dudas y errores que me han surgido las he solucionado gracias al foro:

<https://stackoverflow.com/>

La documentación oficial de angular era necesaria también para el desarrollo

<https://angular.io/docs>

Las peticiones post y get cual era su funcionamiento dentro de angular

<https://www.youtube.com/watch?v=74X18AoZ2Gk&t=349s>

<https://www.youtube.com/watch?v=SYBzE68Ee-g>

Para aprender el funcionamiento de node.js también he necesitado aprender de la formación de un youtuber que hacía pequeñas aplicaciones de las que se podían hacer con node.js:

https://www.youtube.com/watch?v=3J925fRI_UE&t=10s

<https://www.youtube.com/watch?v=3FGRqijmBWI>

Para tener acceso a la API desde mi aplicación necesitaba añadir un CORS Policy para hacer uso de mi aplicación:

<https://victorroblesweb.es/2018/01/31/configurar-acceso-cors-en-nodejs/>