

Ejercicio 2. Programación (6p)

Se desea implementar una aplicación web SPA para consultar los libros con la siguiente funcionalidad:

- Al entrar en la aplicación aparecerá un formulario con un campo de texto para insertar un criterio de búsqueda y un botón para realizar la búsqueda.
- Al pulsar el botón se realizará una petición al servidor web solicitando los libros que tengan en el título el texto introducido por el usuario en el campo de texto.
- Desde que el usuario pulsa el botón hasta que se muestran los resultados se mostrará una imagen de un spinner de carga (consideramos que está disponible en ruta /assets/spinner.gif relativa a la URL en la que se carga la app).
- Cuando llegan los resultados del servidor, se muestran debajo del formulario.
- Si no hay ningún resultado, se deberá mostrar un mensaje “No hay resultados”.
- Cada libro se representa con la siguiente información: Título y precio.
- Aquellos libros que estén en oferta deberán mostrarse con el texto en color verde.
- Se asume que hay libros ya almacenados en una base de datos correctamente configurada en el pom.xml.

Se pide:

- **A) Implementar el backend de la aplicación web con Spring MVC, SpringData y JPA. (2p)**
 - No es necesario escribir el fichero pom.xml, se puede asumir que tiene todas las dependencias correspondientes y una base de datos adecuada y correctamente configurada.
 - No es necesario escribir la clase Application.
 - Es necesario escribir TODOS los demás ficheros de la aplicación.
 - No es necesario incluir los imports en los ficheros Java.
 - No es necesario implementar los getter y setter. Se puede usar Lombok si se considera.
 - Se valorará especialmente que la API REST esté diseñada de forma correcta (formato de las URLs, uso de los códigos de estado, etc.)
- **B) Implementar el frontend usando Angular (2p)**
 - Se puede asumir que se dispone de un proyecto Angular con todos los ficheros necesarios correctamente configurados (index.html, angular.cli, package.json, tsconfig.json, etc.)
 - Hay que implementar un componente y un servicio para las peticiones al backend. Concretamente, hay que implementar los ficheros: app.component.ts, app.component.css, app.component.html y books.service.ts.
 - Se usará HTML plano en el template.
 - No es necesario incluir los imports en los ficheros TypeScript.
 - Se asumirá que el proxy está correctamente configurado y se pueden usar URLs relativas para acceder a la API REST del backend.
- **C) Empaquetar la aplicación en un contenedor Docker (2p)**
 - Escribe un script (en Bash, Bat o PowerShell) que al ejecutarse construya la imagen daw/libros:1.0.0 con la aplicación (tanto backend como frontend) empaquetada. Se deberán implementar los ficheros Dockerfile que sean necesarios.
 - Se asumirá que el código Spring está alojado en una carpeta “backend” y el código Angular está alojado en una carpeta “frontend” y el script y el/los Dockerfile están en la carpeta raíz.
 - Se asumirá que en el sistema donde se ejecute el script o se construya el Dockerfile sólo tiene Docker instalado. No dispone de Node.js ni Maven.
 - Se asumirá que en DockerHub existen las imágenes maven:3.6.0, node:14.0.0 y openjdk-jre:11.0.0