Notas sobre el ejercicio de avantio

Apuntes sobre el ejercicio de Avantio

Resumen de las tareas a realizar

1. Diseñar una aplicacion web de noticias.
2. Vistas a diseñar.
   2.1. Listado de noticias.
   2.2. Detalle de la noticia.
   2.2.1. Dentro de cada noticia hay un slide-out que nos vale para editarla y para crearla.

> Los assets los proporcionan ellos en el repositorio github

3. El back-end tiene que ser una rest-api y con mongoDB.
4. El modelo de la noticia es el siguiente:
   4.1. title
   4.2. body
   4.3. image
   4.4. url
   4.5. publisher
5. Los endpoints que se tendran que crear son los siguientes:
   5.1. Listado de noticias
   5.2. Detalle de noticia
   5.3. Eliminación de noticia
   5.4. Edición de noticia
   5.5. Creación de noticia

Frontal

1. Diseño responsivo opcional

> No es mala idea que sea responsivo ya que es bastante facil con flex box reestructurar la aplicacion gracias a que las noticias van en cajitas. Revisar con mas profundidad a la hora de diseñar las pantallas.
> En la pagina de figma si se inicia sesion las maquetas estan adapatadas a formato movil, asi que es mas facil hacerlo responsive

2. Valoran los siguientes puntos:
   2.1. La arquitectura del proyecto.

   > Hacer un buen diseño y una buena estructura con draw.io

   2.2. La arquitectura de componentes que se creen.

> No se me ocurre de primeras que los componentes se puedan reusar.

2.3. La claridad del codigo y de las hojas de estilo.

> Llenar todo de comentarios
> Estructurar bien el scss y usar los tags buenos de html

2.4. El uso de layout css modernos (flexbox, cssgrid).

> Usar flexbox ya que gridcss aun es muy nuevo y flexbox lo aceptan la mayoria de los navegadores.

3. Se tendra en cuenta:

3.1. Código preparado para producción

> Al ser una aplicacion simple no veo mucho mas inconveniente ya que el angular-cli lo deja preparado con el enviroment.ts

3.2. Entregar una solución que se pueda escalar o añadir funcionalidad con facilidad

> No cerrar mucho los componentes

3.3. Sientete libre a la hora de añadir cualquier mejora de UX/UI

> No se me ocurre nada mas que añadir animaciones

Back-end

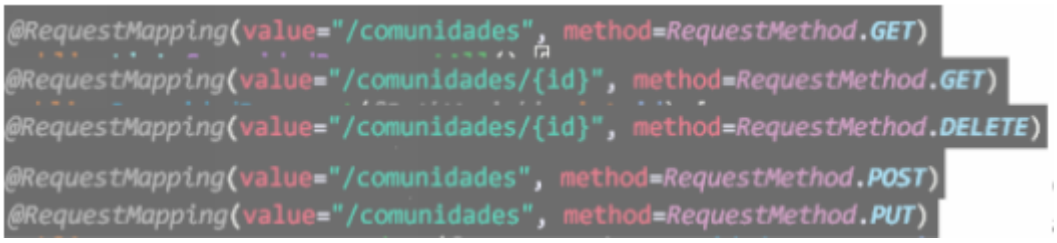
1. La arquitectura de la aplicación

> Buscar por internet algun ejemplo pero yo creo que usare el de siempre
> Controlador (entrada end-point) --> Servicio(acceso bbdd)

2. El diseño del API según el standard

> Reviar el proyecto final de grado, la parte back-end, porque esto ya lo tuve que mirar en su epoca y recuerdo haber sacado unos buenos apuntes al respecto (revisar mis apuntes de OneNote o el github).

> Encontrado:
> El instructor decía que son buenas practicas hacer los end-points de la siguiente manera.
> Da igual que método sea(GET/POST/PUT) el end-point siempre tiene que ser el mismo y ademas tiene que ser en plural.
> El instructor decía que esto eran buenas practicas y que ademas en un futuro el código es mas

> fácil de mantener y de organizar.
>
> 
> ```
> @RequestMapping(value="/comunidades", method=RequestMethod.GET)
> @RequestMapping(value="/comunidades/{id}", method=RequestMethod.GET)
> @RequestMapping(value="/comunidades/{id}", method=RequestMethod.DELETE)
> @RequestMapping(value="/comunidades", method=RequestMethod.POST)
> @RequestMapping(value="/comunidades", method=RequestMethod.PUT)
> ```

3. Las buenas prácticas en el desarrollo de una solución
4. Código preparado para producción

> Hacer pruebas desplegando el docker en la raspberry para probarlo bien.

5. Testing

> Programar test Unitarios (*Una prueba unitaria se utiliza para comprobar que un método concreto del código*)
> Si hay tiempo buscar alguna herramienta para hacer End-to-end (*Simulamos que una aplicacion web consume los servicios*)

6. Dockerización de la aplicación

> Hay que hacerlo si o si.

7. Paginacion?

> Es algo que no habia pensado, pero el tipico boton abajo que pone cargar mas.
> Primero hacerlo sin paginacion y luego añadir esta nueva funcionalidad.

Todo

- [x] Definir tecnicamente la arquitectura/pantallas del proyecto
  - [x] Frontal
    - [x] Cojer la maqueta que hay en figma e ir pantalla por pantalla definiendo los componentes/rutas.
    - [x] Definir una lista de componentes que se van a usar en toda la aplicacion
    - ~~[ ] Añadir storybook?~~

      > No creo que tenga el suficiente tiempo para hacerlo
  - [x] Back
    - [x] Crear un diagrama con la arquitectura que se va a usar
    - [x] Crear los endpoints
      - [x] Listado de noticias
      - [x] Detalle de noticia

- [x] Eliminación de noticia
- [x] Edición de noticia
- [x] Creación de noticia

> El back en si ya esta fino ahora a por el frontal.
> Postman tambien
> Revisar y pasar todas las constantes que se usan al tipico archivo .env

- [ ] Front, pagina 1
  - [x] Crear el componente navegacion izquierda

> Recordar que tiene que tener 2 estados, expandido y no expandido y responsivo
> Revisar los modulos, que creo que he metido muchos y mal organizados

  - [x] Crear el componente expandido
  - [x] Añadirle la fncionalidad de no e xpandido
  - [x] Hacerlo responsivo
- [ ] Componente noticia
  - [x] Hacer 1 solo componente que se comporte de 3 formas distintas
  - [x] Transformar la imagen de base 64 (que es como se guarda en el back) a img, a traves del sanitizer de angulart.
  - [ ] Si hay algun error en las request avisar al usuario de alguna forma?

> URGENTE: Revisar los estilos del componente noticia porque estan super empastrados por las prisas...

- [x] Hacer la pantalla de leer noticia, y crear las ruta
- [x] Hacer el componente modificar noticia (modal)
- [x] Crear el servicio necesario para hacer el POST y enlazarlo con el modal
- [x] Añadido la opcion de subir imagenes (Funciona siempre y cuando la imagen no sea muy grande)

> El problema que hay en el back es que hay un tamaño maximo, asi que el formulario de la foto hay que meterle validaciones etc..