Universidad Rey Juan Carlos

Desarrollo de Aplicaciones Web
3º Grado en Ingeniería del Software

Examen 5 de Julio de 2022

El examen tiene una duración de 1:30 horas.

Ejercicio de Programación (6p)

Se desea implementar una aplicación web para gestionar una lista de tareas pendientes de hacer. Esta aplicación web deberá tener las siguientes características:

- Funcionalidades:
  - Al entrar en la aplicación se mostrará el título “Tareas”, un formulario para dar de alta una nueva tarea y la lista de tareas que se han dado de alta previamente.
  - Una tarea es simplemente su descripción. Por ejemplo: “Comprar pan”, “Estudiar DAW”, “Preparar vacaciones”, etc.
  - Cuando el usuario pone la descripción de la tarea y le de al botón de “Crear”, se enviarán la descripción al servidor, que la guardará en la base de datos.
  - Al guardar una nueva tarea en la base de datos, la lista de tareas se debe actualizar para incluir a la nueva tarea.
  - Las tareas se guardarán en la base de datos como mayúsculas, independientemente de cómo se hayan escrito en el formulario de la web. Esta transformación a mayúsculas se realizará en el servidor (por seguridad).
  - En la lista de tareas, al lado de cada tarea aparecerá un botón “Borrar”. Al pulsar ese botón, la tarea será borrada de la base de datos y se actualizará la lista de tareas para que se elimine también de la lista.
- 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/new/).
  - No es necesario usar ningún tipo de estilo CSS ni librería de componentes.
  - La conversión a mayúsculas de la descripción de la tarea deberá realizarse en un único lugar del código, independientemente de si se trata de la web MVC o de la API REST que permite implementar la web SPA.

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 de Angular 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 fichero **Dockerfile multistage** que construya la imagen daw/tareas:1.0.0 con la aplicación (tanto backend como frontend) empaquetada.
  - Se asumirá que el código Spring está alojado en una carpeta “backend” y el código Angular está alojado en una carpeta “frontend” y el Dockerfile está 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