

Ejercicio Práctico: Aplicación web de reserva de sala (6p)

Se desea implementar una aplicación web SPA para reservar salas con la siguiente funcionalidad:

- Al entrar en la aplicación aparecerá una lista de franjas horarias en las que se puede reservar una sala. Para facilitar la implementación, sólo se podrá reservar la sala en horas enteras desde las 9:00 hasta las 19:00 de un único día (el día actual).
- Cada franja horaria podrá estar en dos estados:
 - **Libre:** Al lado de la hora pondrá la palabra “Libre” en la web y a la derecha aparecerá el botón “Reservar”. Al pulsar en reservar, aparecerá un cuadro de texto justo al lado de la franja en el que el usuario podrá poner su nombre. Además aparecerá un botón “Guardar” para dar de alta la reserva en el backend.
 - **Reservada:** Al lado de la hora pondrá el nombre de la persona que tiene la reserva y a la derecha aparecerá un botón con el nombre “Anular”. Al pulsar el botón, la reserva mostrará un cuadro de diálogo de confirmación (confirm) que avisará al usuario si realmente desea borrar la reserva. Si acepta, se borrará la reserva.
- Cualquiera de las operaciones de gestión de reservas (Creación y borrado) será enviada al backend usando una API REST.
- Las reservas estarán almacenadas en una base de datos.

Se pide:

- **A) Implementar el backend de la aplicación web con Spring, 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 reservations.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