Prueba técnica FullStack

En este documento se incluyen tanto la información demanda como algunos aspectos de las pruebas realizadas.

En cuanto a la información solicitada:

* Tipo de prácticas: curricular.
* Posible fecha de incorporación a jornada completa: Lo antes posible.
* Meses durante los que podrías realizar las prácticas: Disponibilidad total hasta Julio en caso de ser presenciales.

Respecto a la fecha de incorporación, dado que ya he acabado mis estudios a falta de realizar las practicas obligatorias, ahora mismo dispongo de completa disponibilidad para ello. Por otra parte en cuanto a los meses de realización quiero realizar un pequeño apunte. Dado que vivo en Ontinyent (a unos 40 minutos de Valencia en coche) y ahora mismo no dispongo de coche propio, me resultaría complicado desplazarme todos los días a Valencia en caso de que fuera necesario realizar las practicas de forma presencial. Sin embargo, en este momento dispongo de un piso de alquiler en Valencia compartido con otros compañeros estudiantes hasta por lo menos el mes de Julio, por lo que, en el caso de que fuera necesario podría extender el contrato de alquiler para otro año más. Cualquier duda y/o problema que suponga esto último podemos debatirlo sin problema alguno en caso de concertar la entrevista.

Una vez explicado esto, voy a comentar las implementaciones de ambos ejercicios. En primer lugar, quiero destacar que dado que es la primera vez que trabajo tanto con Angular como con Spring (a pesar de que conocía ambas herramientas), las implementaciones pueden resultar un tanto simples, o al menos no tan sofisticadas como me habría gustado por la falta de tiempo para aprender los conceptos básicos de ambos y adicionalmente realizar ambas implementaciones.

Frontend

Para el desarrollo del ejercicio de Frontend se ha creado en primer lugar un servicio “movie” que mediante el uso del api key, interactúa con The Movie Database para obtener tanto la lista de películas como los detalles de las mismas. A continuación, se han creado dos componentes, uno por cada funcionalidad requerida, que utilizan los métodos creados en el servicio. Ambos componentes disponen de sus propios archivos HTML y CSS para implementar la interfaz de cada uno de ellos.

Por tanto, al ejecutar la aplicación veremos una página que mostrará un buscador de películas. Al introducir nuestra búsqueda, nos aparecerán una serie de películas que, al pinchar sobre alguna de ellas, nos permitirá ver otra página con algunos detalles de la misma. Como se ha comentado anteriormente, se trata de dos interfaces muy simples dada la falta de tiempo para realizar un diseño mas complejo. Sin embargo, se han implementado algunos pequeños detalles como la incorporación del poster de la película en el listado, el aumento de tamaño de las tarjetas que incluyen la película al pasar el ratón por encima, o transiciones al pasar el ratón por encima de los enlaces entre otros. Estos pequeños detalles contribuyen a que la interfaz sea más “limpia” a pesar de la simpleza.

Backend

En cuanto al desarrollo del ejercicio de Backend, en primer lugar encontramos un controlador llamado “MovieController” que proporciona un endpoint REST para buscar películas por nombre, utilizando el api key para interactuar con la base de datos como en el anterior ejercicio.

Por otra parte encontramos diferentes clases de configuración. La primera de ellas es “CorsConfig”, que configura la política de CORS. En ella se permite, básicamente, el acceso desde el origen <http://127.0.0.1:5500>. Esta es la dirección de una pequeña página de prueba creada expresamente para poder probar esta configuración. Dado que para probar esta funcionalidad es necesario ejecutar esta página desde un servidor externo, se ha utilizado la extensión incorporada en VSCode para ello llamada Live-Server. Esta extensión nos permite exponer un pagina HTML en un servidor local en la dirección <http://127.0.0.1:5500>.

La segunda de las configuraciones “SecurityConfig”, maneja el acceso autenticado a la API. Para ello, tanto si utilizamos cualquier extensión REST como RESTClient en Firefox, o desde la propia página de prueba, veremos un cuadro de dialogo donde será necesario introducir como credenciales “user” y “password” como usuario y contraseña respectivamente para poder ver la información deseada.

Finalmente, la configuración “CacheConfig” crea una caché que servirá los datos ya demandados previamente. Dado que los datos demandados no son excesivamente pesados y los tiempos de respuesta son muy reducidos, se ha incluido un logger en la aplicación que muestra la traza de la caché con la intención de comprobar que esta funcionando correctamente.