

El examen tiene una duración de 1:30 horas.

## Ejercicio de Programación (6p)

Se desea implementar una aplicación web para anunciar los conciertos de una sala. Esta aplicación web deberá tener las siguientes características:

- Funcionalidades:
  - Al entrar en la aplicación se mostrará un título “Conciertos MusiURJC”, un formulario para dar de alta un nuevo concierto y la lista de conciertos existentes.
  - Los datos de un conciertos son: Título, fecha y sala (A, B o C).
  - Cuando el usuario complete los datos de un concierto y le de al botón de “Crear”, se enviarán los datos del concierto al servidor, que los guardará en la base de datos y se volverá a recargar la página de nuevo. Ese nuevo concierto se mostrará en la lista de conciertos si se ha grabado correctamente en la base de datos. Opcionalmente se puede poner una página informando de que el concierto se ha guardado correctamente antes de mostrar la página principal.
  - Los conciertos no se pueden borrar.
  - Cuando un usuario quiere crear un concierto, el servidor verificará que no hay ningún otro concierto en esa misma sala en esa misma fecha. En caso de error de validación, se le mostrará un error al usuario.
- Cuestiones de implementación:
  - Se deberá implementar la web con arquitectura web tradicional (generando los HTML en el servidor) y también como una aplicación SPA. Es decir, como hemos hecho en la práctica.
  - La aplicación web tradicional estará en la raíz de la URL (<http://localhost/>) y la aplicación SPA estará en (<http://localhost/next/>).
  - No es necesario usar ningún tipo de estilo CSS ni librería de componentes.
  - La validación de fecha y sala se implementará en el servidor. Para ello se usará código Java (no se usarán restricciones de la base de datos). Se valorará que no haya código repetido en el servidor.
  - La forma de mostrar el error al usuario puede ser diferente en la arquitectura web tradicional y en SPA. Por ejemplo, en arquitectura web tradicional puede ser una nueva página de error que muestre el mensaje “Error al crear el concierto” y en SPA puede ser una alerta de JavaScript (`alert("Error al crear el concierto");`).

Se pide:

- **A) Implementar la aplicación web con Spring MVC, SpringData y JPA. (2.5p)**
  - No es necesario escribir el fichero pom.xml, se puede asumir que tiene todas las dependencias correspondientes y está conectada a una base de datos externa 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 pueden dejar indicados con un comentario.
  - En caso de que se necesite definir una API REST, se valorará especialmente que esté diseñada de forma correcta (formato de las URLs, uso de los códigos de estado, etc.)
  - Se valorará que no haya código duplicado.
- **B) Implementar el frontend usando Angular (2p)**
  - Se puede asumir que se dispone de un proyecto Angular con todos los ficheros necesarios (index.html, angular.cli, package.json, tsconfig.json, app.module.ts, etc.).

- Hay que escribir todos los ficheros necesarios para implementar los componentes, servicios y configuración de rutas (en caso de que sean necesarias).
- No es necesario incluir los imports en los ficheros TypeScript.
- Se usará HTML plano en el template de los componentes (sin ninguna librería de componentes como ng-bootstrap o angular-material).
- 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 (1.5p)**
  - Escribe un script (en Bash, Bat o PowerShell) que al ejecutarse construya la imagen daw/alumnos: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 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