En esta pequeña documentación me dispongo a explicar cómo he enfocado esta prueba técnica.

Lo primero ha sido crear el proyecto en angular 19 standalone e instalar Material.

Tras leer las instrucciones y haberme entrevistado con Francisco, lo primero que he hecho ha sido mockear un array de héroes para tener un mock de datos, y crear un servicio para poder tratarlo, con las llamadas pertinentes de un crud, que en este caso serán tratadas con signals porque es un tema al que se le dio mucho énfasis en la entrevista. Es por eso que en el código se podrán ver varias declaraciones de signals y métodos computed para imprimir los héroes.

En este caso he decidido usar material, y en vez de una lista clásica, he decidido usar las material cards para presentar una estructura en grid, con 12 elementos por página, para poder, también con signals, instaurar un paginador que nos permita recorrer 3 páginas. En el tema de estilos he decidido simplemente darle unos estilos con tonos súper heroicos, y procurar que estén centradas y presentables. Estas cartas presentan el nombre, el alter ego y el universo del que proviene el héroe.

En referencia a la botonera que tiene la carta, al clickar en el botón del ojo, podremos entrar en un detalle en el que además nos saldrá una imagen del héroe que hayamos pulsado. Podremos modificar un héroe con el lápiz gracias a la creación de un modal auxiliar con un formulario y podremos borrarlo con el icono de la papelera. En este caso, como ya he usado un modal para crear y modificar, he decidido usar un alert que nos avise de que lo vamos a borrar, con la posterior confirmación.

Por otro lado, arriba tenemos el botón para añadir héroe, que en este caso recicla el modal de modificar, puesto que son iguales, y así no tengo que saturar el proyecto con componentes redundantes. Aquí podremos introducir al héroe, y se nos insertará al final del array de héroes. Tanto en este modal como en el de modificar, al guardar nos encontraremos un spinner, introducido desde un spinner service con un componente, para poder usarlo globalmente, en el que he metido un timeout de 3 segundos de espera para simular el tiempo que tardan las peticiones.

Debajo encontraremos el botón de filtrar, en el que podremos filtrar por cualquiera de los 4 campos (nombre, apellido, alter ego o universo), también realizados con signals y computeds.

Podemos encontrarnos una directiva en la que la primera letra de los campos del formulario de crear y añadir héroe siempre se verán en letra mayúscula.

Si consultamos el app.routes, veremos la creación de una ruta vacía estándar, que será el componente de héroes, y una ruta hacia el detalle del héroe seleccionado.

Por último se ha probado a hacer unos cuántos tests unitarios en el componente y en el servicio, quedando dos sin resolver, uno de ellos en este caso porque me pide crear el servicio desde el constructor, pero prefiero usar inject como las prácticas nuevas requieren.

Estos tests están realizados con karma y jasmine, puesto que vienen integrados en angular y se pueden ejecutar con el comando ng test y ver los errores en el explorador.